UNIVERSIDADE FEDERAL DO CEARÁ DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

ANANDA LIMA FREIRE

A DIMENSÃO TEMPORAL NO PROJETO DE CLASSIFICADORES DE PADRÕES PARA NAVEGAÇÃO DE ROBÔS MÓVEIS: UM ESTUDO DE CASO

Livros Grátis

http://www.livrosgratis.com.br

Milhares de livros grátis para download.

ANANDA LIMA FREIRE

A DIMENSÃO TEMPORAL NO PROJETO DE CLASSIFICADORES DE PADRÕES PARA NAVEGAÇÃO DE ROBÔS MÓVEIS: UM ESTUDO DE CASO

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de Mestre em Engenharia de Teleinformática.

Orientador:

Prof. Dr. Guilherme de Alencar Barreto Co-Orientador:

Prof. Dr. Antonio Themoteo Varela

Em memória de meu pai, Armando Freire

Agradecimentos

A Deus, acima de tudo.

Ao meu orientador, Prof. Guilherme de Alencar Barreto, a quem sou grata pela orientação, paciência e confiança depositada.

Ao meu co-orientador, Prof. Antonio Themoteo Varela por abrir os caminhos à minha frente, permitindo que eu desse início a esta jornada.

Aos colegas de laboratório, por estarem sempre dispostos a ajudar, e por proporcionarem momentos de descontração, criando um ótimo ambiente de trabalho.

Aos professores e funcionários do Departamento de Engenharia de Teleinformática que de forma direta ou indireta contribuíram para o desenvolvimento deste trabalho.

À FUNCAP (Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico) pelo suporte financeiro e ao Governo do Estado do Ceará pelo robô cedido à Universidade Federal do Ceará.

À minha mãe, Odete, e ao meu amado Victor pelo amor, carinho, incentivo e apoio incondicional, que permitiram concentrar minhas energias neste trabalho.

Resumo

Este trabalho investiga o grau de influência que a inclusão de mecanismos de memória de curta duração (MCD) exercem sobre o desempenho de classificadores neurais quando aplicados em tarefas de navegação de robôs. Em particular, trata da navegação do tipo *Wall Following*. Para este fim, quatro conhecidas arquiteturas neurais (Perceptron Logístico, Perceptron Multicamadas, Mistura de Especialistas e rede de Elman) são usadas com o intuito de associar diferentes padrões de leituras sensoriais com quatro classes de ações pré-determinadas. Todas as etapas dos experimentos - aquisição dos dados, seleção e treinamento das arquiteturas em simulador, além da execução das mesmas em robô móvel real (SCITOS G5) - são descritas em detalhes. Os resultados obtidos sugerem que a tarefa de seguir paredes, formulada como um problema de classificação de padrões, é não-linearmente separável, resultado este que favorece a rede MLP quando os classificadores são treinados sem MCD. Contudo, se mecanismos de MCD são usados, então até mesmo uma rede linear é capaz de executar a tarefa de interesse com sucesso.

Palavras-chave: Redes Neurais Artificiais, Robótica Móvel, Estratégia Seguir-Paredes, Memória de Curta Duração

Abstract

This work reports results of an investigation on the degree of influence that the inclusion of short-term memory mechanisms has on the performance of neural classifiers when applied to robot navigation tasks. In particular, we deal with the well-known strategy of navigating by "wall-following". For this purpose, four neural architectures (Logistic Perceptron, Multilayer Perceptron, Mixture of Experts and Elman network) are used to associate different sensory input patterns with four predetermined action categories. All stages of the experiments - data acquisition, selection and training of the architectures in a simulator and their execution on a real mobile robot - are described. The obtained results suggest that the wall-following task, formulated as a pattern classification problem, is nonlinearly separable, a result that favors the MLP network if no memory of input patters are taken into account. If short-term memory mechanisms are used, then even a linear network is able to perform the same task successfully.

Keywords: Artificial Neural Networks, Mobile Robotics, Wall-following Strategy, Short Term Memory

Lista de Figuras

1.1	Ilustração do processo básico de interação com o ambiente por parte de um robô móvel.	3
1.2	Diagrama básico dos níveis de controle neste trabalho	4
1.3	Ilustração do comportamento wall-following	5
1.4	Diagrama de um sistema de classificação genérico	6
1.5	Diagrama de um sistema de classificação de padrões como controlador de alto nível	8
1.6	Exemplo hipotético de um problema linearmente separável	14
1.7	Exemplo de um problema não-linear tratado por um discriminante linear	14
1.8	Exemplo de aplicação bem sucedida com um discriminante não-linear	15
2.1	Modelo do neurônio artificial de McCulloch e Pitts	20
2.2	Ilustração da arquitetura da rede neural perceptron simples	21
2.3	Função de ativação Degrau e Sinal	23
2.4	Função de ativação Sigmóide Logística e Tangente Hiperbólica	26
2.5	Ilustração da rede perceptron multicamadas com uma camada oculta	28
2.6	Diagrama de blocos da Mistura de Especialistas, onde as saídas dos módulos especialistas são mediadas através do seletor.	33
2.7	Fluxo do sinal em um neurônio linear de módulo especialista i	36
2.8	Ilustração da arquitetura da rede de Elman com apenas uma camada oculta	42
3.1	Robô SCITOS G5 no Laboratório de Robótica (LABOR/CENTAURO)	46
3.2	Percepção do ambiente e trajetória do SCITOS G5 realizada pelo Algoritmo 3.3.1	50
3.3	Ilustração das distâncias resumidas	52
3.4	Diagrama do ambiente de navegação	53
3.5	Percepção do ambiente e trajetória do SCITOS G5 realizada pelo Algoritmo 3.3.1	53

Lista de Figuras vi

3.6	Gráfico de dispersão das distâncias resumidas à frente e à esquerda	55
3.7	Superfície de decisão da rede PL(2,4)	58
3.8	Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(2,4)	58
3.9	Percepção do ambiente e trajetória do SCITOS G5 para a rede a) PL(24,4) e b) PL(4,4).	59
3.10	Percepção do ambiente e trajetória do SCITOS G5 para a rede ME(2,4,4)	61
3.11	Superfície de decisão da rede ME(2,4,4)	61
3.12	Percepção do ambiente e trajetória do SCITOS G5 para a rede a) ME(24,4,4) e b) ME(4,4,4)	62
3.13	Superfícies de Decisão com a Rede MLP com a) 3, b) 4, c) 5, d) 6, e) 7 e f) 8 g) 10 e h) 20 neurônios ocultos	64
3.14	Ativações da camada oculta da rede MLP(2,6,4)	66
3.15	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,6,4)	67
3.16	Percepção do ambiente e trajetória do SCITOS G5 para a rede a) MLP(24,6,4) e b) MLP(4,6,4)	68
4.1	Percepção do ambiente e trajetória do SCITOS G5 para a rede PL com a) $\tau=1$, b) $\tau=3$, c) $\tau=5$, d) $\tau=19$ e e) $\tau=34$	75
4.2	Superfícies de decisão da rede perceptron com a) $\tau=1$, b) $\tau=3$ e c) $\tau=5$	76
4.3	Superfícies de decisão da rede perceptron com a) $\tau=19$ e b) $\tau=34$	77
4.4	a) Percepção do ambiente e trajetória do SCITOS G5 e b) Superfícies de decisão da rede perceptron com $\tau=9$	77
4.5	Superfícies de decisão da arquitetura ME com a) $\tau=1$, b) $\tau=4$, c) $\tau=6$	80
4.6	Superfícies de decisão da arquitetura ME com a) $\tau = 9$ e b) $\tau = 19$	81
4.7	Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com a) $\tau=1$, b) $\tau=4$	81
4.8	Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com $ au=6$	82
4.9	Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com a) $\tau = 9$ e b) $\tau = 19$	82

Lista de Figuras vii

4.10	Superfícies de decisão com a rede de Elman com a) $q=2$, b) $q=3$, c) $q=4$, d) $q=5$, e	
	e)q = 6 neurônios ocultos	84
4.11	Percepção do ambiente e trajetória do SCITOS G5 para a rede de Elman(2+4,4,4)	86
4.12	Gráfico das taxas de acerto dos testes das redes PL, PL com memória (PLm), ME, ME com memória (MEm), MLP e Elman	87
5.1	Gráfico das taxas de acerto dos testes em relação à quantidade de amostras passadas (rede PL)	91
5.2	Gráfico das taxas de acerto dos testes em relação à quantidade de amostras passadas (arquitetura ME)	91
5.3	Custo Computacional	92
A.1	Ativações da camada oculta da rede MLP(24,6,4)	95
A.2	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,6,4)	96
A.3	Ativações da camada oculta da rede MLP(4,6,4)	97
A.4	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,6,4)	98
A.5	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,8,4)	99
A.6	Ativações da camada oculta da rede MLP(24,8,4)	100
A.7	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,20,4)	101
A.8	Ativações da camada oculta da rede MLP(24,20,4)	102
A.9	Ativações da camada oculta da rede MLP(4,8,4)	104
A.10	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,8,4)	105
A.11	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,20,4)	106
A.12	Ativações da camada oculta da rede MLP(4,20,4)	107
A.13	Ativações da camada oculta da rede MLP(2,8,4)	109
A.14	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,8,4)	110
A.15	Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,20,4)	111
A.16	Ativações da camada oculta da rede MLP(2,20,4)	112
B.1	Percepção do ambiente e trajetória do SCITOS G5 para a rede de a) Elman(24+6,6,4) e b) Elman(24+5,5,4)	114
	,	1

Lista de Figuras viii

B.2	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(28,4,4)	115
B.3	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(24+3,3,4)	116
B.4	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+5,5,4)	118
B.5	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+4,4,4)	119
B.6	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+3,3,4)	120
B.7	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(2+5,5,4)	121
B.8	Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(5,3,4)	122
C.1	Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(24,4)	124
C.2	Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(4,4)	125
C.3	Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com a) $\tau=1$, b) $\tau=3$ e c) $\tau=5$ amostras passadas	128
C.4	Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com a) $\tau=9$ e b) 19	129
C.5	Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com $tau = 34$.	130
D.1	Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME(24,4,4)	132
D.2	Percepção do ambiente e trajetória do SCITOS G5 para a rede ME(4,4,4)	133
D.3	Percepção do ambiente e trajetória do SCITOS G5 para a mistura de especialistas com $ au=1$ e 4	136
D.4	Percepção do ambiente e trajetória do SCITOS G5 para a mistura de especialistas com $\tau=6$ e 9	137
D.5	Percepção do ambiente e trajetória do SCITOS G5 para a ME(40,4,4)	138

Lista de Tabelas

3.1	Estatísticas do treinamento <i>offline</i> da rede PL(2,4)	57
3.2	Resultado dos testes <i>offline</i> para a rede PL(2,4)	57
3.3	Resumo do resultado dos testes <i>offline</i> para as redes PL(4,4) e PL(24,4)	59
3.4	Estatísticas do treinamento <i>offline</i> para a arquitetura ME(2,4,4)	60
3.5	Resultado dos testes <i>offline</i> para a arquitetura ME(2,4,4)	60
3.6	Resumo dos resultado dos testes $\it offline$ para as arquiteturas $\it ME(24,4,4)$ e $\it ME(4,4,4)$	62
3.7	Estatísticas do treinamento <i>offline</i> para a rede MLP(2,6,4)	65
3.8	Resultado dos testes <i>offline</i> para a rede MLP(2,6,4)	65
3.9	Resultado dos testes <i>offline</i> para as redes MLP(24,6,4) e MLP(4,6,4)	67
4.1	Quantidade de componentes no vetor ${\bf x}$ para cada ${\bf \tau}$ com a rede PL	71
4.2	Quantidade de componentes no vetor ${\bf x}$ para cada τ com a rede ME	71
4.3	Estatísticas do treinamento <i>offline</i> da rede PL com memória $\tau=1,3$ e 5	73
4.4	Estatísticas do treinamento <i>offline</i> da rede PL com memória $\tau=9,19$ e 34	73
4.5	Resultado dos testes <i>offline</i> para a rede PL com $\tau = 1, 3$ e 5	74
4.6	Resultado dos testes <i>offline</i> para a rede PL com $\tau = 9, 19$ e 34	74
4.7	Estatísticas do treinamento <i>offline</i> da rede ME $\tau = 1,4$ e 6	78
4.8	Estatísticas do treinamento <i>offline</i> da rede ME $\tau = 9$ e 19	78
4.9	Resultado dos testes <i>offline</i> da rede ME $ au=1,4$ e 6	79
4.10	Resultado dos testes <i>offline</i> da rede ME com memória $\tau=9$ e 19	79
4.11	Estatísticas do treinamento <i>offline</i> da rede de Elman(2+4,4,4)	85
4.12	Resultado dos testes <i>offline</i> da rede de Elman(2+4,4,4)	85
4.13	Resultado do cálculo dos custos computacionais	88

Lista de Tabelas x

A.1	Estatísticas do treinamento <i>offline</i> da rede MLP(24,6,4)	94
A.2	Resultado dos testes <i>offline</i> da rede MLP(24,6,4)	94
A.3	Estatísticas do treinamento textitoffline da rede MLP(4,6,4)	96
A.4	Resultado dos testes <i>offline</i> da rede MLP(4,6,4)	97
A.5	Estatísticas do treinamento textitoffline da rede MLP(24,8,4)	99
A.6	Resultado dos testes <i>offline</i> da rede MLP(24,8,4)	99
A.7	Estatísticas do treinamento textitoffline da rede MLP(24,20,4)	101
A.8	Resultado dos testes <i>offline</i> da rede MLP(24,20,4)	101
A.9	Estatísticas do treinamento textitoffline da rede MLP(4,8,4)	103
A.10	Resultado dos testes <i>offline</i> da rede MLP(4,8,4)	104
A.11	Estatísticas do treinamento textitoffline da rede MLP(4,20,4)	105
A.12	Resultado dos testes <i>offline</i> da rede MLP(4,20,4)	106
A.13	Estatísticas do treinamento textitoffline da rede MLP(2,8,4)	108
A.14	Resultado dos testes <i>offline</i> da rede MLP(2,8,4)	108
A.15	Estatísticas do treinamento textitoffline da rede MLP(2,20,4)	110
A.16	Resultado dos testes <i>offline</i> da rede MLP(2,20,4)	110
B.1	Estatísticas do treinamento <i>offline</i> da rede Elman com 6 e 5 neurônios ocultos	113
B.2	Resultado dos Testes <i>offline</i> da rede Elman com memória $\tau=6$ e 5	113
B.3	Estatísticas do treinamento offline da rede Elman com 4 neurônios ocultos	114
B.4	Resultado dos testes <i>offline</i> da rede Elman com 4 neurônios ocultos	115
B.5	Estatísticas do treinamento offline da rede Elman com 3 neurônios ocultos	116
B.6	Resultado dos testes <i>offline</i> da rede Elman com 3 neurônios ocultos	116
B.7	Estatísticas do treinamento <i>offline</i> da rede Elman com 6 e 5 neurônios ocultos	117
B.8	Resultado dos Testes <i>offline</i> da rede Elman com memória $\tau=6$ e 5	117
B.9	Estatísticas do treinamento offline da rede Elman com 4 neurônios ocultos	118
B.10	Resultado dos testes <i>offline</i> da rede Elman com 4 neurônios ocultos	118

Lista de Tabelas xi

B.11	Estatísticas do treinamento <i>offline</i> da rede Elman com 3 neurônios ocultos	119
B.12	Resultado dos testes <i>offline</i> da rede Elman com 3 neurônios ocultos	119
B.13	Estatísticas do treinamento <i>offline</i> da rede Elman com 6 e 5 neurônios ocultos	121
B.14	Resultado dos Testes <i>offline</i> da rede Elman com memória $\tau=6$ e 5	121
B.15	Estatísticas do treinamento offline da rede Elman com 3 neurônios ocultos	122
B.16	Resultado dos testes <i>offline</i> da rede Elman com 4 neurônios ocultos	122
C.1	Estatísticas do treinamento offline da rede PL(24,4)	123
C.2	Resultado dos Testes offline da rede PL(24,4)	124
C.3	Estatísticas do treinamento offline da rede PL(4,4)	124
C.4	Resultado dos Testes offline da rede PL(4,4)	125
C.5	Quantidade de componentes no vetor ${\bf x}$ para cada ${\bf au}$ com a rede PL	126
C.6	Estatísticas do treinamento <i>offline</i> da rede PL com 4 distâncias resumidas e memória $\tau = 1, 3$ e 5	126
C.7	Estatísticas do treinamento <i>offline</i> da rede PL com 4 distâncias resumidas e memória $\tau = 9,19$ e 34	
C.8	Resultado dos Testes <i>offline</i> PL com 4 distâncias resumidas e memória $\tau=1,3$ e 5	127
C.9	Resultado dos Testes <i>offline</i> PL com 4 distâncias resumidas e memória $\tau=9,19$ e 34	127
D.1	Estatísticas do treinamento <i>offline</i> da arquitetura ME(24,4,4)	131
D.2	Resultado dos Testes <i>offline</i> da arquitetura ME(24,4,4)	132
D.3	Estatísticas do treinamento <i>offline</i> da arquitetura ME(4,4,4)	133
D.4	Resultado dos Testes <i>offline</i> da arquitetura ME(4,4,4)	133
D.5	Quantidade de componentes no vetor ${\bf x}$ para cada τ com a rede ME	134
D.6	Estatísticas do treinamento <i>offline</i> da rede ME com memória $ au=1,4$ e 6	134
D.7	Estatísticas do treinamento <i>offline</i> da rede ME com memória $\tau=9$ e $\tau=19$	134
D.8	Resultado dos Testes <i>offline</i> ME com $\tau = 1,4$ e 6	135
D.9	Resultado dos Testes <i>offline</i> da arquitetura ME com memória $\tau = 9$ e 19	135

Lista de Símbolos

f	função para variável continua
X	variável vetorial, i.e., $\mathbf{x} \in \mathfrak{R}^n$
\mathbf{w}_i	vetor de pesos associados ao neurônio i em uma rede neural
\mathbf{m}_k	vetor de pesos associados ao neurônio k em uma rede neural
η	taxa de aprendizagem das redes neurais artificiais
X	matriz de dados
d	vetor de saídas desejadas para redes supervisionadas
w_{ij}	peso associado à ligação entre entrada j e neurônio i da camada intermediária de uma
	rede supervisionada
m_{ki}	peso associado à ligação entre neurônio i da camada intermediária e saída k de uma
	rede supervisionada
ε_{med}	erro quadrático médio por época de treinamento da rede supervisionadas
$\phi(\cdot)$	função de ativação de um neurônio de rede supervisionada
θ_i	limiar de ativação de um neurônio i de rede supervisionada
u_i	ativação do neurônio i da camada escondida de uma rede supervisionada
y_i	saída do neurônio i na camada de saída de uma rede supervisionada
δ_k	gradiente local do neurônio k em uma rede neural
δ_i	gradiente local do neurônio i em uma rede neural
p	dimensão do vetor de entrada de uma rede neural
m	dimensão do vetor de saída de uma rede neural
q	número de neurônios da primeira camada escondida
I	dimensão da entrada (input dimension)
O	dimensão da saída (output dimension)
K	quantidade de redes especialistas

τ

quantidade de amostras passadas

xiii

Lista de Siglas

EQM Erro Quadrático Médio

MCD Memória de Curta Duração

ME Mistura de Especialistas

MLP MultiLayer Perceptron

M-P McCulloch-Pitts

PL Perceptron Logístico

RNAs Redes Neurais Artificiais

Sumário

Re	esumo					
Al	bstrac	:t		iv		
Li	sta de	sta de Figuras viii				
Li	sta de	e Tabela	\mathbf{s}	xi		
Li	sta de	e Símbo	los	xii		
Li	sta de	e Siglas		xiii		
1	Int	RODUÇA	ÃO	1		
	1.1	Funda	mentos da Robótica Móvel	2		
	1.2	Recon	hecimento de Padrões	4		
		1.2.1	Arquitetura Padrão dos Classificadores	6		
	1.3	Recon	hecimento de Padrões em Navegação de Robôs	7		
	1.4	Padrõe	es Estáticos e Dinâmicos	9		
		1.4.1	Classes de Problemas em Processamento de Sequências Temporais	10		
	1.5	Memó	ria de Curta Duração	11		
	1.6	Defini	ção dos Problemas de Interesse	13		
	1.7	Objeti	vos da Dissertação	16		
		1.7.1	Objetivo Geral	16		
		172	Objetivos Específicos	16		

Sumário xv

	1.8	Produç	ção Científica	16
	1.9	Organi	ização Geral da Dissertação	17
2	RED	es Nei	URAIS ARTIFICIAIS	19
_				
	2.1		ıção	
	2.2	Rede F	Perceptron Simples	21
		2.2.1	Arquitetura da rede PS	21
		2.2.2	Regras de Ativação e Aprendizagem	23
		2.2.3	Perceptron Simples Logístico	25
	2.3	Rede F	Perceptron Multicamadas	27
		2.3.1	Propagação no sentido direto	29
		2.3.2	Propagação no sentido inverso	30
	2.4	Mistur	ra de Especialistas	31
		2.4.1	Vantagens da Arquitetura Modular	32
		2.4.2	Descrição da Arquitetura de Mistura de Especialistas	32
		2.4.3	Resumo do Algoritmo da Arquitetura ME	40
	2.5	Rede F	Recorrente de Elman	41
	2.6	Conclu	ısão	43
3	RES	ULTAD(OS DOS EXPERIMENTOS COM CLASSIFICAÇÃO ESTÁTICA	45
	3.1	Introdu	lção	45
	3.2		Móvel SCITOS G5	46
		3.2.1	O Hardware	46
	2.2	3.2.2	API do SCITOS G5	47
	3.3		tmo Heurístico para Navegação do tipo Seguir-Paredes	49
	3.4	Metod	ologia para a Coleta de Dados	51
	3.5	Resulta	ados dos Experimentos	54

Sumário xvi

		3.5.1 Resultados com a Rede Perceptron Logístico			56
			3.5.1.1	Resultados para a Rede PL(2,4)	57
			3.5.1.2	Resultados para as Redes PL(4,4) e PL(24,4)	59
		3.5.2	Resultad	os para a Arquitetura de Mistura de Especialistas	60
			3.5.2.1	Resultados para a Rede ME(2,4,4)	60
			3.5.2.2	Resultados para as Arquiteturas $ME(4,4,4)$ e $ME(24,4,4)$	62
		3.5.3	Resultad	os para a Rede MLP	63
			3.5.3.1	Dimensão da Camada Oculta	63
			3.5.3.2	Resultados para a Rede MLP(2,6,4)	65
			3.5.3.3	Resultados para as Redes MLP(4,6,4) e (24,6,4)	67
	3.6	Conclu	ısão		68
4	RES	ULTADO	os dos E	XPERIMENTOS COM CLASSIFICAÇÃO ESPAÇO-TEMPORAL	70
	4.1	Introdu	ıção		70
	4.2	Defini	ção das En	ntradas	71
	4.3	Resulta	ados dos E	Experimentos	72
		4.3.1	Resultad	os da rede PL	73
		4.3.2	Resultad	os com a Mistura de Especialistas	78
		4.3.3	Resultad	os para a Rede de Elman	83
			4.3.3.1	Determinação da Dimensão da Camada Oculta	83
			4.3.3.2	Resultados para rede de Elman(2+4,4,4)	85
	4.4	Desem	penho × 0	Custo Computacional	86
	4.5	Conclu	ısão		88
5	Con	CLUSÕ	es e Per	SPECTIVAS	90
АĮ	pêndio	ee A – T	estes extr	as com a rede MLP	94
	A.1	Resulta	ados da rec	de MLP(24,6,4)	94

Sumário xvii

A.2	Resulta	ados para MLP(4,6,4)	. 96	
A.3	A.3 Resultados do Estudo com Diferentes Quantidades de Neurônios Oculto			
	A.3.1	Resultados para MLP(24,q,4)	. 98	
		A.3.1.1 MLP(24,8,4)	. 98	
		A.3.1.2 MLP(24,20,4)	. 100	
	A.3.2	Resultados para MLP(4,q,4)	. 103	
		A.3.2.1 MLP(4,8,4)	. 103	
		A.3.2.2 MLP(4,20,4)	. 105	
	A.3.3	Resultados para MLP $(2,q,4)$. 108	
		A.3.3.1 MLP(2,8,4)	. 108	
		A.3.3.2 MLP(2,20,4)	. 110	
Apêndio	e B – T	estes extras com a rede de Elman	113	
B.1	Resulta	ados para a rede de Elman $(24+q,q,4)$. 113	
B.2	Resulta	ados da rede de Elman $(4+q,q,4)$. 117	
		B.2.0.3 Resultados para a Rede de Elman $(2+q,q,4)$. 120	
Apêndio	ce C – T	estes Extras com a Rede PL	123	
C.1	Rede F	L sem o uso de MCD	. 123	
	C.1.1	Resultados para a Rede PL(24,4)	. 123	
	C.1.2	Resultados para a rede PL(4,4)	. 124	
C.2	Rede F	L com MCD	. 125	
Apêndio	e D – T	estes Extras com a Arquitetura ME	131	
D.1	Arquit	etura ME sem o uso de MCD	. 131	
	D.1.1	Resultados para a Arquitetura ME(24,4,4)	. 131	
	D.1.2	Resultados para a Arquitetura ME(4,4,4)	. 132	
D.2	Arquit	etura de Mistura de Especialistas com MCD	. 133	

Si	umai	rıo	XV11

Referências Bibliográficas

139

1 Introdução

Dentro da área de pesquisa de Aprendizagem de Máquinas, redes neurais artificiais mostramse bem sucedidas em diversos problemas de classificação de padrões. A maioria dos classificadores neurais visa a construção de um mapeamento estático, no qual o sistema é alimentado com um vetor **x** e retorna um vetor **y**, em que ambos são padrões espaciais, i.e. sem memória. Tais redes não são capazes de tratat as propriedades temporais presentes nos padrões de entrada pois processam as entradas atuais sem levar em consideração o histórico dos dados passados (BARRETO et al., 2003).

Em muitos problemas do mundo real, sejam em sistemas reais ou artificiais (e.g. robôs móveis), a maior parte da informação a ser processada não possui natureza estática, mas depende também da relação temporal entre os dados que compõem a informação. Esse conjunto de dados ordenado temporalmente é chamado de sequência espaço-temporal (BARRETO, 1999). Por exemplo, em Robótica, observa-se que durante a navegação de um robô móvel em um determinado ambiente, é gerada uma rica quantidade de informação espaço-temporal a partir de seus sensores e atuadores (BASHIR et al., 2006). A partir dos dados coletados durante a execução de uma trajetória, percebe-se que determinadas sequências de padrões espaciais podem indicar pontos de referência, como a existência de uma canto entre duas paredes ou obstáculos.

A dimensão temporal dos dados pode ser aproveitada de forma que um classificador de padrões receba as informações do ambiente por meio de sensores, e produza como saída os diferentes movimentos que o robô pode executar. Como exemplo de classificador de padrões útil para este problema, pode-se citar as diversas arquiteturas de redes neurais artificiais. Neste contexto, pode-se entender o processo de aprendizado nas arquiteturas neurais como um tipo de aprendizado por imitação, uma vez que o agente robótico desenvolverá um comportamento semelhante ao que gerou os dados de treinamento, sempre que se encontrar em uma situação ambiental parecida. Trabalhos como o de ANTONELO et al. (2008b) e MUCIENTES et al. (2009a) fazem uso dessa metodologia.

Tanto da perspectiva da Robótica quanto da Classificação de Padrões, a categorização dos

dados referentes às ações de um robô é uma importante área de pesquisa (ELSHAW et al., 2004; MALONE et al., 2005). Desta forma, este trabalho faz uso do problema de imitação de comportamento por meio de classificadores neurais, não só revisitando uma estratégia clássica de navegação, o método de seguir paredes (*wall-following*) (BAUZIL et al., 1981), mas também realizando um estudo de caso como a dimensão temporal influencia o desempenho dos classificadores nesta tarefa.

Ao longo deste capítulo são definidos alguns conceitos básicos necessários para a compreensão deste trabalho. Estes estão relacionados a três tópicos principais: a definição do problema de classificação de padrões estáticos e dinâmicos, a definição do problema da navegação de robôs móveis como um problema de processamento de padrões espaço-temporais, e a aplicação de redes neurais artificiais aos dois problemas anteriores.

Assim, o capítulo se organiza da seguinte forma: na Seção 1.1 é feita uma introdução à Robótica Móvel com alguns conceitos importantes para este trabalho. Na Seção 1.2 é apresentado o problema de Classificação de Padrões, suas características e a estrutura de um classificador. Na Seção 1.3 é feita a contextualização do problema de navegação como um probema de classificação de padrões. Na Seção 1.4 são descritos os padrões de entrada estáticos e dinâmicos. Na Seção 1.5 é feita uma introdução sobre estruturas de memória de curta duração. Na Seção 1.6 são apresentados os problemas de interesse neste trabalho e por fim, nas Seções 1.7, 1.8 e 1.9 são apresentados os objetivos, produção científica e organização desta dissertação, respectivamente.

1.1 Fundamentos da Robótica Móvel

A Robótica pode ser vista como sendo a área da ciência e tecnologia que estuda, desenvolve e aplica robôs em suas mais diversas instâncias (AGUIRRE, 2007). Um robô é um conjunto de dispositivos mecânicos, elétricos e eletrônicos, capaz de substituir o trabalho humano em situações perigosas ou insalubres (BEKEY, 2005), podendo ser autônomo ou controlado remotamente. Embora atualmente existam robôs criados para o entretenimento e atividades domésticas, capazes de reconhecer voz, faces, obstáculos e objetos em movimento, a grande maioria é utilizada na indústria na forma de robôs manipuladores.

Dentro da Robótica, há uma área que trabalha especificamente com mecanismos que podem se locomover pelo seu ambiente, não estando restritos a uma base fixa. Essa área é a Robótica Móvel, que tem entre seus objetivos elaborar trajetórias para robôs móveis, de tal forma que eles possam atingir suas metas, realizar movimentos sem colisões, a localizar e alcançar de alvos,

ter auto-suficiência em termos de suprimento de energia, transportar de objetos, dentre outras aplicações.

Os robôs são constituídos basicamente de sensores, atuadores, um *hardware* microprocessado e um software de controle. Sensores são dispositivos que transferem informações sobre ambiente para o robô, medindo grandezas físicas de forma passiva (termômetros, fotômetro, etc.) ou ativa (laser, sonar, etc.) (PIERI, 2002). Atuadores são responsáveis pela movimentação e articulação das partes móveis. O *hardware* microprocessado, que é normalmente construído em dimensões reduzidas, possui propósitos específicos, devendo funcionar de forma autônoma (JUNG et al., 2005). O software de controle faz a interconexão dos componentes, formando uma configuração que fornece a resposta desejada do sistema (DORF; BISHOP, 2008).

Sabe-se que o projeto de robôs móveis é uma tarefa desafiadora, sendo a principal dificuldade concentrada no projeto do controlador, pois a interação dele com o ambiente pode ser vista como a interação de dois sistemas dinâmicos (AGUIRRE, 2007). Mesmo quando o ambiente é estático, ou seja, quando não há outros agentes móveis ou mudanças na estrutura física desse ambiente, na perspectiva do robô que se movimenta, o ambiente varia. Dessa forma, as informações geradas pelos sensores em um dado instante é função das propriedades do ambiente que o circunda e também das ações anteriores realizadas por ele (AGUIRRE, 2007).

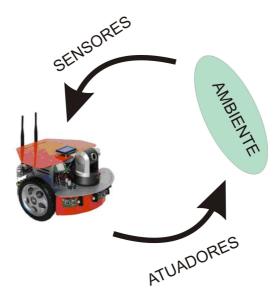


Figura 1.1: Ilustração do processo básico de interação com o ambiente por parte de um robô móvel.

Um sistema de controle para robôs móveis possui no mínimo três etapas (BUGMANN et al., 1998), mostrado na Figura 1.1. Na primeira, chamada de sensoriamento, são coletados dados sobre o ambiente por meio de sensores (câmeras de vídeo, ultra-som, etc.). Em um segundo estágio, denominado de programação ou planejamento, usa-se a informação sobre a

configuração (estado) atual do robô obtida a partir da leitura dos sensores para determinar a próxima ação a ser desenvolvida. Em um terceiro e último estágio, chamado de execução, procedimentos clássicos de controle seriam usados para movimentar o robô de acordo com o que foi "decidido" no segundo estágio.

O interesse desta dissertação está voltado para o segundo estágio, em um nível de tomada de decisão. Em outras palavras, a intenção está em desenvolver um sistema computacional que determine o comportamento do robô com base nas informações coletadas do ambiente. O nível mais baixo de controle é responsável por garantir a resposta por parte dos atuadores de acordo com as decisões feitas. Na Figura 1.2 é mostrada essa divisão, em que no nível correspondente à tomada de decisão, o controle é realizado por meio de uma rede neural artificial (caso explorado neste trabalho).

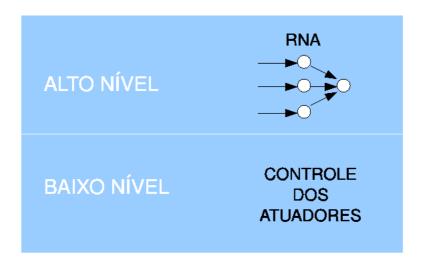


Figura 1.2: Diagrama básico dos níveis de controle neste trabalho.

As decisões tomadas têm como base um determinado comportamento que o robô deve apresentar em certas situações. Neste trabalho, revisita-se uma das estratégias clássicas para navegação de robôs móveis (BAUZIL et al., 1981; KRISHNA; KALRA, 2000); que é o comportamento de seguir-paredes (*wall-following*). Neste, a movimentação do robô é baseada no movimento paralelo à uma parede à uma pequena distância, conforme ilustrado na Figura 1.3

Na próxima seção é feita uma introdução ao problema de Classificação de Padrões.

1.2 Reconhecimento de Padrões

A área de Reconhecimento de Padrões estuda os processos de decisão e de aprendizagem de uma forma que pode ser resumida na seguinte pergunta (MARQUES, 2005):

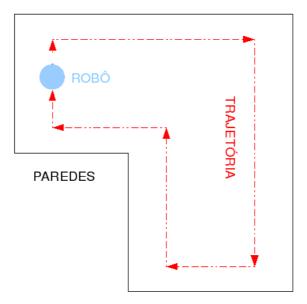


Figura 1.3: Ilustração do comportamento wall-following

como escolher uma de várias hipóteses conhecidas, tendo como informação disponível medidas realizadas por sensores e exemplos de decisões anteriores?

A resposta a esta questão tem aplicações variadas na Engenharia e em muitas outras áreas da Ciência. Dentre os exemplos dessas aplicações, pode-se citar o diagnóstico médico de doenças, reconhecimento de faces, reconhecimento da fala, entre outros. A tomada de decisão é um processo comum aos seres vivos, pois estes estão a todo instante, mesmo que inconscientemente, identificando objetos, sons e cheiros a partir dos sinais captados por seus sistemas sensoriais. Apesar disso, conseguir que um sistema computacional seja capaz de um desempenho semelhante, ainda é um desafio.

Um sistema de reconhecimento de padrões envolve essencialmente os seguintes aspectos mostrados (Figura 1.4):

- 1. observação do fenômeno de interesse e coleta de dados;
- 2. seleção e/ou extração de características;
- 3. classificação.

A coleta de dados é um processo de medição dos fenômenos envolvidos através de qualquer equipamento de aquisição de informação (sensores). A seleção e/ou extração de características consiste na escolha e possível redução da dimensão dos dados preservando a informação relevante para a decisão. Por último, ocorre a tomada de decisão, na qual as características esco-

lhidas alimentam o classificador de padrões, resultando na categorização do objeto de interesse em uma dentre várias classes pré-definidas (MARQUES, 2005).

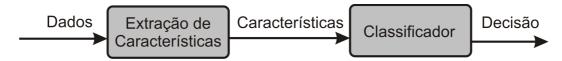


Figura 1.4: Diagrama de um sistema de classificação genérico.

Na seção seguinte são formalizados os conceitos principais relacionados com classificadores de padrões.

1.2.1 Arquitetura Padrão dos Classificadores

Em sistemas de reconhecimento ou classificação de padrões, há c categorias ou classes previamente definidas, nas quais os padrões são enquadradados. Estes últimos são representados por um vetor $\mathbf{x} = [x_1 \ x_2 \ x_3 \ ... \ x_d]^T$ formado por um conjunto de d características ou atributos, e pode ser visto como um ponto, ou vetor, no espaço d-dimensional chamado de espaço de $características <math>S \subset \mathbb{R}^d$. O resultado da classificação é representado pelo vetor $\mathbf{y} = [y_1 \ ... \ y_c]$ pertencente ao espaço de saída $W \subset \mathbb{R}^c$ que tem, normalmente, tantos elementos quanto classes a serem escolhidas. Esse vetor possui apenas um único elemento igual a 1, o qual indica qual a classe decidida, enquanto todos os outros possuem valores menores. Dessa forma, o sistema pode ser visto como um mapeamento do conjunto de atributos dos padrões para a saída $S \to W$ (BISHOP, 1995)

Um classificador de padrões é um tipo de mapeamento que associa a cada observação $\mathbf{x} \in S$ uma classe $\mathbf{y} \in W$ (MARQUES, 2005). Ele realiza uma partição do conjunto S em c subconjuntos disjuntos $R_1, R_2, ..., R_c$ em que R_k é o conjunto dos padrões de S mapeados na classe y_k , de forma que

$$R_k = \{ \mathbf{x} \in S : C(\mathbf{x}) = y_k \}, \tag{1.1}$$

em que os conjuntos R_k definem as regiões de $decis\~ao$ do classificador. Cada uma delas é separada da outra por pontos que fronteiros comuns a outras regiões de decis\~ao, e o conjunto desses pontos comuns a duas regiões é chamado de superf'icie de $decis\~ao$ entre estas classes.

Outro conceito importante é a noção de *função discriminante*. Dado um classificador definido pelas regiões de decisão $R_1, R_2, ..., R_c$, a função discriminante associada à região R_k deve

¹Dois conjuntos são ditos disjuntos se não tiverem nenhum elemento em comum. Em outras palavras, dois conjuntos são disjuntos se sua interseção for o conjunto vazio.

satisfazer

$$g_k(\mathbf{x}) > g_j(\mathbf{x}), \ \forall \mathbf{x} \in R_k, \ \forall j \neq k.$$
 (1.2)

Isto posto, a categorização do padrão \mathbf{x} pode ser realizada por meio da seguinte regra de decisão:

$$\hat{y} = y_k : k = \arg\max g_i(\mathbf{x}), \quad i = 1, ..., c,$$
 (1.3)

em que \hat{y} denota a classe predita e k denota o índice da função discriminante que produz o maior valor para o padrão \mathbf{x} .

Em suma, o desenvolvimento de um classificador consiste na determinação do número de classes, c, e das respectivas funções discriminantes $(g_1, ..., g_c)$ que permitam separar as classes de forma a minimizar o erro de classificação (MARQUES, 2005).

Classificadores de padrões podem ser categorizados quanto ao modo de aprendizado, em: (i) supervisionados, em que os padrões no treinamento apresentam-se com seus rótulos (*labels*) que indicam a que classe a que pertencem, ou (ii) não-supervisionados, em que no treinamento não se dispõe dos rótulos para o aprendizado e o número de classes muitas vezes deve ser também determinado (WATANABE, 1985). Neste último caso, o problema de classificação costuma ser chamado de análise de agrupamentos (*clustering*).

Neste trabalho os sistemas classificadores utilizados são redes neurais artificiais e, portanto, a partir deste ponto as explicações se darão sob a perspectiva dessas redes. Na seção seguinte é feito a contextualização do problema de navegação com classificadores.

1.3 Reconhecimento de Padrões em Navegação de Robôs

Para aplicar um classificador em um problema de robótica faz-se necessário descobrir quais e de que tipo são os padrões a serem categorizados, quais são as informações destes que serão alimentados na entrada do classificador e quais são as classes.

Conforme já mencionado, da perspectiva de um robô móvel, o mundo real é percebido através dos seus sensores que enviam sinais para o software de controle armazenado no processador da unidade robótica. A partir dessas informações sensoriais, a rede neural determina a ação adequada para que o robô alcance um determinado objetivo.

Traçando um paralelo com um sistema de reconhecimento de padrões, as classes a serem categorizadas são as diferentes ações que podem ser executadas e o padrão a ser identificado é o estado do ambiente em um determinado instante. As informações a respeito do estado atual são fornecidos por meio de sensores, que alimentam as unidades de entrada do classificador, o qual fará o papel de controlador de alto nível. Esse sistema pode está ilustrado na Figura 1.5.

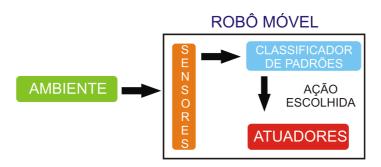


Figura 1.5: Diagrama de um sistema de classificação de padrões como controlador de alto nível.

O modelo adotado neste trabalho é conhecido na literatura como *ciclo sentir-pensar-agir* (*sensing-thinking-acting cycle*) que compõe a abordagem de Processamento da Informação (PFEI-FER; SCHEIER, 2002). Uma outra abordagem para o problema de navegação, conhecida como Ciência Cognitiva Corpórea (*Embodied Cognitive Science*), estabelece que qualquer comportamento deve ser concebido como uma coordenação sensório-motora que estrutura os sinais de entrada do sistema (PFEIFER; SCHEIER, 2002). Nesta perspectiva, a percepção não depende apenas das informações sensoriais, mas também da atividade motora realizada, tentando, com isso, induzir regularidades às informações de entrada do sistema e simplificar a tarefa de classificação (PFEIFER; SCHEIER, 2002). Trabalhos como o de NEHMZOW & SMITHERS (1991), LAMBRINOS & SCHEIER (1997) e LAMBRINOS et al. () fazem uso dessa abordagem dentro do problema de classificação de padrões.

Neste trabalho, as classes do problema são definidas pelos projetistas por meio de tentativa e erro, ajustando as reações do robô de acordo com as informações sensoriais. Esse método resultou em um programa de controle de alto nível baseado em regras SE-ENTÃO, permitindo que o robô navegasse pela sala gerando dados que são coletados como padrões de entrada (perpectiva do ambiente retornada pelos sensores) e as classes (movimentos) correspondentes. Com esses dados, os classificadores são então projetados de forma que ao final do processo, os classificadores possam imitar o comportamento gerado pelo programa baseado em regras.

Por fim, faz-se necessário determinar a natureza dos padrões de entrada, que pode ser estática ou dinâmica. Em outras palavras, é preciso vericar se a dimensão temporal dos dados de entrada é importante para o desempenho dos classificadores. Na seção seguinte são explicadas

as caracteristicas de cada tipo de padrão, assim como os mecanismos que são necessário para o seu processamento.

1.4 Padrões Estáticos e Dinâmicos

No reconhecimento de padrões estáticos, a dimensão do vetor de entrada define a dimensão do espaço de entrada ou espaço de padrões. Em outras palavras, se o vetor de entrada é composto por duas variáveis independentes, então o espaço da entrada tem duas dimensões. As classes são descritas estatisticamente, através de médias, variâncias, etc, e o problema de classificação não é afetado com a sequência na qual os padrões são apresentados (PRINCIPE et al., 2000).

Como exemplos de aplicações, pode-se citar o diagnóstico de doenças, reconhecimento de faces, reconhecimento de caracteres. Em todas estas aplicações, a ordem em que os exemplos são apresentados ao classificador não influencia na decisão.

Já em problemas com características temporais, os dados coletados não são um conjunto de amostras independentes, mas funções no tempo (PRINCIPE et al., 2000). Um único sensor, por exemplo, produz uma sequência de medidas que estão relacionadas no tempo. Se a ordem das medidas for modificada, o sinal será sendo distorcido, mudando as características de frequência, por isso a ordem deve ser preservada no processamento temporal.

Um padrão temporal pode ser contínuo. Neste caso, para posterior processamento por um sistema computacional ele deve ser amostrado e convertido em uma seqüência de padrões discretos no tempo. Doravante nesta dissertação, assume-se que padrões temporais são de tempo discreto, e passarão a ser chamados simplesmente de seqüências temporais.

Essas seqüências temporais surgem naturalmente quando se tenta processar qualquer sinal no domínio do tempo, seja com o propósito de reconhecer ou antecipar a presença de características relevantes em uma aplicação particular, seja com o propósito de reproduzir tais características. A fala humana, sinais biomédicos, música ou qualquer função do tempo originada a partir de sensores constitui uma seqüência, cujo significado ou conteúdo não depende apenas da existência de certas características, mas também de sua ordem temporal (RAY; KARGUPTA, 1996).

Na subseção que se segue, são apresentadas as classes de problemas que realizam o processamento de sequências temporais.

1.4.1 Classes de Problemas em Processamento de Sequências Temporais

Em muitas aplicações científicas e de engenharia é necessário modelar processos dinâmicos que lidam com sequências temporais. O tipo de informação que se deseja obter vai depender da aplicação. Normalmente, quando se processa algum tipo de padrão temporal, se está interessado em (HERTZ et al., 1991):

- Reconhecimento de Sequências: neste caso, deseja-se categorizar uma sequência de entrada específica é apresentada, devendo esta ser apenas identificada. Uma aplicação típica é o reconhecimento de voz, na qual a saída indicaria a palavra que foi falada.
- Reprodução de Sequências Temporais: aqui, o sistema deve ser capaz de gerar a sequência de entrada quando parte dela lhe é apresentada. Este seria o caso apropriado quando se deseja que a rede reproduza uma sequência melódica, sendo capaz de prever o curso futuro de uma série temporal a partir de partes desta melodia.
- Associação Temporal de Sequências: para esta situação, uma sequência de saída deve ser gerada em resposta a uma sequência de entrada. A sequência de entrada e saída podem ser bastante diferentes. Este caso inclui as duas classes anteriores como casos especiais.
- Produção de Sequências Temporais: neste caso, o sistema gera uma sucessão de estados entre dois pontos quaisquer não consecutivos, o ponto inicial e o ponto final. Este conceito esta estreitamente ligado ao de interpolação de estados.

Um tópico que deve ser comentado é como a dimensão temporal pode vir embutida em uma sequência temporal. De acordo com WANG & YUWONO (1995), ela pode vir de duas maneiras:

- Ordem Temporal: se as componentes de um padrão temporal são retirados de um alfabeto específico, a ordem temporal se refere à posição relativa destas componentes dentro da sequência.
- **Duração do tempo:** assumindo uma taxa de amostragem uniforme, a duração do tempo é inversamente proporcional à taxa de aprensentação (ou de observação) da sequência. A duração desempenha um papel crítico em algumas tarefas de processamento temporal, tanto no reconhecimento quanto na reprodução de padrões temporais.

Nesta dissertação, aborda-se o problema de Reconhecimento de Sequências no qual a dimensão temporal manifesta-se na ordem em que as sequências temporais são alimentadas à rede neural.

Para que seja possível explorar o tempo como variável do sistema são necessárias estruturas físicas que armazenem amostras passadas da sequência de padrões espaço-temporais de interesse. Essa estrutura é chamada de memória de curto prazo ou de curta duração. Utilizando o problema de navegação como exemplo, na perspectiva de um robô, vários locais no ambiente parecem iguais. Sendo assim, ele só será capaz de distingui-los se tiver conhecimento (memória) sobre onde já esteve (ZIEMKE, 1999). Dessa forma, o robô possui a capacidade de se localizar no ambiente e navegar em direção a algum objetivo.

Devido a sua importância para esta dissertação, na seção seguinte são apresentados mais detalhes sobre a memória, tanto do ponto de vista biológico como da área de Aprendizagem de Máquinas.

1.5 Memória de Curta Duração

Segundo LENT (2008), o termo *memória* se refere ao processo mediante o qual adquirimos, formamos, conservamos e evocamos informação. A fase de aquisição é coloquialmente chamada de "aprendizagem", enquanto a evocação recebe também as denominações expressão, recuperação e lembrança.

Ao converter a realidade em um complexo código de sinais elétricos e bioquímicos, os neurônios realizam uma tradução. Durante a evocação, ao reverterem essa informação para o meio que nos rodeia, os neurônios reconvertem esses sinais bioquímicos e estruturais em elétricos, de maneira que, novamente, os nossos sentidos e nossa consciência possam vir a interpretá-los como pertencentes ao mundo real.

Do ponto de vista funcional, existe a memória de trabalho, que mantém disponível a informação enquanto está sendo percebida e/ou processada, e não deixa traços. As memórias que persistem ao longo do tempo denominam-se memórias de curta e longa duração. A primeira é encarregada de manter a informação comportamentalmente disponível durante as horas que a memória de longa duração necessita para adquirir sua forma definitiva (LENT, 2008).

Segundo BARRETO (1998), para que se possa estabelecer associações entre padrões apresentados à rede em instantes de tempo distintos, é necessário que padrões anteriores permaneçam presentes, de alguma forma, no instante em que o padrão atual é apresentado. Desta forma,

pode-se criar mecanismos que passem a considerar a ordem temporal com que os padrões ocorrem. Costuma-se chamar esse tipo de mecanismo de modelos de Memória de Curta Duração (MCD).

Há duas abordagens para incorporar MCD em sistemas de processamento da informação:

- incorporar a MCD aos padrões de entrada, reorganizando-os de forma a levar em conta o histórico temporal dos eventos;
- incorporá-la diretamente na dinâmica de ativações (nível do neurônio) e/ou na aprendizagem (nível de sinapse).

Na primeira abordagem, amostras passadas da entrada do sistema são armazenadas em uma "linha de atraso" (*delay line*) finita, a qual alimenta o classificador com o vetor

$$\mathbf{x}_{n} = \begin{pmatrix} \mathbf{x}(n) \\ \mathbf{x}(n-1) \\ \vdots \\ \mathbf{x}(n-N) \end{pmatrix} = \begin{pmatrix} x_{0}(n) \\ \vdots \\ x_{d}(n-1) \\ \vdots \\ x_{d}(n-1) \\ \vdots \\ x_{0}(n-N) \\ \vdots \\ x_{p}(n-N) \end{pmatrix}$$

$$(1.4)$$

em que $\mathbf{x}(n) = [x_0(n) \dots x_p(n)]^T$ e \mathbf{x}_n é o novo vetor constituído pela concatenação dos vetores de características atuais e passados, N é a quantidade de amostras passadas que a linha de atraso apresenta ao classificador e d é a quantidade de informações disponíveis sobre o padrão a ser classificado. Dessa forma, a estrutura de MCD apresentada permite que o sistema trabalhe com várias amostras de uma única vez, como se a entrada fosse estática. Em outras palavras, transforma um fenômeno dinâmico em padrões estáticos multidimensionais (PRINCIPE et al., 2000).

Na segunda abordagem, as estruturas de memória são incorporadas dentro do sistema, neste caso explorado aqui, uma rede neural. Isto significa que no lugar de armazenar as amostras passadas dos padrões de entrada, é armazenado o histórico das ativações dos neurônios (PRINCIPE et al., 2000).

Na seção seguinte está a descrição o problema de interesse do projeto.

1.6 Definição dos Problemas de Interesse

Nesta dissertação, é levantada a hipótese sobre o quanto a dimensão temporal influencia na tarefa de navegação de robôs móveis, sendo introduzida por meio de mecanismos de MCD nos classificadores.

Na literatura, o uso de memória na Robótica, normalmente, é do tipo operacional, ou seja, o robô não armazena o que percebe sobre o mundo se tornando apenas reativo aos estímulos. Nos trabalhos que incorporam a informação temporal, as estruturas de memória deixam disponíveis no instante atual, vestígios dos estados sensoriais e/ou motores e das ativações dos neurônios em instantes passados, sendo que muitos desses trabalhos se inspiram em processos neurocognitivos (BARRETO, 1998). Como exemplos de trabalhos na área pode-se citar o trabalho de NEHMZOW & SMITHERS (1991) sobre localização baseada em memória episódica, BAK-KER et al. (2003) com a identificação e memorização de observações passadas, ANTONELO et al. (2008a) sobre a detecção de eventos e localização com redes neurais recorrentes.

Formulados como um problema de reconhecimento de padrões, podem ser citados os trabalhos em robótica de CAZANGI et al. (2003), MUCIENTES et al. (2009b), ANTONELO et al. (2008b), MALONE et al. (2005) e o de POMERLEAU (1993). Este último, desenvolveu um projeto chamado ALVINN, que consiste em um veículo guiado por uma rede neural, que recebe como entrada imagens obtidas por uma câmera no teto desse veículo, e que está voltada para a estrada. A saída da rede determina a direção do carro, mantendo-o na estrada e imitando um ser humano dirigindo. A sua rede neural, por não possuir nenhum mecanismo de memória de curto prazo, se torna um sistema puramente reativo aos estímulos. Isso significa que ele não tem nenhuma noção do aspecto temporal da tarefa, reagindo independente do contexto em que os estímulos estão inseridos. Dessa forma, o sistema não tem noção do ponto inicial da tarefa, nem para onde vai ou o que vai encontrar pela frente e nem navegar em direção a um objetivo.

Trabalhos de ANTONELO et al. (2008b) e MALONE et al. (2005) fazem uso de estruturas de memória, tendo o primeiro uma metodologia parecida com a adotada nesta dissertação. O diferencial do projeto descrito aqui, está na preocupação com o problema de reconhecimento de padrões, em outras palavras, o foco está no estudo do desempenho dos classificadores na categorização dos dados provenientes da navegação do robô. O problema de navegação neste trabalho passa a ser tratado como um problema de classificação de padrões e um questionamento muito importante surge: a navegação do tipo *wall-following* é linearmente separável ou não?

Um problema linearmente separável permite o uso de classificadores mais simples, uma vez que a decisão é tomada baseada no valor da combinação linear das informações de entrada $\mathbf{x} = (x_1 \dots x_p)^T$, ou seja

$$g_j(\mathbf{x}) = \mathbf{w}_j \cdot \mathbf{x} + w_{j0} = \sum_i w_{ji} x_i + w_{j0}, \quad j = 1, ..., C$$
 (1.5)

em que $\mathbf{w} \in \mathbb{R}^p$ é um vetor de parâmetros do modelo

A superfície de decisão gerada pela função discriminante mostrada na Equação 1.5 que, para $\mathbf{x} \in \mathbb{R}^2$, torna-se uma reta. Um exemplo desse tipo de superfície de decisão é mostrado na Figura 1.6

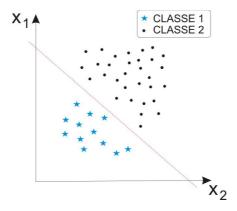


Figura 1.6: Exemplo hipotético de um problema linearmente separável.

Já para um problema não-linearmente separável, um hiperplano não consegue separar totalmente os padrões de classes diferentes, como mostrado na Figura 1.7.

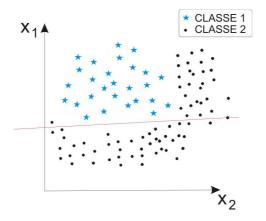


Figura 1.7: Exemplo de um problema não-linear tratado por um discriminante linear.

No entanto, se os padrões $\mathbf{x} \in S$ em que $S \subset \mathbb{R}^n$ sofrerem uma transformação $\phi(\mathbf{x}) = x_i$,

 $\operatorname{com} x_i \in \mathbb{R} \text{ e } \phi(.)$ uma função não-linear definida como

$$\phi(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + w_0), \tag{1.6}$$

então as classes podem ser separadas linearmente no espaço de ϕ (WEBB, 2002).

Com base nisso, a função discriminante não-linear pode ser representada pela seguinte expressão

$$g_j(\mathbf{x}) = \sum_{i=1}^q m_{ji}\phi_i(\mathbf{x}) + m_{j0}, \quad j = 1, ..., c.$$
 (1.7)

Na equação acima, m_{ji} são os parâmetros do classificador, q representa a quantidade de funções ϕ_i e c a quantidade de classes do problema. Assim, a regra do discriminante é dada por

$$\mathbf{x}$$
 pertence à classe k se $g_k(\mathbf{x}) = \max_j g_j(\mathbf{x}),$ (1.8)

em que **x** pertence a classe com o maior valor da função discriminante. Dessa forma, superfícies de decisão não-lineares podem ser formadas, como a da Figura 1.8.

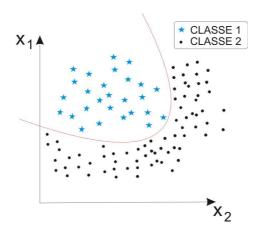


Figura 1.8: Exemplo de aplicação bem sucedida com um discriminante não-linear.

O questionamento levantado no começo desta seção motiva a aplicação de diferentes tipos de classificadores, com o objetivo de descobrir se um discriminante linear é o bastante para separar corretamente as classes do problema. Redes neurais clássicas como o perceptron simples e perceptron multicamadas são muito utilizadas em problemas de reconhecimento de padrões por sua capacidade de aprendizagem e generalização em diversas aplicações (HAYKIN, 1994; PRINCIPE et al., 2000). Essas características as tornam boas candidatas para o problema de navegação wall-following.

Nas próximas seções seguintes são descritos os objetivos, publicações e organização desta dissertação.

1.7 Objetivos da Dissertação

1.7.1 Objetivo Geral

Revisitar um dos problemas clássicos da robótica: a navegação seguir-paredes (*wall-following*) (BE-KEY, 2005), através da avaliação empírica desta tarefa como um problema de classificação de padrões.

1.7.2 Objetivos Específicos

Os objetivos específicos desta dissertação são enumerados a seguir.

- 1. Verificar a hipótese que a memória de curto prazo melhora o desempenho dos classificadores dentro da problemática mencionada. Sendo aquela introduzida de duas formas diferentes neste trabalho: para as redes de classificação estáticas, a memória é introduzida nas unidades de entrada através de um pequeno histórico das amostras que alimentam a rede, e no caso da rede neural dinâmica, a memória é intrínseca à própria arquitetura.
- 2. Realizar a comparação de diferentes paradigmas de aprendizagem e arquitetura de redes neurais artificiais, tais como:
 - redes lineares × não-lineares: como arquiteturas lineares, são exploradas as redes perceptron logístico e a arquitetura de mistura de especialistas. As não-lineares são representadas pelas redes perceptron multicamadas e a rede recorrente de Elman;
 - métodos de aprendizagem locais x globais: os métodos locais de aprendizagem são utilizados com a arquitetura de mistura de especialistas, cujos detalhes podem ser vistos no Capítulo 2, enquanto as outras redes utilizam métodos globais;
 - redes estáticas × redes dinâmicas: este caso é um dos mais explorados dentro deste projeto, uma vez que diz respeito ao desempenho das redes neurais, mencionadas anteriormente, com memória (redes dinâmicas) e sem (redes estáticas).

1.8 Produção Científica

Ao longo do desenvolvimento desta dissertação os seguintes artigos científicos foram publicados:

- Ananda L. Freire, Guilherme A. Barreto, Marcus Veloso e Antonio T. Varela (2009), "A
 Dimensão Temporal no Aprendizado de Tarefas de Navegação de Robôs Móveis Usando
 Redes Neurais Artificiais", aceito para publicação no IX Congresso Brasileiro de Redes
 Neurais / Inteligência Computacional (CBRN'2009), Ouro Preto-MG.
- 2. Ananda L. Freire, Guilherme A. Barreto, Marcus Veloso e Antonio T. Varela (2009), "Comparação de Algoritmos de Classificação de Padrões em Tarefas de Navegação de Robôs Móveis", aceito para publicação no IX Simpósio Brasileiro de Automação Inteligente (SBAI'2009), Brasília-DF.
- 3. Ananda L. Freire, Guilherme A. Barreto, Marcus Veloso e Antonio T. Varela (2009), "Short-Term Memory Mechanisms in Neural Network Learning of Robot Navigation Tasks: A Case Study", aceito para publicação no VI Latin American Robotics Symposium (LARS2009), Valparaíso-Chile.
- 4. Ananda L. Freire, José M. Menezes Jr. e Guilherme A. Barreto, "Redes Neurais Recorrentes para Predição Recursiva de Séries Temporais Caóticas: Um Estudo Comparativo", aceito para publicação no IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (CBRN'2009), Ouro Preto-MG.

1.9 Organização Geral da Dissertação

No Capítulo 2 é feito uma introdução sobre as redes neurais utilizadas neste trabalho, onde, na Seção 2.2 estão os detalhes sobre a arquitetura e aprendizagem da rede perceptron. Na Seção 2.3 descreve a rede perceptron multicamadas, na Seção 2.4 estão os detalhes sobre a arquitetura e algoritmo de aprendizagem da mistura de especialistas e por fim, na Seção 2.5, é descrito a rede recorrente de Elman.

No Capítulo 3 estão os resultados dos testes realizados com as redes neurais estáticas, ou seja, que não fazem uso de memória de curto prazo. Este capítulo incia-se com uma introdução sobre a unidade robótica utilizada neste projeto, na Seção 3.2, seguida pela descrição do programa baseado em heurísticas, na Seção 3.3, e a metodologia de coleta dos dados para o treinamento das redes, na Seção 3.4. A Seção 3.5 é divida em três subseções: 3.5.1, 3.5.2 e 3.5.3, nas quais estão descritos os detalhes dos resultados com as redes perceptron logístico, mistura de especialistas e perceptron multicamadas, respectivamente.

No Capítulo 4 estão os resultados dos testes com as redes dinâmicas. Nas Subseções 4.3.1, 4.3.2 e 4.3.3 mostram os resultados encontrados com as redes perceptron logístico dinâmico,

mistura de especialistas dinâmicos e rede recorrente de Elman, respectivamente. Na Seção 4.4 se encontra uma pequena discussão sobre as redes que foram bem sucedidas e seus custos computacionais.

No Capítulo 5, se encontram as conclusões do trabalho.

No Apêndice A estão os resultados dos testes com a rede MLP que não foram explorados no Capítulo 4. Nesse apêndice, estão os resultados dos testes com diferentes camadas ocultas e outras entradas.

No Apêndice B encontram-se os resultados da rede de Elman com diferentes camadas ocultas, como também diferentes quantidades de unidades de entrada.

No Apêndice C estão os resultados da rede perceptron logístico com e sem memória para os casos com 24 e 4 unidades de entrada.

Por fim, no Apêndice D, estão resultados semelhantes ao anterior, mas relacionados à arqutietura de mistura de especialistas.

2 REDES NEURAIS ARTIFICIAIS

2.1 Introdução

Neste capítulo, são discutidos alguns dos principais conceitos, modelos e propriedades de arquiteturas clássicas de redes neurais artificiais (RNA), tais como Perceptron e Perceptron Multicamadas; além da arquitetura de Mistura de Especialistas.

RNAs são modelos matemáticos inspirados nos mecanismos de processamento da informação dos neurônios biológicos, estes constituídos basicamente de dendritos, axônios e corpo celular. Ao se organizarem em redes, os neurônios podem formar diversas topologias capazes de criar um intricado sistema paralelo dotado da capacidade de aprendizagem interativa e adaptação. Uma arquitetura de RNA pode ser entendida como um processador maciçamente paralelo e distribuído, constituído de unidades de processamento simples¹, e que têm a habilidade intrínseca para armazenar informação e torná-la disponível para o uso, assemelhando-se ao cérebro em dois aspectos (HAYKIN, 1994):

- o conhecimento é adquirido pela rede por meio de dados coletados do ambiente, num processo de aprendizagem. Processo de treinamento envolve um Algoritmo de Aprendizagem, que tem como finalidade ajustar os pesos sinápticos da rede de uma forma ordenada para alcançar um objetivo desejado.
- 2. as conexões entre os neurônios, denotados por analogia de pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Em suma, RNAs são capazes de aprender padrões através de exemplos que lhe são apresentados, e reconhecê-los após o processo de aprendizagem, mesmo que a informação recebida esteja incompleta ou difusa, e também são capazes de generalizar e fazer associações (HAY-KIN, 1994; PRINCIPE et al., 2000; TSOUKALAS; UHRIG, 1997).

¹Simples do ponto de vista de processamento da informação, uma vez que um neurônio é visto simplesmente como uma chave *ON/OFF*.

2.1 Introdução 20

A origem do campo de estudo em RNAs remonta ao trabalho do psiquiatra W. McCulloch e do matemático W. Pitts, no qual propuseram, em 1943, o primeiro modelo matemático de um neurônio artificial (MCCULLOCH; PITTS, 1943), mostrado na Figura 2.1. Com um número suficiente de unidades básicas e conexões sinápticas adequadamente ajustadas, esses pesquisadores sugeriram que uma rede assim constituída poderia sintetizar, em princípio, qualquer função computável (HAYKIN, 1994). A maioria das arquiteturas de redes neurais utilizam o modelo de McCulloch-Pitts (M-P), ou variações deste, como unidade básica de processamento.

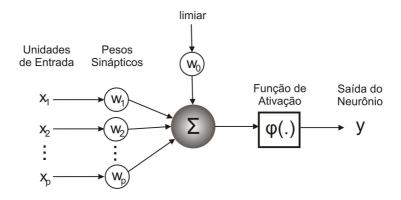


Figura 2.1: Modelo do neurônio artificial de McCulloch e Pitts.

De forma geral, RNAs podem ser divididas quanto ao tipo de aprendizado em duas categorias: (1) redes com aprendizado supervisionado e (2) redes com aprendizado não-supervisionado. No aprendizado supervisionado, cada entrada apresentada à rede possui uma saída desejada associada, de forma a permitir uma modificação dos parâmetros ajustáveis em função do erro entre a resposta fornecida pela rede e a saída real desejada. Ao final da etapa de ajuste dos parâmetros, chamada genericamente de treinamento, as respostas da rede para todos as entradas devem ser próximas das saídas desejadas. No caso não-supervisionado, a rede neural detecta padrões e características estatísticas do espaço de entrada, de forma a construir uma representação reduzida dos dados em seus pesos sinápticos.

As redes descritas neste trabalho utilizam aprendizagem supervisionada, sendo estas: a rede Perceptron Simples Logístico, Perceptron Multicamadas, Mistura de Especialistas com Perceptrons Simples Logísticos e rede neural recorrente de Elman. Estas diferenciam-se, entre si, na capacidade de tratar problemas não-linearmente separáveis, por utilizar métodos de aprendizagem locais ou globais², ou pelo uso ou não de conexões recorrentes.

²Métodos locais capturam sub-estruturas do mapeamento, e métodos globais capturam a estrutura subjacente do mapeamento como um todo.

2.2 Rede Perceptron Simples

A rede perceptron simples é a arquitetura mais básica de rede neural artificial e foi proposto por Rosenblatt (1958). Esta rede é normalmente utilizada para classificação de padrões linearmente separáveis, ou seja, padrões distribuídos em classes que podem ser eficientemente separadas por um hiperplano. A rede PS É constituída de uma única camada de neurônios, mostrada na Figura 2.2, e pode ser subcategorizada de acordo com a sua função de ativação: se esta for a soma ponderada das entradas, tem-se a arquitetura MADALINE; caso seja uma função degrau ou sinal, tem-se perceptron simples (PS); e caso seja uma função logística, tem-se perceptron simples logístico (ou apenas perceptron logístico, PL). Um conjunto de vários neurônios M-P pode ser organizado em camadas, formando outra arquitetura conhecida como perceptron multicamadas, a ser discutida na Seção 2.3.

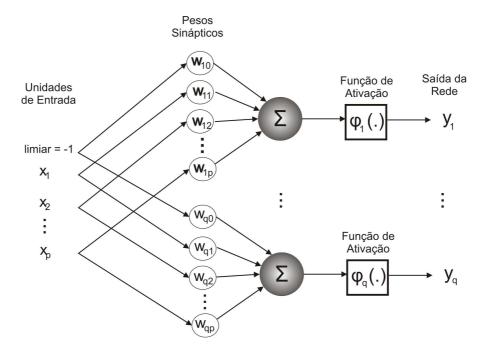


Figura 2.2: Ilustração da arquitetura da rede neural perceptron simples.

2.2.1 Arquitetura da rede PS

Os dados de entrada fornecem a informação disponível sobre o problema em questão. Esta informação é fornecida através de um conjunto de N pares de vetores $\{\mathbf{x}(n), \mathbf{d}(n)\}$, n = 1, 2, ..., N, em que $\mathbf{x}(n) \in \mathbb{R}^{p+1}$ simboliza o vetor de entrada no instante de tempo discreto n e $\mathbf{d}(n)$ denota o vetor de saídas (respostas) desejadas para aquele vetor de entrada. Cada vetor de

entrada é representado como

$$\mathbf{x}(n) = \begin{pmatrix} x_0(n) \\ x_1(n) \\ \vdots \\ x_j(n) \\ \vdots \\ x_p(n) \end{pmatrix} = \begin{pmatrix} -1 \\ x_1(n) \\ \vdots \\ x_j(n) \\ \vdots \\ x_p(n) \end{pmatrix}$$
(2.1)

onde p > 0 representa a quantidade de variáveis de entrada, e $x_0(n) = -1$ é uma componente fixa com o objetivo de ajustar os limiares das ativações dos neurônios. O vetor de resposta desejada $\mathbf{d}(n)$ no instante n pode ser representado como

$$\mathbf{d}(n) = \begin{pmatrix} d_1(n) \\ \vdots \\ d_i(n) \\ \vdots \\ d_q(n) \end{pmatrix}$$
 (2.2)

em que q > 0 indica a quantidade de variáveis de saída da rede neural. Normalmente, na rede Perceptron, q também refere-se a quantidade de neurônios, ou seja, $d_i(n)$ é a saída desejada do i-ésimo neurônio.

O vetor de pesos w associado ao i-ésimo neurônio é representado por

$$\mathbf{w}_{i} = \begin{pmatrix} w_{i0} \\ w_{i1} \\ \vdots \\ w_{ij} \\ \vdots \\ w_{ip} \end{pmatrix} = \begin{pmatrix} \theta_{i} \\ w_{i1} \\ \vdots \\ w_{ij} \\ \vdots \\ w_{ip} \end{pmatrix}$$

$$(2.3)$$

em que w_{ij} é o peso sináptico que conecta a j-ésima entrada ao i-ésimo neurônio e θ_i define um limiar (threshold ou bias) de ativação. Cada neurônio da rede possui seu próprio vetor de pesos \mathbf{w}_i , resultando em uma matriz de pesos de dimensão $(p+1) \times q$, formada pelos pesos sinápticos w_{ij} e limiares θ_i , os quais são parâmetros ajustáveis que devem ser modificados por meio de uma regra de atualização recursiva denominada Regra de Aprendizagem do Perceptron.

2.2.2 Regras de Ativação e Aprendizagem

Após a apresentação de um vetor de entrada \mathbf{x} , na iteração n, a ativação u_i do i-ésimo neurônio é calculada de acordo como,

$$u_{i}(n) = \sum_{j=1}^{p} w_{ij}(n)x_{j}(n) - \theta_{i}$$

$$= \sum_{j=1}^{p} w_{ij}(n)x_{j}(n) + w_{i0}(n)x_{0}(n)$$

$$= \sum_{j=0}^{p} w_{ij}(n)x_{j}(n)$$

$$= \mathbf{w}_{i}(n)^{T}\mathbf{x}(n)$$
(2.4)

onde o sobrescrito T indica a operação de transposição de vetores.

A saída atual da rede PS pode ser dada por .

$$y_i(n) = sinal(u_i(n)) = \begin{cases} +1, & u_i(n) \ge 0 \\ -1 & u_i(n) < 0 \end{cases}$$
 (2.5)

$$y_i(n) = degrau(u_i(n)) = \begin{cases} 1, & u_i(n) \ge 0 \\ 0 & u_i(n) < 0 \end{cases}$$
 (2.6)

representadas na Figura 2.3.

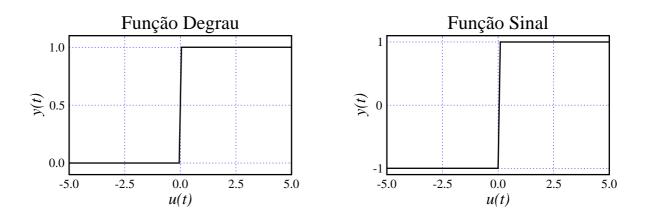


Figura 2.3: Função de ativação Degrau e Sinal.

A rede fornece uma saída binária $\{0,1\}$ ou bipolar $\{-1,+1\}$. Em virtude disto, esta rede é normalmente empregada em problemas de classificação de padrões.

Os pesos são ajustados de forma que a saída $y_i(n)$ gerada seja próxima da saída desejada

 $d_i(n)$ associada a entrada $\mathbf{x}(n)$. No algoritmo utilizado, somente os padrões classificados erroneamente contribuem para a atualização dos pesos.

A título de derivação teórica, considera-se apenas um neurônio de saída, de modo que o problema de classificação envolve apenas duas classes (classe 1: ω_1 e classe 2: ω_2). Por existir apenas um neurônio, o vetor de pesos deste é simplesmente representado por w, eliminando o índice i. Para a análise que se segue, adota-se a função sinal como função de transferência.

Cada vetor de entrada x é redefinido de modo que este passe a ser representado como

$$\mathbf{z} = \begin{cases} \mathbf{x}, & \text{se } \mathbf{x} \in \omega_1, \\ -\mathbf{x} & \text{se } \mathbf{x} \in \omega_2 \end{cases}$$
 (2.7)

O propósito da definição acima é fazer com que o produto $\mathbf{w}^T\mathbf{z}$ seja sempre positivo para todo \mathbf{z} . Por definição, se $\mathbf{w}^T\mathbf{z} > 0$ para todo \mathbf{z} disponível, então o problema de classificação é dito ser linearmente separável. Contudo, na prática, mesmo que não se consiga obter $\mathbf{w}^T\mathbf{z} > 0$ para todos os vetores de entrada, busca-se uma solução para que w produza $\mathbf{w}^T\mathbf{z} > 0$ para tantos vetores de entrada quanto possível. Busca-se, por tanto, minimizar o erro de classificação dos vetores de entrada. Isto posto, a função custo a ser minimizada a cada passo do treinamento é representada por

$$J[\mathbf{w}] = \sum_{\mathbf{z}_k \in \mathbb{Z}} (-\mathbf{w}^T \mathbf{z}_k)$$
 (2.8)

em que \mathbf{z}_k denota o k-ésimo vetor de entrada mal-classificado (i.e. classificado erroneamente) e \mathbb{Z} é o conjunto dos vetores mal-classificados. Como J é uma função contínua, faz-se uso do método iterativo do gradiente descendente para derivar a regra de ajuste recursivo do vetor de pesos \mathbf{w} , mostrado abaixo:

$$\mathbf{w}^{novo} = \mathbf{w}^{atual} + \Delta \mathbf{w}$$

$$= \mathbf{w}^{atual} - \alpha \frac{\partial J[\mathbf{w}]}{\partial \mathbf{w}}.$$
(2.9)

A expressão \mathbf{w}^{novo} indica o vetor de pesos já ajustado e α ($0 < \alpha \ll 1$) é chamada de passo ou taxa de aprendizagem. A derivada $\frac{\partial J[\mathbf{w}_i]}{\partial \mathbf{w}_i}$ é dada por

$$\frac{\partial J[\mathbf{w}]}{\partial \mathbf{w}} = \sum_{\mathbf{z}_k \in \mathbb{Z}} (-\mathbf{z}_k), \tag{2.10}$$

que é a soma dos vetores mal-classificados. Substituindo este resultado na Equação (2.9), temos

$$\mathbf{w}^{novo} = \mathbf{w}^{atual} + \alpha \sum_{\mathbf{z}_k \in \mathbb{Z}} \mathbf{z}_k \tag{2.11}$$

O tipo de treinamento que usa a regra de aprendizagem monstrada na Equação 2.11, algumas vezes chamada de treinamento época-à-época ou em lote (*batch*), uma vez que todos os vetores mal-classificados são usados ao mesmo tempo para atualizar w. Outra forma muito utilizada é realizar a atualização logo que um erro de classificação ocorre. Este procedimento é comumente chamado de treinamento sequencial. Neste caso, a regra de aprendizagem passa a ser escrita como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha \mathbf{z}(n). \tag{2.12}$$

em que *n* denota o instante de apresentação do vetor de entrada **z**

Em termos da notação original, a regra de aprendizagem mostrada na Equação (2.12) pode ser escrita como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha e(n)\mathbf{x}(n). \tag{2.13}$$

em que e(n) = d(n) - y(n) corresponde ao erro de classificação do vetor de entrada mathbfx(n).

Para o caso em que há q neurônios, a regra de ajuste do vetor de pesos do i-ésimo neurônio é dada por

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \alpha e_i(n)\mathbf{x}(n). \tag{2.14}$$

em que $e_i(n) = d_i(n) - y_i(n)$ corresponde ao erro de classificação do *i*-ésimo neurônio.

2.2.3 Perceptron Simples Logístico

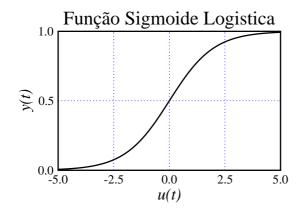
Esta rede é uma variante da rede PS em que se faz uso de uma função de ativação sigmoidal, representada por $\phi(.)$, tal como a função Sigmóide Logística,

$$y_i(n) = \phi_i(n) = \frac{1}{1 + \exp(-u_i(n))},$$
 (2.15)

pertencente ao intervalo (0,1) e a Tangente Hiperbólica

$$y_i(n) = \phi_i(n) = \frac{1 - \exp(u_i(n))}{1 + \exp(-u_i(n))},$$
 (2.16)

pertencente ao intervalo (-1,1). Os gráficos destas funções estão mostrados na Figura 2.4.



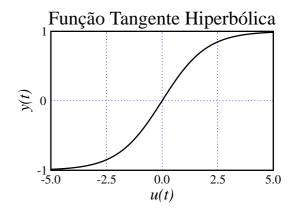


Figura 2.4: Função de ativação Sigmóide Logística e Tangente Hiperbólica

Por serem funções contínuas, a derivada $\frac{\partial J[\mathbf{w}_i]}{\partial \mathbf{w}_i}$ é encontrada agora a partir da regra da cadeia, ou seja,

$$\frac{\partial J(n)}{\partial w_{ij}(n)} = \frac{dJ(n)}{de_i(n)} \frac{de_i(n)}{dy_i(n)} \frac{dy_i(n)}{du_i(n)} \frac{du_i(n)}{dw_{ij}(n)}.$$
(2.17)

Em que

$$\frac{dJ(n)}{de_i(n)} = e_i(n) \tag{2.18}$$

$$\frac{dJ(n)}{de_i(n)} = e_i(n)$$

$$\frac{de_i(n)}{dy_i(n)} = -1$$
(2.18)

$$\frac{dy_i(n)}{du_i(n)} = \frac{d\phi(u_i(n))}{du_i(n)} = \phi'_i(n)$$
(2.20)

$$\frac{du_i(n)}{dw_{ij}(n)} = x_j(n) \tag{2.21}$$

na qual, $\phi_i'(n)$ é a derivada da função de ativação $\phi_i(u_i(n))$ em relação à ativação $u_n(n)$. Fazendo com que a regra recursiva do ajuste dos pesos, w_{ij} , seja dada por

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n)$$

$$= w_{ij}(n) - \alpha \frac{\partial J(n)}{\partial w_{ij}(n)}$$

$$= w_{ij}(n) + \alpha x_j(n) e_i(n) \phi_i'(n)$$

$$= w_{ij}(n) + \alpha \delta_i(n) x_j(n)$$

$$(2.22)$$

em que δ_i é o gradiente local do i-ésimo neurônio. A derivada $\phi_i'(n)$ para a função Sigmóide Logística é dada por

$$\phi_i'(n) = \phi_i(n)[1 - \phi_i(n)] = y_i(n)(1 - y_i(n))$$
(2.23)

e para a função tangente Hiperbólica é dada por

$$\phi_i'(n) = 1 - \phi_i^2(n) = 1 - y_i^2(n)$$
(2.24)

Essa regra de aprendizagem é chamada de Regra LMS Generalizada e é mais estável do que a regra LMS original, porque a derivada $\phi[u_i(n)]$ atenua os efeitos causados por elevados valores do erro e_i , em geral causados por elevados valores da ativação $u_i(n)$ (PRINCIPE et al., 2000). Além disso, permite que a rede possa ser aplicada não só em problemas de classificação, como também em problemas de aproximação de funções, uma vez que suas ativações não assumem mais apenas valores binários. A desvantagem é a convergência mais lenta em relação à regra original.

2.3 Rede Perceptron Multicamadas

A rede perceptron multicamadas (*Multilayer Perceptron*, MLP), é uma generalização da rede PS. A rede MLP possui uma ou mais camadas ocultas de neurônios em sua arquitetura, enquanto a rede PS possui apenas a camada de saída. Por esta característica, a rede MLP se enquadra na categoria dos algoritmos não-lineares, e desta forma está preparada para solucionar problemas não-linearmente separáveis.

Os neurônios pertencentes à camada oculta desempenham um papel crucial na operação da rede MLP porque agem como detectores de características (HAYKIN, 1994). Conforme o processo de aprendizagem avança, esses neurônios começam gradualmente a "descobrir" as características salientes presentes nos dados de treinamento. Realizam uma transformação não-linear dos dados de entrada para um novo espaço, chamado espaço oculto ou espaço de características. Neste novo espaço, as classes de interesse em uma tarefa de classificação de padrões podem ser mais facilmente separadas entre si do que no espaço original de entrada (HAYKIN, 1994).

Tipicamente, uma rede MLP é constituída de uma camada de entrada que recebe os sinais, uma ou mais camadas escondidas ou ocultas, compostas por neurônios com função de ativação não-linear e uma camada de saída, também composta por um ou mais neurônios somadores (que podem ser lineares ou não). Os neurônios da camada intermediária são chamados de neurônios ocultos ou escondidos por não terem acesso direto à saída da rede MLP. É na camada de saída que são calculados os sinais de erro necessários para guiar o ajuste dos pesos. A Figura 2.5 exemplifica a arquitetura de uma MLP com uma única camada de saída.

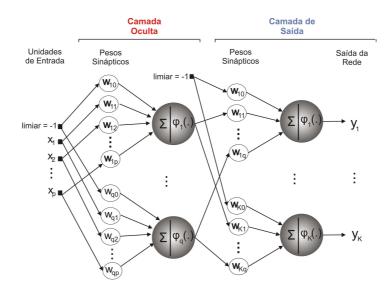


Figura 2.5: Ilustração da rede perceptron multicamadas com uma camada oculta

O algoritmo utilizado para treinar esta rede é chamado de algoritmo de retropropagação do erro (*Error Backpropagation*). Possui este nome devido ao fato de propagar o sinal de erro a partir da saída em direção a todos os neurônios ocultos, percorrendo o caminho inverso do processamento dos dados das entradas. Foi desenvolvido por WERBOS (1994) e redescoberto independentemente por PARKER (1982) e RUMELHART et al. (1986) e possui duas fases distintas.

Na primeira fase, o fluxo do sinal vai da entrada para a saída, propagando-se para frente (sentido direto). Nesse momento, são calculadas as ativações e saídas dos neurônios de todas as camadas ocultas e de saída. Na segunda fase, os gradientes locais são propagados para trás (camada por camada), atualizando os pesos.

O primeiro passo antes de executar as fases mencionadas acima, é definir a arquitetura. O tamanho do vetor de entradas depende de quantas características o objeto possui, mais a unidade de limiar. A camada oculta é definida empiricamente, tendo em conta que utilizando unidades demais pode levar a rede a memorizar os exemplos de treinamento. Este problema é chamado de sobreajustamento (*overfitting*); por outro lado, com pequeno número pode ocorrer o subajustamento (*underfitting*), em que os pesos da rede convergem, mas por conta dos poucos neurônios ocultos, ela não consegue aprender o problema corretamente. A camada de saída é, usualmente, a quantidade de classes nas quais o objeto pode ser classificado.

O vetor de pesos associado a cada neurônio i da camada oculta é representado da mesma forma mostrada na Equação (2.3), em que θ_i é o limiar associado ao i-ésimo neurônio de entrada; i=1,2,...,q, sendo q o número de neurônios na camada oculta. O vetor associado à

camada de saída é dado por

$$\mathbf{m}_{k} = \begin{pmatrix} m_{k0} \\ m_{k1} \\ \vdots \\ m_{kj} \\ \vdots \\ m_{kp} \end{pmatrix} = \begin{pmatrix} \theta_{k} \\ m_{k1} \\ \vdots \\ m_{kj} \\ \vdots \\ m_{kp} \end{pmatrix}$$

$$(2.25)$$

em que a denominação m é dada para diferenciar os pesos da camada oculta da de saída, e tal que θ_k é o limiar associado ao k-ésimo neurônio de saída; k=1,...,M, sendo M o número de neurônios na camada de saída.

2.3.1 Propagação no sentido direto

Nesta fase, é analisado o fluxo dos dados da entrada em direção à camada oculta e depois para a camada de saída. Após a apresentação de um vetor de entrada \mathbf{x} , na iteração t, o primeiro passo é calcular as ativações dos neurônios da camada oculta como

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n) x_j(n) = \mathbf{w}_i^T(n) \mathbf{x}(n).$$
 (2.26)

Em seguida, calculam-se as saídas dos neurônios ocultos de acordo com

$$s_i(n) = \phi[u_i(n)] = \phi\left[\sum_{j=0}^p w_{ij}(n)x_j(n)\right] = \phi[\mathbf{w}_i^T(n)\mathbf{x}(n)], \qquad (2.27)$$

em que ϕ é a função de ativação, geralmente a função sigmóide logística, ou a função tangente hiperbólica. Em seguida, calculam-se as ativações dos neurônios da camada de saída

$$u_k(n) = \sum_{i=0}^{q} m_{ki}(n) s_i(n) = \mathbf{m}_k^T(n) \mathbf{s}(n),$$
 (2.28)

em que k = 1,...,M, com M representando a quantidade total de neurônios de saída. Dessa forma, a saída da rede é dada por

$$y_k(n) = \phi[u_k(n)] = \phi\left[\sum_{i=0}^q m_{ki}(n)s_i(n)\right] = \phi[\mathbf{m}_k^T(n)\mathbf{s}(n)], \qquad (2.29)$$

tal que a função de ativação $\phi(.)$ é geralmente uma função sigmoidal ou uma função linear.

2.3.2 Propagação no sentido inverso

Nesta fase são realizados os cálculos dos gradientes locais e o ajuste dos pesos dos neurônios de todas as camadas. Para isso, o fluxo de informações, dessa vez, se propaga da camada de saída em direção à entrada. Daí vem o termo *backpropagation* do nome do algoritmo de aprendizagem.

O primeiro passo consiste em calcular os gradientes locais dos neurônios da camada de saída de acordo com

$$\delta_k(n) = e_k(n)\phi_k'[u_k(n)], \quad k = 1,...,M.$$
 (2.30)

O termo $e_k(n)$ representa o erro do neurônio de saída, ou seja, a diferença entre a resposta desejada $d_k(n)$ e a saída da rede $y_k(n)$,

$$e_k(n) = d_k(n) - y_k(n), k = 1,...,M.$$
 (2.31)

A derivada $\phi'[u_k(n)]$ pode assumir uma entre duas formas de acordo com a função de ativação do neurônio oculto. Caso seja uma função sigmóide logística, tem-se a seguinte expressão:

$$\phi_k'(n) = \phi_k(n)[1 - \phi_k(n)] = y_k(n)[1 - y_k(n)]$$
(2.32)

Caso seja uma função tangente hiperbólica, a derivada $\phi'_k(n)$ é dada por

$$\phi_k'(n) = 1 - \phi_k^2(n) = 1 - y_k^2(n). \tag{2.33}$$

Em seguida, calculam-se os gradientes dos neurônios ocultos por meio da seguinte expressão:

$$\delta_i(n) = \phi_i'[u_i(n)] \sum_{k=1}^M m_{ki}(n) \delta_k(n), \quad i = 1, ..., q.$$
 (2.34)

Da mesma forma que as derivadas da função da camada de saída, as da camada oculta também dependem do tipo de função de ativação. No caso da sigmóide logística, tem-se

$$\phi_i'(n) = \phi_i(n)[1 - \phi_i(n)] = s_i(n)[1 - s_i(n)], \tag{2.35}$$

e no caso da tangente hiperbólica, tem-se

$$\phi_i'(n) = 1 - \phi_i^2(n) = 1 - s_i^2(n). \tag{2.36}$$

Após os cálculos dos gradientes locais, é realizada a atualização dos pesos. Para a arquite-

tura MLP com uma camada oculta, a regra para o ajuste dos pesos é dada por

$$w_{ij}(n+1) = w_{ij}(n+1) + \Delta w_{ij}(n) = w_{ij}(n+1) + \alpha \delta_i(n) x_j(n)$$
 (2.37)

A taxa de aprendizagem α , da mesma forma que na rede PS, é uma constante compreendida entre $0 < \alpha < 1$. A regra para a atualização dos pesos da camada de saída é dada por

$$m_{ki}(n+1) = m_{ki}(n+1) + \Delta m_{ki}(n)$$

= $m_{ki}(n+1) + \alpha \delta_k(n) s_i(n)$ (2.38)

2.4 Mistura de Especialistas

Durante o treinamento, se uma única rede neural deve realizar diferentes subtarefas em momentos distintos, haverá, geralmente, grande interferência por parte dos erros, os quais levam a um aprendizado lento e generalização pobre (JACOBS et al., 1991). Essa interferência pode ser reduzida tendo em consideração que se há diferentes tarefas embutidas no conjunto de dados para treinamento, então existem subconjuntos de dados que correspondem a cada uma delas. Dessa forma, uma arquitetura composta de um certo número de redes neurais como especialistas aliadas a um mecanismo seletor, que decide qual desses especialistas deve ser usado em cada caso de treinamento, pode levar a uma melhor performance (JACOBS et al., 1991). Esse tipo de arquitetura modular é chamada de mistura de especialistas (ME).

Fazer uso da arquitetura ME é uma forma de combinar as vantagens que métodos locais e globais possuem (HAYKIN, 1994). Os métodos locais oferecem aprendizagem rápida com a capacidade para operar em tempo real, enquanto os métodos globais oferecem como vantagens uma necessidade menor de armazenamento e melhor generalização. A idéia foi originalmente proposta por Jacobs e Hinton, em 1988 na *Connectionist Summer School* em Pittsburgh (JACOBS et al., 1991)

De particular interesse para esta dissertação, o trabalho de TANI & NOLFI (1999) envolve uma aplicação de mistura de especialistas em robótica móvel para coordenação sensóriomotora. Nessa aplicação, várias módulos de RNAs competem para se tornarem especialistas na predição do fluxo sensório-motor de um determinado comportamento. O trabalho de TANI & NOLFI (1999) motivou o uso da arquitetura ME nos testes desenvolvidos nesta dissertação.

2.4.1 Vantagens da Arquitetura Modular

Algumas das vantagens de arquiteturas modulares, tal como mistura de especialistas, reside na sua velocidade de aprendizagem, que é maior em problemas nos quais existe uma decomposição natural da tarefa em funções mais simples. Isso se deve à capacidade deste tipo de arquitetura de dividir o espaço de entrada entre as diversas redes que a compõe. Uma outra vantagem que a modularização oferece é a facilidade de entender melhor a representação dos dados de entrada uma vez que essa passa a ser dividida em tarefas mais simples (HAYKIN, 1994).

A título de curiosidade, BALLARD (1986) afirma que, do ponto de vista neurobiológico, existe uma limitação física em relação à quantidade de neurônios que pode-se acomodar em determinados espaços no cérebro. Dessa forma, tarefas que requerem um espaço de alta dimensionalidade passam a ser distribuídas em diferentes áreas do cérebro, que computam funções diferentes. OLIVEIRA & AMARAL (1997) enfatizam que os processos mentais humanos são sistemas complexos e que não estão localizados em áreas restritas, mas sim, que acontecem pela atuação harmoniosa de diversas áreas, cada uma delas oferecendo sua contribuição particular. Assim, pode-se fazer uma analogia às redes modulares artificiais.

2.4.2 Descrição da Arquitetura de Mistura de Especialistas

A arquitetura de mistura de especialistas utiliza métodos probabilísticos para dividir o espaço de entrada em regiões que se sobrepõem e nas quais os especialistas atuam. Além de uma rede selecionadora, o seletor, que pondera as saídas desses especialistas e forma a saída combinada da rede.

Essa arquitetura também mistura paradigmas de aprendizagem supervisionada e não-supervisionada de uma forma intrínseca (HAYKIN, 1994). O aprendizado supervisionado ocorre em função das saídas desejadas necessárias para treinar os diferentes módulos da rede. No entanto, as saídas não especificam qual módulo deve produzir cada resposta desejada. A função de determinar qual das redes deve fornecê-la cabe ao aprendizado não-supervisionado. Este último, promove uma competição dos módulos entre si pelo direito de responder a uma determinada entrada e o seletor tem a tarefa de mediar a competição entre os diferentes módulos, ver Figura 2.6.

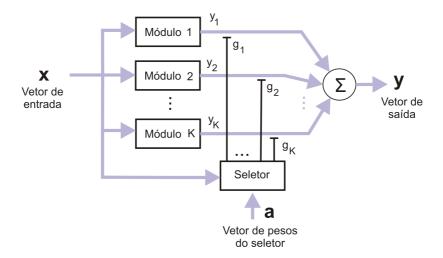


Figura 2.6: Diagrama de blocos da Mistura de Especialistas, onde as saídas dos módulos especialistas são mediadas através do seletor.

A arquitetura ME possui K módulos, cada um representando uma rede neural especialista, além de uma rede que media esses módulos, chamada seletor. Seja o conjunto de dados para treinamento representado pelo vetor $\mathbf{x} \in \mathbb{R}^p$ e a saída desejada representada pelo vetor $\mathbf{d} \in \mathbb{R}^q$. O seletor é formado por uma rede de única camada possuindo tantos neurônios quanto o número de especialistas, no caso, K neurônios.

O vetor de entrada \mathbf{x} é apresentado a todos os módulos e ao seletor simultaneamente, gerando o vetor $\mathbf{y}_i \in \mathbb{R}^q$, que é o vetor de saída do i-ésimo especialista. Uma observação a ser feita é que o seletor pode ter entradas diferentes para facilitar a escolha dos especialistas (JACOBS et al., 1991). Seja g_i a ativação do i-ésimo neurônio do seletor e \mathbf{y} a saída do sistema, de dimensão q, descrita como

$$\mathbf{y} = \sum_{i=1}^{K} g_i \mathbf{y}_i. \tag{2.39}$$

Observe que as saídas y_i de todos os especialistas contribuem, ainda que em proporções diferentes, para a saída final do sistema.

O objetivo do algoritmo de treinamento desta arquitetura é modelar a distribuição de probabilidade dos padrões do conjunto de treinamento, $\{\mathbf{x}, \mathbf{d}\}$, utilizando sua estrutura modular (HAYKIN, 1994). O seletor escolhe a saída \mathbf{y} dentre as K fornecidas pelos especialistas, com base na probabilidade de cada saída \mathbf{y}_i , com i = 1, ..., K, dada pelo valor de g_i .

A probabilidade (ou ativação) g_i , do i-ésimo neurônio do seletor, é dada por

$$g_i = \frac{\exp u_i}{\sum_{j=1}^K \exp\left\{u_j\right\}}$$
 (2.40)

em que $u_i = \mathbf{a}_i^T \mathbf{x}$. Os diferentes g_i devem obedecer às restrições

$$0 < g_i < 1 \tag{2.41}$$

e

$$\sum_{i=1}^{K} g_i = 1. (2.42)$$

A Equação (2.40) pode ser vista como uma generalização da função logística no caso de múltiplas entradas, diferenciando-se por ser normalizada e obedecer às restrições já comentadas acima (PRINCIPE et al., 2000). Essa equação é conhecida como *softmax* e permite que as saídas sejam consideradas como estimativas a posteriori, $P(i|\mathbf{x})$, de que uma determinado vetor \mathbf{x} de entrada pertença à alguma classe. É interessante notar que a normalização feita pode ser considerada como uma competição, porque se a ativação u_i for grande, as outras ativações u_j com $j \neq i$ são forçadas a diminuírem suas influências, uma vez que a soma tem valor restrito (PRINCIPE et al., 2000).

O aprendizado da arquitetura ME é feito a partir de um treinamento simultâneo dos especialistas e do seletor. Para tanto, a seguinte sequência de passos deve ser obedecida.

- 1. Um vetor de entrada **x** é selecionado;
- 2. um módulo é escolhido da distribuição $P(i|\mathbf{x})$, que é a probabilidade de um especialista i seja escolhido, dado uma entrada \mathbf{x} ;
- 3. gera-se um vetor de resposta desejada \mathbf{d} , de dimensão q, selecionando um módulo i de acordo com o processo regressivo

$$\mathbf{d} = \mathbf{F}_i(\mathbf{x}) + \varepsilon_i, \tag{2.43}$$

com i = 1, 2, ..., K, em que $\mathbf{F}_i(\mathbf{x})$ é uma função determinística do vetor de entrada \mathbf{x} , e ε_i é um vetor aleatório que representa erros de modelagem. Para simplificar, assume-se que esse vetor tem distribuição gaussiana com média zero e matriz de covariância $\sigma^2 \mathbf{I}$, com \mathbf{I} denotando a matriz identidade e σ^2 é a variância, a qual pode-se simplificar ainda mais e considerá-la $\sigma^2 = 1$;

- 4. modifica-se os pesos sinápticos da rede especialista escolhida, e
- 5. simultaneamente modifica-se os pesos do seletor.

O vetor de saída de cada módulo é modelado como como média condicional de uma distribuição gaussiana multivariada, ou seja,

$$\mathbf{y}_i = \mu_i \tag{2.44}$$

$$\mu_i = E[\mathbf{d}|\mathbf{x}, i] = \mathbf{F}_i(\mathbf{x}). \tag{2.45}$$

Onde μ_i é a média condicional da resposta desejada **d** dado um vetor **x** e que o especialista i foi escolhido para ser treinado.

Normalmente, as saídas dos diferentes especialistas são correlacionadas, sendo a matriz de covariância de \mathbf{y}_i a mesma de ε_i . Mas para simplificar, assume-se para todos os K especialistas

$$\varepsilon_1 = \varepsilon_2 = ... = \varepsilon_K$$

e a matriz de covariância de ε_i é a matriz identidade. Sendo assim, a função densidade de probabilidade (FDP) gaussiana multivariada do vetor de resposta desejada \mathbf{d} , dado o vetor de entrada \mathbf{x} e escolhido o i-ésimo módulo, é expressa como

$$f(\mathbf{d}|\mathbf{x},i) = \frac{1}{(2\pi)^{q/2}} \exp\left\{-\frac{1}{2}\|\mathbf{d} - \mathbf{y}_i\|^2\right\},\tag{2.46}$$

em que ||.|| denota a norma euclidiana.

Pensando na Equação (2.46) como uma FDP condicional, pode-se tratar a densidade de probabilidade de um vetor resposta desejada **d** como um modelo "misturado", ou seja, como uma combinação linear de *K* diferentes densidade gaussianas multivariadas, ou seja,.

$$f(\mathbf{d}|\mathbf{x}) = \sum_{i=1}^{K} g_i f(\mathbf{d}|\mathbf{x}, i) = \frac{1}{(2\pi)^{q/2}} \sum_{i=1}^{K} g_i \exp\left\{-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i\|^2\right\}.$$
 (2.47)

Pode-se, ainda, ver a FDP condicional $f(\mathbf{d}|\mathbf{x})$ como uma função de verossimilhança, tendo o vetor dos pesos sinápticos de todos os especialistas, \mathbf{w}

$$\mathbf{w} = [\mathbf{w}_1 \ \mathbf{w}_2 \dots \mathbf{w}_K]^T, \tag{2.48}$$

e o vetor de ativação g

$$\mathbf{g} = [g_1 \ g_2 \ \dots \ g_K]^T, \tag{2.49}$$

como parâmetros desconhecidos.

Em formulações como a descrita pela Equação (2.47), é preferível trabalhar com o logaritmo natural de $f(\mathbf{d}|\mathbf{x})$ do que com $f(\mathbf{d}|\mathbf{x})$ simplesmente (JACOBS et al., 1991). Assim, define-se a função log-verossimilhança como

$$l(\mathbf{w}, \mathbf{g}) = \ln f(\mathbf{d}|\mathbf{x}). \tag{2.50}$$

Substituindo Equação (2.47) em (2.50), e ignorando o termo constante $-\ln(2\pi)^{q/2}$, pode-se expressar a função do logaritmo da verossimilhança $l(\mathbf{w}, \mathbf{g})$ como

$$l(\mathbf{w}, \mathbf{g}) = \ln \sum_{i=1}^{K} g_i \exp\left\{-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i\|^2\right\}$$
 (2.51)

A função $l(\mathbf{w}, \mathbf{g})$ pode ser vista como uma função objetivo. A maximização desta fornece a estimativa de máxima verossimilhança dos parâmetros livres da arquitetura ME, que são os pesos das redes especialistas e da rede seletora (HAYKIN, 1994). Nessa perspectiva pode-se concluir dois pontos a respeito desses parâmetros:

- 1. os vetores de saída $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K$ dos módulos otimizados são vetores de médias condicionais desconhecidas;
- 2. as saídas otimizadas do seletor, $g_1, g_2, ..., g_K$ são as probabilidades condicionais a priori de que os respectivos módulos geraram o padrão de entrada atual, e que são calculados como mostrado na Seção 2.4.2.

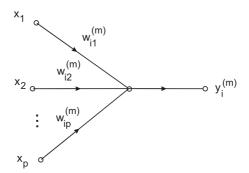


Figura 2.7: Fluxo do sinal em um neurônio linear de módulo especialista i

Para a atualização dos pesos dos especialistas, utiliza-se a maximização da função $l(\mathbf{w}, \mathbf{g})$ para obter a estimativa de máxima verossimilhança dos parâmetros da rede. Para tanto, usa-se o gradiente ascendente ao derivar a função l em relação aos pesos sinápticos, e, assim, modificar

esses pesos aplicando um pequeno ajuste $\Delta \mathbf{w}_i^{(m)}$ definido por

$$\Delta \mathbf{w}_{i}^{(m)} = \eta \frac{\partial l}{\partial \mathbf{w}_{i}^{(m)}} \begin{cases} i = 1, 2, ..., K \\ m = 1, 2, ... q \end{cases}$$
 (2.52)

em que η é um pequena taxa de aprendizagem. Um exemplo de neurônio pode ser visto na Figura 2.7, que mostra o fluxo do sinal que passa por ele. Podendo a derivada parcial $\frac{\partial l}{\partial \mathbf{w}_i^{(m)}}$ ser expressa como

$$\frac{\partial l}{\partial \mathbf{w}_{i}^{(m)}} = \frac{\partial l}{\partial y_{i}^{(m)}} \frac{\partial y_{i}^{(m)}}{\partial \mathbf{w}_{i}^{(m)}}.$$
(2.53)

Para auxiliar alguns cálculos, define-se a função h_i como

$$h_i = \frac{g_i \exp\left\{-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i\|^2\right\}}{\sum_{j=1}^K g_j \exp\left\{-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i\|^2\right\}}, \quad i = 1, 2, ..., K$$
(2.54)

que é definida como a probabilidade a posteriori associada à saída do i-ésimo módulo especialista, sendo esta função condicional tanto em relação ao vetor de entrada \mathbf{x} , quanto da resposta desejada \mathbf{d} , sujeita às restrições

$$0 < h_i < 1$$
 (2.55)

e

$$\sum_{i=1}^{K} h_i = 1. (2.56)$$

Ao calcular a derivada da função do log-verossimilhança em relação ao vetor de saída \mathbf{y}_i , após uma simplificação, tem-se

$$\frac{\partial l}{\partial \mathbf{y}_i} = h_i(\mathbf{d} - \mathbf{y}_i) \tag{2.57}$$

com i = 1,...,K. Observe que, durante o processo de aprendizagem, os pesos do i-ésimo especialista são ajustados para corrigir o erro existente na diferença entre a saída desejada e a saída do módulo especialista, mas em proporção à probabilidade a posteriori h_i de que o i-ésimo módulo gerou o padrão de treinamento em uso.

Essa mesma derivada parcial, pode ser colocada em relação a cada m-ésimo elemento do vetor \mathbf{y}_i , de acordo com a seguinte equação:

$$\frac{\partial l}{\partial y_i^{(m)}} = h_i \left(d^{(m)} - y_i^{(m)} \right) = h_i e_i^{(m)}, \tag{2.58}$$

em que $e_i^{(m)}$ é o erro produzido pela saída do m-ésimo neurônio do i-ésima rede especialista,

definido como

$$e_i^{(m)} = d^{(m)} - y_i^{(m)}. (2.59)$$

Se o especialista for a rede PS, a derivada de $y_i^{(m)}$ em relação vetor de pesos $\mathbf{w}_i^{(m)}$ é dada por

$$\frac{\partial y_i^{(m)}}{\partial \mathbf{w}_i^{(m)}} = \mathbf{x}. \tag{2.60}$$

Caso seja a rede PL, a derivada de $y_i^{(m)}$ em relação vetor de pesos $\mathbf{w}_i^{(m)}$ passa a ser dividida em duas derivadas parciais por conta da função de transferência logística, ou seja

$$\frac{\partial y_i^{(m)}}{\partial \mathbf{u}_i^{(m)}} = \frac{\partial \phi(\mathbf{u}_i^{(m)})}{\partial \mathbf{u}_i^{(m)}} = \phi_i'$$

$$\frac{\partial \mathbf{u}_i^{(m)}}{\partial \mathbf{w}_i^{(m)}} = \mathbf{x}.$$
(2.61)

Se ϕ é a função logística, mostrada na Equação (2.15), a sua derivada ϕ' é dada na Equação (2.23).

Dessa forma, substituindo as Equações (2.58) e (2.60) na Equação (2.52), obtém-se a regra de atualização dos pesos dos especialistas PS na iteração n,

$$\mathbf{w}_{i}^{(m)}(n+1) = \mathbf{w}_{i}^{(m)}(n) + \Delta \mathbf{w}_{i}^{(m)}(n)$$

$$\mathbf{w}_{i}^{(m)}(n+1) = \mathbf{w}_{i}^{(m)}(n) + \eta h_{i} e_{i}^{(m)}(n) \mathbf{x}(n)$$
(2.62)

para i = 1, 2, ..., K e m = 1, 2, ...q.

Para o caso dos especialistas PL, substitui-se na Equação (2.52), as Equações (2.58), (2.61) e (2.23), obtendo assim, a seguinte regra de atualização dos pesos do *i*-ésimo neurônio do *m*-ésimo especialista:

$$\mathbf{w}_{i}^{(m)}(n+1) = \mathbf{w}_{i}^{(m)}(n) + \Delta \mathbf{w}_{i}^{(m)}(n)$$

$$\mathbf{w}_{i}^{(m)}(n+1) = \mathbf{w}_{i}^{(m)}(n) + \eta h_{i} e_{i}^{(m)}(n) y_{i}(n) [1 - y_{i}(n)] \mathbf{x}(n)$$
(2.63)

para i = 1, 2, ..., K e m = 1, 2, ...q.

Sobre o treinamento dos pesos do seletor, existem algumas diferenças em relação ao dos

pesos dos especialistas. A começar, a quantidade de neurônios de saída presentes na rede seletora é igual a K, enquanto nos módulos é igual a q. Apesar de, tanto os especialistas quanto o seletor serem redes de camada única, os especialistas possuem saídas lineares, enquanto o seletor usa a função softmax como função de ativação.

Lembrando que a ativação g_i é em função da soma ponderada u_i , substitui-se a Equação (2.40) na Equação (2.51), levando em consideração que o somatório $\sum_{j=1}^{K} \exp\{u_j\}$ é o mesmo para todo i, pode-se reescrever a função log-verossimilhança de acordo com

$$l = \ln \sum_{i=1}^{K} \exp\{u_i\} \cdot \exp\left\{-\frac{1}{2} \|\mathbf{d} - \mathbf{y}_i\|^2\right\} - \ln \sum_{j=1}^{K} \exp\{u_j\}\right\}.$$
 (2.64)

Assim como na atualização dos pesos dos especialistas, calcula-se a derivada parcial da Equação (2.64), pelo vetor de pesos \mathbf{a}_i do seletor, ou seja,

$$\frac{\partial l}{\partial \mathbf{a}_i} = \frac{\partial l}{\partial u_i} \frac{\partial u_i}{\partial \mathbf{a}_i}, \qquad i = 1, 2, ..., K.$$
 (2.65)

Sabendo que

$$u_i = \mathbf{x}^T \mathbf{a}_i, \tag{2.66}$$

então a derivada parcial de u_i em relação aos pesos é igual a

$$\frac{\partial u_i}{\partial \mathbf{a}_i} = \mathbf{x}, \qquad i = 1, 2, ..., K. \tag{2.67}$$

A derivada parcial de l em função de u_i , depois de simplificada, é dada por

$$\frac{\partial l}{\partial u_i} = h_i - g_i. \tag{2.68}$$

A Equação (2.68) mostra que os pesos do i-ésimo neurônio do seletor são ajustados de tal forma que as ativações, ou probabilidades a priori, tendem em direção às probabilidades a posteriori, h_i (JACOBS et al., 1991).

Sendo assim, substituindo as Equações (2.68) e (2.67), na Equação (2.65), tem-se que

$$\frac{\partial l}{\partial \mathbf{a}_i} = (h_i - g_i)\mathbf{x}, \qquad i = 1, 2, ..., K$$
(2.69)

então o ajuste correspondente se torna

$$\Delta \mathbf{a}_{i} = \eta \frac{\partial l}{\partial \mathbf{a}_{i}} \\
= \eta (h_{i} - g_{i}) \mathbf{x} .$$

em que o η pode ter um valor diferente do usado no treinamento dos pesos dos especialistas. Dessa forma, a atualização dos pesos passa a ser feita de acordo com a seguinte regra:

$$\mathbf{a}_i(n+1) = \mathbf{a}_i(n) + \eta \left(h_i(n) - g_i(n)\right) \mathbf{x}(n). \tag{2.70}$$

2.4.3 Resumo do Algoritmo da Arquitetura ME

Nesta seção, apresenta-se um resumo do procedimento para a adaptação da arquitetura de mistura de especialistas, construídas por módulos de redes de camada única com função de ativação logística e uma rede seletora também de camada única, mas com uma função de ativação softmax.

- 1. Inicialização: atribuir valores pequenos e aleatórios para os pesos dos especialistas $\mathbf{w}_i^{(m)}(n)$ e do seletor $\mathbf{a}_{i}(n)$, i = 1, 2, ..., K e m = 1, 2, ..., c;
- 2. Adaptando os Especialistas e o Seletor: apresentar o padrão de entrada $\mathbf{x}(n)$ e a resposta desejada $\mathbf{d}(n)$ para o instante atual n:

$$u_i(n) = \mathbf{x}^T(n)\mathbf{a}_i(n), \tag{2.71}$$

$$u_{i}(n) = \mathbf{x}^{T}(n)\mathbf{a}_{i}(n), \qquad (2.71)$$

$$g_{i}(n) = \frac{\exp\{u_{i}(n)\}}{\sum_{j=1}^{K} \exp\{u_{j}(n)\}}, \qquad (2.72)$$

$$y_{i}^{(m)}(n) = \mathbf{x}^{T}(n)\mathbf{w}_{i}^{(m)}(n), \qquad (2.73)$$

$$\mathbf{y}_{i}^{(m)}(n) = \mathbf{x}^{T}(n)\mathbf{w}_{i}^{(m)}(n), \tag{2.73}$$

$$\mathbf{y}_{i}(n) = [y_{i}^{(1)}(n) y_{i}^{(2)}(n) \cdots y_{i}^{(c)}(n)]^{T}, \tag{2.74}$$

$$\mathbf{y}_{i}(n) = [y_{i}^{(1)}(n) y_{i}^{(2)}(n) \cdots y_{i}^{(c)}(n)]^{T}, \qquad (2.74)$$

$$h_{i}(n) = \frac{g_{i}(n) \exp\left\{-\frac{1}{2} \|\mathbf{d}(n) - \mathbf{y}_{i}(n)\|^{2}\right\}}{\sum_{j=1}^{K} g_{j}(n) \exp\left\{-\frac{1}{2} \|\mathbf{d}(n) - \mathbf{y}_{i}(n)\|^{2}\right\}}, \qquad (2.75)$$

$$e_i^{(m)}(n) = d^m(n) - y_i^{(m)}(n),$$
 (2.76)

$$\delta_i^{(m)}(n) = e_i^{(m)}(n) \left[y_i^{(m)}(n) (1 - y_i^{(m)}(n)) \right], \tag{2.77}$$

$$\mathbf{w}_{i}^{(m)}(t+1) = \mathbf{w}_{i}^{(m)}(n) + \eta h_{i}(n) \delta_{i}^{(m)}(n) \mathbf{x}(n), \tag{2.78}$$

$$\mathbf{a}_i(t+1) = \mathbf{a}_i(n) + \eta \left[h_i(n) - g_i(n) \right] \mathbf{x}(n). \tag{2.79}$$

- 3. Repetir passo 2 para todos os padrões de treinamento, i.e. n = 1, 2, ..., N.
- 4. Repetir passos 2 e 3 até o processo convergir.

2.5 Rede Recorrente de Elman

Até o presente momento, as arquiteturas descritas são do tipo estática, i.e. não possuem nenhuma forma de realimentação e, consequentemente, nenhuma memória. Nesta seção, é descrita a rede de Elman que possui realimentações das ativações anteriores de sua camada oculta e, dessa forma, consegue processar a informação temporal contida nos dados de entrada.

ELMAN (1990) propôs uma rede recorrente baseada na rede MLP, com o diferencial de apresentar na camada de entrada, neurônios chamados de *unidades de contexto*, as quais consistem nas ativações dos neurônios ocultos na iteração anterior. Essa mudança na arquitetura coloca esta rede em uma categoria, conhecida como redes recorrentes simples.

A existência de realimentação de uma camada para outra leva à influência das ativações neurais (estados) anteriores na resposta atual do sistema, caracterizando-se como uma forma de memória de curto prazo (PRINCIPE et al., 2000). Dessa forma, mecanismos de memória permitem que o mapeamento do espaço de entrada no de saída inclua a dimensão temporal no tratamento dos dados. Em outras palavras, a sequência na qual estes são apresentados, passa a ser levada em consideração. As redes que fazem uso de memória, seja através da recorrência ou atrasadores na camada de entrada, são conhecidas como redes dinâmicas.

As redes recorrentes simples são caracterizadas por terem *conexões de realimentação*³ fixas, em outras palavras, seus valores não são ajustados, e pode-se fazer a atualização dos pesos restantes com o algoritmo de retropropagação.

A primeira parte da camada de entrada corresponde ao vetor de entrada propriamente dito, conforme definido

$$\mathbf{x}(n) = [x(n) \ x(n-\tau) \ \cdots \ x(n-(d_E-1)\tau)]^T.$$
 (2.80)

A segunda parte, chamada de unidades de contexto, consiste na cópia das saídas dos neurônios da camada oculta no instante n-1. O termo "cópia", na verdade, é implementado computacionalmente através de um conjunto de conexões de realimentação com pesos fixos e iguais a 1. Desta forma, os valores exatos das ativações dos neurônios da camada oculta no instante n-1 são utilizados pelas unidades de contexto para compor o vetor de entrada no instante n (ver Figura 2.8).

Se uma rede MLP tem duas camadas ocultas pode-se escolher de qual camada oculta realimentar as ativações neurais a fim de gerar uma rede recorrente de Elman. Três opções são

³Pesos sinápticos com orientação da camada de saída para a entrada. Eles realimentam para uma dada camada, a saída de neurônios de outras

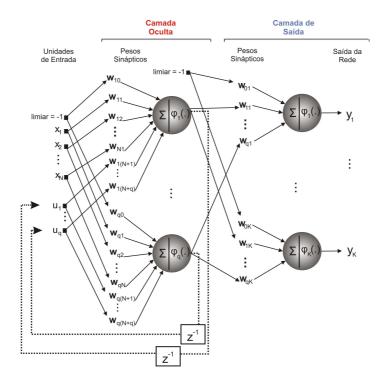


Figura 2.8: Ilustração da arquitetura da rede de Elman com apenas uma camada oculta.

possíveis: (i) realimentar as ativações da primeira camada oculta para as unidades de contexto, (ii) realimentar as ativações da segunda camada oculta para as unidades de contexto, ou (iii) realimentar as ativações de ambas as camadas ocultas para as unidades de contexto. Após experimentação, observou-se melhores resultados para a segunda opção. Dessa forma, utiliza-se nessa dissertação apenas as ativações da segunda camada oculta que são realimentadas para as unidades de contexto.

Com relação à quantidade de unidades de contexto, esta depende diretamente de qual das opções listadas no parágrafo anterior for escolhida. No primeiro caso, a quantidade de unidades de contexto é igual ao número de neurônios da primeira camada oculta (q_1) . No segundo caso, a quantidade de unidades de contexto é igual ao número de neurônios da segunda camada oculta (q_2) . No terceiro e último caso, a quantidade de unidades de contexto é igual a $q_1 + q_2$. Todas as outras conexões da rede de Elman são ajustáveis e do tipo *feedforward*, de tal forma que esta arquitetura pode ser treinada pelo algoritmo *backpropagation* padrão. Do ponto de vista das conexões, a rede de Elman possui apenas recorrências globais, sendo a maior parte de suas conexões sinápticas do tipo *feedforward*.

As ativações dos neurônios da rede de Elman com uma camada oculta são calculadas como

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n)x_j(n) + \sum_{l=1}^{q_1} w_{il}(n)v_l(n-1), \qquad i = 1, \dots, q_1,$$
 (2.81)

2.6 Conclusão 43

tal que a saída dos neurônios ocultos é dada por $v_i(n) = \phi[u_i(n)]$. As ativações e as saídas dos neurônios da última camada são calculadas como nas Equações (2.28) e (2.29). Durante o treinamento os pesos w_{ij} e w_{il} são ajustados segundo as regras do algoritmo *backpropagation* padrão.

O do número de parâmetros da rede recorrente de Elman com uma única camada oculta é dado por

$$M = ((p+q_1+1) \times q_1) + ((q_1+1) \times m). \tag{2.82}$$

Como exemplo hipotético, considere as seguintes constantes p=4, $q_1=10$ e m=1. Assim, uma rede de Elman, terá M=150+11=161 parâmetros ajustáveis, enquanto uma rede MLP teria

$$M = (p+1)q_1 + (q_1+1)M = (4+1)10 + (10+1)1 = 51.$$
 (2.83)

2.6 Conclusão

Neste capítulo foram descritas as redes perceptron simples e logístico, perceptron multicamadas, mistura de especialistas e rede recorrente de Elman.

A rede PS é a forma mais simples de rede neural artificial, normalmente utilizada para classificação de padrões linearmente separáveis, sendo constituída por uma única camada de neurônios. A rede PL é uma variação desta, fazendo uso da função sigmóide logística como função de ativação e seu treinamento é realizado com a regra LMS generalizado.

A rede MLP é constituída de duas ou mais camadas de neurônios. Por conta dessa característica, se enquadra na categoria dos algoritmos não-lineares e, desta forma, está preparada para solucionar problemas não-linearmente separáveis. O treinamento de seus pesos é realizado com o algoritmo de retropropagação do erro (*backpropagation*)

A arquitetura de mistura de especialistas é um tipo de arquitetura modular, a qual é composta de um certo número de redes neurais como especialistas aliadas a um mecanismo seletor, que decide qual desses especialistas deve responder ao estímulo na camada de entrada. Essa arquitetura possui uma forte inspiração neurobiológica, na qual se afirma que o próprio cérebro faz uso de estruturas modulares e divisões de tarefas afim de evitar que muitos neurônios se aglomerem em uma única região (BALLARD, 1986).

Como principal vantagem, permite que problemas complexos seja divididos em subpro-

2.6 Conclusão 44

blemas mais simples. O treinamento é feito de forma que cada especialista seja ativado com maior intensidade somente em determinados momentos. Apesar dos especialistas fornecerem uma ativação a todos os estímulos, essas saídas serão ponderadas pela probabilidade de cada especialista ser o mais adequado ao dado vetor de entrada. Da mesma forma, eles são treinados em proporção à sua probabilidade de responderem a uma dada entrada.

Diferente das arquiteturas descritas até então, a rede Elman possui estruturas de memória embutidas em sua arquitetura. A sua estrutura básica é semelhante a da rede MLP, com a exceção da existência das unidades de contexto, as quais incluem na camada de entrada as ativações da camada oculta do momento anterior. Por conta dessa realimentação, esta rede é capaz de tratar a informação temporal contida nos dados de treinamento.

No capítulo seguinte estão a descrição dos resultados dos testes de classificação estática com as redes perceptron, perceptron multicamadas e mistura de especialistas.

3 RESULTADOS DOS EXPERIMENTOS COM CLASSIFICAÇÃO ESTÁTICA

3.1 Introdução

Neste capítulo, são detalhados os testes realizados com as três redes neurais escolhidas para o problema de classificação de padrões estáticos na navegação de um robô móvel. A perspectiva utilizada nesses testes é baseada na formulação do problema de navegação "seguir paredes" para robôs móveis como um problema de classificação de padrões, no qual o robô deve distinguir dentre as quatro opções possíveis de movimentação, dada uma determinada configuração do ambiente.

A título de abreviar a denominação das arquiteturas avaliadas neste capítulo, adota-se a seguinte notação: PL(I,O) para rede perceptron logístico, em que I é a dimensão da entrada e O o número de neurônios de saída; ME(I,K,O) para rede de mistura de especialistas, em que K é o número de especialistas; e MLP(I,q,O) para rede perceptron multicamadas com q neurônios ocultos.

Este capítulo está organizado da seguinte forma. Na Seção 3.2 detalha-se a plataforma robótica utilizada para a coleta dos dados e testes de desempenho das redes neurais utilizadas. Nas Seções 3.3 e 3.4 é detalhado o programa baseado em heurísticas, que permitiu a geração dos dados estudados; a maneira como é realizada a coleta destes e a descrição do ambiente de testes. Na Seção 3.5 encontram-se os resultados do treinamento/teste *offline* das redes perceptron logístico, mistura de especialistas e perceptron multicamadas, assim como a descrição do comportamento do robô com cada uma delas. Por fim, na Seção 3.6 estão as conclusões dos testes.

3.2 Robô Móvel SCITOS G5

Desenvolvido pela equipe da empresa MetraLabs Robotics localizada em Ilmenau, Alemanha, o robô SCITOS G5 é uma plataforma mecatrônica móvel profissional para pesquisa e desenvolvimento que combina as vantagens de robôs industriais, tais como robustez e longevidade, com a mobilidade e flexibilidade de um robô de pesquisa. A Figura 3.1 mostra o robô no laboratório de Robótica (LABOR), laboratório integrante do Centro de Referência em Automação e Robótica (CENTAURO), instalado nas dependências da Universidade Federal do Ceará, onde foram realizados os testes de navegação.



Figura 3.1: Robô SCITOS G5 no Laboratório de Robótica (LABOR/CENTAURO).

Nas seções seguintes estão detalhados o *hardware* e as bibliotecas de *software* disponíveis com esta plataforma.

3.2.1 O Hardware

O robô SCITOS G5 possui vários sensores diferentes, estações auto-recarregáveis e interface homem-máquina para aplicações em ambientes fechados, com a possibilidade de mudar algumas outras interfaces (mecânicas e elétricas) de acordo com a necessidade do usuário.

O robô move-se através de um sistema motor diferencial¹, que permite movimentar a plataforma de 60kg a uma velocidade de até 1,4m/s, capaz de rotacionar 360 graus e empurrar cargas de até 50kg.

¹Significa que é um sistema motor de duas rodas, onde cada uma possui um atuador independente.

O robô SCITOS G5 tem 570mm × 735mm × 610mm de altura, comprimento e largura, respectivamente. Possui ainda uma autonomia de 12h com um PC embarcado. Este último é constituído de uma placa-mãe Mini-ITX, processador Intel Core Duo 1,6 ou 2.0GHz, com memória RAM de 2GB, HD de 120GB, com entrada PS/2 para teclado e mouse, 5 entradas USB, 3 entradas Firewire², TV-out, RS232, VGA, saída SPDIF, LVDS, e entrada RJ-45 para Ethernet 10/100, além de rede wireless *on-board*, padrão IEEE 802.11a/b/g. Como sistema operacional, utiliza Linux Fedora Core 8, com MetraLabs C++ Robot-API.

O robô possui *encoders* para cálculo de posição com 460 "ticks" por rotação da roda, sensor de colisão e 24 sensores de ultra-som com alcance de 20cm - 300cm. Além desses sensores, há ainda um sensor a laser (Hokuyo URG-04LX, SICK S300 or LMS-200), uma cabeça robótica, câmera omnidirecional e portas de I/O digitais e analógicas.

Para interface humano-máquina o robô possui um monitor *touch screen* de 15 polegadas com resolução de 1024×768 pixels. Também possui 2 caixas de som embutidas e 2 microfones omnidirecionais. Além disso, uma cabeça robótica envolta por uma redoma de acrílico transparente com 300mm de diâmetro, na qual estão inseridos um par de olhos falsos e pálpebras móveis. Os movimentos que a cabeça robótica é capaz de realizar são:

- elevação e abaixamento da cabeça (20°, -7°);
- rotação de toda a cabeça em até 350°;
- movimentos sincronizados dos olhos para cima, para baixo e para os lados;
- abertura e fechamento das pálpebras de forma independente uma da outra.

3.2.2 API do SCITOS G5

Nesta seção é apresentada a API *MLRobotics* disponibilizada para leitura de dados sensoriais e envio de comandos de controle (*setpoints*) relacionados ao robô SCITOS G5. A *ML-Robotics* é uma coleção de bibliotecas escritas em C++ e que permitem o desenvolvimento de programas para robôs móveis produzidos pela empresa MetraLabs. A maioria dessas bibliotecas dependem dos pacotes básicos *MetraLabsBase* e *MetraLabsMath*.

O pacote *MetraLabsBase* consiste em algumas classes genéricas que podem ser utilizadas para operações gráficas, operações de leitura e escrita em arquivos comuns e XML, comunicação via *socket*, funções de criptografia e segurança, entre outros.

²também conhecido como i.Link, IEEE 1394 ou HigH Performance Serial Bus/HPSB

O segundo pacote, *MetraLabsMath*, oferece classes para tratamento de vetores, matrizes e operações numéricas, tais como inversão de matrizes, implementação de filtros de Kalman e filtros de Kalman estendidos, geração de números aleatórios, resolução de *n* equações lineares, entre outros métodos e técnicas.

A API *MLRobotics* oferece vários pacotes, os quais consistem em pelo menos uma única classe compartilhada. Os principais componentes desta API são listados a seguir:

- *MLRoboticBase*: fornece um *framework* de aplicações, que consiste em uma classe *factory*, uma *blackboard*, um servidor de eventos e as classes de gestão de parâmetros.
- MLRoboticRobot: este pacote permite a aquisição de alguns tipos de dados, que são necessários no contexto da Robótica Móvel, tais como posição, velocidade, odometria, detector de colisões e medidas dos sensores de distância.
- MLRoboticIP: fornece algumas classes para processamento de imagens, além de dois diferentes tipos de imagem e algumas operações básicas de filtragem, métodos para conversão de cores e operações de I/O para diferentes tipos de arquivos de imagem.
- *MLRoboticOpenCV*: este pacote oferece uma classe *wrapper* para o tipo de imagem IpII-mage a partir do *Intel*® *Open Source Computer Vision Library*. Permite utilizar as funcionalidades do *OpenCV*, juntamente com o *framework* oferecido pela MLRobotic.
- *MLRoboticGUI*: este pacote contém algumas classes GUI simples para exibição de imagens, vídeos e dados de distâncias. Essas classes são baseadas em Qt®toolkit da Trolltech®.
- MLRoboticSystem: neste pacote há algumas classes de baixo nível em que estão localizadas informações sobre a instalação da CPU e o sistema ALSA (Advanced Linux Sound Architecture) para controle do sistema de som.
- MLRoboticHardware: este pacote oferece algumas classes de controle adicional para componentes de hardware externos, que não pertencem ao robô propriamente dito. Quatro diferentes dispositivos como câmara Vídeo-4-Linux, Video-4-Linux-2, DC1394 e DFG1394 são suportadas e uma classe para porta serial está disponível.

Sensores de distância a laser, câmeras e braços robóticos são suportados através de *Add-ons* opcionais.

Na seção seguinte são apresentados os detalhes sobre o programa baseado em heurísticas responsável pela coordenação das quatro movimentações possíveis, de maneira a permitir que o robô apresente o comportamento de seguir paredes.

3.3 Algoritmo Heurístico para Navegação do tipo Seguir-Paredes

Para o desenvolvimento do sistema de navegação e controle de um robô, duas abordagens podem ser seguidas: *top-down* e *bottom-up* (PFEIFER; SCHEIER, 2002).

Na abordagem *top-down* (descendente), a tarefa que o robô deve executar é particionada em subtarefas, as quais são determinadas pelos projetistas. Neste paradigma, um especialista projeta o sistema de navegação e controle com base no seu conhecimento (AGUIRRE, 2007). De tal forma que o robô fica limitado a apenas resolver a tarefa para a qual foi programado.

Já na abordagem *bottom-up* (ascendente), muito pouco conhecimento é imposto pelo projetista do sistema de navegação e controle, exigindo que o sistema seja dotado de capacidade de aprendizagem e adaptação ao ambiente. Assim, os comportamentos observados emergem da interação do robô com o seu ambiente. Esses conceitos estão explicados com maior nível de detalhes e profundidade em ARKIN (1998), PFEIFER & SCHEIER (2002) e Tani (2007).

Nesta seção, é apresentado um algoritmo desenvolvido, na filosofia da abordagem *top-down*, para que o robô SCITOS G5 seja capaz de navegar seguindo paredes (*Wall-Following*). A metodologia de projeto do algoritmo é dita ser do tipo descendente devido ao fato de que as classes de movimentos e as condições sensório-motoras para que os movimentos sejam realizados são contruídas a partir de diversos testes realizados com o robô e a partir do conhecimento de especialistas. O algoritmo baseado em regras heurísticas com melhor resultado, pode ser visto no Algoritmo 3.3.1.

A lógica implementada pelo o Algoritmo 3.3.1 leva em consideração duas distância resumidas: a da frente e da esquerda. Sendo assim, o robô realiza a volta na sala no sentido horário, cuja trajetória criada e percepção sensorial do ambiente podem ser vistos na Figura 3.2.

A tarefa de Seguir-Paredes é dividida em quatro subtarefas: seguir em frente, girar para a direita em uma curva aberta, girar para a direita em uma curva fechada e girar à esquerda. O chaveamento das decisões é feita de acordo com a forma que o robô está percebendo o ambiente com seus sensores de ultra-som.

Algorithm 3.3.1: SEGUEPAREDE(frontDistance, leftDistance)

```
then

(if frontDistance <= 0,9
then Parar e girar à direita

else Reduz a velocidade e gira à esquerda

(if frontDistance <= 0,9
then Parar e girar à direita

else else if leftDistance < 0,55
then Reduz a velocidade e gira à direita

else Segue em Frente
```

Para que ele siga em frente, é necessário que esteja emparelhado com a parede a mais de 55cm desta e que não haja obstáculos à frente a menos de 90cm.

Para que realize a curva à esquerda, é necessário que não haja obstáculos a menos de 90*cm* da sua frente e da sua esquerda, por exemplo, quando este se encontra uma esquina.

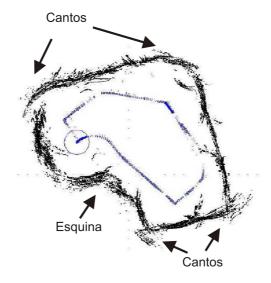


Figura 3.2: Percepção do ambiente e trajetória do SCITOS G5 realizada pelo Algoritmo 3.3.1.

A curva aberta à direita é utilizada apenas quando o robô está muito próximo da parede, ou seja, com uma distância à esquerda menor que 90*cm*, mas a frente está livre. Já a curva fechada à direita é justamente a manobra que evita colisões nos cantos.

Na seção seguinte são descritos os detalhes sobre a coleta dos dados referentes à navegação do robô, os quais são destinados ao treinamento e teste das redes neurais.

3.4 Metodologia para a Coleta de Dados

Para coletar dados destinados ao treinamento das redes neurais, foi criada uma rotina em C++ para guiar e rotular o comportamento de seguir paredes, dadas as leituras sensoriais naquele instante. Detalhes desta rotina foram mostrados na Seção B e a sua lógica de funcionamento é apresentada no Algoritmo 3.3.1. A rotina, por ser a responsável por gerar as decisões que o robô precisa tomar sobre a sua movimentação, é colocada dentro de uma *thread* que é chamada a cada vez que os sensores de ultra-som captam alguma mudança significativa. Caso não haja nenhuma atividade sensorial em 500*ms*, o robô pára.

A lógica do algoritmo de navegação faz uso de regras heurísticas baseadas em condições pré-estabelecidas, as quais foram determinadas a partir de diversas tentativas e erros até que chegasse a um comportamento adequado ao problema. Ela atua sobre as medidas de distância captadas à frente e à esquerda do robô, embora este também calcule as distâncias referentes ao seu lado direito e atrás.

O vetor de leituras sensoriais no instante n, $\mathbf{x}_s(n)$ é representado por

$$\mathbf{x}_{s}(n) = [x_{f}(n) \ x_{e}(n) \ x_{d}(n) \ x_{t}(n)]^{T}, \tag{3.1}$$

em que os índices f, e, d e t indicam, respectivamente, a leitura correspondente aos sensores localizados na parte da frente, à esquerda, à direita e atrás. Cada uma dessas distâncias é extraída de um conjunto de sensores que formam juntos um feixe de 60° , e a distância utilizada é a menor encontrada pelos sensores que fazem parte do conjunto. Utilizando a Figura 3.3 para melhor entender o conceito, a leitura referente à frente , $x_f(n)$, é dada por

$$x_f(n) = \min\{x_0(n) \ x_1(n) \ x_{11}(n)\},\tag{3.2}$$

de forma similar, a leitura referente ao lado esquerdo pode ser representada como

$$x_e(n) = \min\{x_8(n) \ x_9(n) \ x_{10}(n)\}. \tag{3.3}$$

As quatro distâncias são doravante referidas ao longo do texto como *distâncias resumidas*, e a Figura 3.3 mostra de forma intuitiva como são escolhidos os valores que vão representá-las em cada lado. Os números em vermelho, representam os sensores que obtêm a menor distância do

seu grupo, e consequentemente são os escolhidos para representá-los. Essa abordagem também é a mesma utilizada no trabalho de KRISHNA & KALRA (2000).

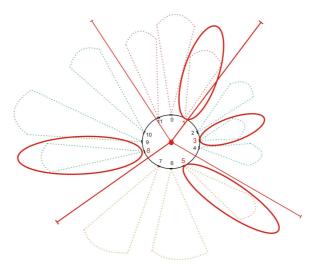


Figura 3.3: Ilustração das distâncias resumidas.

Além da lógica que determina a movimentação, a *thread* também possui a responsabilidade de enviar, através da rede wireless, as seguintes informações ao banco de dados localizado no computador de apoio: (i) as 24 leituras dos sensores ultra-som, (ii) as coordenadas x, y e posição angular a partir do *encoder* do robô, (iii) a velocidade translacional e rotacional do robô naquele momento, (iv) o rótulo numérico da ação que foi realizada pelo robô de acordo com as configurações do ambiente e (v) as quatro distâncias resumidas.

No caso da rede MLP e da Elman, são ainda acrescentados as ativações dos neurônios da camada oculta. Essas informações são coletadas à medida que o SCITOS G5 navega pela sala, realizando quatro voltas. A coleta de dados é realizada a uma taxa de 9 amostras por segundo e gerou um banco de dados com 5456 exemplos. Estes dados são então utilizados para o treinamento dos classificadores neurais apresentados anteriormente, a fim de verificar qual deles melhor "imita" o Algoritmo 3.3.1.

A disposição dos objetos localizados na sala de teste pode ser visto na Figura 3.4.

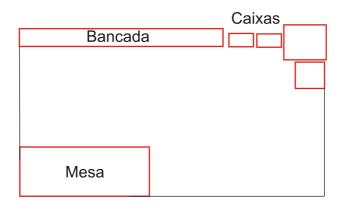


Figura 3.4: Diagrama do ambiente de navegação.

Os movimentos executados a partir da análise das distâncias resumidas à frente e à esquerda são: seguir em frente, curva aberta à direita, curva fechada à direita e curva aberta à esquerda. Mais detalhes a respeito das condições necessárias para a escolha dessas ações no programa heurístico foram vistos na Seção B.

Como já foi mencionado, o problema de navegação é formulado como um problema de classificação de padrões. Neste caso, as leituras sensoriais dos ultra-sons são utilizadas como entradas e as classes do problema são os quatro movimentos que podem ser realizados: seguir em frente (classe 1), curva aberta à direita (classe 2), curva fechada à direita (classe 3) e curva aberta à esquerda (classe 4).

Na Figura 3.5 é mostrada a trajetória que o robô percorreu, em pontos azuis, e a sua perspectiva do ambiente (paredes, obstáculos, etc) em pontos pretos, dada pelas distâncias retornadas pelos sensores de ultra-som.

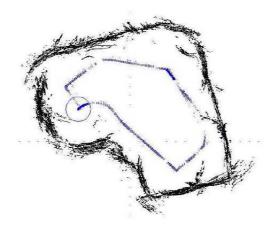


Figura 3.5: Percepção do ambiente e trajetória do SCITOS G5 realizada pelo Algoritmo 3.3.1.

3.5 Resultados dos Experimentos

Nos experimentos que se seguem, a rede PL é formada por uma única camada de quatro neurônios de saída, cada um representando uma ação diferente. Esta rede é treinada por 500 épocas a uma taxa de aprendizagem fixa de $\eta = 0.02$.

A arquitetura ME é constituída por uma rede seletora constituída de uma rede PS com quatro neurônios, mais K=4 redes especialistas que também são formadas por redes PS com quatro neurônios. O treinamento é realizado com 35 épocas e uma taxa fixa de aprendizagem de 0,02.

Com a rede MLP, foram realizados testes preliminares com diferentes quantidades de neurônios ocultos para verificar qual a quantidade mínima capaz de realizar a classificação com sucesso. Observou-se que q=6 é essa quantidade mínima, conforme será mostrado adiante. A camada de saída contém 4 neurônios com função de ativação logística e o treinamento é realizado por 500 épocas com taxa de aprendizagem fixada em $\eta=0,05$.

São realizadas 50 repetições do treinamento/teste de cada rede, sempre reiniciando os pesos aleatoriamente entre [-0,01;0,01] a cada novo treinamento. Estas repetições têm por objetivo fazer uma avaliação estatística das taxas de acerto. Os pesos da rede cujo treinamento resultou na melhor taxa de acerto são armazenados para uso durante o teste. Das 5456 amostras coletadas, 2205 pertencem à classe 1, 826 pertencem à classe 2, 2097 pertencem à classe 3 e 328 pertencem à classe 4. Do total, 4000 amostras são destinadas para treinamento e as 1455 restantes para teste dos classificadores. Essas amostram não passam por nenhum tipo de préprocessamento visto que 70% dos padrões de entrada estão entre [0,1], sendo 23,5% dos valores restantes entre [1,2] e 6,5% de valores entre [2,5].

Na Figura 3.6 pode ser visto o gráfico de dispersão dos dados da distância resumida à frente versus a resumida à esquerda, em que cada ponto tem a sua cor determinada de acordo com a classe a que pertence.

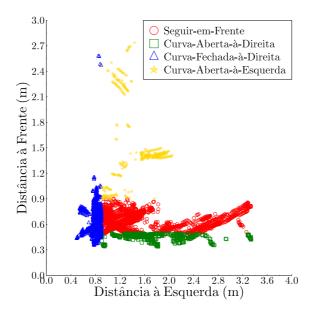


Figura 3.6: Gráfico de dispersão das distâncias resumidas à frente e à esquerda.

Para cada uma das arquiteturas, três Casos são avaliados, variando a dimensão do vetor de entrada. As componentes destes vetores consistem nas distâncias obtidas pela leitura dos sensores ultra-som. Cada mudança na dimensão dos dados de entrada, resulta em uma nova rede e é realizado um novo treinamento.

Para o Caso 1, utiliza-se todas as distâncias lidas pelos 24 sensores de ultra-som do robô SCITOS G5,

$$\mathbf{x}(n) = \begin{bmatrix} x_1 \\ \vdots \\ x_{24}. \end{bmatrix}. \tag{3.4}$$

Para o Caso 2, as quatro distâncias resumidas (frente, esquerda, direita e traseira),

$$\mathbf{x}(n) = [x_f(n) \ x_e(n) \ x_d(n) \ x_t(n)]^T. \tag{3.5}$$

A escolha desse teste é feita na tentativa de diminuir a dimensionalidade da entrada, como também verificar o quanto essa diminuição da complexidade afetará o comportamento do robô.

Por fim, para o Caso 3, apenas as distâncias resumidas referentes à frente e à esquerda do SCITOS, são alimentadas à rede, como é mostrado abaixo

$$\mathbf{x}(n) = [x_f(n) \ x_e(n)]^T. \tag{3.6}$$

Desses três Casos, apenas o que utiliza duas distâncias resumidas é que será tratado em de-

talhes neste capítulo. Todavia, os detalhes dos outros testes podem ser encontrados nos Apêndices A, B, C e D.

Cada experimento para diferentes dimensões do vetor de entrada foi dividido em duas fases. A primeira consiste no treinamento e teste *offline* de uma rede em um computador (Pentium 4, 2.4GHz com memória de 500Mb, Sistema Operacional Ubuntu 8.04), resultando em um arquivo texto contendo os valores dos pesos sinápticos da rede com melhor desempenho durante os testes. Em seguida, embarca-se a rede gerada para que esta controle a movimentação do robô pela sala, comparando seu desempenho com o do algoritmo heurístico que gerou os dados para treinamento.

A avaliação dos resultados é feita através da taxa de acerto médio no treinamento, calculada a partir das taxas geradas a cada repetição do treinamento. Também são utilizadas taxas de acerto médio nos testes *offline*, o erro quadrático médio (EQM) e a matriz de confusão gerada. O EQM é calculado de acordo com

$$EQM = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{d}(n) - \mathbf{y}(n))^{2}(n).$$
 (3.7)

onde N é a quantidade de dados destinados ao teste, nesse caso, N = 1455; o vetor $\mathbf{d}(n)$ são as saídas desejadas no instante n e o vetor $\mathbf{y}(n)$ são as saídas da rede no mesmo instante.

A matriz de confusão é uma matriz quadrada, com tantas linhas e colunas quanto classes a serem classificadas, neste caso é uma matriz 4×4 , onde as linhas representam as respostas desejadas e as colunas, a resposta da rede. Dessa forma, pode-se deduzir que a quantidade de respostas corretas estão na diagonal principal desta matriz, formada pelos elementos [(1,1);(2,2);(3,3);(4,4)], neste caso.

É importante ressaltar que para todas as redes testadas, independente de onde se posiciona o robô dentro da sala, ele é capaz de dar continuidade ao trajeto. Com exceção apenas se o posicionar muito distante de uma parede, mas isto deve-se à uma limitação no algoritmo utilizado para treino, e não por um aprendizado inadequado por parte da rede neural.

3.5.1 Resultados com a Rede Perceptron Logístico

Nesta subseção estão os resultados do treinamento e teste *offline* da rede perceptron logístico com 2 unidades de entrada (Caso 3) e um resumo dos resultados para o Caso 1 e 2. Mais detalhes destes dois últimos, podem ser encontrados no Apêndice C.

3.5.1.1 Resultados para a Rede PL(2,4)

Nesta subseção são descritos os resultados do experimento com a rede PL que utiliza apenas as duas distâncias resumidas à frente e à esquerda do robô SCITOS para compor o vetor de entrada. Na Tabela 3.1 estão as estatísticas dos testes *offline*, enquanto na Tabela 3.2 estão os resultados da execução da rede PL(2,4) no robô SCITOS G5.

Tabela 3.1: Estatísticas do treinamento *offline* da rede PL(2,4).

PL(2,4)	
Média de acertos	14,12%
Taxa Máxima de acertos	14,12%
Taxa Mínima de acertos	14,12%
Variância dos resultados	0

Tabela 3.2: Resultado dos testes *offline* para a rede PL(2,4).

TESTE PL(2,4)								
Erro Quadrático Médio	0,1612							
Taxa de acerto	76,01%							
Matriz de Confusão		Classes Preditas						
	es	525	0	0	0			
	lass eais	193	14	0	0			
	C &	0	0	534	13			
		69	0	5	33			

A rede PL(2,4) apresenta 14,12% como taxa de acerto, a qual não apresentou nenhuma variação durante as 50 repetições de treinamento/teste. Isso mostra que esta rede está convergindo sempre para a mesma solução, apesar do embaralhamento a cada novo treinamento. Na matriz de confusão na Tabela 3.2 observam-se erros na classificação dos movimentos *curva aberta à direita* e *curva à esquerda*.

Na Figura 3.7 mostram-se o gráfico de dispersão dos dados e a superfície de decisão definida pela rede PL(2,4). Esta arquitetura linear produz a reta em preto da figura. Como é possível observar, há a discriminação de apenas duas classes, embora não seja possível distinguir visualmente quais classes a rede está separando. Ao analisar a matriz de confusão, é possível concluir que sejam as classes *seguir em frente* e *curva fechada à direita*, pois são as que possuem menos padrões erroneamente classificados. Além disso, os padrões que não são corretamente categorizados, são associados a uma dessas duas classes predominantes.

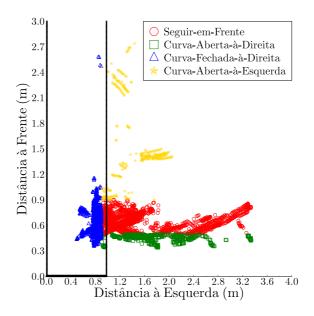


Figura 3.7: Superfície de decisão da rede PL(2,4).

Na Figura 3.8 os pontos pretos representam os obstáculos percebidos com os 24 sensores de ultra-som, que mostram o quão ruidosa é a perspectiva do robô sobre o seu ambiente. A trilha em azul é a trajetória que ele segue durante a volta que realiza pela sala e o círculo em vermelho indica a sua posição final. Durante a navegação com a rede PL(2,4), não houve nenhuma colisão, porém o robô não foi capaz de fazer a curva à esquerda, como é possível inferir pelos resultados da matriz de confusão.

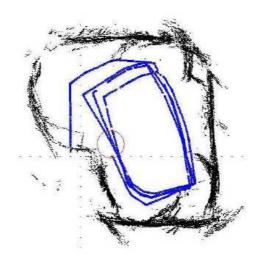


Figura 3.8: Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(2,4).

3.5.1.2 Resultados para as Redes PL(4,4) e PL(24,4)

Nesta subseção encontra-se um resumo dos resultados dos testes realizados com as redes PL(24,4) e PL(4,4). A Tabela 3.3 mostra a média dos acertos do treinamento, na qual é possível observar que a rede PL(24,4) teve o melhor desempenho, embora durante a realização do teste, a rede com 4 unidade de entrada teve uma taxa de 76,49% de acerto, enquanto a que faz uso de 24 teve apenas 28,84%.

		TESTE PL(24,4)			TES	TE	PL(4,4)	
Média de acertos do Treino		16,67%				9,52%			
Taxa de acerto		28,84%				76,49%			
Matriz de Confusão	Classes Preditas Classes Predi			Classes Preditas				Predita	S
	es	144	177	273	0	535	0	59	0
	lasses eais	26	17	164	0	201	6	0	0
ξ		107	183	258	0	0	0	534	13
		48	19	39	1	65	0	4	38

Analisando as matrizes de confusão, percebe-se que ambas as redes tiveram problemas na categorização das classes 2 e 4. Apesar disso, para a rede PL(4,4), ocorre uma quantidade menor de padrões fora da diagonal principal da matriz, ou seja, há menos padrões erroneamente classificados do que a rede PL(24,4).

Nas Figuras 3.9(a) e 3.9(b) são mostradas as trajetórias das duas redes mencionadas acima.

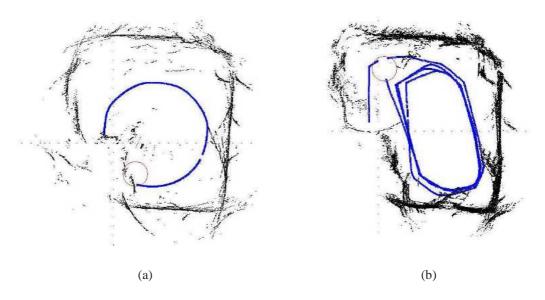


Figura 3.9: Percepção do ambiente e trajetória do SCITOS G5 para a rede a) PL(24,4) e b) PL(4,4).

Com a rede PL(24,4), o robô é quase capaz de realizar uma volta inteira na sala de teste,

colidindo antes de fazer a última curva. Já com a rede PL(4,4), há uma visível melhora no seu desempenho, não só pelo fato de não haver colisões, como foi observado uma tentativa de fazer a curva à esquerda, uma vez que até então o robô não estava realizando esse movimento.

Na subseção seguinte, estão os detalhes dos resultados obtidos com a arquitetura de mistura de especialistas.

3.5.2 Resultados para a Arquitetura de Mistura de Especialistas

Nesta subseção encontram-se os resultados do treinamento e teste da Mistura de Especialistas com 2 unidades de entrada, assim como um resumo sobre os testes com os casos que fazem uso de 4 e 24 sensores. Mais detalhes sobre estes dois últimos, podem ser encontrados no Apêndice D.

3.5.2.1 Resultados para a Rede ME(2,4,4)

Usando as duas distâncias resumidas como entrada da rede, são mostradas na Tabela 3.4 as estatísticas dos testes *offline*, enquanto na Tabela 3.5 é mostrado o resultado do teste *offline* da arquitetura ME(2,4,4).

Tabela 3.4: Estatísticas do treinamento *offline* para a arquitetura ME(2,4,4).

ME(2,4,4)	
Média de acertos	84,22%
Taxa Máxima de acertos	84,22%
Taxa Mínima de acertos	84,22%
Variância dos resultados	0

Tabela 3.5: Resultado dos testes *offline* para a arquitetura ME(2,4,4).

ME(2,4,4)								
Erro Quadrático Médio	0,149403							
Taxa de acerto	76,35%							
Matriz de Confusão		Classes Preditas						
	es	530	0	64	0			
	lasses eais	201	6	0	0			
	S 8	0	0	534	13			
		62	0	4	41			

Pela análise das Tabelas 3.4 e 3.5, verifica-se ainda o mesmo problema com as classes 2 e 4 onde apenas 2,9% $\left(\frac{6}{201+6}\right)$ e 38,31% $\left(\frac{41}{41+4+62}\right)$ dos padrões, respectivamente, são classificados corretamente, em contraste aos 89,22% $\left(\frac{530}{530+64}\right)$ da classe 1 e 97,62% $\left(\frac{534}{534+13}\right)$ da classe 3.

Para a arquitetura ME(2,4,4), o robô efetuou uma trajetória circular, terminando-a com uma colisão antes da curva à esquerda, como é mostrado na Figura 3.10. Dessa forma conclui-se que essa rede só foi capaz de classificar as informações do ambiente como condições para a *curva* à direita. Mesmo ao ser reposicionado um pouco depois do ponto de colisão, onde ele deveria realizar uma curva à esquerda, o robô permaneceu em uma curva aberta à direita.

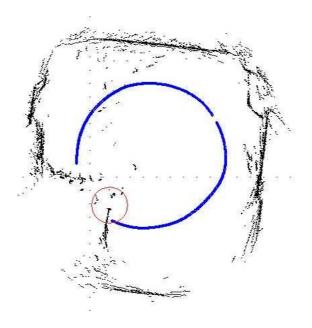


Figura 3.10: Percepção do ambiente e trajetória do SCITOS G5 para a rede ME(2,4,4).

A Figura 3.11 corrobora o comportamento do robô, pois fica claro que as superfícies de decisão geradas colocam todos os padrões em uma única classe, que pelo resultado discutido anteriormente, é possível concluir que seja a classe *curva fechada à direita*.

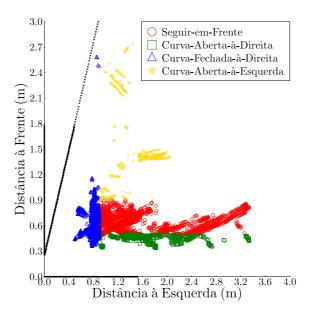


Figura 3.11: Superfície de decisão da rede ME(2,4,4).

3.5.2.2 Resultados para as Arquiteturas ME(4,4,4) e ME(24,4,4)

Nesta subseção encontra-se um resumo sobre os resultados do treinamento e teste *offline* com as redes ME(24,4,4) e ME(4,4,4), mostrados na Tabela 3.6, assim como a descrição do comportamento do robô e as trajetórias geradas por essas duas redes.

		ME(24,4,4)				N	ЛЕ(4,	4,4)	
Média de acertos do Treino		57%					84,35	5%	
Taxa de acerto		9,95%				,	75,8	7%	
Matriz de Confusão		Classes Preditas			Clas	ses P	reditas	S	
5	S	228	177	211	2	523	1	70	0
9	lasses eais	131	31	50	1	195	12	0	0
Ę	Re	395	7	139	0	0	0	534	13
		55	2	26	0	69	0	3	35

Tabela 3.6: Resumo dos resultado dos testes *offline* para as arquiteturas ME(24,4,4) e ME(4,4,4).

Pelos resultados da Tabela 3.6, percebe-se que a rede ME(4,4,4) tem melhor desempenho que a ME(24,4,4) tanto no treinamento, quanto no teste. Isso se reflete na matriz de confusão. Também se observa a dificuldade na discriminação das classes 2 e 4. No geral, a arquitetura ME(4,4,4) tem menos padrões classificados erroneamente.

Nas Figuras 3.12(a) e 3.12(b) são mostradas as trajetórias do robô com as redes ME(24,4,4) e ME(4,4,4). Durante a navegação com a rede ME(24,4,4), o robô colidiu 3 vezes, mostrando que não houve uma boa especialização da tarefa por parte dos especialistas. Por conta das colisões, só foi possível gerar uma imagem da trajetória (ver Figura 3.12(a)).

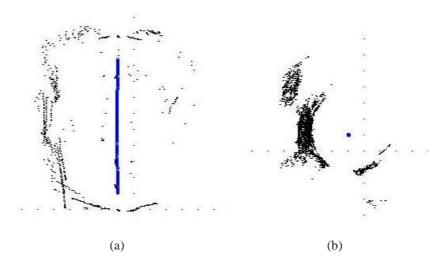


Figura 3.12: Percepção do ambiente e trajetória do SCITOS G5 para a rede a) ME(24,4,4) e b) ME(4,4,4).

Como é possível ser verificado na Figura 3.12(b), o ponto azul mostra que o robô perma-

neceu girando sobre seu próprio eixo, pois realizava somente a *curva fechada à direita*, que é a classe que teve mais padrões corretamente classificados.

Na subseção seguinte estão os resultados com a rede perceptron multicamadas.

3.5.3 Resultados para a Rede MLP

Nesta subseção são detalhados os resultados do treinamento e teste *offline* da rede MLP(2,6,4). Para as redes MLP(4,6,4) e MLP(24,6,4) são feitos apenas resumos dos resultados. Mais detalhes sobre estas últimas, podem ser encontrados no Apêndice A.

Também são discutidos os testes realizados em busca da quantidade mínima de neurônios na camada oculta que permite separar as quatro classes do problema.

3.5.3.1 Dimensão da Camada Oculta

A partir dos resultados anteriores pode-se concluir que as arquiteturas lineares não são capazes de fazer a separação correta das classes dentro do problema abordado. Este fato permite uma antecipação no prognóstico de um melhor desempenho por parte da rede MLP, uma vez que ela é capaz de gerar superfícies de decisão não-lineares. Mas para que a rede MLP possa gerar essas superfícies de forma adequada, é necessário determinar a quantidade de neurônios que ficam na camada oculta e que são responsáveis pela não-linearidade da rede.

Neste trabalho, a quantidade escolhida foi determinada variando o número de neurônios ocultos até encontrar um valor que resultasse em superfícies suficientes para dividir corretamente as classes do problema. Na Figura 3.13 são mostradas as várias superfícies geradas para diferentes quantidades de neurônios ocultos.

A partir dos resultados mostrados na Figura 3.13(d), percebe-se que q=6 é a quantidade mínima que gera superfícies suficientes para separar corretamente as quatro classes. Com 8 neurônios ocultos ainda se obtém uma configuração semelhante, mas a partir desse valor, é possível verificar a ocorrência de *overfitting* a partir das divisões no espaço em que não há dados a serem categorizados.

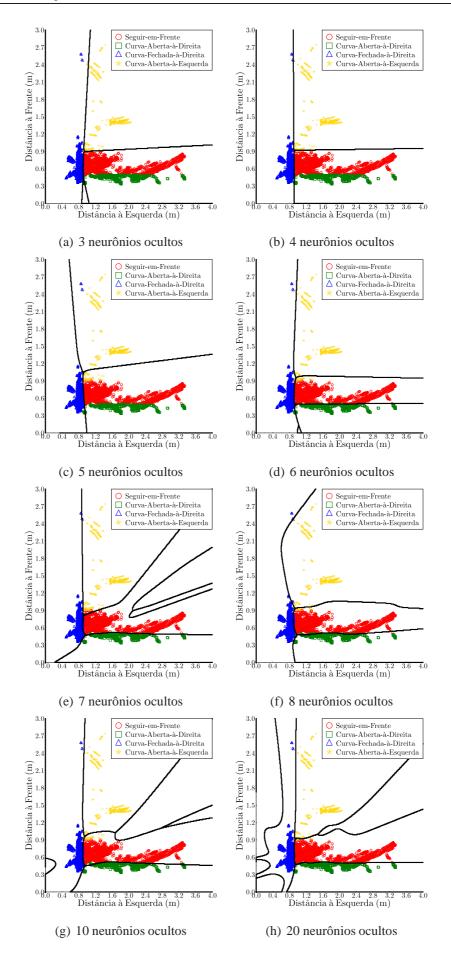


Figura 3.13: Superfícies de Decisão com a Rede MLP com a) 3, b) 4, c) 5, d) 6, e) 7 e f) 8 g) 10 e h) 20 neurônios ocultos

Ao longo deste trabalho, a rede MLP foi avaliada com 6, 8 e 20 neurônios ocultos para verificar se o aumento dos neurônios ocultos causa alguma melhora significativa no desempenho do robô durante a navegação. Neste capítulo aborda-se apenas os testes realizados com 6 neurônios ocultos, os outros resultados podem ser encontrados no Apêndice A.

3.5.3.2 Resultados para a Rede MLP(2,6,4)

Com a MLP que utiliza dois parâmetros de entrada, os resultados da análise inicial com 6 neurônios ocultos estão nas Tabelas 3.7 e 3.8.

Nestas tabelas, observam-se boas taxas de acerto (maiores que 95%) tanto no treinamento quanto no teste. A matriz de confusão mostra uma porcentagem de padrões corretamente classificados de 97,7% (classe 1), 96,4% (classe 2), 99,8% (classe 3) e 85,7% (classe 4).

Tabela 3.7: Estatísticas do treinamento *offline* para a rede MLP(2,6,4).

MLP(2,6,4)							
Média de acertos	96,82%						
Taxa Máxima de acertos	97,95%						
Taxa Mínima de acertos	82,07%						
Variância dos resultados	$9,32 \times 10^{-4}$						

Tabela 3.8: Resultado dos testes *offline* para a rede MLP(2,6,4).

TESTE MLP(2,6,4)								
Erro Quadrático Médio	0.0294259							
Taxa de acerto	97,59%							
Matriz de Confusão		Classes Preditas						
	es	571	7	6	0			
	lass eais	0	213	8	0			
	Z &	1	0	559	0			
		11	0	2	78			

Na Figura 3.14 são mostradas as ativações dos neurônios ocultos da rede MLP(2,6,4) ao longo de uma volta na sala testes, na qual, percebe-se que um neurônio permaneceu com ativações próximas a zero. Apesar deste neurônio não disparar em momento algum, acredita-se que ele possua alguma importância na dinâmica da rede, pelo fato de que com 5 neurônios ocultos, a rede MLP não é capaz de formar superfícies de decisão suficientes para separar todas as classes, como mostra a Figura 3.13(c). Este é um fato a ser investigado com maior profundidade em trabalhos futuros.

Verificou-se ainda que as ativações dos neurônios ocultos não coincidem com nenhuma ação em específico, independente da quantidade de unidades de entrada ou de neurônios ocultos.

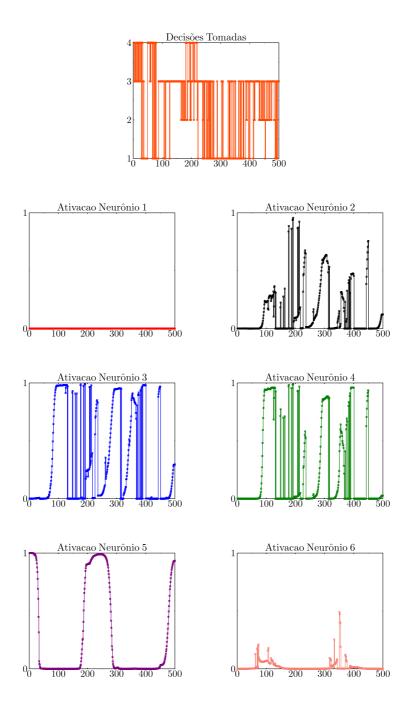


Figura 3.14: Ativações da camada oculta da rede MLP(2,6,4).

A trajetória do robô para a rede MLP(2,6,4) é mostrada na Figura 3.15. Neste caso, não houve colisões e o robô teve ainda menos momentos de hesitação do que os observados com a rede MLP(4,6,4). Essa trajetória gerada é muito próxima a gerada pelo programa baseado em regras, ao ser comparada com a Figura 3.5.

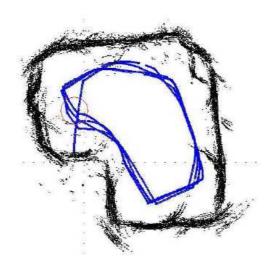


Figura 3.15: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,6,4).

3.5.3.3 Resultados para as Redes MLP(4,6,4) e (24,6,4)

Nesta subseção encontra-se um resumo dos resultados do treinamento e teste *offline* com as redes MLP(24,6,4) e MLP(4,6,4), mostrados na Tabela 3.9, bem como a descrição do comportamento do robô e a trajetória com essas duas redes.

Tabela 3.9: Resultado de	os testes <i>offline</i> para as red	les MLP(24,6,4) e MLP(4,6,4).
---------------------------------	--------------------------------------	-------------------------------

		MLP(24,6,4)				1LP(4	,6,4)	
Média de acertos do Treino		88,78%				84,83	%	
Taxa de acerto		85,92%				97,04	.%	
Matriz de Confusão	C	Classes Preditas				sses Pi	reditas	
e	520	13	19	55	559	9	15	1
asses	S 28	161	3	36	0	228	1	0
—	2 19	6	485	19	12	0	545	0
	2	0	5	85	2	0	3	81

Apesar de durante o treinamento a rede MLP(24,6,4) ter apresentado um resultado melhor, no teste *offline* verifica-se, tanto pela taxa de acerto de 97,04%, como pela matriz de confusão, que a rede MLP(4,6,4) teve um melhor desempenho.

Nas Figuras 3.16(a) e 3.16(b) são mostradas as trajetórias do robô com as redes mencionadas. Com a rede MLP(24,6,4), mostrada na Figura 3.16(a), das quatro voltas que o robô realizou, em apenas uma delas ocorreu colisão. Isso foi devido à uma curva muito fechada à esquerda que foi seguida de uma curva à direita, o que fez com que a cauda do robô encostasse na parede. No restante do percurso, o robô se mostrou muito hesitante e em vários momentos

3.6 Conclusão 68

ficou parado por dois ou três segundos devido a decisões conflitantes. Mesmo assim, o robô foi capaz de fazer a curva à esquerda, que até então não havia sido realizada com sucesso.

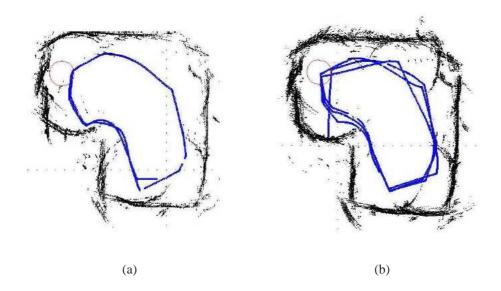


Figura 3.16: Percepção do ambiente e trajetória do SCITOS G5 para a rede a) MLP(24,6,4) e b) MLP(4,6,4)

Com a rede MLP(4,6,4), o robô foi capaz de realizar as voltas na sala com menos momentos de hesitação e sem colisões, demonstrando uma boa imitação do algoritmo heurístico que gerou o comportamento original, como é possível ver na Figura 3.16(b).

3.6 Conclusão

Ao longo dos testes, percebe-se que o acréscimo de unidades de entrada³, em relação às utilizadas no algoritmo heurístico, não traz nenhum benefício à capacidade de classificação das arquiteturas testadas. As informações extras, como o uso de todos os 24 sensores ou de apenas 4 distâncias resumidas, na verdade fizeram com que as redes tivessem um desempenho pior em relação às redes que só fazem uso de apenas 2 distâncias resumidas. Acredita-se que isto se deve à natureza bastante ruidosa dos sinais dos sensores.

Com a rede PL(2,4), observa-se que as taxas de acerto durante o treinamento são um pouco melhores em relação às redes PL(24,4) e PL(4,4), embora o resultado dos testes no robô real não tenham sido muito diferentes. A rede sempre converge para a mesma solução, apesar da aleatoriedade dos pesos a cada novo treinamento. Na matriz de confusão resultante são observados erros na classificação dos movimentos *curva aberta à direita* e *curva à esquerda*.

³Neste caso, o aumento da quantidade de distâncias a serem analisadas.

3.6 Conclusão 69

A arquitetura de ME apresentou problemas para fazer a divisão das quatro classes entre os quatro especialistas e, principalmente, para fazer a classificação das classes 2 e 4, correspondentes às ações menos expressivas durante a navegação.

A rede MLP foi a única das arquiteturas testadas que conseguiu com sucesso fazer a curva à esquerda, dando indícios de tratar-se de um problema não-linearmente separável.

4 RESULTADOS DOS EXPERIMENTOS COM CLASSIFICAÇÃO ESPAÇO-TEMPORAL

4.1 Introdução

Neste capítulo, verifica-se a influência que a introdução de MCD (MCD) tem sobre a capacidade de classificação das redes PL e ME, e se este mecanismo permite que discriminantes lineares alcancem um desempenho similar ao da rede MLP.

A quarta rede neural a ser avaliada nesta dissertação é a rede de Elman. A escolha desta deve-se à estrutura de sua arquitetura, a qual faz uso de MCD ao realimentar as ativações da camada oculta para as unidades de entrada. Desta forma, mecanismos de memória são explorados neste trabalho de duas formas: com o acréscimo de atrasadores nas unidades de entrada para as redes PL e ME, e através da recorrência na rede de Elman. A rede MLP não é incluída neste capítulo, devido aos bons resultados já obtidos e pelo fato de que a rede de Elman tem como arquitetura base uma rede MLP.

Da mesma forma como foi estruturado o Capítulo 4, neste capítulo só são abordados os resultados com 2 distâncias resumidas. Os detalhes dos testes com 4 distâncias resumidas podem ser vistos no Apêndice C para a rede PL e no Apêndice D para a arquitetura ME.

Este capítulo é dividido da seguinte forma. Na Seção 4.2 são descritos os novos vetores de entrada para as redes avaliadas. A Seção 4.3 está dividida na seguintes subseções referentes à rede PL, 4.3.1, à rede ME, 4.3.2 e à rede de Elman na Subseção 4.3.3. Segue-se pela Seção 4.4, onde é feita uma discussão a respeito do custo computacional das redes que tiveram melhor desempenho e, por fim, as conclusões do capítulo na Seção 4.5.

4.2 Definição das Entradas

Nesta seção, são definidas as dimensões dos vetores de entrada para as redes PL, ME e de Elman.

Com a rede PL e ME dotadas de MCD, são realizados 6 e 5 testes, respectivamente, com diferentes quantidades de amostras passadas (τ) para os casos da rede alimentada com duas e quatro distâncias resumidas, mas apenas os testes com duas são apresentados neste capítulo. A escolha das quantidades de amostras é arbitrária, feita apenas no intuito de verificar o comportamento da rede a medida que se aumenta a "memória" dela.

Para os testes com duas distâncias resumidas o vetor de entrada é

$$\mathbf{x} = [x_f(n) \dots x_f(n-\tau) \ x_e(n) \dots x_e(n-\tau)]^T$$
(4.1)

e para o teste com quatro distâncias resumidas o vetor é

$$\mathbf{x} = [x_f(n) \dots x_f(n-\tau) \ x_e(n) \dots x_e(n-\tau) \ x_d(n) \dots x_d(n-\tau) \ x_t(n) \dots x_t(n-\tau)]^T. \tag{4.2}$$

As diferentes quantidades de componentes do vetor de entrada das redes PL e ME para os casos com duas e quatro distâncias resumidas e com diferentes valores de τ podem ser vistas nas Tabelas 4.1 e 4.2.

Tabela 4.1: Quantidade de componentes no vetor \mathbf{x} para cada τ com a rede PL.

Rede PL								
τ	1	3	5	9	19	34		
Dimensão de x:								
Caso com 2 distâncias Caso com 4 distâncias	4	8	12	20	40	70		
Caso com 4 distâncias	8	16	24	40	80	140		

Tabela 4.2: Quantidade de componentes no vetor \mathbf{x} para cada τ com a rede ME.

Rede ME										
τ	1	4	6	9	19					
Dimensão de x:										
Caso com 2 distâncias Caso com 4 distâncias	4	10	14	20	40					
Caso com 4 distâncias	8	20	28	40	80					

Uma observação a ser feita, é o teste que faz uso dos 24 sensores do robô nas unidades de entrada não é aplicado às redes PL e nem para a arquitetura ME. Esta decisão é tomada pelo fato de que a dimensão do espaço de entrada fica muito grande, mesmo com o uso de

poucas amostras passadas de cada um dos sensores. Como os resultados descritos no Capítulo 4 mostram que as redes que fazem uso apenas das distâncias resumidas, têm melhor desempenho que as que utilizam os 24 sensores de ultra-som, então são testado apenas com 4 e 2 dessas distâncias.

A última rede avaliada, rede de Elman, possui um vetor de entrada para o caso com duas distâncias resumidas com a seguinte configuração

$$\mathbf{x} = [x_f(n) \ x_e(n) \ s_1(n-1) \dots s_q(n-1)]^T, \tag{4.3}$$

em que $s_i(n-1)$ são as ativações dos neurônios ocultos no instante anterior e q é o número de neurônios ocultos da rede.

Para o caso com quatro distâncias resumidas, o vetor de entrada é

$$\mathbf{x} = [x_f(n) \ x_e(n) \ x_d(n) \ x_t(n) \ s_1(n-1) \dots s_q(n-1)]^T, \tag{4.4}$$

e para o caso que faz uso das vinte e quatro distâncias, o vetor é

$$\mathbf{x} = [x_1(n) \ x_2(n) \ \dots \ x_{24}(n) \ s_1(n-1) \ \dots \ s_q(n-1)]^T. \tag{4.5}$$

4.3 Resultados dos Experimentos

Nos experimentos que se seguem com o uso de MCD, as mesmas arquiteturas são mantidas, em que a rede PL é formada por uma única camada de quatro neurônios de saída, cada um representando uma ação (classe) diferente: seguir em frente (classe 1), curva aberta à direita (classe 2), curva fechada à direita (classe 3) e curva aberta à esquerda (classe 4). A arquitetura ME é constituída por uma rede seletora formada por uma rede PL de quatro neurônios, mais K = 4 especialistas que também consistem em redes PL com quatro neurônios.

No caso da rede PL, o treinamento é feito com 500 épocas a uma taxa de aprendizagem fixa de $\eta = 0, 2$, e a rede ME com 200 épocas a uma taxa de aprendizagem fixa de $\eta = 0, 2$.

Por fim, a rede de Elman, assim como a rede MLP, passou por uma série de testes afim de conhecer qual a quantidade mínima de neurônios ocultos capaz de resolver o problema. O treinamento da rede de Elman é realizado em 500 épocas a uma taxa de aprendizagem fixa de $\eta=0,05$.

A notação para as redes continua a mesma: PL(I, O) e ME(I, K, O). Para a rede de Elman,

adota-se a seguinte notação é adotada: Elman(I+q,q,O), em que q denota o número de neurônios ocultos.

Da mesma forma que nos testes do capítulo anterior, são realizadas 50 repetições do treinamento/teste de cada rede, sempre reiniciando os pesos aleatoriamente entre [-0,01;0,01] a cada novo treinamento, embaralhando os dados para treino a cada época. Das 5456 amostras coletadas, 4000 são destinadas para treinamento e as 1455 restantes para teste. O conjunto de pesos com melhor desempenho dentre os gerados nas repetições é escolhido para teste.

4.3.1 Resultados da rede PL

Nesta seção, são mostrados os detalhes para a rede PL que faz uso de apenas 2 distâncias resumidas com as quantidades de amostras passadas mencionadas na seção anterior. Nas Tabelas 4.3 e 4.4 são mostradas as taxas médias, máxima e mínima de acerto, determinadas a partir das 50 repetições do treinamento, assim como a variância destes resultados.

Tabela 4.3: Estatísticas do treinamento *offline* da rede PL com memória $\tau = 1, 3$ e 5.

	PL(4,4)	PL(8,4)	PL(12,4)
Média de acertos	13,56%	13,55%	13,55%
Taxa Máxima de acertos	14,47%	14%	14%
Taxa Mínima de acertos	13,55%	13,55%	13,55%
Variância dos resultados	$2,7910 \times 10^{-4}$	$6,75 \times 10^{-6}$	$7,3633 \times 10^{-6}$

Tabela 4.4: Estatísticas do treinamento *offline* da rede PL com memória $\tau = 9,19$ e 34.

	PL(20,4)	PL(40,4)	PL(70,4)
Média de acertos	13,57%	13,58%	13,57%
Taxa Máxima de acertos	14,7%	15,1%	14,77%
Taxa Mínima de acertos	13,55%	13,55%	13,55%
Variância dos resultados	$4,3330 \times 10^{-5}$	$4,805 \times 10^{-6}$	$4,8813 \times 10^{-5}$

Observa-se que não houve uma variação significativa em nenhuma das taxas mostradas nas tabelas mencionadas, embora, dentre os resultados obtidos, a rede PL que teve melhores resultados é a com $\tau = 19$ (i.e., PL(40,4)), com média de acerto de 13,58% e máxima de 15,1%. As redes que tiveram pior desempenho são as que fazem uso de 3 e 5 amostras anteriores, com uma média de 13,55% de acerto e máxima de 14%.

Nas Tabelas 4.5 e 4.6, estão os resultados dos testes *offline*. Ao analisar as taxas de acerto, percebe-se que estas também decrescem à medida que se aumenta o valor de τ , embora, como aconteceu com as redes PL que fazem uso de quatro distâncias resumidas com $\tau = 19$ e 35, essas

taxas não indicam necessariamente que a rede foi bem sucedida em classificar as categorias mais problemáticas (classe 2 e 4).

Tabela 4.5: Resultado dos testes *offline* para a rede PL com $\tau = 1, 3$ e 5.

		PL(4,4)				PL(8,4)				PL(12,4)				
Erro Quadrático Médio			0,175				0,193				0,201			
Taxa de acerto			73,93%			72,58%				71,65%				
Matriz de Confusão		Cla	Classes Preditas			\mathbf{C}	Classes Preditas			S	Classes Preditas			S
	es	522	0	71	0	51	15	0	78	0	481	0	110	0
	Classes Reais	195	11	1	0	19	99	6	2	0	186	19	2	0
	E E	7	0	527	13	1:	5	0	518	12	17	0	515	13
		87	0	5	15	8'	7	0	5	15	74	0	9	24

Tabela 4.6: Resultado dos testes *offline* para a rede PL com $\tau = 9,19$ e 34.

		PL(20,4)				PL(40	0,4)		PL(70,4)				
Erro Quadrático Médio			0,232				0,235			0,252			
Taxa de acerto		67,08%		64,76%				58,97%					
Matriz de Confusão		Cl	Classes Preditas			Cla	Classes Preditas			Classes Preditas			as
	es	287	135	146	19	333	2	183	59	253	0	256	58
	lasses eais	93	110	4	0	117	48	13	29	67	4	28	108
	E E	10	0	522	13	19	0	516	10	18	0	520	2
		47	0	9	51	72	0	2	33	24	0	22	61

No caso da rede PL(4,4), a classe 1 tem 88,02% $\left(\frac{522}{522+71}\right)$ de padrões classificados com sucesso, a classe 2 possui apenas 5,31% $\left(\frac{11}{195+1+11}\right)$, a classe 3 tem 96,34% $\left(\frac{527}{527+13+7}\right)$ e a classe 4 tem 14,01% $\left(\frac{15}{5++15+87}\right)$. Como já foi dito, apesar da taxa de acerto do teste desta rede ter sido a maior dentre as redes perceptron, ela teve um desempenho pobre na classificação das categorias 2 e 4.

A rede PL(8,4) apresentou as seguintes taxas de acerto. Para a classe 1 há 86, 84% $\left(\frac{515}{515+78}\right)$ de exemplos corretamente classificados, a classe 2 tem 2,89% $\left(\frac{6}{6+2+199}\right)$, na classe 3 há 95,04% $\left(\frac{518}{518+12+15}\right)$ e a classe 4 tem 14,01% $\left(\frac{15}{15+5+87}\right)$. Este caso teve em relação ao anterior, uma piora em todas as classes, ficando estável apenas para a classe 4, que é a "curva à esquerda".

Para a rede PL(12,4), a classe 1 apresenta 81,38% $\left(\frac{481}{481+110}\right)$ de padrões corretamente classificados, a classe 2 tem 9,17% $\left(\frac{19}{186+19+2}\right)$, a classe 3 possui 94,49% $\left(\frac{515}{515+13+17}\right)$ e a classe 4 tem 22,42% $\left(\frac{24}{24+74+9}\right)$. As classes mais problemáticas (i.e., classes 2 e 4) já apresentam uma melhora em relação às duas redes anteriores, PL(4,4) e PL(8,4), embora para as classes 1 e 3 tenha ocorrido um decréscimo nas taxas de acerto.

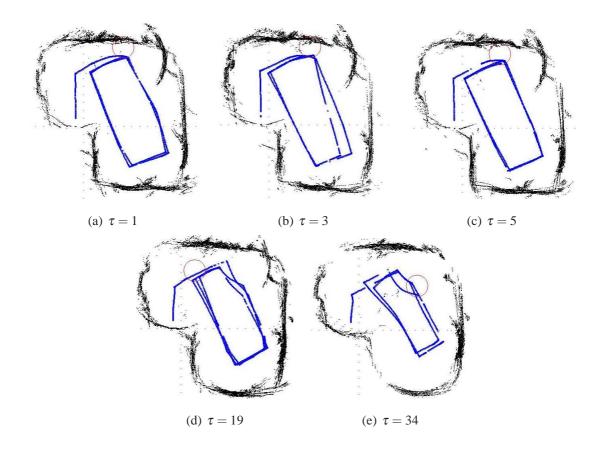


Figura 4.1: Percepção do ambiente e trajetória do SCITOS G5 para a rede PL com a) $\tau = 1$, b) $\tau = 3$, c) $\tau = 5$, d) $\tau = 19$ e e) $\tau = 34$.

No caso da PL(20,4), a classe 1 apresenta 48,89% $\left(\frac{287}{287+135+146+19}\right)$ de exemplos corretamente classificados, a classe 2 tem 53,14% $\left(\frac{110}{110+93+4}\right)$, a classe 3 possui 95,77% $\left(\frac{522}{522+13+10}\right)$ e a classe 4 tem 47,66% $\left(\frac{51}{51+9+47}\right)$. Esta rede tem resultados mais equilibrados, com a maioria dos sucessos das classes em torno de 50%. Em relação às outras redes PL, apenas a classe 1 apresentou decréscimo na sua taxa de acerto.

Com a rede que PL(40,4), a classe 1 tem 57,71% $\left(\frac{333}{333+2+183+59}\right)$ de acertos, a classe 2 possui 23,18% $\left(\frac{48}{48+117+13+29}\right)$, a classe 3 tem 94,67% $\left(\frac{516}{516+10+19}\right)$ e a classe 4 apresenta 30,84% $\left(\frac{33}{33+72+2}\right)$. Analisando estas porcentagens em relação à rede PL(20,4), percebe-se que apenas para a classe 1 teve melhora em seu desempenho.

Por fim, para a rede PL(70,4), a classe 1 apresenta 44,62% $\left(\frac{253}{253+256+58}\right)$ de acerto, a classe 2 tem apenas 1,93% $\left(\frac{4}{28+4+67+108}\right)$, a classe 3 tem 96,29% $\left(\frac{520}{520+18+2}\right)$ e a classe 4 possui 57% $\left(\frac{61}{61+22+24}\right)$ de sucesso.

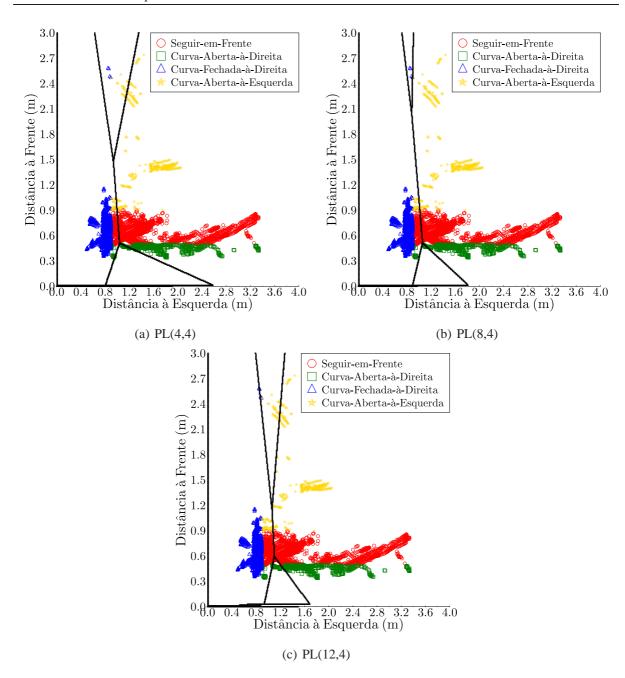


Figura 4.2: Superfícies de decisão da rede perceptron com a) $\tau = 1$, b) $\tau = 3$ e c) $\tau = 5$.

Nas Figuras 4.1 são mostradas as trajetórias do robô para diferentes valores de τ . Nestes casos apresentados, o robô teve comportamento similar ao gerado pela rede perceptron sem memória, diferenciando-se pelos momentos de "hesitação" que se agravam à medida que se aumenta o valor de τ . Esse comportamento deve-se ao aumento da dimensão do vetor de entrada.

As Figuras 4.2 e 4.3 apresentam as superfícies de decisão geradas pelas redes PL com $\tau = 1, 3, 5, 19$ e 35. Comparando-as com as da Figura 3.7, nota-se que existe uma tentativa de formar mais de duas regiões para a classificação dos padrões sensoriais observados.

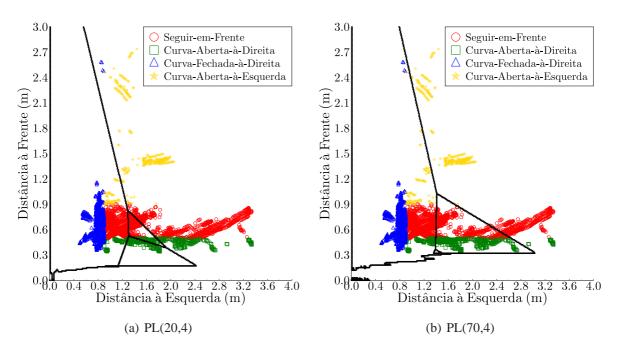


Figura 4.3: Superfícies de decisão da rede perceptron com a) $\tau = 19$ e b) $\tau = 34$

O motivo pelo qual a trajetória da robô com a rede PL(20,4) não está incluída no grupo da Figura 4.1 é por ter sido a única a conseguir realizar a curva à esquerda. A trajetória e a superfície de decisão gerada por essa rede podem ser vistas na Figura 4.4.

É importante ressaltar que o uso de memória, neste caso, permitiu que uma arquitetura linear fosse capaz de classificar a movimentação do robô nos momentos corretos, e consequentemente, imitar de forma razoável o desempenho do algoritmo heurístico.

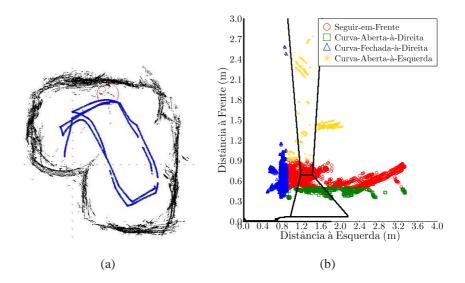


Figura 4.4: a) Percepção do ambiente e trajetória do SCITOS G5 e b) Superfícies de decisão da rede perceptron com $\tau = 9$.

4.3.2 Resultados com a Mistura de Especialistas

Com a arquitetura ME dotada de memória, são realizados 5 testes com diferentes valores de τ para os casos da rede alimentada com duas e quatro distâncias resumidas, das quais apenas a rede que usa duas distâncias resumidas é tratada neste capítulo.

Nesta subseção estão descritos os detalhes dos treinamento e testes *offline* realizados com a arquitetura ME com $\tau = 1,4,6,9$ e 19. As Tabelas 4.7 e 4.8 mostram as taxas de acerto média, máxima e mínima, além da variação delas nas 50 repetições do treinamento.

Tabela 4.7: Estatísticas do treinamento *offline* da rede ME $\tau = 1,4$ e 6.

	ME(4,4,4)	ME(10,4,4)	ME(14,4,4)
Média de acertos	84,25%	86,69%	86,67%
Taxa Máxima de acertos	84,25%	86,7%	86,67%
Taxa Mínima de acertos	84,25%	86,67%	86,67%
Variância dos resultados	0	$2,33 \times 10^{-8}$	0

Tabela 4.8: Estatísticas do treinamento *offline* da rede ME $\tau = 9$ e 19.

	ME(20,4,4)	ME(40,4,4)
Média de acertos	88,32%	88,05%
Taxa Máxima de acertos	88,325%	88,12%
Taxa Mínima de acertos	88,3%	88,05%
Variância dos resultados	$2,04 \times 10^{-8}$	$1,63 \times 10^{-7}$

Ao analisar as taxas médias de acerto, observa-se que a rede que obteve melhor desempenho no treinamento é a ME(210,4,4), com 88,32% de acerto médio. Interessante notar que a a rede PL(20,4) também foi a que se destacou em seu grupo. Vale lembrar que a rede ME é formada por 4 redes PL.

Seguindo a ordem decrescente das taxas de acerto, a que teve melhores resultados depois da rede mencionada anteriormente, é a que faz uso de $\tau = 19$ com 88,05% de sucesso. Seguida pela rede com $\tau = 4$ com 86,69%, pela rede que faz uso de $\tau = 6$ amostras com 86,67% e por último a que utiliza apenas tau = 1 amostra com 84,25%.

As Tabelas 4.9 e 4.10 apresentam os resultados dos testes *offline* realizados com as redes geradas pelo treinamento descrito acima.

A matriz de confusão da rede ME(4,4,4) mostra que a classe 1 tem 89,56% $\left(\frac{532}{532+62}\right)$ de padrões corretamente classificados. A classe 2 apresenta 4,34% $\left(\frac{9}{198+9}\right)$, a classe 3 tem 96,52% $\left(\frac{528}{528+13+6}\right)$ e a classe 4 tem 33,64% $\left(\frac{36}{69+36+2}\right)$ de sucesso. Analisando os resultados, percebese que a classe 2 é a mais problemática.

]	ME(4,4,4)		N	IE(10	,4,4)		ME(14,4,4)					
Erro Quadrático Médio			0,156				0,174				0,178			
Taxa de acerto			75,94%				74,43%			74,57%				
Matriz de Confusão		Classes Preditas		Clas	Classes Preditas				Classes Preditas					
	es	532	0	62	0	520	1	73	0	512	3	79	0	
	Classes Reais	198	9	0	0	183	24	0	0	168	39	0	0	
	E C	6	0	528	13	17	2	516	12	19	2	514	12	
		69	0	2	36	80	0	4	23	83	0	4	20	

Tabela 4.9: Resultado dos testes *offline* da rede ME $\tau = 1.4$ e 6.

Tabela 4.10: Resultado dos testes *offline* da rede ME com memória $\tau = 9$ e 19.

		ME(20,4,4)				M	ME(40,4,4)				
Erro Quadrático Médio			0, 1	187			0,201				
Taxa de acerto			73,	74%			72,3%				
Matriz de Confusão		Classes Preditas				Clas	Classes Preditas				
	es	509	6	79	0	509	9	76	0		
	lasses eais	176	31	0	0	164	43	0	0		
		22	2	510	13	42	1	491	13		
		80	0	4	23	98	0	0	9		

O caso com $\tau=4$ (ME(10,4,4)) apresenta a classe 1 com 87,54% $\left(\frac{520}{520+1+73}\right)$ de taxa de acerto, a classe 2 com 11,59% $\left(\frac{24}{24+183}\right)$, a classe 3 tem 94,33% $\left(\frac{516}{516+12+2+17}\right)$ e a classe 4 com 21,49% $\left(\frac{23}{23+4+80}\right)$. Em relação à rede mencionada anteriormente, a classe 2 teve mais sucesso, embora todas as outras tenham sofrido uma pequena diminuição em suas taxas.

Para a rede ME(14,4,4), as taxas da classe 1 continuaram a diminuir, caindo para 86,19% $\left(\frac{512}{512+3+79}\right)$ de sucesso, a classe 2 tem uma pequena melhora de 18,84% $\left(\frac{39}{168+39}\right)$, a classe 3 tem 93,96% $\left(\frac{514}{514+2+12+19}\right)$ e a classe 4 mostra 18,69% $\left(\frac{20}{20+4+83}\right)$.

A ME(20,4,4) teve 85,69% $\left(\frac{509}{509+6+79}\right)$ de sucesso para a classe 1, 14,97% $\left(\frac{31}{31+176}\right)$ para a classe 2, que sofreu um pequeno decréscimo em relação à rede anterior. A classe 3 apresentou 93,23% $\left(\frac{510}{510+13+2+22}\right)$ de padrões corretamente classificados e a classe 4 tem 21,49% $\left(\frac{23}{23+4+80}\right)$.

Por fim, a ME(40,4,4) tem a classe 1 com 85,69% $\left(\frac{509}{509+9+76}\right)$ de acerto, a classe 2 apresenta 20,77% $\left(\frac{43}{43+164}\right)$, a classe 3 tem 89,76% $\left(\frac{491}{491+13+1+42}\right)$ e a classe 4 tem 8,41% $\left(\frac{9}{9+98}\right)$.

As superfícies de decisão podem ser vistas nas Figuras 4.5 e 4.6 onde é possível perceber que as classes não são bem divididas.

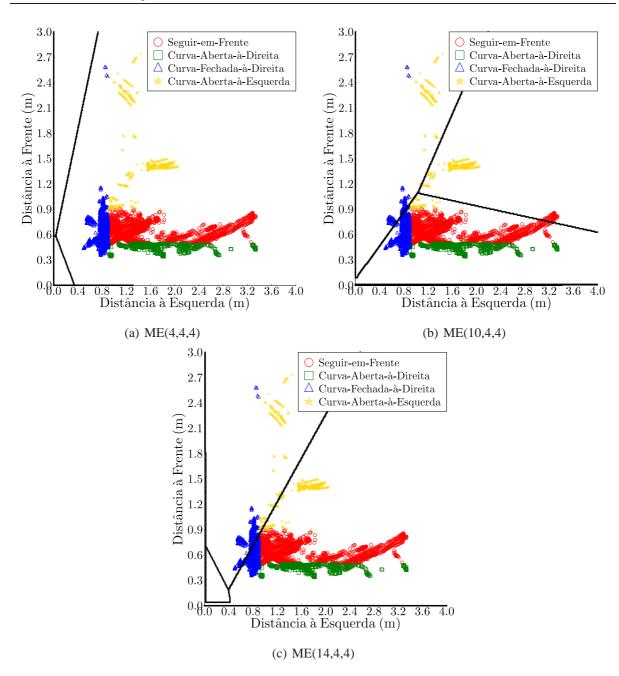


Figura 4.5: Superfícies de decisão da arquitetura ME com a) $\tau = 1$, b) $\tau = 4$, c) $\tau = 6$.

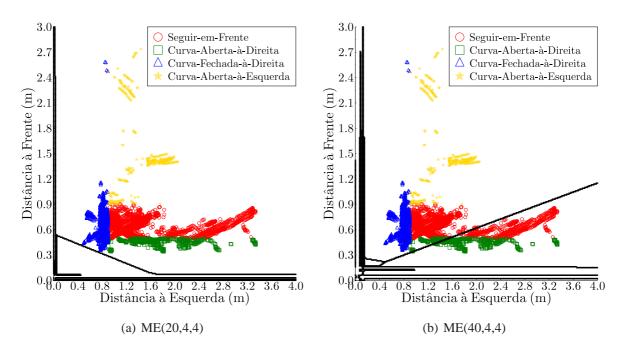


Figura 4.6: Superfícies de decisão da arquitetura ME com a) $\tau = 9$ e b) $\tau = 19$.

Nas Figuras 4.7 e 4.9 são mostradas as trajetórias que o robô realizou com as redes descritas anteriormente.

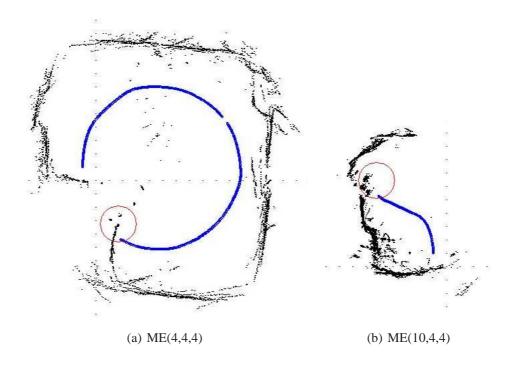


Figura 4.7: Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com a) $\tau = 1$, b) $\tau = 4$.

A Figura 4.7(a) mostra a trajetória do robô com a rede ME(4,4,4), em que o robô efetuou uma trajetória circular, terminando-a com uma colisão antes da curva à esquerda. Isso mostra

que essa rede, para essas configurações, só é capaz de classificar as informações do ambiente como condições para a curva à direita e, em raros momentos, seguindo em frente.

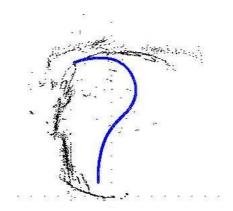


Figura 4.8: Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com $\tau = 6$.

Com a rede ME(10,4,4), o robô iniciou seus movimentos com uma curva à esquerda, mas não foi capaz de realizar a curva à direita para evitar a parede, como mostra a Figura 4.7(b), e com $\tau = 6$ o robô começou seguindo em frente, mas no lugar de fazer uma curva à direita, fez uma à esquerda e colidiu com a parede.

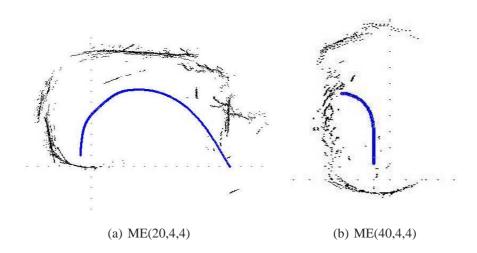


Figura 4.9: Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME com a) $\tau = 9$ e b) $\tau = 19$.

Na Figura 4.9(a) é mostrada a trajetória do robô com ME(20,4,4), na qual começou os movimentos corretamente, mas colidiu por não fazer uma curva fechada o suficiente. Já o teste da ME(40,4,4), o comportamento foi similar ao caso da ME(10,4,4), em que o robô inicia com uma curva à esquerda e colide em seguida.

Dentre os resultados obtidos com a arquitetura ME, nenhuma combinação desta arquitetura,

com ou sem memória consegue dividir as classes entre as redes especialistas. Pelas matrizes de confusão, é possível perceber a dificuldade em aprender corretamente as classes *curva aberta à direita* e *curva à esquerda*, que são as classes com menos exemplos em relação às outras duas. Mas apesar de não ter sido bem sucedido em imitar o comportamento original e ainda ocorrer colisões, o uso de memória permitiu que o robô conseguisse realizar outros movimentos durante a sua navegação.

4.3.3 Resultados para a Rede de Elman

Como já foi mencionado, a rede de Elman não faz uso de amostras passadas das unidades de entrada como recurso de memória, mas sim das ativações passadas de sua camada oculta. Nesta seção são mostrados os resultados dos treinamentos e testes *offline*, assim como o comportamento do robô com 3,4,5 e 6 neurônios ocultos, e com duas distâncias resumidas como entradas. Detalhes sobre os testes com 24 entradas e 4 entradas resumidas estão no Apêndice D.

4.3.3.1 Determinação da Dimensão da Camada Oculta

A escolha do tamanho da camada oculta é feita da mesma forma que com a rede MLP, variando o número de neurônios ocultos até encontrar um valor que gerasse superfícies de decisão suficientes para dividir corretamente as classes do problema. Iniciou-se com a quantidade mínima encontrada para a rede MLP, q=6 neurônios, diminuindo-a para verificar se a rede de Elman funcionaria com menos neurônios. A Figura 4.10 mostra as diferentes superfícies geradas pela rede de Elman para quantidades diferentes de neurônios ocultos.

Observando as várias superfícies de decisão obtidas, e também o comportamento do robô enquanto navega com essas diferentes redes, percebe-se que a rede de Elman com 4 neurônios ocultos (Elman(2+4,4,4)) divide o espaço de entrada de forma semelhante à rede MLP, resultando em comportamentos também semelhantes.

Os resultados obtidos no treinamento e teste *offline*, além do comportamento do robô com as redes de Elman com q = 2, 3, 5 e 6 neurônios ocultos, podem ser encontrados no Apêndice D.

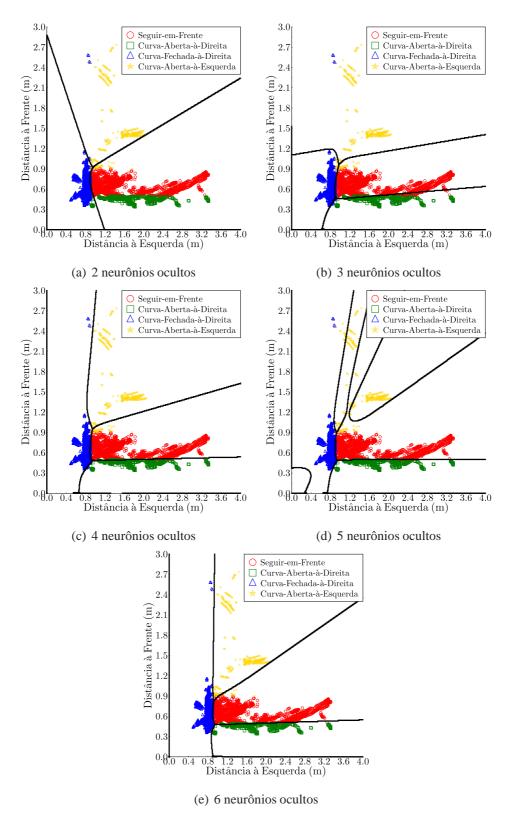


Figura 4.10: Superfícies de decisão com a rede de Elman com a)q = 2, b)q = 3, c)q = 4, d)q = 5, e e)q = 6 neurônios ocultos

4.3.3.2 Resultados para rede de Elman(2+4,4,4)

A rede de Elman com apenas duas distâncias resumidas como entradas, tem os resultados dos treinamentos e testes *offline* mostrados nas Tabelas 4.11 e 4.12.

Tabela 4.11: Estatísticas do treinamento *offline* da rede de Elman(2+4,4,4)

Elman(2+4,4,4)								
Média de acertos	85,16%							
Taxa Máxima de acertos	96,42%							
Taxa Mínima de acertos	75,52%							
Variância dos resultados	0,00390116							

Tabela 4.12: Resultado dos testes *offline* da rede de Elman(2+4,4,4).

	Elman(2+4,4,4)								
Erro Quadrático Médio	0,0555338								
Taxa de acerto	96,22%								
Matriz de Confusão	578	9	7	0					
	10	197	0	0					
	6	3	539	0					
	17	0	3	87					

Comparando os resultados da Tabela 4.11 com os obtidos para a da rede MLP(2,6,4) no capítulo anterior, percebe-se que os primeiros são um pouco mais baixos. As taxas de acerto no teste da rede de Elman(2+4,4,4) e a MLP(2,6,4) são bem mais próximas, com uma diferença em torno de 1%.

A matriz de confusão mostra que a classe 1 tem 97,3% $\left(\frac{578}{578+9+7}\right)$ de acerto, a classe 2 apresenta 95,16% $\left(\frac{197}{197+10}\right)$, a classe 3 tem 98,35 $\left(\frac{539}{539+3+6}\right)$ e a classe 4 tem 81,3% $\left(\frac{87}{87+3+17}\right)$. As boas taxas de acerto indicam que o robô não terá problemas durante a navegação. Na Figura 4.10(c) são mostradas as superfícies de decisão geradas por esta rede, e como é possível observar, elas são muito parecidas com a rede MLP(2,6,4) que teve excelente desempenho no problema de navegação formulado como um problema de classificação de padrões estáticos.

A Figura 4.11 mostra a trajetória do robô com a rede discutida aqui. Neste caso, como a matriz de confusão permitiu antecipar, o robô navegou corretamente, sem colisões e conseguindo realizar todos os movimentos nos momentos corretos e com pouca hesitação. Além disso, seu desempenho e trajetória são muito parecidos com a da rede MLP(2,6,4) e do próprio programa baseado em heurísticas destinado à coleta de dados para treinamento.

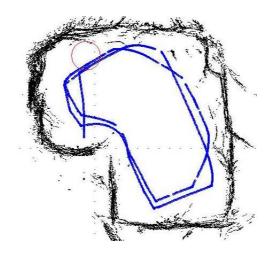


Figura 4.11: Percepção do ambiente e trajetória do SCITOS G5 para a rede de Elman(2+4,4,4).

4.4 Desempenho × Custo Computacional

Ao longo dos Capítulos 3 e 4, algumas redes conseguiram se destacar das demais ao conseguir imitar corretamente o comportamento do programa que gerou os dados de treinamento: a rede MLP com 6 neurônios ocultos, independente da quantidade de entradas utilizadas, a rede PL(20,4) e a rede de Elman(2+4,4,4) distâncias resumidas na entrada. Isso deve-se por serem capazes de classificar corretamente os movimentos adequados de acordo com a sua percepção do ambiente.

Em termos de taxas de acerto médio, a rede MLP(2,6,4) teve o melhor desempenho com 97,59%, seguida pela rede MLP(4,6,4) com 97,04%, depois pela rede de Elman(2+4,4,4) com 96,22%, seguida pela MLP(24,6,4) com 85,2%, depois pela Elman(4+4,4,4) com 96% e, por último, a rede PL(20,4), com 67,08% (ver Figura 4.12).

Um outro aspecto que deve ser avaliado é o custo computacional dessas redes. Uma vez que todas elas apresentam resultados semelhantes durante a navegação, é interessante utilizar a rede que exige menos recursos computacionais do robô. Dessa forma, o robô leva menos tempo na tomada de decisão e, consequentemente, é mais rápido para reagir em caso de uma mudança drástica no ambiente, tal como o aparecimento súbito de um obstáculo.

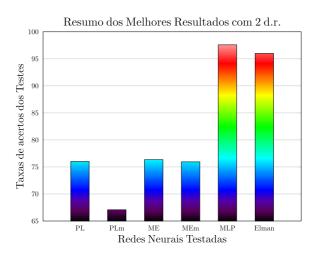


Figura 4.12: Gráfico das taxas de acerto dos testes das redes PL, PL com memória (PLm), ME, ME com memória (MEm), MLP e Elman

Para o cálculo do custo é levado em consideração o número de operações matemáticas (multiplicações e adições) e a existência ou não de execuções das funções sigmóides logísticas realizadas pelas redes. O cálculo do número de operações para a rede PL com memória é dado por

$$c = [p(\tau+1)+1] + (p\tau)q + q = q[2p(\tau+1)+1] + 1$$
(4.6)

em que p é a dimensão do vetor de entrada, τ é a quantidade de amostras passadas por unidade de multiplicações, a segunda parcela, $(p\tau)$ à quantidade de somas. Como estas duas últimas são as mesmas para todos os neurônios, então a soma dessas quantidades é multiplicada pelo número de neurônios. Por fim, é acrescentada a quantidade de funções de transferência, que é a mesma dos neurônios.

O cálculo para a rede MLP é dado por

$$c = q[(2p+1)+1] + M[(2q+1)+1], (4.7)$$

em que p é a dimensão do vetor de entrada, q é a quantidade de neurônios ocultos e M é a quantidade de neurônios na saída. A primeira parcela da equação, delimitada por colchetes, calcula a quantidade de operações de soma e multiplicação da camada oculta. Em seguida, é somado a quantidade de funções de transferência da mesma camada, depois pela quantidade de somas e multiplicações da camada de saída e, por último, a quantidade de funções de transferência da mesma.

4.5 Conclusão 88

Por fim, para a rede de Elman

$$c = q[2(p+q)+1]+1+M[(2q+1)+1], (4.8)$$

na qual percebe-se que a única diferença em relação ao cálculo da rede MLP, é que se acrescenta ao número de entradas, o número de saídas da camada oculta.

Redes	Custo
MLP(2,6,4)	92
Elman(2+4,4,4)	96
Elman(4+4,4,4)	112
MLP(4,6,4)	116
PL(20,4)	168
MLP(24,6,4)	356

Tabela 4.13: Resultado do cálculo dos custos computacionais.

Sendo assim, os resultados encontram-se na Tabela 4.13, na qual percebe-se que a rede MLP(2,6,4) é a que tem menos operações dentre as bem sucedidas. Por essa característica, mais o fato de ter a melhor taxa de acerto no teste *offline* e imitar muito bem a navegação e trajetória do programa baseado em heurísticas conclui-se que a rede MLP com 6 neurônios na camada oculta, 4 na de saída e 2 distâncias resumidas na camada de entrada, é a melhor dentre as testadas ao considerar o item custo computacional estritamente.

Se, por outro lado, for feita uma analogia da rede neural como o "cérebro" do robô, a rede de Elman(2+4,4,4) executa a mesma operação tão bem quanto a rede MLP(2,6,4), porém usando menos neurônios.

4.5 Conclusão

Neste capítulo são mostrados os resultados do desempenho das redes PL e mistura de especialistas acrescidas de estruturas de memória, em que amostras passadas recentes dos padrões de entrada são alimentados junto com as amostras atuais. Além dessas redes, é apresentado também a rede de Elman, a qual também utiliza uma estrutura de memória, mas diferente das outras mencionadas, ela realimenta as ativações da camada oculta para a camada de entrada.

Observou-se que, para a rede com mistura de especialistas, o acréscimo de amostras passadas das distâncias resumidas também aumentou a taxa de acerto de uma das classes mais problemáticas, que é a classe referente ao movimento de curva aberta à direita. Em consequência, as classes restantes tiveram suas taxas de sucesso diminuídas, o que comprometeu o desempe4.5 Conclusão 89

nho geral da rede durante a navegação. Apesar desse ponto negativo, apenas quando inserido a memória nesta arquitetura, o robô foi capaz de realizar mais movimentos antes de colidir, diferente dos resultados vistos na Subseção 3.5.2.

Com relação às redes que obtiveram sucesso neste problema, observa-se que o acréscimo de memória permitiu uma melhor separação das classes por parte das superfícies de decisão, mesmo no caso de uma arquitetura linear, como a rede PL, como pode ser observado nas Fig. 3.7 e 4.4(b). Na última Figura, percebe-se uma melhora na distinção das quatro classes em relação às superfícies geradas pela mesma rede sem o uso de memória. Já as redes MLP (ver Fig. 3.13(d)) e Elman (ver Fig. 4.10(c)) possuem uma distribuição semelhante das superfícies de decisão, embora a rede de Elman possua dois neurônios a menos em sua camada oculta.

5 CONCLUSÕES E PERSPECTIVAS

Este trabalho comparou os desempenhos de quatro arquiteturas neurais usadas como controladoras da navegação ao estilo *wall-following* de um robô móvel real. Para isso, a tarefa de navegação foi formulada como um problema de classificação de padrões, em que os padrões consistem nas leituras sensoriais e as classes em ações a serem tomadas pelo robô. Foram avaliadas a rede PL, a arquitetura ME, com especialistas formados por redes PL, MLP e a rede recorrente de Elman.

Como resultado dessa análise, verificou-se que uma tarefa de aparentemente simples de navegação, constitui-se na verdade em uma complexa tarefa de tomada de decisões, e que o acréscimo do histórico das leituras sensoriais à rede linear PL permitiu que esta fosse capaz de realizar com sucesso uma tarefa de classificação não-linear.

Percebeu-se também que a introdução de mais dados sobre o ambiente, como o uso das 24 distâncias dos sensores e das 4 distâncias resumidas, não trouxe nenhuma vantagem para o classificador, degradando a capacidade de classificação deste. Como neste trabalho foram utilizados os dados sem nenhum pré-processamento, o acréscimo de informação também gerou um acréscimo no nível de ruído para o classificador, contribuindo para as baixas taxas de acerto.

Sobre a rede PL sem estrutura de memória, os resultados dos testes *offline* e da navegação mostram que esta rede não consegue identificar todas as classes disponíveis. As categorias de ação mais problemáticas são: a *curva aberta à direita* e a *curva à esquerda*. Em relação às taxas médias de acerto e ao desempenho da navegação, a rede PL(4,4) se mostra melhor que as outras devido a uma única tentativa de realizar o movimento curva à esquerda, que não é executado pelas outras.

Com o uso de MCD através do acréscimo de 9 amostras passadas nas unidades de entrada, a rede PL, que utiliza 2 distâncias resumidas, foi capaz de reconhecer todas as quatro classes de ações do problema. Interessante observar que a essa rede nessas configurações não apresentou o melhor resultado em relação às taxas de acerto (ver Figura 5.1).

Na tentativa de verificar se o aprendizado local mostrar-se-ia melhor quem uma única PL

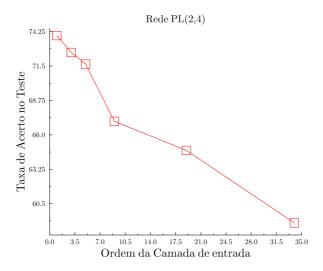


Figura 5.1: Gráfico das taxas de acerto dos testes em relação à quantidade de amostras passadas (rede PL).

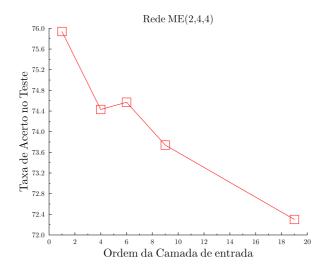


Figura 5.2: Gráfico das taxas de acerto dos testes em relação à quantidade de amostras passadas (arquitetura ME).

com método de aprendizagem local, foram feitos testes com a arquitetura ME. Esta era constituída de quatro redes PL que são responsáveis por aprender uma das 4 ações disponíveis. Pelos testes realizados, percebeu-se que a especialização nos diferentes movimentos não foi sendo bem sucedida, uma vez que esta rede só estava sendo capaz de categorizar os dados em apenas uma única classe, como visto nas superfícies de decisão.

Com o acréscimo de memória, apesar de o robô ainda não ter conseguido realizar uma volta na sala sem colidir, foi possível observar outros movimentos sendo realizados. Na Figura 5.2 é mostrada a relação das taxas de acerto obtidas versus a dimensão do vetor de entrada. Da mesma forma que a rede PL, as taxas de acerto da rede ME também tendem a decair à medida que a dimensão aumenta.

Uma vez testadas redes lineares com métodos de aprendizagem globais e locais, são testadas arquiteturas não-lineares; ou seja, rede MLP e rede recorrente de Elman.

A rede MLP não apresentou dificuldades para a separação das classes independente da quantidade de unidades de entrada utilizados, contanto que a quantidade mínima de neurônios na sua camada oculta seja igual a 6. Já para a rede de Elman, essa quantidade pode ser reduzida a 4 neurônios ocultos, obtendo, mesmo assim, desempenho semelhante à MLP.

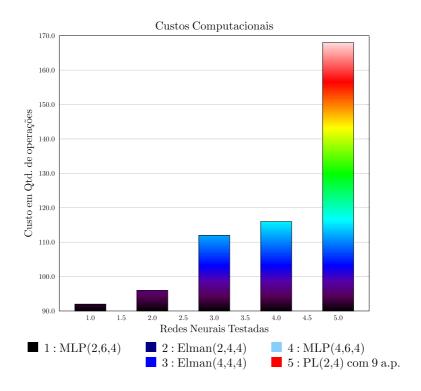


Figura 5.3: Custo Computacional

Por conta do custo computacional mais baixo, mais o fato de ter a melhor taxa de acerto no teste *offline* e imitar muito bem a navegação e trajetória do programa baseado em heurísticas conclui-se que a rede MLP com 6 neurônios na camada oculta, 4 na de saída e 2 distâncias resumidas na camada de entrada, é a melhor dentre as testadas ao considerar o item custo computacional estritamente.

Se, por outro lado, for feita uma analogia da rede neural como o "cérebro" do robô, a rede de Elman(2+4,4,4) executa a mesma operação tão bem quanto a rede MLP(2,6,4), porém usando menos neurônios.

Outra contribuição deste trabalho será a disponibilização dos dados coletados para a comunidade científica, e em trabalhos futuros pretende-se verificar como a coordenação sensóriomotora pode melhorar o desempenho dos classificadores, investigar o desempenho de classificadores SVM na navegação do tipo *wall-following*, bem como avaliar outros tipos de mecanismos de memória de curta duração, tais como o implementado pela arquitetura *Echo-state network* (ANTONELO et al., 2008b).

APÊNDICE A - Testes extras com a rede MLP

Neste apêndice são apresentados os resultados da rede MLP com 6,8 e 20 neurônios na camada oculta e 2,4 e 24 unidades de entrada. No caso de 6 neurônios ocultos, são mostrados apenas os resultados com 4 e 24 padrões de entrada.

A.1 Resultados da rede MLP(24,6,4)

Esta seção apresenta os detalhes dos testes realizados com a rede MLP com 6 neurônios ocultos e que faz uso das distâncias informadas pelos 24 sensores de ultra-som como padrão de entrada.

Tabela A.1: Estatísticas do treinamento *offline* da rede MLP(24,6,4).

MLP(24,6,4)				
Média de acertos	88,78%			
Taxa Máxima de acertos	89,45%			
Taxa Mínima de acertos	86,15%			
Variância dos resultados	$4,41 \times 10^{-5}$			

Tabela A.2: Resultado dos testes *offline* da rede MLP(24,6,4).

TESTE MLP(24,6,4)					
Erro Quadrático Médio		0	.0817		-
Taxa de acerto		85	5,92%		
Matriz de Confusão		Cl	asses I	Predita	S
	es	520	13	19	55
	asso eais	28	161	3	36
	C S	19	6	485	19
		2	0	5	85

Pelas Tabelas A.1 e A.2 verifica-se que as taxas de acerto são mais altas que as das redes perceptron logístico e mistura de especialistas, e que as classes 2 e 4 já possuem mais padrões

corretamente classificados, 70,6% e 92,4% respectivamente, e as classes 1 e 3 possuem 85,6% e 90%.

Na Figura A.1, observa-se as ativações da camada oculta ao longo de uma volta do robô no seu ambiente de testes. A imagem com nome "Decisões Tomadas" indica qual a classe que foi escolhida num determinado período de tempo, e é possível verificar qual a intensidade da ativação dos neurônios ocultos no mesmo período em outras imagens. É interessante notar que a ativação máxima dos neurônios ocultos não coincidem com a escolha de nenhuma classe em específico.

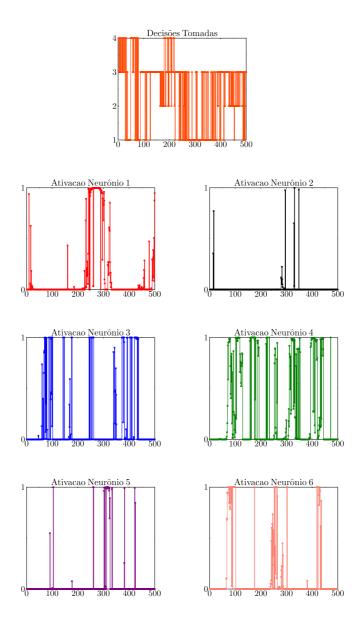


Figura A.1: Ativações da camada oculta da rede MLP(24,6,4).

A respeito da navegação das quatro voltas que o robô realizou, em apenas uma delas ocorreu

uma colisão. Isso foi devido à uma curva muito fechada à esquerda que foi seguida de uma curva à direita, o que fez com que a pequena cauda do robô encostasse na parede. No restante do percurso, o robô se mostrou muito hesitante e em vários momentos ficou parado por uns dois ou três segundos, devido a decisões conflitantes, mas foi capaz de fazer a curva à esquerda, que até então não havia sido realizada com sucesso. A trajetória do robô para a rede MLP(24,6,4) é mostrada na Figura A.2.

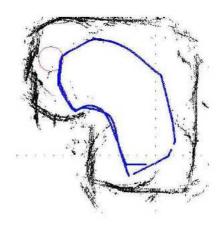


Figura A.2: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,6,4).

A.2 Resultados para MLP(4,6,4)

Nesta seção, são mostrados os resultados com a rede MLP(4,6,4), que podem ser vistos nas Tabelas A.3 e A.4. É possível observar uma melhora na taxa máxima de acerto e no erro quadrático médio, em relação ao teste realizado com 24 parâmetros de entrada.

Tabela A.3: Estatísticas do treinamento textitoffline da rede MLP(4,6,4).

MLP(4,6,4)				
Média de acertos	84,83%			
Taxa Máxima de acertos	96,27%			
Taxa Mínima de acertos	82,37%			
Variância dos resultados	$6,44 \times 10^4$			

Pela Tabela A.4 é possível observar que a matriz de confusão tem pouquíssimos elementos fora da diagonal principal, o que indica uma boa capacidade de classificação. Neste caso são 95,7% de padrões corretamente classificados na classe 1, 99,5% na classe 2, 99,8% na classe 3 e 94,8% na classe 4. Observa-se também que a rede MLP(4,6,4) não apresentou problemas com as classes 2 e 4, como as redes lineares apresentaram, e um melhor desempenho em relação à rede MLP(24,6,4).

abela A.4: Resultado dos testes <i>offline</i> da rede MLP(4,6,4					
TESTE MLP(4,6,4)					
Erro Quadrático Médio		0	.0488		
Taxa de acerto		97	7,04%		
Matriz de Confusão		Cl	asses I	Predita	S
	es	559	9	15	1
	lasses eais	0	228	1	0
	C. S.	12	0	545	0
		2	Ω	2	01

Tabela A.4: Resultado dos testes *offline* da rede MLP(4,6,4).

Da mesma forma que a rede mencionada anteriormente, as ativações dos neurônios ocultos não mostram nenhuma relação com alguma ação em específico, mas diferente das redes vistas até então, este caso não apresentou nenhum neurônio cuja ativação fosse totalmente em torno de zero.

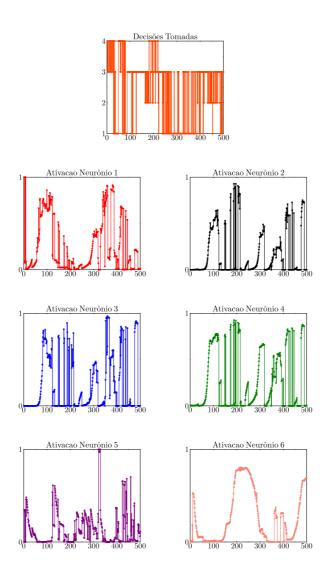


Figura A.3: Ativações da camada oculta da rede MLP(4,6,4).

Durante a navegação o robô foi capaz de realizar as voltas na sala com menos momentos de hesitação e sem colisões, demonstrando uma boa imitação do código que gerou o comportamento original. A trajetória do robô com a rede MLP(4,6,4) é mostrada na Figura A.4.

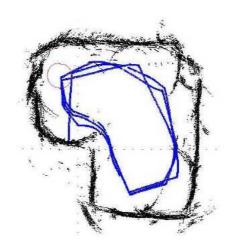


Figura A.4: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,6,4).

A.3 Resultados do Estudo com Diferentes Quantidades de Neurônios Ocultos

Nesta seção, são apresentados os resultados da rede MLP com 8 e 20 neurônios na camada oculta, para os casos com 24,4 e 2 unidades de entrada.

A.3.1 Resultados para MLP(24,q,4)

Nesta subseção são descritos o resultados dos testes da rede MLP com 8 e 20 neurônios ocultos para o caso que faz uso das distâncias encontradas pelos 24 sensores do robô.

A.3.1.1 MLP(24,8,4)

A análise sobre o treinamento e teste textitoffline com rede com 8 neurônios ocultos e 24 unidades de entrada, é mostrado nas Tabelas A.5 e A.6.

Por essas tabelas, percebe-se que as taxas de acerto são menores do que as da rede MLP(24,6,4). Da mesma forma, a quantidade de padrões corretamente classificados diminuíram: 86,6% $\left(\frac{505}{505+14+52+12}\right)$ da classe 1, 82,7% $\left(\frac{191}{191+32+5+3}\right)$ da classe 2, 87,6% $\left(\frac{494}{494+2+1+67}\right)$ da classe 3 e 57,7% $\left(\frac{45}{45+10+23}\right)$ da classe 4.

Tabela A.5: Estatísticas do treinamento textitoffline da rede MLP(24,8,4).

MLP(24,8,4)				
Média de acertos	86,84%			
Taxa Máxima de acertos	87,45%			
Taxa Mínima de acertos	82,62%			
Variância dos resultados	6.90765e - 05			

Tabela A.6: Resultado dos testes *offline* da rede MLP(24,8,4).

TESTE MLP(24,8,4)					
Erro Quadrático Médio		(), 101		
Taxa de acerto		84	1,82%		
Matriz de Confusão		Cl	asses I	Predita	S
	es	505	14	52	12
	asso	32	191	5	3
	Z &	67	1	494	2
		23	0	10	45

Com relação à navegação do robô, nas 6 voltas realizadas utilizando esta arquitetura, nas duas primeiras, ele não só não foi capaz de realizar à curva à esquerda, mostrado na Figura A.5(a), como também colidiu com a quina. Mas durante as quatro voltas restantes, ele conseguiu fazê-las sem colidir e, ainda, foi capaz de fazer uma curva à esquerda mais fechada do que todos os testes realizados, como mostra da Figura A.5(b)).

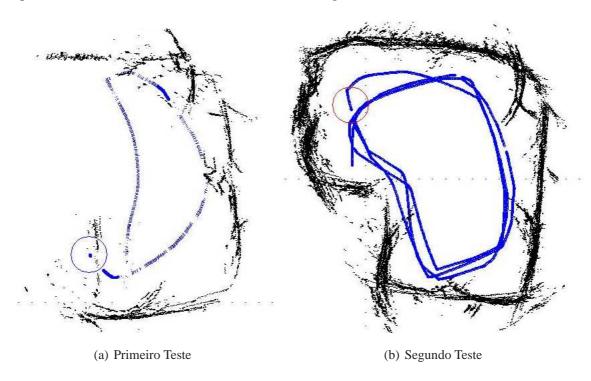


Figura A.5: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,8,4)

Na Figura A.6 verificam-se as ativações da camada oculta ao longo de uma volta do robô no seu ambiente de testes, com a rede MLP(24,20,4). Outra vez, é possível observar a falta de coincidência entre as ativações e uma ação em específico, e a presença de um único neurônio cujas ativações são muito próximas de zero.

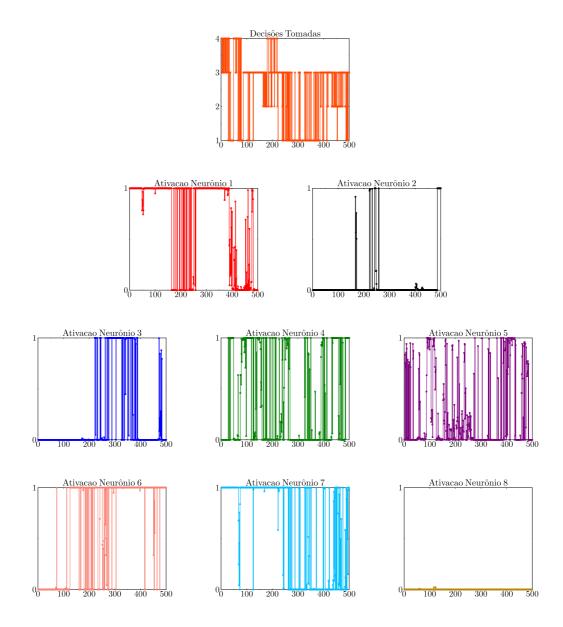


Figura A.6: Ativações da camada oculta da rede MLP(24,8,4).

A.3.1.2 MLP(24,20,4)

Nas Tabelas A.7 e A.8 são mostrados os resultados do treinamento e teste textitoffline da rede MLP com 20 neurônios ocultos e 24 unidades de entrada.

Tabela A.7: Estatísticas do treinamento textitoffline da rede MLP(24,20,4).

MLP(24,20,4)				
Média de acertos	89,27%			
Taxa Máxima de acertos	94,75%			
Taxa Mínima de acertos	82,9%			
Variância dos resultados	0,00107062			

Tabela A.8: Resultado dos testes *offline* da rede MLP(24,20,4).

TESTE MLP(24,20,4)					
Erro Quadrático Médio			0,06		
Taxa de acerto		92	2,17%		
Matriz de Confusão		Cl	lasses I	Predita	S
	es	537	16	14	16
	lasses eais	24	186	1	0
	Se C	25	6	532	2
		8	0	2	87

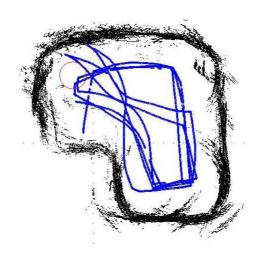


Figura A.7: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(24,20,4).

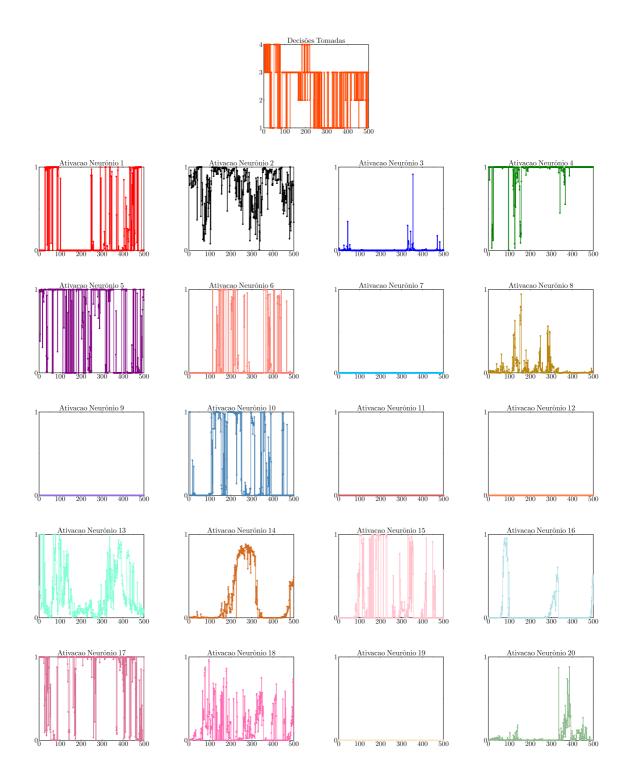


Figura A.8: Ativações da camada oculta da rede MLP(24,20,4).

Por meio dessas tabelas, percebe-se as melhores taxas de acerto até então. Com 92,1% $\left(\frac{537}{537+16+14+16}\right)$ de padrões corretamente classificados na classe 1, 88,1% $\left(\frac{186}{186+24+1}\right)$ na classe 2, 94,1% $\left(\frac{532}{532+2+6+25}\right)$ na classe 3 e 89,7% $\left(\frac{87}{87+2+8}\right)$ na classe 4. As boas taxas de acerto podem ser antecipadas pelo fato das superfícies de decisão na Figura 3.13(h) estarem bem

ajustadas ao conjunto de dados.

Na Figura A.8, verifica-se as ativações da camada oculta ao longo de uma volta do robô no seu ambiente de testes, com a rede MLP(24,20,4). Da mesma forma que nos casos anteriores, essas ativações não coincidem com nenhuma classe em específico, além disso, muitos desses neurônios possuem ativações muito próximas de zero. Apesar de alguns deles desempenharem alguma função dentro desta arquitetura, a falta de ativação na maioria é um indício de que há mais neurônios do que o necessário.

Assim como nos outros testes com a rede MLP, o desempenho da navegação do robô é semelhante ao do programa com heurísticas e sem colisões, como mostrado na Figura A.7. Embora tenha apresentado vários momentos de hesitação, que são devidos ao chaveamento de decisões conflitantes em um curto período de tempo.

Nesta mesma figura, é importante fazer uma observação sobre alguns pontos que sugerem uma colisão. Nesses casos, apesar do robô ter se aproximado bastante da parede, ele apenas girou em seu próprio eixo, desviando com sucesso.

A.3.2 Resultados para MLP(4,q,4)

Nesta subseção são descritos o resultados dos testes da rede MLP com 8 e 20 neurônios ocultos para o caso que faz uso das quatro distâncias resumidas.

A.3.2.1 MLP(4,8,4)

Para a MLP com quatro parâmetros de entrada e 8 neurônios ocultos, os resultados da análise inicial estão nas Tabelas A.9 e A.10.

Tabela A.9: Estatísticas do treinamento textitoffline da rede MLP(4,8,4).

MLP(4,8,4)				
Média de acertos	97,47%			
Taxa Máxima de acertos	98,52%			
Taxa Mínima de acertos	93,15%			
Variância dos resultados	0,000167964			

Tabela A.10: Resultado dos testes *offline* da rede MLP(4,8,4).

TESTE MLP(4,8,4)					
Erro Quadrático Médio		(0,022		
Taxa de acerto		97	7,59%		
Matriz de Confusão	Classes Preditas			ıS	
	es	625	7	12	0
	ass	5	188	1	0
	Z &	0	0	539	0
		10	0	0	69

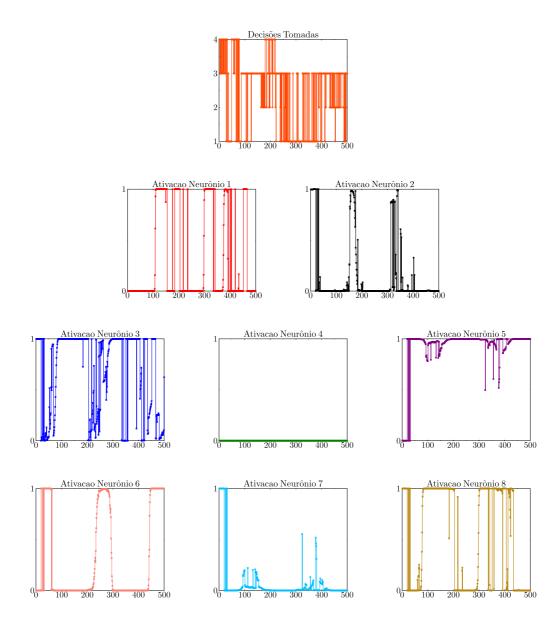


Figura A.9: Ativações da camada oculta da rede MLP(4,8,4).

Pela Tabela A.10, verifica-se uma taxa de acerto média do treinamento, menor que a rede

MLP(4,6,4), mas nas Tabelas A.9 e A.3, observa-se taxas de acerto bem próximas no período de testes, com 97% de padrões corretamente classificados na classe 1 e 2, 100% na classe 3, e 87,3% na classe 4. Nas ativações dos neurônios ocultos, percebe-se que apenas um deles permanece com ativações próximas de zero, mostrado na Figura A.9.

Com 8 neurônios, o robô foi capaz de navegar pela sala sem colisões, mas com alguns momentos de hesitação marcantes, em outras palavras, o robô chegava a fazer um parada brusca antes de dar continuidade ao movimento. Esse comportamento é resultante de decisões opostas em um curto intervalo de tempo e o percurso pode ser visto na Figura A.10.

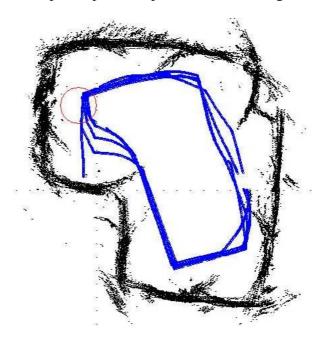


Figura A.10: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,8,4)

A.3.2.2 MLP(4,20,4)

Com a MLP com quatro parâmetros de entrada, os resultados da análise inicial com 20 neurônios ocultos estão nas Tabelas A.11 e A.12.

Tabela A.11: Estatísticas do treinamento textitoffline da rede MLP(4,20,4).

MLP(4,20,4)				
Média de acertos	97,59%			
Taxa Máxima de acertos	98,65%			
Taxa Mínima de acertos	92,32%			
Variância dos resultados	0,000142278			

Nessas tabelas, observa-se valores de taxa de acerto bem próximas de 100%, assim, como é mostrado na matriz de confusão, há pouquíssimos dados mal classificados. As seguintes

porcentagens de sucesso são verificadas: 98,4% $\left(\frac{566}{566+9}\right)$ na classe 1, 98,6% $\left(\frac{219}{219+1+2}\right)$ na classe 2, 99,4% $\left(\frac{565}{565+3}\right)$ na classe 3 e 90% $\left(\frac{82}{9+82}\right)$ na classe 4. Pela Figura A.12, percebe-se que há três neurônios que não são ativados durante a navegação do robô.

Com relação à navegação, o desempenho foi praticamente o mesmo do caso anterior, com exceção no final da terceira volta do percurso, onde ele deveria fazer uma curva à esquerda após passar de uma quina, a unidade robótica se afastou uns 40 centímetros desta e começou a rodar em um pequeno círculo em busca da parede para seguir, como é mostrado na Figura A.11. Esta situação pode ser vista também com o programa de heurísticas, mas somente quando o robô é posicionado a 1 metro ou mais de alguma parede.

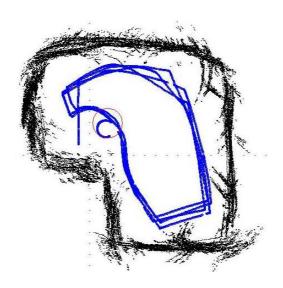


Figura A.11: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(4,20,4).

Tabela A.12: Resultado dos testes *offline* da rede MLP(4,20,4).

TESTE MLP(4,20,4)					
Erro Quadrático Médio		(0,019		
Taxa de acerto	98,35%				
Matriz de Confusão	Classes Preditas				
	es	566	9	0	0
	lasses eais	1	219	2	0
	Se C	3	0	565	0
		9	0	0	82

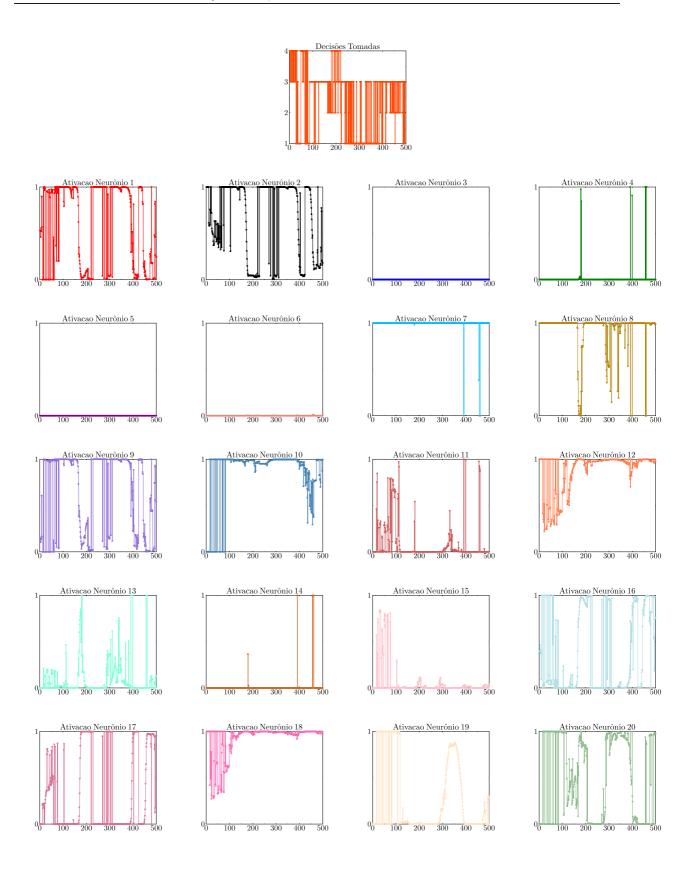


Figura A.12: Ativações da camada oculta da rede MLP(4,20,4).

A.3.3 Resultados para MLP(2,q,4)

Nesta subseção são descritos o resultados dos testes da rede MLP com 8 e 20 neurônios ocultos para o caso que faz uso das duas distâncias resumidas.

A.3.3.1 MLP(2,8,4)

Os resultados da avaliação inicial da rede MLP com dois parâmetros de entrada com 8 neurônios na camada oculta estão nas Tabelas A.13 e A.14.

Tabela A.13: Estatísticas do treinamento textitoffline da rede MLP(2,8,4).

MLP(2,8,4)				
Média de acertos	95,12%			
Taxa Máxima de acertos	96,87%			
Taxa Mínima de acertos	81,85%			
Variância dos resultados	0,00113494			

Tabela A.14: Resultado dos testes offline da rede MLP(2,8,4)

TESTE MLP(2,8,4)					
Erro Quadrático Médio 0,04					
Taxa de acerto	97,04%				
Matriz de Confusão	Classes Preditas				
	es	522	6	4	0
	lasses eais	12	218	3	0
	Z &	8	0	586	0
		9	0	1	87

Nessas tabelas, observa-se resultados semelhantes à rede MLP(2,8,4), com boas taxas de acerto tanto no treinamento e teste. A matriz de confusão apresenta uma porcentagem de padrões corretamente classificados de 98,2% $(\frac{522}{522+6+4})$ na classe 1, 93,5% $(\frac{218}{218+12+3})$ na classe 2, 98,6% $(\frac{586}{586+8})$ na classe 3 e 89,6% $(\frac{87}{87+1+9})$ na classe 4.

Pela Figura A.13, verifica-se a mesma condição das ativações das redes passadas e que tem dois neurônios com ativações próximas de zero.

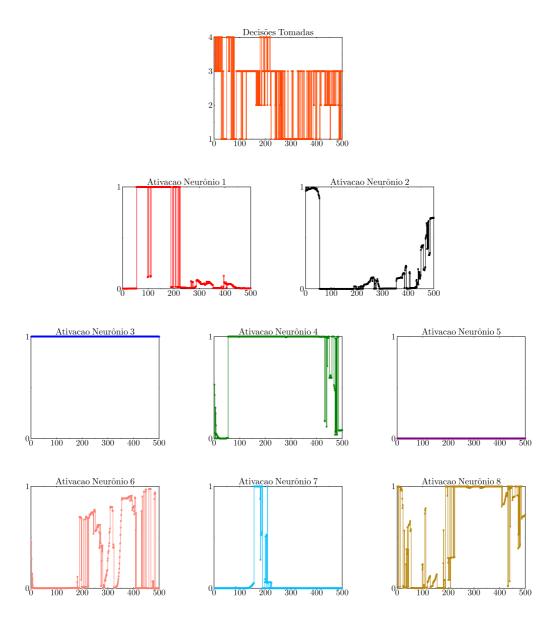


Figura A.13: Ativações da camada oculta da rede MLP(2,8,4).

Com relação à navegação, assim como a rede MLP(2,6,4), esta teve um comportamento semelhante ao programa original, sem colisões e com menos momentos de hesitação, cuja trajetória pode ser vista na Figura A.14.

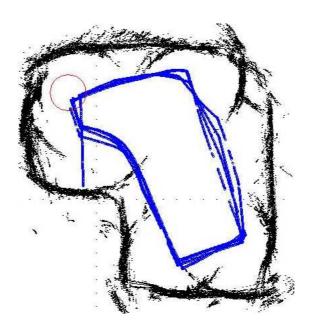


Figura A.14: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,8,4).

A.3.3.2 MLP(2,20,4)

Por fim, os resultados da avaliação inicial da rede MLP com dois parâmetros de entrada com 20 neurônios na camada oculta estão nas Tabelas A.15 e A.16.

Tabela A.15: Estatísticas do treinamento textitoffline da rede MLP(2,20,4).

MLP(2,20,4)				
Média de acertos	97,36%			
Taxa Máxima de acertos	98,32%			
Taxa Mínima de acertos	95,02%			
Variância dos resultados	6.65839e - 05			

Tabela A.16: Resultado dos testes *offline* da rede MLP(2,20,4).

TESTE MLP(2,20,4)					
Erro Quadrático Médio		(0,024		
Taxa de acerto	97,87%				
Matriz de Confusão	Classes Preditas				
	es	575	13	8	0
	lasses eais	0	213	1	0
	C Re	0	0	552	0
		6	0	3	85

Estas mostram os resultados do treinamento e teste textitoffline. As taxas de acerto se mantém em valores bem próximos da rede MLP(2,6,4) e MLP(2,8,4), com 96,4% $(\frac{575}{575+13+8})$

de padrões corretamente classificados na classe 1, 99,5% $(\frac{213}{213+1})$ na classe 2, 100% na classe 3 e 90,4% $(\frac{85}{85+3+6})$ na classe 4.

Sobre a navegação, da mesma forma que as outras redes, a MLP(2,20,4) imitou muito bem o comportamento que deu origem aos dados de treinamento, sem colisões e poucos momentos de hesitação.

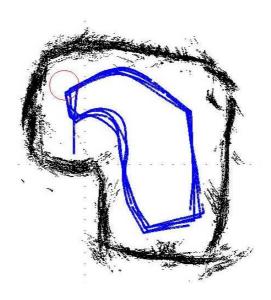


Figura A.15: Percepção do ambiente e trajetória do SCITOS G5 para a rede MLP(2,20,4).

As ativações dos neurônios ocultos, apresentados na Figura A.16, mostram que todos os neurônios são ativados, tendo um único com ativação estável próxima a 1.

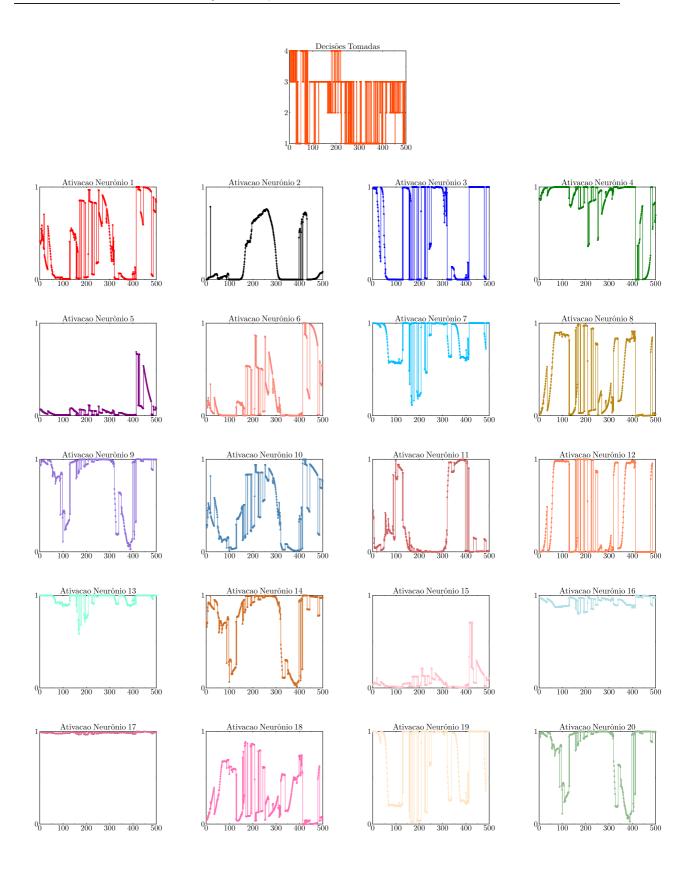


Figura A.16: Ativações da camada oculta da rede MLP(2,20,4).

APÊNDICE B - Testes extras com a rede de Elman

Neste apêndice, são apresentados os resultados encontrados com a rede de Elman para os casos que utilizam 24 e 4 distâncias como entradas tanto para o caso com 4 neurônios ocultos, como também para 3,5 e 6. No caso que faz uso de apenas 2 distâncias como entradas, apenas os resultados dos testes com 3,5 e 6 neurônios na camada oculta são abordados.

B.1 Resultados para a rede de Elman(24+q,q,4)

As Tabelas B.1 e B.2 mostram as estatísticas do treinamento e teste da rede Elman com 6 e 5 neurônios na camada oculta.

Tabela B.1: Estatísticas do treinamento offline da rede Elman com 6 e 5 neurônios ocultos.

	Elman(24+6,6,4)	Elman(24+5,5,4)
Média de acertos	76,93%	75,24%
Taxa Máxima de acertos	87,82%	89,2%
Taxa Mínima de acertos	45,8%	57,67%
Variância dos resultados	0,00654518	0,00481853

Tabela B.2: Resultado dos Testes *offline* da rede Elman com memória $\tau = 6$ e 5.

		Elman(24+6,6,4)			Elm	an(24+	-5,5,4)		
Erro Quadrático Médio		0,105				0,1			
Taxa de acerto		83,37%				82,629	%		
Matriz de Confusão		Classes Preditas			Clas	ses Pr	editas		
	es	472	41	54	27	501	39	52	2
	Class Reais	31	145	0	31	33	173	0	1
	C S	18	1	529	0	17	0	526	5
		26	0	13	68	92	0	12	3

Pela Tabela B.1, observa-se que a taxa de acerto média entre as duas redes analisadas são bem próximas, com 76,93% e 75,24%. Na Tabela B.2, encontram-se os resultados do teste *offline*, no qual nota-se que a rede com 6 neurônios ocultos teve o melhor desempenho em

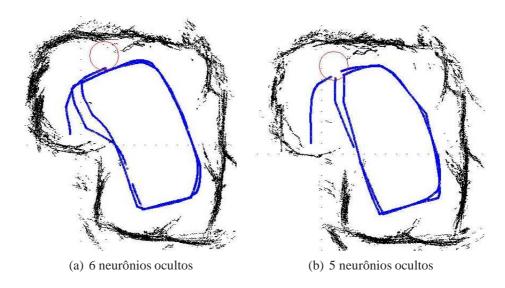


Figura B.1: Percepção do ambiente e trajetória do SCITOS G5 para a rede de a) Elman(24+6,6,4) e b) Elman(24+5,5,4)

relação à taxa de acerto. Quanto à matriz de confusão, é possível ver que a classe 1 teve 79,46% $\left(\frac{472}{472+41+54+27}\right)$ de acerto, a classe 2 teve 70% $\left(\frac{145}{145+31+31}\right)$, a classe 3 apresentou 96,53% $\left(\frac{529}{529+18+1}\right)$ e a classe 4 teve 63,55% $\left(\frac{68}{68+13+26}\right)$.

Já com a rede de Elman(24+5,5,4), a classe 1 teve 84,34% $\left(\frac{501}{501+39+52+2}\right)$ de sucesso, a classe 2 com 83,57% $\left(\frac{173}{173+33+1}\right)$, a classe 3 com 95,98% $\left(\frac{526}{526+5+17}\right)$ e a classe 4 teve 2,8% $\left(\frac{3}{3+12+92}\right)$. Esta rede apresenta problemas na classificação no momento da curva à esquerda, sendo uma candidata a ter o mesmo comportamento da rede perceptron sem memória.

As Figuras B.1(a) e B.1(b) mostram a trajetória que o robô seguiu utilizando as duas redes acima.

Com a rede Elman(24+6,6,4), o robô conseguiu realizar a movimentação correta e sem colisões, enquanto que com a rede Elman(29,5,4) a navegação foi bastante hesitante e não foi capaz de realizar a curva à esquerda, como havia sido previsto.

As Tabelas B.3 e B.4 mostram as taxas médias, máximas e mínimas de acerto durante as repetições do treinamento e teste da rede Elman com 4 neurônios ocultos, que usa como entrada as 24 distâncias retornadas pelos sensores de ultra-som.

Tabela B.3: Estatísticas do treinamento *offline* da rede Elman com 4 neurônios ocultos.

Elman(24+4,6,4)			
Média de acertos	71,88%		
Taxa Máxima de acertos	85,02%		
Taxa Mínima de acertos	41,5%		
Variância dos resultados	0,0121999		

Teste Elman(24+4,4,4)					
Erro Quadrático Médio 0, 108					
Taxa de acerto	82,48%				
Matriz de Confusão Classes Preditas					5
	es	518	33	43	0
	ass ais	55	152	0	1
	E E	15	2	531	0
		98	6	3	0

Tabela B.4: Resultado dos testes *offline* da rede Elman com 4 neurônios ocultos.

A taxa média de acertos do treinamento é de 71,88%, menor do que a rede MLP que também faz uso de 24 distâncias, com 88,78%. A taxa máxima encontrada pela rede Elman é equivalente à minima da rede MLP, a qual tem uma variância menor nas taxas analisadas. Em relação ao teste, a rede Elman continua com um resultado inferior comparado à MLP, com 82,48% de acerto e o EQM de 0,108379.

A matriz de confusão na Tabela B.4 mostra que a classe 1 tem 87,2% $\left(\frac{518}{518+33+43}\right)$ de sucesso, 2% a mais que a rede MLP. A classe 2 tem 73,07% $\left(\frac{152}{152+55+1}\right)$, a classe 3 apresenta 96,89% $\left(\frac{531}{15+2+531}\right)$ e a classe 4 não tem nenhum acerto. Com exceção desta última, todas as outras classes tiveram um desempenho melhor que a rede MLP.

A Figura B.2 mostra a trajetória do robô com a rede descrita acima.

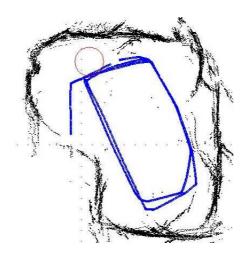


Figura B.2: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(28,4,4).

Pela matriz de confusão, é possível antecipar que o robô tem problemas com a curva à esquerda. Apesar de não ocorrer colisões, ele não fez essa curva e ainda foi muito hesitante quando havia necessidade de fazer curvas à direita. Essa hesitação.

Nas Tabelas B.5 e B.6 mostram os resultados do treinamento e teste offline.

Fazendo uma análise sobre as taxas médias de acerto do treinamento, percebe-se que estas

Tabela B.5: Estatísticas do treinamento *offline* da rede Elman com 3 neurônios ocultos.

Elman(24+3,6,4)				
Média de acertos	66,85%			
Taxa Máxima de acertos	84,12%			
Taxa Mínima de acertos	41,5%			
Variância dos resultados	0,0150793			

Tabela B.6: Resultado dos testes *offline* da rede Elman com 3 neurônios ocultos.

Teste Elman(24+3,4,4)					
Erro Quadrático Médio	Erro Quadrático Médio 0, 139				
Taxa de acerto	76,78%				
Matriz de Confusão	Classes Preditas				
	es	509	56	29	27
	lasses eais	271	180	0	31
	E E E	86	33	429	0
		107	0	0	68

diminuem a medida que a quantidade de neurônios ocultos também diminui. A matriz de confusão na Tabela B.6 mostra que a classe 1 tem 81,96% $\left(\frac{509}{509+56+29+27}\right)$ de sucesso, a classe 2 tem 37,34% $\left(\frac{180}{180+271+31}\right)$, a classe 3 apresentou 78,28% $\left(\frac{429}{429+33+86}\right)$ e a classe 4 tem 38,85% $\left(\frac{68}{68+107}\right)$. Em relação à ultima, houve uma melhora na taxa de acerto, apesar de ser baixa, mostrando que no caso com o uso de 24 distâncias como entradas, o problema de classificação das classes 2 e 4 é acentuado.

Na Figura B.3 é mostrada a trajetória do robô, onde é possível ver que ele colidiu antes de realizar a curva à esquerda.

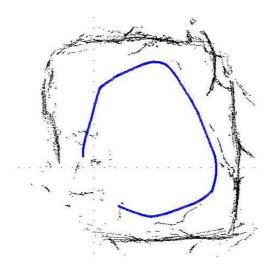


Figura B.3: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(24+3,3,4).

B.2 Resultados da rede de Elman(4+q,q,4)

Nesta Seção são apresentados os resultados dos treinamentos e teste *offline*, assim como a descrição a navegação do robô SCITOS G5 com as várias redes de Elman geradas com o uso de 4 distâncias resumidas como entradas. Os resultados com as redes que possuem 5 e 6 neurônios na camada oculta podem ser vistos Tabelas B.7 e B.8.

Tabela B.7: Estatísticas do treinamento *offline* da rede Elman com 6 e 5 neurônios ocultos.

	Elman(4+6,6,4)	Elman(4+5,5,4)
Média de acertos	89,38%	87,81%
Taxa Máxima de acertos	95,95%	96,1%
Taxa Mínima de acertos	62,52%	62,27%
Variância dos resultados	0,00483967	0,00612614

Tabela B.8: Resultado dos Testes *offline* da rede Elman com memória $\tau = 6$ e 5.

		Elman(4+6,6,4)			Eln	nan(4+	-5,5,4)		
Erro Quadrático Médio		0,045				0,04	4		
Taxa de acerto		92,99%				94,23	%		
Matriz de Confusão		Classes Preditas		Cla	sses Pr	editas			
	es	590	1	3	0	584	2	8	0
	ass ais	60	147	0	0	34	173	0	0
	CL Re	9	0	529	10	8	0	527	13
		19	0	0	88	19	0	0	88

Com essas redes, observa-se que não há muita diferença entre os dados estatísticos levantados, como a taxa média, mínima e máxima de acerto, assim como a variância desses resultados. Pelo fato das matrizes de confusão serem também muito parecidas apenas a da rede com 6 neurônios na camada é descrita a seguir. Nela, percebe-se que a classe 1 apresenta 99,32% $\left(\frac{590}{590+1+3}\right)$ de sucesso, a classe 2 tem 71% $\left(\frac{147}{147+60}\right)$, a classe 3 tem 96,53% $\left(\frac{529}{529+10+9}\right)$ e a classe 4 apresenta 82,24% $\left(\frac{88}{88+19}\right)$.

Em relação às redes de Elman que fazem uso das distâncias dadas pelos 24 sensores, houve uma melhora na capacidade de classificação das classes 2 e 4, que até então tinham taxas de acerto em torno de 30 a 40%.

Na Figura B.4 é apresentada a trajetória do robô com a rede de Elman(4+5,5,4). Com esta rede, o robô foi capaz de navegar sem colisões, mas mostrou hesitação ao realizar as curvas. Isso significa um chaveamento de decisões conflitantes, neste caso, entre fazer a curva ou continuar em frente.

Nas Tabelas B.9 e B.10 são mostradas as estatísticas do treinamento e teste offline da rede

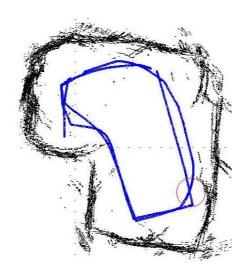


Figura B.4: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+5,5,4).

Elman que faz uso das 4 distâncias resumidas como entradas do sistema com 4 neurônios na camada oculta.

Tabela B.9: Estatísticas do treinamento *offline* da rede Elman com 4 neurônios ocultos.

Elman(4+4,6,4)				
Média de acertos	83,17%			
Taxa Máxima de acertos	96%			
Taxa Mínima de acertos	72,9%			
Variância dos resultados	0,00403204			

Tabela B.10: Resultado dos testes *offline* da rede Elman com 4 neurônios ocultos.

Teste Elman (4+4,4,4)						
Erro Quadrático Médio		C	,045			
Taxa de acerto	94,23%					
Matriz de Confusão		Classes Preditas			S	
	es	586	2	5	1	
	lasses eais	58	149	0	0	
	Z &	6	0	539	3	
		9	0	0	98	

Analisando a tabela referente ao treinamento, percebe-se que os resultados são similares aos da rede MLP(4,6,4), com uma taxa média de acerto de 83,17%. Já no teste, na Tabela B.10, a taxa de acerto foi 3% menor, com 94,23%, mas os erros quadráticos médio são próximos.

A matriz de confusão mostra que a classe 1 apresentou 98,65% $\left(\frac{586}{586+2+5}\right)$ de sucesso, a classe 2 tem 71,98% $\left(\frac{149}{149+58}\right)$, a classe 3 tem 98,35% $\left(\frac{539}{539+3+6}\right)$ e a classe 4 tem 91,58% $\left(\frac{98}{98+9}\right)$. Apesar de serem porcentagens menores que a rede MLP equivalente, ainda podem ser consideradas taxas altas.

Na Figura B.5 é mostrada a trajetória do robô com a rede de Elman(4+4,4,4). Como é possível observar, o SCITOS G5 foi capaz de navegar sem colisões, tendo uma trajetória bem próxima da baseada em heurísticas, mas sofrendo uma hesitação ao terminar a curva à esquerda.

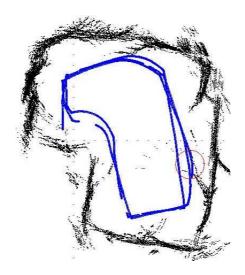


Figura B.5: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+4,4,4).

A seguir, nas Tabelas B.11 e B.12, são mostrados os resultados do treinamento e teste *offline* da rede de Elman com 3 neurônios ocultos.

Tabela B.11: Estatísticas do treinamento *offline* da rede Elman com 3 neurônios ocultos.

Elman(4+3,6,4)				
Média de acertos	80,04%			
Taxa Máxima de acertos	80,04%			
Taxa Mínima de acertos	63,67%			
Variância dos resultados	0,00281282			

Tabela B.12: Resultado dos testes *offline* da rede Elman com 3 neurônios ocultos.

Teste Elman (4+3,4,4)						
Erro Quadrático Médio		(0,074			
Taxa de acerto	92,99%					
Matriz de Confusão	Classes Preditas				S	
	es	581	1	10	2	
	lasses eais	35	172	0	0	
	C Re	11	0	537	0	
		37	0	6	64	

Da mesma forma que no caso com 24 distâncias na entrada, percebe-se que a diminuição dos neurônios ocultos, também ocasiona a diminuição das taxas de acerto. Outro ponto a ser observado é que apesar dessa diminuição nos acertos, em alguns casos vistos neste trabalho, não implica que a navegação do robô deixará de imitar o comportamento adequadamente.

A matriz de confusão da rede de Elman(4+3,3,4) mostra que a classe 1 tem 97,81% $\left(\frac{581}{581+1+10+2}\right)$ de padrões corretamente classificados, a classe 2 apresenta 83% $\left(\frac{172}{172+35}\right)$, a classe 3 tem 98% $\left(\frac{537}{537+11}\right)$ e a classe 4 apresenta 59,81% $\left(\frac{64}{64+6+37}\right)$. Com o classificador descrito, o robô executou a trajetória mostrada na Figura B.6.

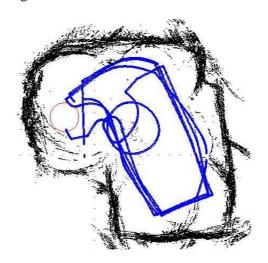


Figura B.6: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(4+3,3,4).

Neste caso, o robô realizou a primeira volta na sala sem colisões e fazendo a curva à esquerda corretamente. Na segunda volta, ele apresentou uma hesitação de 10s durante a curva à esquerda, antes de voltar a trajetória correta. A partir disso, em todas as curvas à esquerda passaram a ser muito hesitantes por conta das decisões conflitantes entre girar à esquerda ou à direita ou girar quase sobre seu próprio eixo para a esquerda. Este último faz com que o robô acabe rotacionando mais do que o necessário para encontrar a parede, não permitindo que o robô ande o suficiente para frente para encontrar a parede à esquerda ou mesmo encontrar a parede à frente e dar continuidade à trajetória.

B.2.0.3 Resultados para a Rede de Elman(2+q,q,4)

Nesta Subseção, são apresentados os resultados dos testes com a rede de Elman, variando a quantidade de neurônios ocultos no caso de 2 distâncias resumidas na entrada. Diferente das Subseções anteriores, o caso com 4 neurônios ocultos está detalhado no Capítulo 5.

Nas Tabelas B.13 e B.14 que seguem, apresentam os dados relacionados ao treinamento e teste *offline* das redes de Elman com 6 e 5 neurônios ocultos.

Como é possível notar, as taxas de acerto são superiores a todos os outros casos estudados anteriormente com a mesma arquitetura. Com relação ao teste, cujos resultados são mostrados na Tabela B.14, apesar das taxas de acerto terem valores próximos, as matrizes de confusão mostram uma diferença. Por exemplo, na rede de Elman(2+6,6,4), durante a classificação de

Tabela B.13:	Estatísticas de	o treinamento	offline d	la rede I	Elman	com 6	e 5 neu	rônios	ocultos.

	Elman(2+6,6,4)	Elman(2+5,5,4)
Média de acertos	89,98%	88,88%
Taxa Máxima de acertos	96,22%	96,2%
Taxa Mínima de acertos	71,67%	75,75%
Variância dos resultados	0,00317442	0,0038332

Tabela B.14: Resultado dos Testes *offline* da rede Elman com memória $\tau = 6$ e 5.

		Elman(2+6,6,4)			Elr	nan(2+	-5,5,4)		
Erro Quadrático Médio			0,	05		0,051			
Taxa de acerto		96,9%				96,35	%		
Matriz de Confusão		Classes Preditas		Cla	sses Pi	reditas			
	es	574	9	10	1	587	0	7	0
	lasses eais	17	190	0	0	10	197	0	0
	Z &	4	0	544	0	3	1	544	0
		3	0	1	103	27	0	5	75

padrões da classe 1, estes são categorizados como pertencentes, na sua maioria, às classes 2 e 3. Enquanto para a rede de Elman(2+5,5,4), essa má categorização é dada apenas para a classe 3.

A Figura B.7 mostra a trajetória que o robô fez com a rede de Elman(7,5,4). Com esta rede, ele navegou sem colisões, mas um pouco hesitante nas curvas.

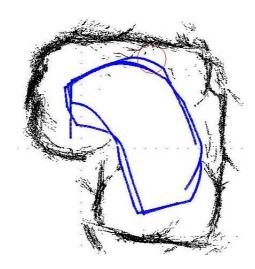


Figura B.7: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(2+5,5,4).

Por fim, nas Tabelas B.15 eB.16 mostram os resultados do treinamento e teste realizados com a rede de Elman com 3 neurônios ocultos.

Esta rede apresentou uma taxa média de acerto durante o treinamento de 80,81% e de 90,52% no teste *offline*. A matriz de confusão gerada com este teste mostra que a classe 1

Tabela B.15: Estatísticas do treinamento *offline* da rede Elman com 3 neurônios ocultos.

Elman(2+3,6,4)				
Média de acertos	80,81%			
Taxa Máxima de acertos	93,57%			
Taxa Mínima de acertos	72,85%			
Variância dos resultados	0,00257872			

Tabela B.16: Resultado dos testes *offline* da rede Elman com 4 neurônios ocultos.

Teste Elman (2+3,4,4)						
Erro Quadrático Médio		(0,063			
Taxa de acerto		90,52%				
Matriz de Confusão		Classes Preditas			S	
	es	557	31	6	0	
	lasses eais	64	143	0	0	
	Z &	4	0	530	14	
		15	0	4	88	

tem 93,77% $\left(\frac{557}{557+31+6}\right)$ de sucesso, a classe 2 apresenta 69% $\left(\frac{143}{143+64}\right)$, a classe 3 tem 96,71% $\left(\frac{530}{530+14+4}\right)$ e a classe 4 tem 82,24% $\left(\frac{88}{88+4+15}\right)$.

Com essa rede, a trajetória seguida pelo robô SCITOS G5 pode ser vista na Figura B.8.

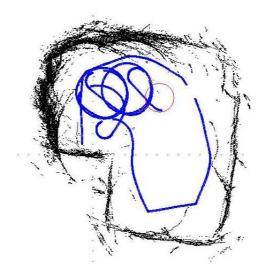


Figura B.8: Percepção do ambiente e trajetória do SCITOS G5 para a rede Elman(5,3,4).

Neste caso, ele começou a navegação sem colisões e hesitações, mas após tentar fazer a curva à esquerda, não conseguiu dar continuidade à trajetória esperada. Além disso, percebe-se um comportamento em que ele começa a rotacionar para a esquerda, quase em torno do seu próprio eixo, e tal comportamento não é definido dentro das opções de movimento. Assim, o robô ficou fazendo círculos, como se estivesse à procura da parede para seguir, mas em várias oportunidades em que esteve próximo a esta, ele não mudou de comportamento.

APÊNDICE C - Testes Extras com a Rede PL

Neste apêndice são detalhados os testes realizados com a rede PL que utiliza 2 e 4 distâncias resumidas como entrada e que realiza classificação estática, sem estruturas de MCD, e classificação dinâmica, que utiliza MCD.

C.1 Rede PL sem o uso de MCD

Nesta seção são descritos os resultados da rede PL com vinte e quatro distâncias e quatro distâncias resumidas como vetor de entrada.

C.1.1 Resultados para a Rede PL(24,4)

O primeiro experimento a ser avaliado utiliza as distâncias lidas pelos 24 sensores de ultrasom para compor o vetor de entrada. Na Tabela C.1 são mostrados as estatísticas do treinamento offline, enquanto na Tabela C.2 são mostrados os resultados da execução offline.

Tabela C.1: Estatísticas do treinamento *offline* da rede PL(24,4).

PL(24,4	4)
Média de acertos:	16,67%
Taxa Máxima de acertos:	99,42%
Taxa Mínima de acertos:	0%
Variância dos resultados:	$1,05889 \times 10^6$

Pela análise das tabelas, é possível concluir que o desempenho da rede PL(24,4) não foi bem sucedido. As taxas de acerto, tanto do treinamento quanto a de teste, são abaixo de 50%, e pela matriz de confusão, observa-se que a maior parte dos padrões alimentados à rede são classificados erroneamente, uma vez que há menos dados na diagonal principal da matriz do que no restante dela.

A trajetória percorrida pelo robô pode ser vista na Figura C.1, onde se observa que ele é quase capaz de realizar uma volta inteira na sala de teste, colidindo antes de fazer a última

Ta	Tabela C.2: Resultado dos Testes <i>offline</i> da rede PL(24,4).													
	Teste PL(24,4)													
	Erro Quadrático Médio		0.	,727										
	Taxa de acerto		28	,84%										
	Matriz de Confusão		Cla	asses P	reditas	5								
		es	144	177	273	0								
		lass eais	26	17	164	0								
		2 %	107	183	258	0								
			48	19	39	1								

curva. Apesar do início promissor, os únicos movimentos que o robô está realizando são: curva aberta à direita e curva fechada à direita.

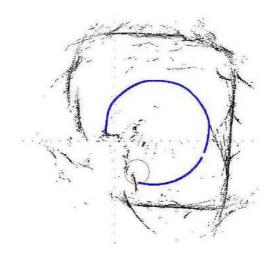


Figura C.1: Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(24,4).

Resultados para a rede PL(4,4) **C.1.2**

O segundo experimento com a rede PL usa as 4 distâncias resumidas para compor o vetor de entrada. Na Tabela C.3 são mostrados as estatísticas do treinamento offline, enquanto na Tabela C.4 são mostrados os resultados do teste PL(4,4) no robô SCITOS G5.

Tabela C.3: Estatísticas do treinamento *offline* da rede PL(4,4).

PL(4,4)	
Média de acertos:	9,52%
Taxa Máxima de acertos:	9,52%
Taxa Mínima de acertos:	9,52%
Variância dos resultados:	0

Tabela C.4: Resultado dos Testes <i>offline</i> da	reae F	PL(4.4)	١.
---	--------	---------	----

Teste PL(4,4)											
Erro Quadrático Médio		0	, 15								
Taxa de acerto	76,49%										
Matriz de Confusão	Classes Preditas										
	es	535	0	59	0						
	lass eais	201	6	0	0						
	Se Ci	0	0	534	13						
		65	0	4	38						

Analisando as tabelas, verifica-se pela matriz de confusão que esta rede foi melhor sucedida, em relação ao caso anterior, em classificar corretamente a maioria dos padrões apresentados, apesar do valor médio da taxa de acerto ser menor. O desempenho do robô comprova essa melhora, uma vez que não houve colisões e em uma das quatro voltas realizadas foi observado uma tentativa de fazer a curva à esquerda, como é mostrada na Figura C.2. Até então o robô não estava realizando esse movimento, embora esse sucesso possa ser antecipado por conta dos 35% de acerto da classe 4 na matriz de confusão.

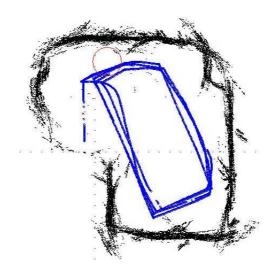


Figura C.2: Percepção do ambiente e trajetória do SCITOS G5 para a rede PL(4,4).

C.2 Rede PL com MCD

Nesta seção são descritos os testes da rede PL dotadas de MCS e que utilizam como entrada as quatro distâncias resumidas mais τ amostras anteriores de cada uma delas. A Tabela C.5 mostra a dimensão do vetor de entrada para cada valor de τ avaliado.

Na Tabela C.6 estão as estatísticas do treinamento da rede PL para o caso em que se utiliza, além da amostra atual, uma única amostra anterior de cada distância resumida (ver 1ª coluna

Tabela C.5: Quantidade de componentes no vetor \mathbf{x} para cada τ com a rede PL.

Rede PL								
τ	1	3	5	9	19	34		
Dimensão de x:								
Caso com 4 distâncias	8	16	24	40	80	140		

da Tabela), 3 amostras anteriores (2ªcoluna) e 5 amostras passadas (3ªcoluna). Na Tabela C.7 estão as estatísticas dos casos com 9, 19 e 35 amostras passadas.

Tabela C.6: Estatísticas do treinamento *offline* da rede PL com 4 distâncias resumidas e memória $\tau = 1,3$ e 5.

	PL(20,4)	PL(26,4)	PL(24,4)
Média de acertos	8,63%	6,24%	6,24%
Taxa Máxima de acertos	11,8%	14,35%	14,6%
Taxa Mínima de acertos	8,55%	6%	6%
Variância dos resultados	$3,738 \times 10^{-4}$	0.00225	0.002204

Tabela C.7: Estatísticas do treinamento *offline* da rede PL com 4 distâncias resumidas e memória $\tau = 9,19$ e 34.

	PL(40,4)	PL(80,4)	PL(140,4)
Média de acertos	14,27%	14,26%	13,96%
Taxa Máxima de acertos	15,2%	14,75%	14,25%
Taxa Mínima de acertos	14,25%	14,25%	0%
Variância dos resultados	2.25×10^{-5}	5×10^{-7}	0.000406

Pelas taxas médias de acerto, percebe-se que, apesar de baixas, elas são maiores que a taxa de acerto da mesma rede sem memória (ver Tabela C.3). Em especial, a rede PL(40,4), com uma média de 14,27% de acerto e máxima de 15,2%.

Em termos de ordem crescente pela taxa de acerto, o pior resultado é da rede PL(16,4)¹ com uma taxa média de acerto no treinamento de 6,24%, seguida pela rede PL(24,4) com a mesma taxa. Depois a rede PL(8,4), com taxa de 8,63%, a rede com 34 amostras com taxa de 13,96%, seguida pela PL(80,4) e por fim, a rede com melhor resultado no treinamento é a PL(40,4) com taxa de 14,27%. Nas Tabelas C.8 e C.9 estão os dados dos testes *offline* realizados. Interessante reparar que apesar das taxas de acerto durante o treinamento tenham melhorado em relação ao caso sem memória, as taxas de acerto do teste foram piores. A rede com melhor desempenho no treinamento, com uma média de 14,27% de acerto, teve o pior resultado no teste com apenas 57,53% de acerto.

¹O método de desempate para esta afirmação, dado o fato de que tanto a rede com 3 e 5 amostras anteriores possuem taxas iguais de médias e mínima de acerto, foi a taxa máxima, na qual a rede com 3 teve a menor.

			PL((8,4)			PL(10	6,4)]	PL(24	4,4)	
Erro Quadrático Médio		0.158				0.169			0.174				
Taxa de acerto		76,49%		74,38%				74,13%					
Matriz de Confusão		Cla	Classes Preditas		Classes Preditas			S	Classes Preditas				
	es	532	0	61	0	513	1	79	0	509	0	82	0
	Classes Reais	194	13	0	0	177	30	0	0	171	35	1	0
	Se Ci	7	0	527	13	15	0	517	13	19	0	514	12
		70	0	3	34	84	0	3	20	86	0	4	17

Tabela C.8: Resultado dos Testes *offline* PL com 4 distâncias resumidas e memória $\tau = 1,3$ e 5.

Tabela C.9: Resultado dos Testes *offline* PL com 4 distâncias resumidas e memória $\tau = 9,19$ e 34.

			PL((40,4)		P	L(80	0,4)		PI	ر14	0,4)	
Erro Quadrático Médio		0,26		0,312			0,201						
Taxa de acerto		57,53%		61,97%				73,18%					
Matriz de Confusão		Cla	Classes Preditas		Classes Preditas		5	Classes Preditas		S			
	es	282	0	305	0	359	0	218	0	513	0	54	0
	Class Reais	144	2	61	0	178	0	29	0	205	0	2	0
	2 %	8	0	537	0	21	0	524	0	13	0	527	0
		43	0	53	11	54	0	46	0	101	0	6	0

Nessas mesmas tabelas, observa-se que não é apenas a taxa de acerto que indica o desempenho das redes. Pelas matrizes de confusão, percebe-se que a medida que se aumenta a quantidade de amostras passadas na entrada das redes, o problema na classificação das classes "curva aberta à direita" e "curva à esquerda" tende a piorar.

No caso da rede PL(8,4), a taxa de acerto para a classe 1 é de 89,71% $\left(\frac{532}{532+61}\right)$, de 6,28% $\left(\frac{13}{13+194}\right)$ para a classe 2, 96,34% $\left(\frac{527}{527+13+7}\right)$ para a classe 3 e 31,77% $\left(\frac{34}{34+3+70}\right)$ para a classe 4.

Com a rede com 3 amostras anteriores (PL(16,4)), as taxas passam a ser: 86,5% $\left(\frac{513}{513+1+79}\right)$ para a classe 1, 14,49% $\left(\frac{30}{30+177}\right)$ para a classe 2, 94,86% $\left(\frac{517}{517+13+15}\right)$ para a classe 3 e 18,69% $\left(\frac{20}{20+3+84}\right)$ para a classe 4. Observa-se que em relação ao caso anterior, houve melhora apenas na classificação para a classe 2 que é a "curva aberta à direita", enquanto nas outras, houve uma pequena piora.

No caso da rede PL(24,4), a classe 1 tem uma taxa de acerto de 86, 12% $\left(\frac{509}{509+82}\right)$, a classe 2 tem 16,9% $\left(\frac{35}{35+171+1}\right)$, a classe 3 tem 94,31% $\left(\frac{514}{514+12+19}\right)$ e a classe 4 tem 15,88% $\left(\frac{17}{17+4+86}\right)$. A redução da capacidade de classificação continua a ser observada.

A rede PL(40,4) possui na classe 1 48,04% $\left(\frac{282}{282+305}\right)$ de amostras corretamente classificadas. Na classe 2, 0,09% $\left(\frac{2}{2+144+61}\right)$ de acerto, a classe 3 com 98,53% $\left(\frac{537}{537+8}\right)$ e a classe 4 com 10,28% $\left(\frac{11}{11+53+43}\right)$. Apenas para a classe 3 houve um aumento na taxa de acertos, para todo o

resto, observa-se a mesma deterioração na capacidade de classificação.

Nos casos com a rede perceptron com 19 e 34 amostras passadas (ver Tabela C.9), elas não conseguem mais distinguir as classes 2 e 4, classificando melhor as outras duas e, por conta disso, a taxa de acerto do teste fica mais alta que o caso anterior. Essas matrizes de confusão mostram a importância de se ter mais de um parâmetro para a avaliação do desempenho durante o teste.

Nas Figuras C.3 são mostradas as trajetórias que o robô percorreu fazendo uso das redes, cujo desempenho está descrito nas Tabelas C.6 e C.8. Como é possível observar, em nenhuma das trajetórias apresentadas nas Figuras C.3, o robô foi capaz de realizar a curva à esquerda.

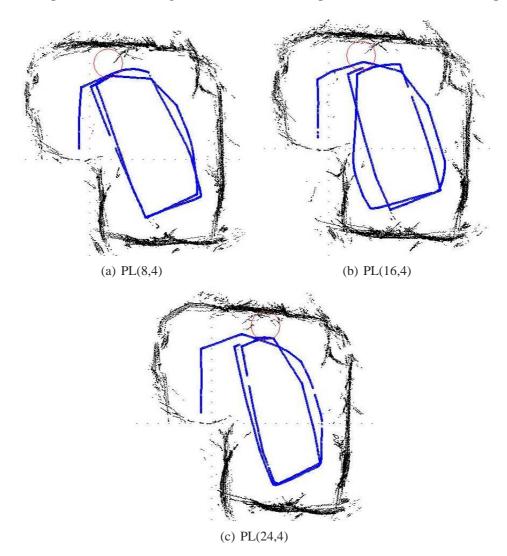


Figura C.3: Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com a) $\tau = 1$, b) $\tau = 3$ e c) $\tau = 5$ amostras passadas

Nas Figuras C.4, estão as trajetórias das redes que fazem uso de 9 e 19 amostras passadas. Com 9, mostrada na Figura C.4(a), o robô faz o seu percurso a uma distância 20cm maior que a

normal, apesar de não haver colisões, ele não foi capaz de fazer a curva à esquerda, embora não se tenha certeza se o motivo pelo qual ele não realiza esta curva seja devido à curta distância da quina até a parede mais próxima, ou se realmente ele não foi capaz de aprender essa classe.

No caso com 19 amostras, mostrada na Figura C.4(b), não houve colisões, mas também não fez a curva à esquerda. Um observação importante é que a segunda volta realizada na sala, o robô reduziu o seu trajeto, em outras palavras, o robô reagiu como se estivesse em uma sala mais estreita.

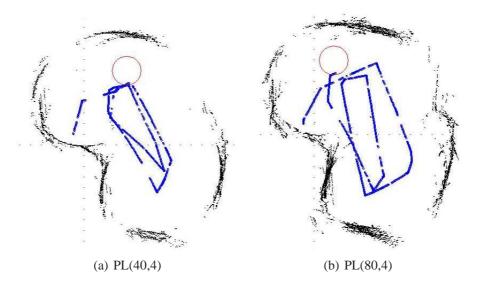


Figura C.4: Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com a) $\tau = 9$ e b) 19.

Por fim, no caso da rede perceptron com 35 amostras anteriores de cada uma das 4 distâncias resumidas, a sua trajetória está na Figura C.5. Neste caso, o robô colidiu 3 vezes. A primeira foi ao final da segunda reta percorrida, onde ele não fez a curva à direita e bateu de frente contra a parede. Após esse incidente, o robô foi posicionado a partir do ponto de colisão, de onde ele seguiu o restante do trajeto até colidir novamente, mas dessa vez por estar muito perto da parede e ao girar seu corpo para desviar da esquina, ele bateu com a traseira. Essas duas primeiras colisões estão na Figura C.5(a). Por último, na tentativa de verificar se ele seria capaz de realizar a curva à esquerda, foi colocado no começo da parede que termina na quina na qual ele deve realizar esse movimento, no entanto, ele não só não fez a curva desejada como continuou em linha reta até colidir com a parede do outro lado, ver Figura C.5(b).

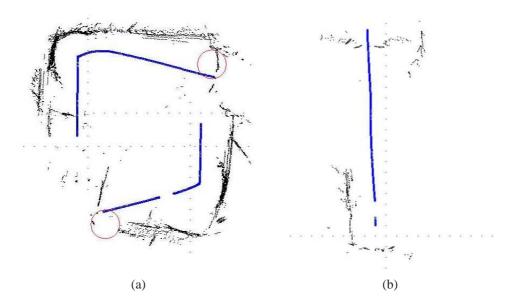


Figura C.5: Percepção do ambiente e trajetória do SCITOS G5 para a rede perceptron com tau = 34.

APÊNDICE D – Testes Extras com a Arquitetura ME

Neste apêndice estão os resultados dos diversos testes realizados com a arquitetura de mistura de especialistas. Utilizando estruturas de memória, os testes são feitos com 2 e 4 distâncias resumidas, e sem MCD os testes são feitos com 2,4 e 24 unidades de entrada.

D.1 Arquitetura ME sem o uso de MCD

Nesta seção são descritos os resultados das redes ME que fazem uso de vinte e quatro distâncias e quatro distâncias resumidas como vetor de entrada.

D.1.1 Resultados para a Arquitetura ME(24,4,4)

Inicialmente são utilizadas as distâncias dadas pelos 24 sensores de ultrassom como entrada da rede, as quais são alimentadas à todos os quatro especialistas da rede, como também a rede seletora. Na Tabela D.1 são mostrados as estatísticas do treinamento *offline*, enquanto na Tabela D.2) são mostrados os resultados do teste *offline* da arquitetura ME(24,4,4) no robô SCITOS G5.

Tabela D.1: Estatísticas do treinamento *offline* da arquitetura ME(24,4,4).

ME(24,4,4)									
Média de acertos	57%								
Taxa Máxima de acertos	57%								
Taxa Mínima de acertos	57%								
Variância dos resultados	$5,90195 \times 10^{-16}$								

Pelas baixas taxas de acerto, é possível antecipar o desempenho pobre da arquitetura ME(24,4,4). Observa-se na matriz de confusão uma grande quantidade de padrões mal-classificados. Apenas 36% dos dados da classe *seguir em frente*, 14% da classe *curva aberta à direita*, 25,7%

a D.Z. Resultado dos Testes offante da arquitetara 1412(2)												
Teste ME(24,4,4)												
Erro Quadrático Médio		0.	,329									
Taxa de acerto	9,95%											
Matriz de Confusão		Classes Preditas										
	es	228	177	211	2							
	lasses eais	131	31	50	1							
	Z &	395	7	139	0							
		55	2	26	0							

Tabela D.2: Resultado dos Testes *offline* da arquitetura ME(24,4,4).

da classe *curva fechada à direita* são corretamente classificadas, e a classe *curva à esquerda* não obteve nenhum sucesso. Durante a navegação, o robô colidiu 3 vezes, mostrando que não houve uma boa especialização da tarefa por parte dos especialistas e por conta das colisões, só foi possível gerar uma imagem da trajetória (Figura D.1) até a primeira colisão.

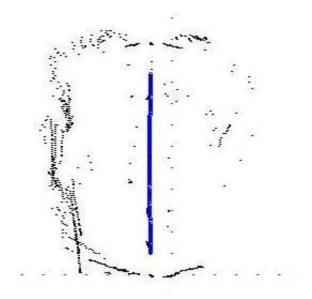


Figura D.1: Percepção do ambiente e trajetória do SCITOS G5 para a arquitetura ME(24,4,4).

D.1.2 Resultados para a Arquitetura ME(4,4,4)

Usando as 4 distâncias resumidas, nas Tabelas D.3 e D.4 são mostrados as estatísticas do treinamento e teste *offline* da arquitetura ME(4,4,4).

Verifica-se uma melhora na taxa de acerto em relação ao caso com 24 unidades de entrada. Com a rede ME(4,4,4), 88% $\left(\frac{523}{523+1+70}\right)$ dos dados da classe *seguir em frente*, 5,8% $\left(\frac{12}{12+195}\right)$ da classe *curva aberta à direita*, 97,2% $\left(\frac{534}{534+13}\right)$ da classe *curva fechada à direita*, e 32,7% $\left(\frac{35}{35+3+69}\right)$ da classe *curva à esquerda* são corretamente classificadas. Novamente, é possível notar problemas para realizar a classificação das classes 2 e 4.

Tabela D.3: Estatísticas do treinamento *offline* da arquitetura ME(4,4,4).

ME(4,4,4)										
Média de acertos	84,35%									
Taxa Máxima de acertos	84,35%									
Taxa Mínima de acertos	84,35%									
Variância dos resultados	0									

Tabela D.4: Resultado dos Testes *offline* da arquitetura ME(4,4,4).

Teste ME(4,4,4)									
Erro Quadrático Médio 0.121									
Taxa de acerto 75,87%									
Matriz de Confusão	Matriz de Confusão Classes Preditas								
	es	523	1	70	0				
	lasses eais	2 3 195 12 0							
	5 % %	0	0	534	13				
		69	0	3	35				

Apesar da melhora nas taxas, o robô simplesmente permaneceu girando sobre seu próprio eixo por realizar somente a *curva fechada à direita*, que é a classe que teve mais padrões corretamente classificados. Na Figura D.2, pode-se observar a percepção do ambiente na perspectiva do robô, e o ponto em azul confirmando que ele não fez nenhum outro movimento além de girar.

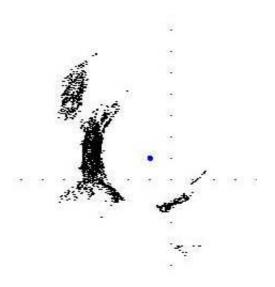


Figura D.2: Percepção do ambiente e trajetória do SCITOS G5 para a rede ME(4,4,4).

D.2 Arquitetura de Mistura de Especialistas com MCD

Nesta Subseção estão os resultados do treinamento e teste *offline* para a rede com arquitetura de mistura de especialistas, cujas entradas são as quatro distâncias resumidas mais τ amostras

passadas de cada uma delas. A quantidade dessas amostras e a quantidade de componentes que elas geram no vetor de entrada podem ser vistas na Tabela D.5.

Tabela D.5: Quantidade de componentes no vetor \mathbf{x} para cada τ com a rede ME.

Rede ME									
τ 1 4 6 9 19									
Dimensão de x:									
Caso com 4 distâncias	8	20	28	40	80				

Tabela D.6: Estatísticas do treinamento *offline* da rede ME com memória $\tau = 1,4$ e 6.

	ME(8,4,4)	ME(20,4,4)	ME(28,4,4)
Média de acertos	87,8%	80,325%	79,54%
Taxa Máxima de acertos	87,8%	80,35%	80,45%
Taxa Mínima de acertos	87,8%	80,32%	79,52%
Variância dos resultados	0	$3,58 \times 10^{-8}$	2.82×10^{-5}

Tabela D.7: Estatísticas do treinamento *offline* da rede ME com memória $\tau = 9$ e $\tau = 19$.

	ME(40,4,4)	ME(80,4,4)
Média de acertos	75,8%	37,5%
Taxa Máxima de acertos	75,87%	59,65%
Taxa Mínima de acertos	75,8%	36,77%
Variância dos resultados	$1,6333 \times 10^{-7}$	0.016911

Nesto resultados mostrados nas Tabelas D.6 e D.7, também é possível perceber que a relação *taxa de acertos* × *quantidade de amostras passadas* da rede perceptron logístico¹, se repete neste caso. A mistura de especialistas com apenas 1 amostra tem a maior taxa média de acertos com 87,8%, enquanto a rede que faz uso de 19 amostras tem a menor, 37,5%.

Essa tendência, observada nos resultados dos testes *offline*, também pode ser vista nas matrizes de confusão.

No caso da ME(8,4,4), a classe *seguir em frente* tem 82,49% $\left(\frac{490}{490+3+101}\right)$ de padrões corretamente classificados, a classe *curva aberta à direita* apresentou 13,04% $\left(\frac{27}{27+1+179}\right)$, a classe *curva fechada à direita* tem 96,52% $\left(\frac{528}{528+13+6}\right)$ e a classe *curva à esquerda* tem 26,16% $\left(\frac{28}{28+6+73}\right)$. Desempenho similar à rede PL(2,4) com 5 amostras passadas.

Com a rede ME(20,4,4), a classe 1 tem 29,2% $\left(\frac{172}{172+196+190+31}\right)$ de sucesso, a classe 2 tem 94,11% $\left(\frac{208}{208+11+2}\right)$, a classe 3 apresenta 98,92% $\left(\frac{553}{553+6}\right)$ e a classe 4 mostra 87,2% $\left(\frac{75}{75+11}\right)$. Interessante observar a piora na capacidade de classificação da classe 1, que é uma das classes que as redes já testadas conseguem categorizar com mais facilidade, e a grande melhora para

¹Comentada no Apêndice C

Tabela D.8: Resultado dos Testes *offline* ME com $\tau = 1.4$ e 6.

			ME(8,4,4)	<i>J.</i>		ME(20,	4,4)			ME(2	8,4,4)	
Erro Quadrático Médio		0,17			0,196			0,204					
Taxa de acerto		73,74%			69,27%			61,71%					
Matriz de Confusão		Classes Preditas		Cla	Classes Preditas			Classes Preditas			ıS		
	es	490	3	101	0	172	196	190	31	53	241	187	108
	Classes Reais	179	27	1	0	2	208	11	0	0	217	4	0
	E E	6	0	528	13	0	0	553	6	0	0	554	5
		73	0	6	28	0	0	11	75	0	0	12	74

Tabela D.9: Resultado dos Testes *offline* da arquitetura ME com memória $\tau = 9$ e 19.

		ME(40,4,4)			N	Æ(80,4,4)		
Erro Quadrático Médio			0,299						
Taxa de acerto			68,59% 37,2					25%	
Matriz de Confusão		Cl	Cla	sses	Predit	tas			
	es	349 1 49 189				5	0	567	17
	ass	192	15	7	7	3	0	211	7
	E E E	0	0	548	12	7	0	517	35
		0	0	0	86	1	0	65	20

as classe 2 e 4 que se mostram as mais complicadas para categorizar corretamente. A classe 3 continuou bem representada.

Para o caso com ME(28,4,4), a classe 1 apresenta 8,99% $\left(\frac{53}{53+241+187+108}\right)$ de padrões corretamente classificados, tendo muitos padrões pertecentes a esta classe categorizados de forma quase uniforme entre as outras classes, especialmente para a classe 2. Assim como no caso anterior, onde a classe 1 tem uma grande piora em relação à todas as outras redes já testadas, as classes mais problemáticas apresentam bons resultados: 98,19% $\left(\frac{217}{217+4}\right)$ de acerto para a classe 2 e 86,04% $\left(\frac{74}{74+12}\right)$ para a classe 4. A classe 3, com 99,1% $\left(\frac{554}{554+5}\right)$, permanece com boas taxas de sucesso.

Analisando o caso ME(40,4,4), a classe 1 sofre menos com os erros de classificação em relação às redes ME(28,4,4) e ME(20,4,4), tendo uma taxa de sucesso de 59,35% $\left(\frac{349}{349+1+49+189}\right)$. A classe 2 também volta ao padrão normal dos casos estudados, com uma baixa taxa de acerto de 6,78% $\left(\frac{15}{15+7+7+192}\right)$. A classe 3 tem 97,85% $\left(\frac{548}{548+12}\right)$ e a classe 4 apresenta 100% de sucesso.

Por fim, com a rede ME(80,4,4) percebe-se problemas, pois a coluna referente à classe 2 está em zero, ou seja, nenhum padrão foi classificado como pertencente a esta classe, nem corretamente e nem erroneamente, sobrando a maioria para as classes 3 e 4. Para a classe 1 temse 0,84% $\left(\frac{5}{5+567+17}\right)$ de padrões corretamente classificados, a classe 3 tem 92,48% $\left(\frac{517}{517+35+7}\right)$

e a classe 4 com 23,25% $\left(\frac{20}{20+65+1}\right)$.

A Figura D.3(a) mostra a trajetória do robô com a rede ME(8,4,4), em que permaneceu realizando uma curva aberta à direita, colidindo antes de formar um círculo. Uma observação interessante é a suvidade com que ele percorreu o trajeto, isso devido a ausência de decisões conflitantes, pois apenas um único movimento era escolhido.

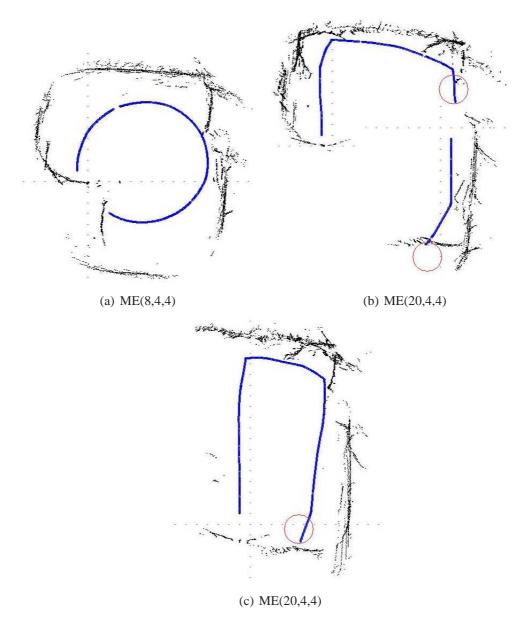


Figura D.3: Percepção do ambiente e trajetória do SCITOS G5 para a mistura de especialistas com $\tau = 1$ e 4.

As Figuras D.3(b) e D.3(c) mostram o comportamento do robô utilizando a rede ME(20,4,4). Interessante ressaltar que esta rede teve um desempenho pobre para a classe 1, enquanto as outras obtiveram uma boa taxa de classrificação. Com esta configuração, o robô começou seguindo as duas primeiras paredes, mas colidiu após a segunda curva por ter se aproximado

demais. Ainda dando continuidade ao trajeto, o robô colidiu novamente ao ir de encontro com a parede à frente. Ambas colisões são mostradas na Figura D.3(c). Como última tentativa, posiciona-se o robô para verificar se ele realiza a curva à esquerda, a qual ele não faz, mas dá continuidade à volta na sala, colidindo apenas ao chegar no último ponto de colisão comentado, como pode ser visto na Figura D.3(c).

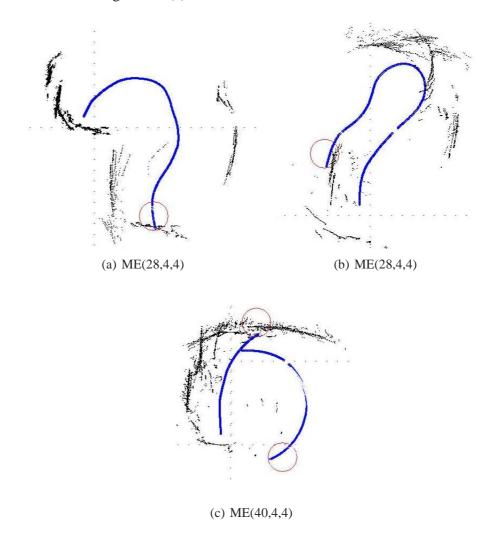


Figura D.4: Percepção do ambiente e trajetória do SCITOS G5 para a mistura de especialistas com $\tau = 6$ e 9.

As Figuras D.4(a) e D.4(b) mostram a trajetória do robô com a rede ME(28,4,4), em que o robô iniciou com muita hesitação por conta de dois movimentos conflitantes: seguir em frente e girar à direita. Depois de 7 segundos de pouca movimentação, ele deu início a sua trajetória que começou como um semi-círculo, desviou da parede corretamente e deu continuidade com uma reta, mas por ter se afastado bastante da parede, ele realizou uma curva a esquerda q o posicionou de tal forma que não pode desviar mais, e colidiu (ver Figura D.4(a)). Na tentativa de verificar se ele realiza a curva à esquerda no momento certo, é dado continuidade ao trajeto que realizava, mas quando chegou o momento de fazer a curva esperada, no lugar de girar para

a esquerda, ele foi para a direita e se afastou muito da parede. Mais a frente, passou a girar para a esquerda em busca de uma parede para seguir e terminou colidindo com a quina, como mostra a Figura D.4(b).

Para o caso da ME(40,4,4), o robô começou a sua movimentação normalmente, mas colidiu por fazer uma curva aberta no lugar de uma fechada. Dando continuidade ao teste, o robô realizou apenas uma curva aberta e colidiu com a quina. Ambas as colisões estão mostradas na Figura D.4(c).

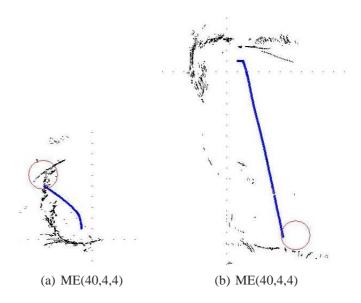


Figura D.5: Percepção do ambiente e trajetória do SCITOS G5 para a ME(40,4,4)

Por último, a trajetória feita com a rede ME(80,4,4) e que, durante o teste *offline*, não foi capaz de categorizar nenhum padrão como pertencente à classe 2, é mostrada nas Figuras D.5(a) e D.5(b). O robô iniciou a sua movimentação com uma curva à esquerda e não fazia tentativa de desviar da parede, colidindo com ela (ver Figura D.5(a)). Foram feitas duas tentativas no mesmo ponto inicial, e nas duas o mesmo comportamento. Posicinou-se o robô em outro ponto mais adiante de seu trajeto, de onde ele deveria seguir em frente, mas ele começou sua movimentação girando para a direita e em seguida continuou em linha reta, atravessando a sala e colidindo na outra ponta, como é possível verificar na Figura D.5(b). suavidadeclassificação.Posicionou

Referências Bibliográficas

AGUIRRE, L. A. (Ed.). *Enciclopédia de Automática: Controle e Automação*. [S.l.]: Blucher, 2007.

ANTONELO, E.; SCHRAUWEN, B.; STROOBANDT, D. Event detection and localization for small mobile robots using reservoir computing. *Neural Networks*, v. 21, n. 6, p. 862–871, 2008.

ANTONELO, E.; SCHRAUWEN, B.; STROOBANDT, D. Mobile robot control in the road sign problem using reservoir computing networks. In: *IEEE International Conference on Robotics and Automation, ICRA 2008.* [S.l.: s.n.], 2008. v. 19, n. 23, p. 911 – 916.

ARKIN, R. Behavior-Based Robotics. [S.1.]: MIT Press, 1998.

BAKKER, B. et al. A Robot that Reinforcement-Learns to Identify and Memorize. [S.1.], 2003.

BALLARD, D. Cortical connections and parallel processing: structure and function. *Behavioral and Brain Sciences*, n. 9, p. 67–120, 1986.

BARRETO, G. *Redes Neurais Não-Supervisionadas para Processamento de Sequências Temporais*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, Julho 1998.

BARRETO, G. A. Unsupervised learning and recall of temporal sequences: An application to robotics. *International Journal of Neural Systems*, v. 9, p. 9–3, 1999.

BARRETO, G. A.; ARAÚJO, A. F. R.; KREMER, S. C. A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case. *Neural Computation*, v. 15, p. 1255–1320, 2003.

BASHIR, F.; KHOKHAR, A.; SCHONFELD, D. View-invariant motion trajectory-based activity classification and recognition. *Multimedia Systems*, Springer, v. 12, n. 1, p. 45–54, 2006.

BAUZIL, G.; BRIOT, M.; RIBES, P. A navigation subsystem using ultrasonic sensors for the mobile robot hilare. In: *1st International Conference on Robot Vision and Sensory Controls*. UK: [s.n.], 1981. p. 47–58.

BEKEY, G. A. Autonomous Robots. [S.1.]: MIT Press, 2005.

BISHOP, C. Neural Networks for Pattern Recognition. [S.l.]: Oxford University Press, 1995.

BUGMANN, G. et al. Stable encoding of robot trajectories using normalized radial basis functions: Application to an autonomous wheelchair. In: 29th International Symposium On Robotics, Advanced Robotics: Beyond 2000. Birmingham, UK: [s.n.], 1998.

CAZANGI, R.; VON ZUBEN, F.; FIGUEIREDO, M. A classifier system in real applications for robot navigation. In: SARKER, R. et al. (Ed.). *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*. Canberra: IEEE Press, 2003. p. 574–580.

DORF, R.; BISHOP, R. *Modern Control Systems*. 11^a. ed. Estados Unidos: Pearson Prentice Hall, 2008.

ELMAN, J. L. Finding structure in time. *Cognitive Science*, v. 14, p. 179–211, 1990.

ELSHAW, M. et al. An associaor network approach to robot learning by imitation through vision, motor control and language. In: *IEEE International Joint Conference on Neural Networks*. [S.l.: s.n.], 2004. v. 1, p. – 596.

HAYKIN, S. Neural Networks: A Comprehensive Foundation. 1^a. ed. [S.l.]: Prentice Hall, 1994.

HERTZ, J.; KROGH, A.; PALMER, R. *Introduction to the theory of neural computation*. [S.l.]: Addison-Wesley Publishing Company, 1991.

JACOBS, R. et al. Adaptive mixtures of local experts. *Neural Computation*, n. 3, p. 79–87, 1991.

JUNG, C. R. et al. Implementação de veículos autônomos inteligentes. In: *XXV Congresso da Sociedade Brasileira de Computação*. São Leopoldo: [s.n.], 2005.

KRISHNA, K.; KALRA, P. Solving the local minima problem for a mobile robot by classification of spatio-temporal sensory sequences. *Journal of Robotic Systems*, v. 17, n. 10, p. 549–564, 2000.

LAMBRINOS, D.; SCHEIER, C. Adaptive classification in autonomous agents. *Applied Artificial Intelligence*, v. 11, n. 2, p. 119–130, 1997.

LAMBRINOS, D.; SCHEIER, C.; PFEIFER, R. In: . [S.l.: s.n.].

LENT, R. *Neurociência da mente e do comportamento*. Rio de Janeiro: Guanabara Koogan S.A., 2008.

MALONE, J. et al. Spatio-temporal neural data mining architecture in learning robots. In: *IEEE International Joint Conference on Neural Networks*. [S.l.: s.n.], 2005. v. 5, p. 2802–2807.

MARQUES, J. Reconhecimento de padrões: métodos estatísticos e neuronais. 2ª. ed. Lisboa: Instituto Superior Técnico, 2005.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 1943.

MUCIENTES, M. et al. Learning weighted linguistic rules to control an autonomous robot. *International Journal of Intelligent Systems.*, John Wiley & Sons, Inc., New York, NY, USA, v. 24, n. 3, p. 226–251, 2009.

MUCIENTES, M.; ALCALÁ, R.; ALCALÁ-FDEZ J.AND CASILLAS, J. Learning weighted linguistic rules to control an autonomous robot. *International Journal of Intelligent Systems*, v. 24, n. 3, p. 226–251, 2009.

NEHMZOW, U.; SMITHERS, T. Using motor actions for location recognition. In: *Proceedings of the First European Conference on Artificial Life*. [S.l.]: MIT Press, 1991. p. 96–104.

OLIVEIRA, J.; AMARAL, J. Princípios de Neurociência. São Paulo: TECNOPRESS, 1997.

PARKER, D. Learning Logic. Stanford, CA, 1982.

PFEIFER, R.; SCHEIER, C. Understanding Intelligence. Londres: MIT Press, 2002.

PIERI, E. R. Curso de Robótica Móvel - Universidade Federal de Santa Catarina. Abril 2002.

POMERLEAU, D. A. Neural Network Perception for Mobile Robot Guidance. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

PRINCIPE, J.; EULIANO, N.; LEFEBVRE, W. *Neural and Adaptative Systems: Fundamentals through simulation*. [S.l.]: John Wiley and Sons, INC, 2000.

RAY, S.; KARGUPTA, H. A temporal sequence processor based on the biological reaction-diffusion process. *Complex Systems*, v. 9, n. 4, 1996.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, November 1958.

RUMELHART, D.; HILTON, G.; WILLIAMS, R. Learning representations by backpropagation errors. *Nature*, v. 323, p. 533–566, 1986.

TANI, J. On the interactions between top-down anticipation and bottom-up regression. *Frontiers in Neurorobotics*, v. 1, n. 2, Novembro 2007.

TANI, J.; NOLFI, S. Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems. *Neural Computation*, n. 12, p. 1131–1141, 1999.

TSOUKALAS, L. H.; UHRIG, R. Fuzzy and Neural Approaches in Engineering. Nova York: John Wiley and Sons, 1997.

WANG, D.; YUWONO, B. Antecipation-based temporal pattern generation. *IEEE Transactions on Systems, Man, Cybernetics*, v. 25, p. 615–628, 1995.

WATANABE, S. Pattern Recognition: Human and Mechanical. Nova York: Willey, 1985.

WEBB, A. Statistical Pattern Recognition. 2^a. ed. [S.l.]: John Wiley & Sons, 2002.

WERBOS, P. J. The roots of backpropagation: from ordered derivatives to neural networks and political forecasting. New York, NY, USA: Wiley-Interscience, 1994.

ZIEMKE, T. Remembering how to behave: Recurrent neural networks for adaptive robot behavior. 1999.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

<u>Baixar</u>	livros	de	Adm	inis	tra	ção

Baixar livros de Agronomia

Baixar livros de Arquitetura

Baixar livros de Artes

Baixar livros de Astronomia

Baixar livros de Biologia Geral

Baixar livros de Ciência da Computação

Baixar livros de Ciência da Informação

Baixar livros de Ciência Política

Baixar livros de Ciências da Saúde

Baixar livros de Comunicação

Baixar livros do Conselho Nacional de Educação - CNE

Baixar livros de Defesa civil

Baixar livros de Direito

Baixar livros de Direitos humanos

Baixar livros de Economia

Baixar livros de Economia Doméstica

Baixar livros de Educação

Baixar livros de Educação - Trânsito

Baixar livros de Educação Física

Baixar livros de Engenharia Aeroespacial

Baixar livros de Farmácia

Baixar livros de Filosofia

Baixar livros de Física

Baixar livros de Geociências

Baixar livros de Geografia

Baixar livros de História

Baixar livros de Línguas

Baixar livros de Literatura

Baixar livros de Literatura de Cordel

Baixar livros de Literatura Infantil

Baixar livros de Matemática

Baixar livros de Medicina

Baixar livros de Medicina Veterinária

Baixar livros de Meio Ambiente

Baixar livros de Meteorologia

Baixar Monografias e TCC

Baixar livros Multidisciplinar

Baixar livros de Música

Baixar livros de Psicologia

Baixar livros de Química

Baixar livros de Saúde Coletiva

Baixar livros de Serviço Social

Baixar livros de Sociologia

Baixar livros de Teologia

Baixar livros de Trabalho

Baixar livros de Turismo