



Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

**Implementação de um módulo de reconhecimento de fala como interface de
comando para robôs LEGO® MINDSTORMS® NXT 2.0**

Cezar Leandro Manica

CASCAVEL
2014

CEZAR LEANDRO MANICA

**IMPLEMENTAÇÃO DE UM MÓDULO DE RECONHECIMENTO DE
FALA COMO INTERFACE DE COMANDO PARA ROBÔS LEGO®
MINDSTORMS® NXT 2.0**

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Ciência da
Computação, do Centro de Ciências Exatas e Tec-
nológicas da Universidade Estadual do Oeste do
Paraná - Campus de Cascavel

Orientador: Prof. Josué Pereira de Castro

CASCADEL
2014

CEZAR LEANDRO MANICA

**IMPLEMENTAÇÃO DE UM MÓDULO DE RECONHECIMENTO DE
FALA COMO INTERFACE DE COMANDO PARA ROBÔS LEGO[®]
MINDSTORMS[®] NXT 2.0**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,
aprovada pela Comissão formada pelos professores:

Prof. Josué Pereira de Castro (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Adriana Postal
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Suzan Kelly Borges Piovezan
Colegiado de Engenharia com Ênfase em
Controle de Automação, FAG

Cascavel, 13 de novembro de 2014

AGRADECIMENTOS

Agradeço a todos que me apoiaram em toda a minha graduação, principalmente aos meus pais e irmãs, que nos momentos difíceis fizeram com que eu persistisse e chega-se até aqui. Agradeço aos meus amigos da faculdade Thiago Junior Vacari, Mauriverti da Silva Junior, Gean Carlo Peixoto, Alexandre Unfried parceiros nos momentos de estudos e confraternizações. E sou grato a todos que me ajudaram neste trabalho, Eduarda Bordini, Francielly Fruhauf, Juliano Richetti e a todos que se dedicaram um tempo para realização deste trabalho.

Lista de Figuras

2.1	Processo de aquisição do sinal de fala.	4
2.2	Sistema básico de reconhecimento de fala utilizando comparação de padrões.	5
2.3	Estrutura de um reconhecimento por comparação de padrões.	8
2.4	Diagrama de blocos do sistema de reconhecimento de voz acústico-fonético	9
2.5	Rede Perceptron.	12
2.6	Redes Perceptron Multi-Camadas.	12
2.7	Redes Recorrentes.	13
3.1	Neurônio biológico.	18
3.2	Neurônio artificial.	18
3.3	Funções de ativação	20
4.1	Configuração rede neural.	26
4.2	Matriz de confusão.	27
5.1	Configuração rede neural padrão nntool.	32
5.2	Matriz de confusão rede neural padrão nntool.	33
5.3	Configuração rede neural <i>radial basis (exact fit)</i>	34
5.4	Matriz de confusão rede neural <i>radial basis</i>	35
5.5	Configuração rede neural <i>feed-forward backpropagation</i> com função de ativação sigmóide.	36
5.6	Matriz de confusão rede neural <i>feed-forward backpropagation</i> com função de ativação sigmóide.	37

Lista de Tabelas

4.1 Saída da rede neural.	25
5.1 Resultado dos testes.	31

Lista de Abreviaturas e Siglas

A/D	Analógico - Digital
DFT	Transformada Discreta de Fourier
IBGE	Instituto Brasileiro de Geografia e Estatística
LPC	<i>Linear Predictive Coding</i>
MLP	Multi-layer Perceptron
RNA	Redes Neurais Artificiais

Sumário

Lista de Figuras	v
Lista de Tabelas	vi
Lista de Abreviaturas e Siglas	vii
Sumário	viii
Resumo	x
1 Introdução	1
1.1 Definição do problema	1
1.2 Objetivos	1
1.2.1 Geral	1
1.2.2 Específicos	2
1.3 Justificativas	2
1.4 Organização do trabalho	3
2 Sistemas de reconhecimento de fala	4
2.1 Sistema básico de reconhecimento de fala utilizando comparação de padrões . .	5
2.1.1 Processamento de sinal de fala	6
2.1.2 Comparação de padrões	6
2.1.3 Padrões de referência	6
2.1.4 Pós-processador	7
2.2 Reconhedores por comparação de padrões	7
2.3 Reconhedores baseados em análise acústico-fonética	9
2.4 Reconhedores que utilizam inteligência artificial	10
2.4.1 Principais arquiteturas	11
2.5 Problemas do reconhecimento de fala	13

3	Redes neurais aplicadas ao reconhecimento de fala	16
3.1	Principais características	16
3.2	Neurônio biológico	17
3.3	Neurônio artificial	18
3.4	Função de ativação	19
3.5	Algoritmo de treinamento	20
3.5.1	Aprendizado supervisionado	21
3.5.2	Aprendizado não-supervisionado	21
3.6	Redes neurais no reconhecimento de fala	22
4	Desenvolvimento	24
4.1	Aquisição Sonora	24
4.2	Rede neural	25
4.3	Envio dos comandos	27
4.3.1	<i>Bluetooth</i> [®]	28
4.3.2	Comandos de movimento	29
5	Considerações finais	30
5.1	Testes	30
5.2	Redes treinadas	31
5.3	Trabalhos futuros	37
	Referências Bibliográficas	38

Resumo

Neste trabalho descreve-se a implementação de um reconhecimento de fala para ROBÔS LEGO MINDSTORMS NXT 2.0, este reconhecedor após desenvolvido foi utilizado para realizar a movimentação do robô. Foi realizado então um levantamento de várias técnicas de reconhecimento de fala, adquirindo-se então o conhecimento necessário para se implementar o reconhecedor, optamos por utilizar redes neurais para realizar o reconhecimento. Começamos este trabalho com a aquisição dos comandos vocais, sendo estes sons utilizados para treinamento da rede neural. Para estas tarefas e também da comunicação com o robô e o envio dos comandos foi utilizada a ferramenta MATLAB e a biblioteca RWTH-Mindstorms NXT toolbox. Para concluir, descreve-se como foi a implementação, citando desde o treinamento da rede, ao envio dos comandos para o robô, finalizando com os resultados obtidos, testes, e sugestões para trabalhos futuros.

Palavras-chave: Reconhecimento de fala, Redes Neurais, MATLAB, LEGO Mindstorm, Reconhecimento de Padrões.

Capítulo 1

Introdução

1.1 Definição do problema

O trabalho apresenta o desenvolvimento de um interpretador de comandos vocais com base em redes neurais artificiais (RNA). Foi implementada e treinada uma rede capaz de identificar comandos vocais básicos e, a partir daí, efetuar o acionamento de um robô LEGO[®] Mindstorms[®]. O interpretador neural desenvolvido neste trabalho não depende do locutor, isto é, não há necessidade de se preocupar com as características vocais de um locutor.

Existem várias soluções para resolução do problema citado, muitas destas soluções utilizam redes neurais artificiais para identificar os comandos. Alguns autores como Amaral (2005) e Rodrigues (2009) utilizaram o robô LEGO Mindstorms como base para seus testes, ambos utilizaram como linguagem de programação LeJOS, Robolab e o MATLAB separadamente para desenvolvimento de RNA. Outros vão mais além utilizando como ferramenta de testes cadeiras de rodas (DECOM, 2012) para auxiliar portadores de necessidades físicas através do reconhecimento de fala. Existem soluções para as mais diversas áreas, o reconhecimento de fala nos proporciona um leque gigantesco de opções e aplicações para auxiliar e até mesmo facilitar a vida das pessoas.

1.2 Objetivos

1.2.1 Geral

Realizar a implementação de um sistema de reconhecimento de fala que seja capaz de traduzir comandos vocais independente de locutor em movimentos realizados por um robô LEGO

Mindstorms. Os movimentos a serem realizados incluem: frente, ré, direita, esquerda.

1.2.2 Específicos

O objetivo geral deste trabalho pode ser dividido nos seguintes objetivos específicos:

- Realizar um levantamento bibliográfico sobre o tema.
- Analisar as técnicas de reconhecimento de fala mais usadas, para identificar a mais adequada para implementação em robôs LEGO NXT.
- Avaliar a biblioteca RWTH do MATLAB[®] como ferramenta de implementação.
- Escolher uma técnica eficiente para resolver o problema, e que possa ser implementada em tempo hábil.
- Implementação da técnica de reconhecimento de fala escolhida.
- Implementar um módulo de reconhecimento de comandos, capaz de reconhecer os comandos de voz produzidos pelo usuário.
- Implementar um módulo de execução de comandos, capaz de transformar os comandos de voz do usuário em instruções interpretáveis pelo LEGO Mindstorms NXT 2.0 e enviá-las ao robô.
- Integrar os módulos e realizar testes com o sistema.

1.3 Justificativas

Aproximadamente 13,3 milhões de brasileiros são portadores de necessidades motoras especiais segundo o IBGE (Instituto Brasileiro de Geografia e Estatística) (censo 2010)¹. Podemos aplicar os conhecimentos obtidos neste trabalho em melhorias na mobilidade de cadeirantes, principalmente os deficientes tetraplégicos, podendo ampliar o reconhecimento para que um usuário dotado de necessidades especiais possa controlar muito mais que sua própria cadeira, podendo acionar portas, janelas, cortinas, etc. Para aplicações em cadeiras de rodas, podemos

¹http://biblioteca.ibge.gov.br/visualizacao/periodicos/99/cd_2010_resultados_gerais_amostra.pdf

aplicar os comandos rápido, médio, lento, elevar, entre outros tudo depende da necessidade de cada cadeirante (DECOM, 2012).

1.4 Organização do trabalho

O capítulo 2, inicia falando sobre o processo de aquisição do sinal sonoro, logo a seguir abordamos sobre o sistema básico de reconhecimento de fala, algumas das principais técnicas e alguns problemas referentes ao reconhecimento de fala.

No capítulo 3, são apresentadas as redes neurais artificiais, suas principais propriedades, e os métodos de aprendizado.

No capítulo 4, estão descritos os detalhes do desenvolvimento do trabalho.

O capítulo 5 encerra o trabalho, citando as considerações finais a respeito do desenvolvimento do projeto, apresenta-se os testes e as redes neurais utilizadas e para finalizar, algumas sugestões são citadas para trabalhos futuros.

Capítulo 2

Sistemas de reconhecimento de fala

Um sistema de reconhecimento de fala tem como entrada ondas sonoras produzidas pelo ser humano, captadas com um microfone ligado à placa de som, a qual é responsável também pela transformação do sinal analógico em sinal digital (SILVA, 2009), mostrada na Figura 2.1.

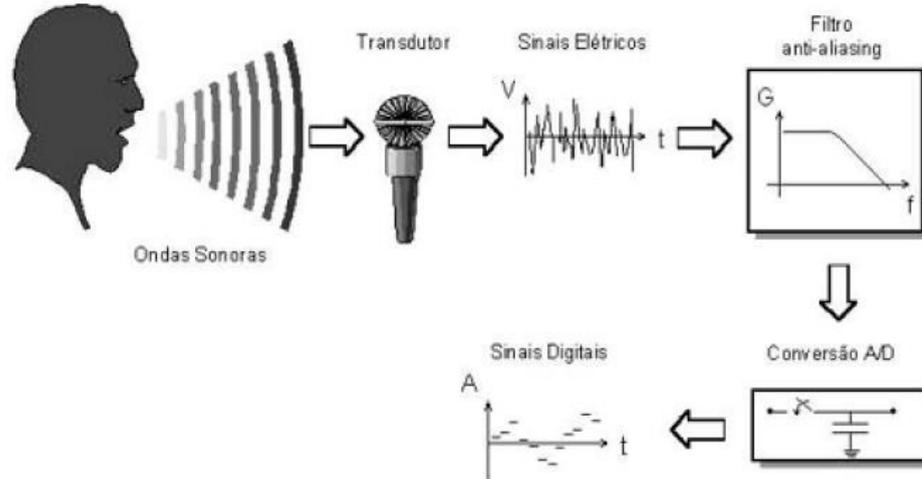


Figura 2.1: Processo de aquisição do sinal de fala.
Fonte: (BRAGA, 2006)

Após a aquisição do sinal digital, deve-se utilizar o reconhecedor de fala mais adequado. Existem basicamente três classes de reconhecedores:

- Reconhecedores por comparação de padrões;
- Reconhecedores baseados em análise acústico-fonética;

- Reconhecedores que utilizam inteligência artificial.

Levando em conta que os sistemas que apresentam os melhores resultados em reconhecimento de fala utilizam a técnica de comparação de padrões, utilizaremos redes neurais com esta técnica para a implementação do trabalho proposto.

2.1 Sistema básico de reconhecimento de fala utilizando comparação de padrões

Neste sistema, comparamos as características do sinal de fala com padrões previamente armazenados, e então escolhe-se o padrão mais próximo do sinal de entrada, sendo que ele pode estar sujeito a algumas restrições do pós-processador.

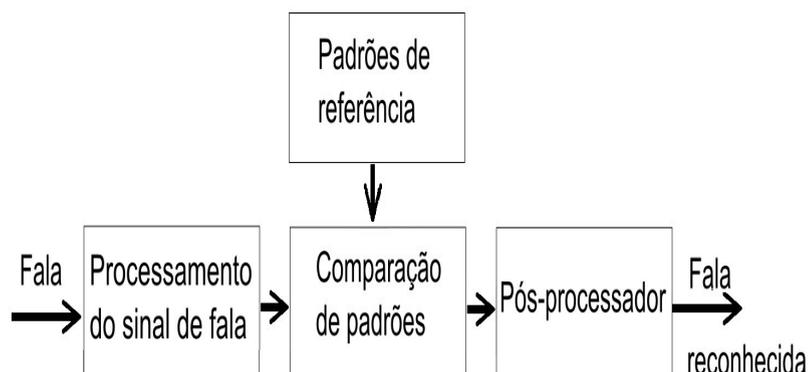


Figura 2.2: Sistema básico de reconhecimento de fala utilizando comparação de padrões.

Fonte: (MARTINS, 1997)

Na Figura 2.2 adaptada de Martins (1997), podemos observar o sistema básico para reconhecimento de fala utilizando a técnica de comparação de padrões. O uso de modelos representando palavras é o mais indicado para vocabulários pequenos pois a quantidade de cálculos necessários aumenta a medida que o tamanho do vocabulário cresce. Para grandes vocabulários e reconhecimento de fala contínua o uso de unidades menores de palavras é mais apropriado. A composição do sistema básico é a seguinte: Processamento de sinal de fala, padrões de referência, comparação de padrões e pós-processador (MARTINS, 1997).

2.1.1 Processamento de sinal de fala

O sinal de fala é digitalizado e convertido em um conjunto de parâmetros espectrais e temporais para que seja possível a comparação com os diferentes padrões. Usa-se parâmetros pelo fato de que comparações entre formas de onda de sinais de fala são muito complicadas e não relevantes, como exemplo, a distorção de fase (sons emitidos em distâncias desiguais podem chegar ao ouvido de uma pessoa em instantes suficientemente diferentes para ocorrer um fenômeno de interferência, ocorrendo então o que se denomina distorção de fase) altera a forma de onda mas o ouvido humano é insensível a essa distorção (MARTINS, 1997).

Muitos parâmetros tem sido propostos para representação do sinal de fala. Os mais usados são os derivados dos coeficientes LPC (*Linear Predictive Coding*) e os derivados diretamente do espectro do sinal.

Para reconhecedores de palavras isoladas é necessária a detecção dos pontos limitantes de cada palavra, existem vários algoritmos de detecção início/fim, os quais utilizam parâmetros como energia e taxa de cruzamento de zero (espectro do som tem variação de frequência de -1 a 1 decibéis, é analisado então esta mudança de frequência) para separar o sinal de fala do ruído (SILVA, 2009).

2.1.2 Comparação de padrões

O conjunto de parâmetros representando a fala desconhecida é comparado com os diversos padrões de referência. Esse conjunto de parâmetros é da mesma natureza daquelas usadas durante o treinamento.(MARTINS, 1997).

2.1.3 Padrões de referência

Obtemos os padrões de referência a partir de exemplos das unidades a serem reconhecidas. Para reconhecedores independentes do locutor, é necessária a apresentação de vários exemplos de cada unidade, faladas por diferentes locutores e considerado os diferentes sotaques, conseguindo desta forma um sistema mais robusto (MARTINS, 1997).

2.1.4 Pós-processador

As distâncias ou probabilidades obtidas na comparação com os padrões de referência são usadas para escolher o padrão que melhor corresponde ao padrão desconhecido. Quando são usados padrões de referência que não contém um ou mais parâmetros a serem seguidos, para que seja possível a comparação, é necessário o uso de uma medida de distância e um procedimento de alinhamento temporal (MARTINS, 1997).

2.2 Reconhedores por comparação de padrões

Um exemplo de reconhedor de fala por comparação de padrões é o modelo Oculto de Markov. Podemos dividir este reconhedor em 4 etapas (BRESOLIN, 2003):

Características de medida: Uma sequência de medida é feita na entrada do sinal para definir o "teste padrão". Para sinais de voz, as características de medida são geralmente a saída de algum tipo de analisador espectral, tal como um Banco de Filtros, ou a Transformada Discreta de Fourier (DFT);

Padrões de treinamento: Um ou mais padrões de teste correspondentes ao sinal da fala são usados para criar um padrão representativo das características deste sinal. O padrão resultante pode ser derivado da média dos sinais, ou um valor estatístico das características;

Classificação dos padrões: Aqui um sinal desconhecido é comparado com cada padrão de referência e a medida de similaridade entre o sinal de teste e cada padrão de referência é computado;

Decisão lógica: Neste passo vários padrões de referência com valores similares são usados para decidir qual padrão de referência melhor combina com o sinal de teste desconhecido. A Figura 2.3 (BRESOLIN, 2003) mostra um diagrama em blocos de um reconhecimento por comparação de padrões.

Em geral os pontos fortes e fracos da aproximação por reconhecimento de padrões são, segundo Bresolin (2003):

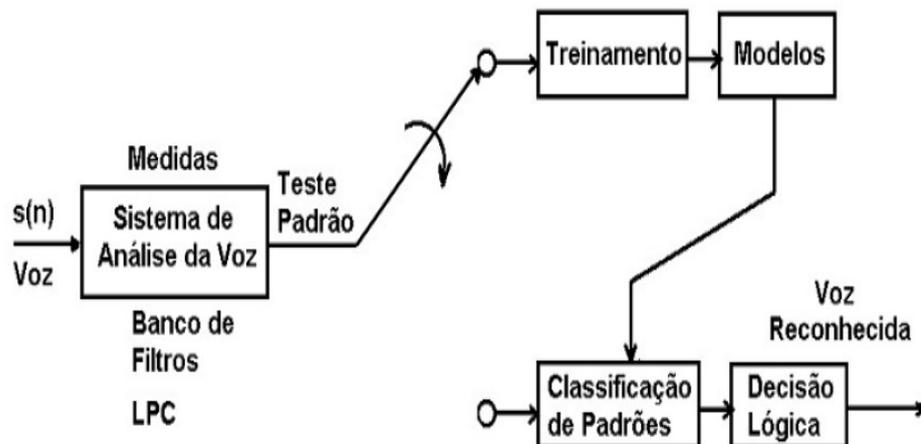


Figura 2.3: Estrutura de um reconhecimento por comparação de padrões.
 Fonte: (BRESOLIN, 2003)

Pontos fortes:

- Quanto maior o treinamento, mais alto é o desempenho na realização da tarefa;
- Um som de voz não específico e conhecido é usado explicitamente no sistema. Então este método é relativamente insensível à escolha de vocabulários, palavras, perguntas, sintaxe e semântica;
- Devido ao sistema ser insensível às classes de sons, as técnicas básicas são aplicadas a uma vasta gama de sons da fala, incluindo frases, palavras, fonemas e sílabas. Então um conjunto básico de técnicas desenvolvidas para uma classe de som pode ser diretamente aplicado para diferentes classes de sons, com pequenas ou nenhuma modificações dos algoritmos;
- Uma sintaxe pré-definida melhora a precisão e diminui o tempo de computação.

Ponto fraco:

- Os padrões são sensíveis ao meio ambiente onde se fala e também às características do meio onde foi criada a fala. Isto significa que as características espectrais da fala são afetadas pela transmissão e por ruídos de fundo;

2.3 Reconhedores baseados em análise acústico-fonética

Os sistemas baseados em análise acústico-fonética decodificam o sinal da fala com base nas suas características acústicas e nas relações entre essas características. A Figura 2.4 representa a estrutura de um sistema baseado em análise acústico-fonética (MARTINS, 1997). O primeiro passo no processamento é o sistema de análise do sinal de voz (passo comum para todas as aproximações), o qual produz uma representação apropriada (espectro) das características de variação no tempo do sinal de voz.

As técnicas mais comuns de análise espectral são: Banco de Filtros, "Linear Predictive Coding" (LPC) e a DFT (Transformada Discreta de Fourier).

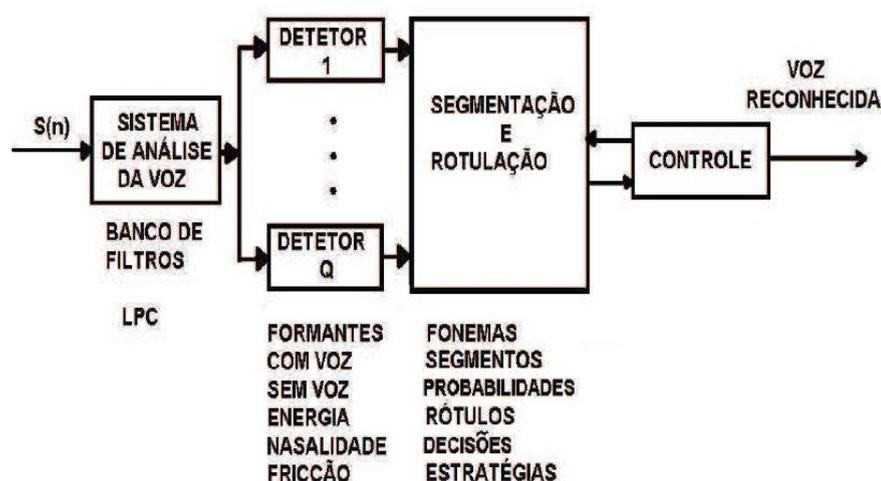


Figura 2.4: Diagrama de blocos do sistema de reconhecimento de voz acústico-fonético
Fonte: (BRESOLIN, 2003)

O próximo passo é o estágio de detecção das características do sinal. Converte-se as medidas espectrais em um conjunto de características que descrevem as amplas propriedades das diferentes unidades fonéticas. Bresolin (2003) e Alencar (2005) citam as seguintes características propostas para o reconhecimento da voz:

- Nasal (presença ou não de ressonância nasal);
- Vocálicos (sons das vogais);
- Fricção (presença ou não de excitação da fala);

- Oclusivas (são caracterizados por um excesso de pressão criado em um ponto do aparelho vocal, seguido de um desprendimento repentino de ar) (ALENCAR, 2005);
- Formantes (frequências do sinal e suas três primeiras ressonantes);
- Classificação com ou sem voz (excitação do sinal periódico ou não periódico);
- Porções de alta e baixa energia (amplitude da frequência).

O estágio de detecção das características geralmente consiste de um conjunto de detectores que operam em paralelo e usam o processamento apropriado para fazer a decisão da presença ou não da característica do som analisado.

O próximo passo é a fase de segmentação e rotulação pelo qual o sistema tenta encontrar regiões estáveis (onde as características mudam muito pouco) e então marcar a região segmentada de acordo com as unidades fonéticas individuais.

Bresolin (2003) cita alguns problemas que estão associados a esta técnica, sendo eles:

- Este método requer um profundo conhecimento das propriedades acústicas da unidade fonética;
- As escolhas das características são feitas com base na intuição;
- O projeto dos classificadores de som não é otimizado;
- A diversidade cultural é muito grande, existem diversos sotaques e variações na entonação vocal o que dificulta muitas vezes o reconhecimento da fala, sendo que muitas palavras mudam o significado de acordo com o contexto.

2.4 Reconhedores que utilizam inteligência artificial

Este tipo de reconhedor explora os conceitos dos reconhedores por comparação de padrões e acústico-fonética. Dois aspectos importantes deste tipo de reconhecimento envolvem os conceitos de aquisição de conhecimento, ou seja, a aprendizagem e também a adaptação (RODRIGUES, 2009).

2.4.1 Principais arquiteturas

A arquitetura de uma RNA é um parâmetro importante na sua criação, devido à particularidade de cada problema. Redes sem camada intermediária só conseguem resolver problemas linearmente separáveis, problemas que envolvem processamento temporal são melhores resolvidas com redes recorrentes (RODRIGUES, 2009).

Alguns parâmetros foram definidos por Braga; Carvalho; Ludemir (2000), sendo eles número de camadas, conexões e conectividade:

- **Números de camadas:** Uma rede neural pode ser classificada quanto ao número de camadas. Podendo ser dividida em uma camada única ou múltiplas camadas.
- **Conexões:** Classificação quanto ao tipo de conexão, sendo divididas em *feedforward*, que são redes onde a saída de um neurônio em qualquer camada da rede não pode ser usada como entrada de neurônios de camadas anteriores. Outro tipo de conexão são as redes *feedback*, em que ao contrário das redes tipo *feedforward*, as saídas de alguma camada da rede podem ser usadas como entradas em algum neurônio de camadas anteriores.
- **Conectividade:** As redes podem ser classificadas em rede fracamente conectada e rede completamente conectada, que é quando todos os neurônios estão conectados entre si.

Podemos identificar três tipos básicos de estrutura de redes neurais (HAYKIN, 2000):

Redes Perceptron: Os nós da camada de entrada são conectados aos nós da camada de saída.

Essa conexão pode ser parcial, na qual nem todos os neurônios de entrada estão conectados com os neurônios de saída, ou total, no qual existe conexões entre todos neurônios entrada-saída. A Figura 2.5 (BRAGA, 2006) exhibe esta arquitetura.

Redes Perceptron Multi-Camadas: A diferença desta arquitetura para a anterior é o acréscimo de camadas intermediárias, com seus respectivos neurônios ocultos. O número de camadas determina, além de outras coisas, a eficiência da rede. Redes de camadas múltiplas conseguem resolver problemas não linearmente separáveis. Contudo, a determinação da quantidade de camadas intermediárias é realizado na base de tentativa-e-erro. Alguns autores apontam que apenas uma única camada oculta é suficiente para a obtenção de uma

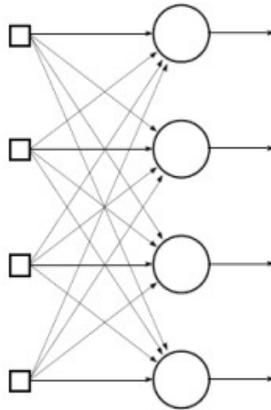


Figura 2.5: Rede Perceptron.
 Fonte: (BRAGA, 2006)

solução satisfatória, enquanto outros estudos indicam que duas camadas, ou mesmo três delas atinjam melhores resultados (BRAGA; CARVALHO; LUDERMIR, 2000). Apesar disso, o problema em questão é quem de fato determina o número de camadas a ser utilizado. Complexidade, tempo de treinamento, tempo de classificação, e outros parâmetros possíveis devem ser levados em consideração quando da escolha da quantidade de camadas ocultas. As conexões entre os neurônios das camadas ocultas e das camadas de saída, podem ser totalmente ou parcialmente conectados. A Figura 2.6 ilustra um modelo de redes perceptron multi-camadas.

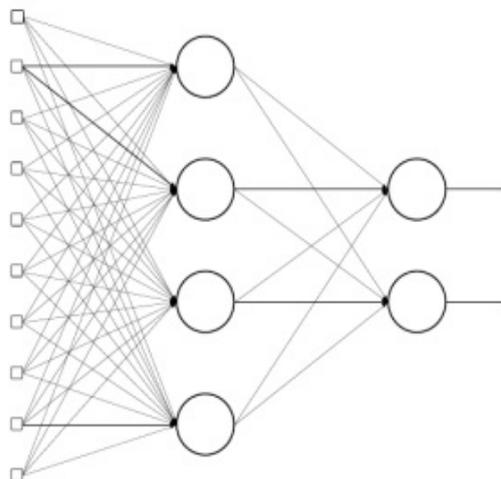


Figura 2.6: Redes Perceptron Multi-Camadas.
 Fonte: (BRAGA, 2006)

Redes Recorrentes: São Redes neurais que utilizam um ou mais laços de realimentação, esse tipo de rede permite a presença de memória, que por sua vez introduz um comportamento dinâmico ao sistema (BRAGA; CARVALHO; LUDERMIR, 2000) (HAYKIN, 2000). A complexidade das conexões da rede tende a ser aumentada, se comparada às redes não-recorrentes. A realimentação pode ser local, na qual a saída de um neurônio serve de entrada para outro neurônio, ou global, na qual a saída de uma camada serve de entrada para outra. A Figura 2.7 ilustra um exemplo dessa arquitetura de rede, onde Z^{-1} representa elementos de atraso unitário o que resulta em um comportamento dinâmico não-linear (BRAGA, 2006).

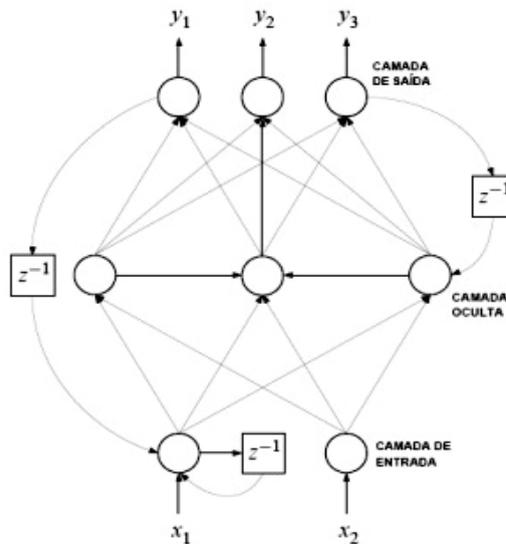


Figura 2.7: Redes Recorrentes.
 Fonte: (BRAGA, 2006)

2.5 Problemas do reconhecimento de fala

Encontramos muitas dificuldades para implementar sistemas de reconhecimento de fala devido à sua complexidade e limitações, o que consequentemente impede que o uso destes sistemas se ampliem. A seguir, descrevemos os problemas mais comuns.

Dependência do emissor versus independência do emissor: Um sistema dependente do emissor é definido por um sistema capaz de reconhecer a voz do locutor ou dos locutores, os quais treinaram o sistema. Deste modo sistemas independentes do emissor são capazes

de reconhecerem qualquer pessoa falando e traduzir os sons em textos e comandos. Chegar a um sistema que utiliza a independência do emissor requer um maior esforço, pois há amplas variações de entonação e pronúncia de cada pessoa, sendo mais simples criar um sistema dependente de emissor, treinado a partir de padrões vocais de um único locutor (AMARAL, 2005) (RODRIGUES, 2009).

Fala Contínua versus Fala Isolada: Esse conceito está relacionado à existência ou não de pausas entre uma palavra e outra e aos métodos de reconhecimento da fala conhecidos como método global e método analítico.

O método global, também chamado de método de reconhecimento de palavras, utiliza técnicas de reconhecimento de formas a comparar, globalmente, a palavra a ser reconhecida com as diversas amostras de referência armazenadas (HUGO, 1995).

Já o método analítico, utilizado no reconhecimento da fala contínua, consiste em segmentar a mensagem em constituintes elementares, como fonemas e sílabas, identificá-los e reconstruir a mensagem para que, posteriormente, possa ser completamente analisada (AMARAL, 2005). Segundo Rich & Knight (1993), a eficiência desse método é dificultada porque a mesma palavra poder ser pronunciada de modos diferentes em contextos distintos.

Tempo real versus processamento *offline*: Há determinadas aplicações em que se exige reconhecimento em tempo real, neste tipo de aplicação palavras devem ser interpretadas à medida que o locutor as pronuncia. Em outras situações isto já não é exigido, permitindo então um tempo maior no processamento computacional no reconhecimento das palavras (AMARAL, 2005) (RODRIGUES, 2009) (RICH; KNIGHT, 1994).

Vocabulário grande versus pequeno: Um aspecto importante que afeta o desempenho de um sistema de reconhecimento de fala é o seu vocabulário. Um sistema com vocabulário pequeno apresenta um melhor desempenho, elimina ambiguidades dentre outras coisas. O vocabulário grande pode levar o sistema a uma possível ambiguidade, se contiver palavras que o algoritmo reconhecedor considera semelhantes (AMARAL, 2005) (RODRIGUES, 2009) (RICH; KNIGHT, 1994).

Gramática ampla versus restrita: Restringir a gramática é importante nos sistemas de reconhecimento de fala porque torna o sistema mais simples, limitando o espaço de busca do algoritmo responsável pelo reconhecimento (AMARAL, 2005) (RODRIGUES, 2009) (RICH; KNIGHT, 1994).

Capítulo 3

Redes neurais aplicadas ao reconhecimento de fala

Redes neurais artificiais são modelos computacionais inspirados no sistema nervoso de seres vivos. Possuem capacidade de aquisição e manutenção do conhecimento e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais, que são interligados por um grande número de interconexões (sinapses artificiais), sendo as mesmas representadas por vetores/matrizes de pesos sinápticos (SILVA; SPATTI; FLAUZINO, 2010).

3.1 Principais características

Os autores Silva, Spatti e Flauzino (2010), citam algumas das principais características sobre redes neurais:

Adaptação por experiência: As adaptações dos parâmetros internos da rede, tipicamente seus pesos sinápticos, são ajustadas a partir da apresentação sucessiva de exemplos (padrões, amostras, medidas) relacionados ao comportamento do processo, possibilitando a aquisição do conhecimento por experimentação;

Capacidade de aprendizado: Por intermédio da aplicação de um método de treinamento, a rede consegue extrair o relacionamento existente entre as diversas variáveis que compõem a aplicação;

Habilidade de generalização: Após o processo de treinamento da rede, essa é capaz de generalizar o conhecimento adquirido, possibilitando estimar soluções que eram até então desconhecidas;

Organização de dados: Baseada em características intrínsecas envolvendo determinado conjunto de informações a respeito de um processo, a rede é capaz de realizar a sua organização interna visando possibilitar o agrupamento de padrões que apresentam particularidades em comum;

Tolerância a falhas: Devido ao elevado nível de interconexões entre os neurônios artificiais, a rede neural torna-se um sistema tolerante a falhas quando parte de sua estrutura interna é sensivelmente corrompida;

Armazenamento distribuído: O conhecimento a respeito do comportamento de determinado processo dentro de uma arquitetura neural é realizada de forma distribuída entre as diversas sinapses de seus neurônios artificiais, permitindo então um incremento da robustez da arquitetura frente a eventuais neurônios que se tornem inoperantes;

Facilidade de prototipagem: Dependendo da especificidade da aplicação, a implementação da maioria das arquiteturas neurais pode ser facilmente prototipada em hardware ou em software, pois, após o processo, os seus resultados são normalmente obtidos por algumas operações matemáticas elementares.

3.2 Neurônio biológico

O cérebro humano tem a capacidade de processar informações de forma não-linear e paralela permitindo a resolução de tarefas complexas. Esse poder de processamento é creditado aos neurônios e suas conexões. A partir da “experiência” acumulada com o tempo, são reforçadas ou inibidas as conexões entre eles. O potencial do cérebro humano, então, não provém da simplicidade de cada neurônio em si, mas da complexidade e da grande quantidade das interconexões formadas ao longo do tempo (HAYKIN, 2000) (BRAGA, 2006).

Os trabalhos de Ramon e Cajal foram determinantes para a compreensão do funcionamento do cérebro (HAYKIN, 2000), os quais introduziram o conceito de que neurônios, na realidade, são os constituintes do cérebro. O processo de aprendizado está fortemente relacionado aos próprios neurônios. Para que esse processo ocorra, uma ligação entre os neurônios é necessária. Essa ligação é denominada sinapse, cuja finalidade é impor ao neurônio receptivo um grau de excitação ou inibição. Contudo, esses sinais inibitórios e excitatórios trafegam através dos

axônios, que nada mais são que as linhas de transmissão, e dos dendritos, que são as zonas receptivas (REIS, 1999) (BRAGA, 2006) (HAYKIN, 2000). Na Figura 3.1 vemos um exemplo de neurônio biológico (VALIATI, 2000).

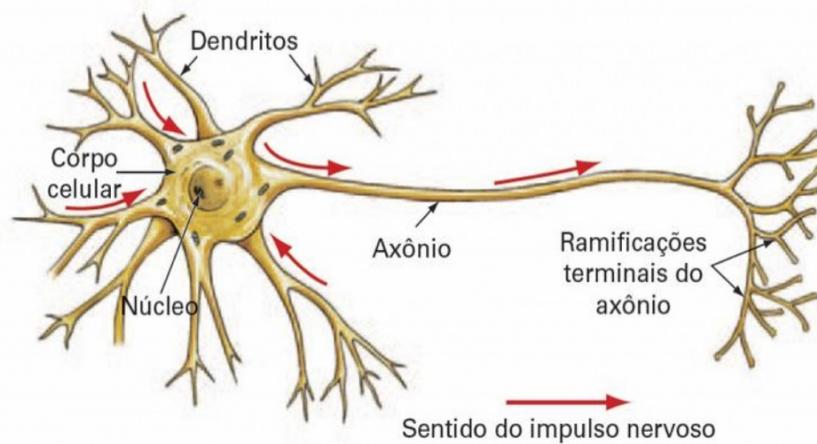


Figura 3.1: Neurônio biológico.
Fonte: (BORGES, 2012)

3.3 Neurônio artificial

O modelo geral de neurônio, criado por McCulloch e Pitts, representa o funcionamento básico do neurônio artificial, exibido na Figura 3.2.

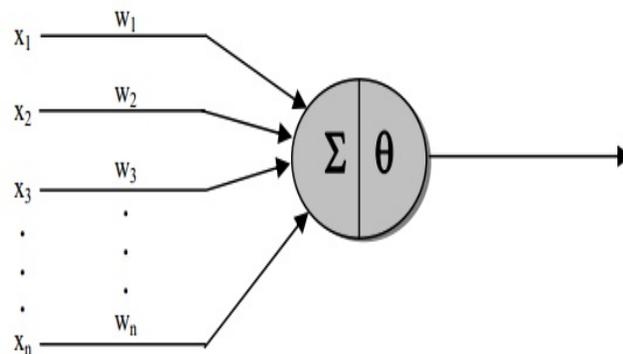


Figura 3.2: Neurônio artificial.
Fonte: (MCCULLOCH; PITTS, 1943)

Este modelo de neurônio artificial é uma simplificação do neurônio biológico, sua descrição matemática resultou em um modelo com n terminais de entrada $x_1, x_2, x_3, \dots, x_n$ (representando

os dendritos) e um terminal de saída (correspondente ao axônio). Com a finalidade de emular o comportamento das sinapses, os terminais de entrada possuem pesos acoplados $w_1, w_2, w_3, \dots, w_n$, estes valores podem ser tanto positivos quanto negativos.

A ativação do neurônio é obtida através da aplicação de uma "função de ativação", que habilita ou não a saída, dependendo do valor da soma ponderada das suas entradas (VALIATI, 2000) (MARTINS, 1997).

3.4 Função de ativação

Existem várias funções de ativação que podem ser aplicadas aos nós para produzir uma saída qualquer e não necessariamente zero ou um (VALIATI, 2000). As funções de ativação definem, segundo o modelo exposto, o valor da saída em função do campo local. As funções mais utilizadas são descritas a seguir, podendo ser encontradas com variações nos limites de definição (BRAGA, 2006) (BRAGA; CARVALHO; LUDERMIR, 2000) (HAYKIN, 2000). A Figura 3.3 exibe estas funções.

Função de limiar: Sua saída é 1 quando a entrada é positiva e 0 caso contrário, mostrada na função 3.1. Essa é a função utilizada no neurônio perceptron multi-camadas. A Figura 3.3(a) (BRAGA, 2006) ilustra esse tipo de função.

$$\begin{aligned} f(x) &= 1, x \geq 0 \\ f(x) &= 0, x < 0 \end{aligned} \tag{3.1}$$

Função linear por partes: produz valores constantes (v) em um determinado intervalo. Um exemplo de uma função linear por partes é a função 3.2, também representada na Figura 3.3(b) (BRAGA, 2006).

$$\begin{aligned} f(x) &= 1, x \geq 1/2 \\ f(x) &= x, -1/2 < x < 1/2 \\ f(x) &= 0, x < -1/2 \end{aligned} \tag{3.2}$$

Função sigmóide: É o tipo de função mais utilizada na elaboração de RNAs devido ao fato de ser diferenciável. Um exemplo de função sigmóide é a função logística, que é dada pela equação 3.3. A Figura 3.3(c) (BRAGA, 2006) (VALIATI, 2000) ilustra a função sigmóide, sendo que α determina a inclinação da curva.

$$f(x) = \frac{1}{1 + \exp(-\alpha x)} \quad (3.3)$$

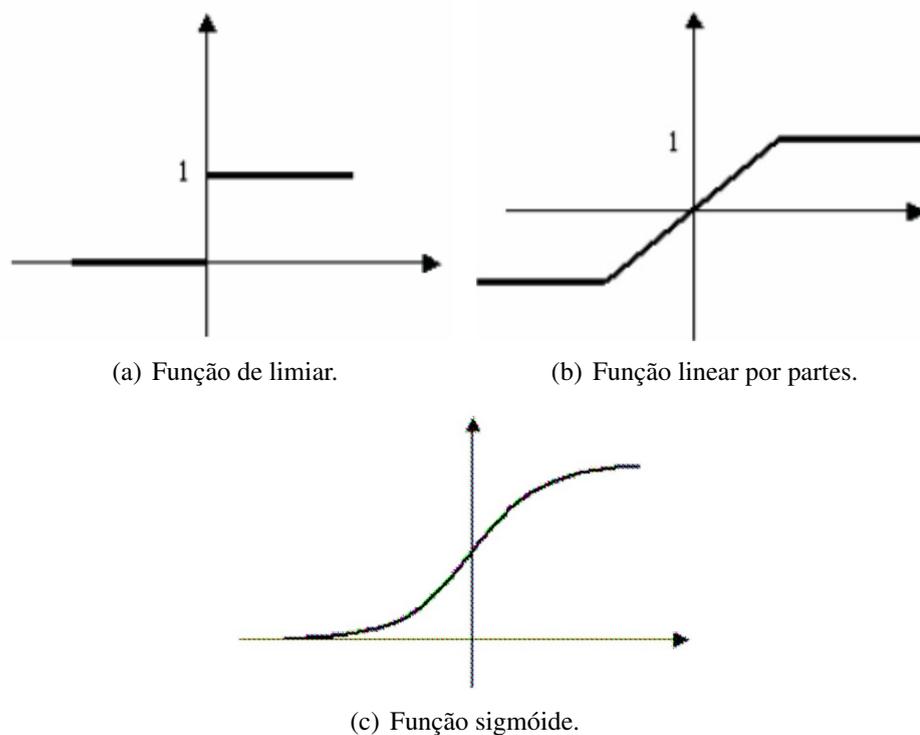


Figura 3.3: Funções de ativação
 Fonte: (BRAGA, 2006) (VALIATI, 2000)

3.5 Algoritmo de treinamento

A capacidade de aprender a partir de apresentação de padrões que exprimem o comportamento do sistema é um grande destaque para as RNAs. Após a rede ter aprendido o relacionamento entre as camadas e saídas, esta é capaz de generalizar soluções. A partir de sinais inseridos na entrada da rede ela será capaz de produzir uma saída próxima da esperada.

O processo de treinamento consiste da aplicação de passos ordenados que sejam necessários para sintonização dos pesos sinápticos e limiares de seus neurônios, tendo como objetivo final

a generalização de soluções a serem produzidas pelas suas saídas, cujas respostas são representativas do sistema físico em que estas estão mapeando.

O conjunto desses passos ordenados visando o treinamento da rede é denominado de algoritmo de aprendizagem. Enquanto está ocorrendo o processo de treinamento da rede, cada apresentação completa das amostras pertencentes ao subconjunto de treinamento, visando o ajuste dos pesos sinápticos e limiares de seus neurônios, será denominada de época de treinamento (SILVA; SPATTI; FLAUZINO, 2010).

3.5.1 Aprendizado supervisionado

Também conhecida como aprendizagem supervisionada, nesse tipo de paradigma um conjunto de exemplos entrada-saída é fornecido à rede por meio de um professor que conhece o ambiente (domínio do problema). No entanto, a rede neural não conhece esse ambiente. Nesse processo de aprendizagem, uma medida utilizada para incorporar conhecimento à rede, é o erro, o qual consiste na diferença entre a resposta apresentada $y(t)$ pela rede e a resposta desejada $d(t)$. O erro $e(t)$ é usado para o ajuste dos parâmetros da rede, através de um processo iterativo (BRAGA, 2006) (SILVA; SPATTI; FLAUZINO, 2010) (RODRIGUES, 2009).

$$e(t) = d(t) - y(t) \quad (3.4)$$

3.5.2 Aprendizado não-supervisionado

Diferente do supervisionado, durante a aplicação de um algoritmo de aprendizado baseado em treinamento não-supervisionado inexitem as respectivas saídas desejadas. Em consequência disso, a própria rede deve se auto organizar em relação às particularidades existentes entre os elementos componentes do conjunto total de amostras, identificando subconjuntos que contenham similaridade. Os pesos sinápticos e limiares dos neurônios da rede são então ajustados pelo algoritmo de aprendizado de forma a refletir esta representação internamente dentro da própria rede (SILVA; SPATTI; FLAUZINO, 2010).

3.6 Redes neurais no reconhecimento de fala

As redes neurais podem ser uma alternativa viável se comparada aos modelos estatísticos tradicionais, se mostrando adequada na extração de características (KOHONEN, 1988) (BRAGA, 2006).

Vários modelos têm sido apresentados e os mais utilizados para reconhecimento de fala são (MARTINS, 1997):

Redes Recorrentes: São sistemas em que as entradas de cada elemento consistem de uma combinação das entradas da rede com as saídas de outros elementos da rede. Essas redes são apropriadas para sistemas com entradas que podem ser representadas por valores binários (BRAGA; CARVALHO; LUDERMIR, 2000) (HAYKIN, 2000);

Multi-layer Perceptron (MLP): Redes MLP têm sido usadas para reconhecimento de fala contínua e palavras isoladas (MARTINS, 1997).

As vantagens são as seguintes:

- Uso de processamento paralelo;
- Não são feitas suposições sobre as propriedades do sinal de fala;
- Treinamento discriminativo, isto é, ao mesmo tempo em que se maximiza o valor de saída desejada, minimiza-se os valores das outras saídas da rede;
- Pode-se produzir qualquer função não-linear das entradas;
- Coeficientes da rede podem ser adaptados em tempo real para melhorar o desempenho da mesma.

Mapas auto-organizáveis de Kohonen: Por terem inspirações biológicas, as redes auto-organizadas podem ser aplicadas na resolução de diversos tipos de problemas, principalmente em problemas de reconhecimento de padrões e categorização de dados em que as classes não são conhecidas antecipadamente. A ideia de utilizarmos a rede auto-organizada para reconhecimento de padrões é buscar agrupar padrões que compartilham características comuns apenas em uma classe. Para realizar este agrupamento (*clusters*),

um algoritmo auto-organizado necessita encontrar características significativas nos conjuntos de dados de entrada, sem o auxílio de um professor. A utilização deste algoritmo só é possível em casos em que houver redundância nos conjuntos de dados de entrada. A redundância dos dados de entrada fornece conhecimentos à rede sobre semelhanças e diferenças entre estes dados, enquanto que a ausência da redundância torna impossível encontrar similaridade nas características dos padrões (KOHONEN, 1988) (BRAGA, 2006) .

Capítulo 4

Desenvolvimento

Como dito no capítulo 1 a implementação será separada em dois módulos, um responsável por reconhecer os comandos de voz emitidos pelo usuário, e o outro que realiza a conversão dos comandos de voz em instruções de movimentação e o envio destas para o LEGO Mindstorms NXT 2.0.

4.1 Aquisição Sonora

O MATLAB é uma ferramenta matemática que tem como principal finalidade a criação de cálculos matriciais e funções matemáticas. Também oferece várias formas de captura de sons e a criação de interface através do comando `GUIDE` (MATHWORKS, 2014).

Antes da captura do som é necessário criar um objeto para a gravação do áudio. Utiliza-se então a função `audiorecorder`, esta função que tem como parâmetros de entrada um `Fs` que define a taxa de amostragem (em Hz), um `nBits` que se refere ao tamanho da amostra, e o número de canais (`nChannels`).

Fs: Os valores válidos dependem do hardware de áudio específico instalado. Os valores típicos suportados pela maioria das placas de som são 8000, 11025, 22050, 44100, 48000 e 96000 Hz, sendo o padrão 8000 Hz.

nBits: Bits por amostra. Os valores válidos dependem do hardware de áudio instalado: 8, 16, ou 24 bits, sendo o padrão 8 bits.

nChannels: Número de canais sendo 1 (mono) e 2 (estéreo), padrão mono.

Após criar o objeto e atribuí-lo a uma variável inicia-se a gravação com a função `recordblocking`, esta função que tem como entrada o objeto e o tempo de gravação. Para armazenar os dados e posteriormente desenhá-los com a função `plot` é necessário armazená-los em uma matriz os valores dos sinais gravados, fazemos isso utilizando a função `getaudiodata` tendo como entrada o objeto da gravação.

Para que posteriormente possa-se fazer novos testes com os dados de som gravados, foram salvas as informações de cada faixa de áudio separadas, utilizando a função `save`, e para se carregar estas informações a função `load`.

4.2 Rede neural

Como a taxa de amostragem do áudio é alta, foi utilizado a média para reduzi-la e os valores adquiridos foram utilizados como neurônios de entrada. A cada 80 valores tira-se a média, caindo de 8000 entradas para 100, tornando-se então viável a utilização da rede neural.

Para um conjunto pequeno de palavra a média se torna útil, mas com o aumento destas palavras a média pode não ser a melhor opção para diminuição da taxa de amostragem, pois os valores podem acabar se tornando muito parecidos para palavras semelhantes.

Para representar a saída da rede inicialmente propõem-se 4 bits, sendo o suficiente para representar 4 comando básicos, caso seja necessário pode-se aumentar o tamanho da saída para que ela suporte mais comandos, as saídas da rede estão representadas na Tabela 4.1.

Tabela 4.1: Saída da rede neural.

Saída da rede neural	
0001	Ré
0010	Frente
0100	Direita
1000	Esquerda

A rede neural foi criada a partir da função `nnTool`. Para treinar a rede deve-se criar uma matriz com todos os valores de entrada, e outra matriz com suas saídas correspondentes (MATHWORKS, 2014).

Todas as entradas devem ter o mesmo tamanho, cada valor da amostra será um neurônio da

camada de entrada, para esta implementação obteve-se melhores resultados com as configurações da rede neural exibidas na Figura 4.1.

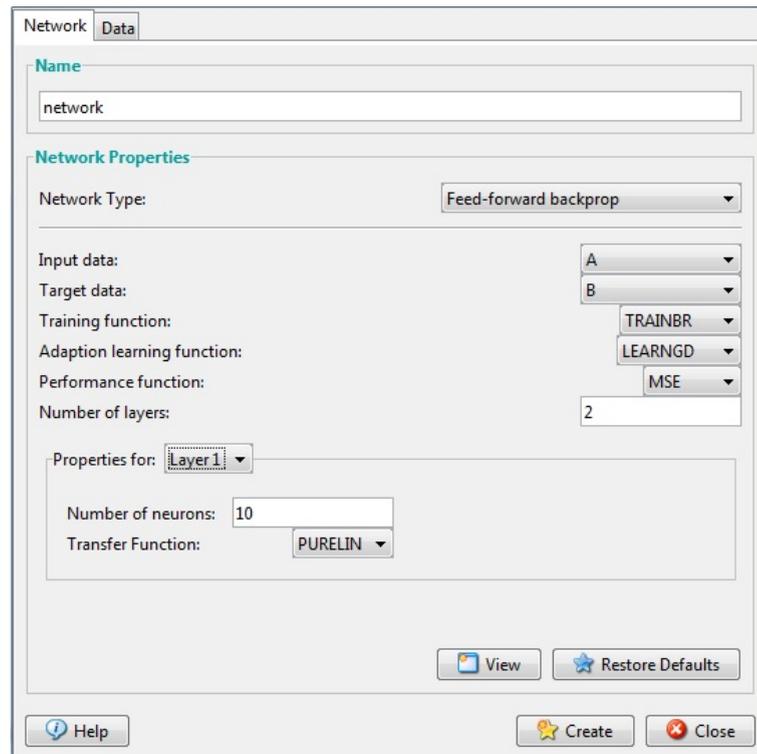


Figura 4.1: Configuração rede neural.

A rede utilizada para comparar os padrões foi a *Feed-forward backpropagation*, esta rede é formada por unidades de processamento interligadas, a qual chamamos de neurônios, tendo como tendência natural armazenar conhecimento experimental e torná-lo disponível para uso (HAYKIN, 2000). Dentre os principais tipos de redes *feed-forward* a MPL é a que se destaca. A função de treinamento escolhida dentre as testadas foi a `trainbr`, que determina a combinação correta de modo a produzir uma rede generalizada. A função de adaptação de aprendizagem escolhida foi a `learngd`, sendo que esta função faz o do gradiente descendente para buscar o vetor de pesos que melhor se ajuste ao conjunto de treinamento, a função de desempenho da rede escolhida foi a `mse`, mede o desempenho da rede de acordo com a média dos quadrados dos erros. O número de camadas escondidas foram duas, sendo ambas definidas com a função de ativação `purelin` (função de ativação linear)(MATHWORKS, 2014). A taxa de aprendizado da rede *Feed-forward backpropagation* foi de 0,005. Essas combinações resultaram nas

maiores taxas de acerto, outras combinações foram testadas, tanto com novas redes quanto com o restante das configurações, estas combinações que não foram satisfatórias estarão descritas no próximo capítulo.

Como resultado destas configurações obteve-se uma taxa de acerto de 86.7%. Estas informações podem ser visualizadas na Figura 4.2. Esta taxa de acerto é calculada em relação as amostras existentes, pois quando testado com amostras não treinadas, esta taxa cai consideravelmente. No próximo capítulo estão descritos os motivos por que desta queda de acertos.



Figura 4.2: Matriz de confusão.

4.3 Envio dos comandos

Após realizado o treinamento da rede, é necessário que se obtenha a saída com novas entradas, a solução encontrada para se obter esta saída foi utilizar a função `sim` (*Simulate dynamic*

system), sendo necessário como parâmetros de entrada a RNA treinada e a nova amostra, retornando então a saída descrita na tabela 4.1. Esta saída nem sempre será um número inteiro, sendo necessário então a realização de um arredondamento, a ferramenta MATLAB oferece a função `round` para esta tarefa (MATHWORKS, 2014). Após o arredondamento, compara-se então cada neurônio de saída e atribui-se um comando, descrito na Tabela 4.1. Após definido qual será o comando, deve se enviar a informação para o LEGO Mindstorms, optamos pelo envio via *bluetooth*[®]. Para realizar a comunicação deve-se realizar os seguintes passos (SILVA, 2013):

1. Instalar o MATLAB (MATHWORKS, 2014);
2. Instalar a ferramenta RWTH Toolbox for MATLAB (RWTH, 2014);
3. Instalar o drive de comunicação MINDSTORMS NXT Driver para utilização da porta USB (LEGO Mindstorms, 2014);
4. Transferir o arquivo `MotorControl.rxe` para o *smart brick*;
5. Conectar o LEGO com o computador via USB ou *Bluetooth*[®];
6. E, no MATLAB, inserir os códigos necessários para controle do protótipo.

4.3.1 *Bluetooth*[®]

Para conectar o NXT via *Bluetooth*[®] devemos seguir os seguintes passos (SILVA, 2013):

1. Ir a Painel de Controle; Hardware e sons; Dispositivos e Impressoras; Adicionar um dispositivo
2. Selecionar o NXT e entrar com o código "1234"; Depois que o dispositivo for adicionado, verificar em propriedades de hardware em qual porta de comunicação ele está conectado;
3. Acessar o MATLAB e digitar o código: `COM_MakeBTConfigFile` e clique em "yes" para confirmar;
4. Configurar a "SerialPort" em que o NXT está conectado e clicar em "Ok".

5. Depois acessar o MATLAB entrar com o seguinte código:

```
h = COM_OpenNXT('bluetooth.ini');
```

```
COM_SetDefaultNXT(h)
```

```
NXT_PlayTone(440, 100)
```

Comandos para o robô vão aqui.

```
COM_CloseNXT(h);
```

Após a realização destes passos, a conexão deve estar concluída.

4.3.2 Comandos de movimento

Após identificado o comando, e realizada a conexão com o robô, deve-se enviar as informações necessárias para que haja a movimentação. Podem ser configurados os valores de potência, direção e rotação dos motores. O primeiro passo é definir uma variável para herdar o vetor dos motores. Define-se a entrada do motor conectado no *smart brick* que se deseja controlar (NXT-Motor, 2014).

```
mBC = NXTMotor([MOTOR_B;MOTOR_C])
```

Define-se então a potência do motor, sendo que a potência varia entre -100 e 100%, o valor negativo corresponde ao giro no sentido anti-horário.

```
mBC.Power = 50
```

E para enviar as configurações e as executar, utiliza-se o seguinte comando:

```
mBC.SendToNXT()
```

Pode-se enviar comandos para todos os componentes do LEGO via MATLAB, como por exemplo para os sensores de toque, cor, ultrassônico, entre outros. Nos limitamos aos motores, mas pode-se ampliar a rede para novos comandos, dependendo da finalidade desejada.

Capítulo 5

Considerações finais

Como esperado a rede neural tem comportamentos diferentes quando submetida a entradas que não foram treinadas, tendo grandes variações com trocas de microfones, ruídos e sons externos, mudanças de equalização sonora, amplitude do som gravado. Então para que possa-se ter uma rede neural confiável, deve-se prever todas as variações sonoras possíveis. Para que a rede tenha bons resultados independente de locutor, várias amostras devem ser coletadas. Para o treinamento da rede foram coletadas 600 amostras, gravadas de 6 voluntários, separadas as gravações em 25 amostras por comando/pessoa. Nos casos em que há apenas um falante, não há a necessidade de se obter todas estas gravações, sendo necessário apenas treinar a rede para reconhecer os comandos de um único indivíduo.

5.1 Testes

Novas gravações foram realizadas, tanto com indivíduos que treinaram a rede, como com novos, esse novo conjunto de gravação foi utilizado para realização de testes. Os resultados obtidos estão descritos na Tabela 5.1, sendo exibido as taxas de acerto, foram coletadas 40 novas amostras de cada indivíduo, 10 por comando.

Como podemos notar, a rede aceitou comandos de novos indivíduos. Para que se tenha melhores taxas de acertos, pode-se treinar a rede com mais amostras, até chegar a resultados mais aceitáveis, ou também aumentar o número de épocas de treinamento. Obteve-se mais acertos com indivíduos que treinaram a rede. Com as entradas treinadas, não houve erro nos testes.

Tabela 5.1: Resultado dos testes.

Indivíduo	1	2	3	4	5	6
Ré	5	3	5	4	3	2
Frente	2	4	5	5	3	1
Direita	1	4	3	2	1	3
Esquerda	0	1	4	2	2	1
Taxa de acerto	20%	30%	42.5%	22.5%	22.5%	17.5%
Ambiente	Em silêncio	Em silêncio	Em silêncio	Em silêncio	Com ruído	Com ruído
Treinou a rede	Não	Sim	Sim	Sim	Não	Não

5.2 Redes treinadas

Para se chegar-se a configuração atual do reconhecedor neural foram realizados vários testes com arquiteturas diferentes de redes neurais, e algoritmos de treinamento. Descrevemos abaixo, as tentativas que não obtiveram sucesso.

Feed-forward backprop: Primeira rede testada. Inicialmente utilizou-se os parâmetros de configuração *default*: Como a rede apresentou um nível de acertos muito baixo, resolvemos abandoná-la e tentar outras arquiteturas. Estas configurações estão exibidas na Figura 5.1.

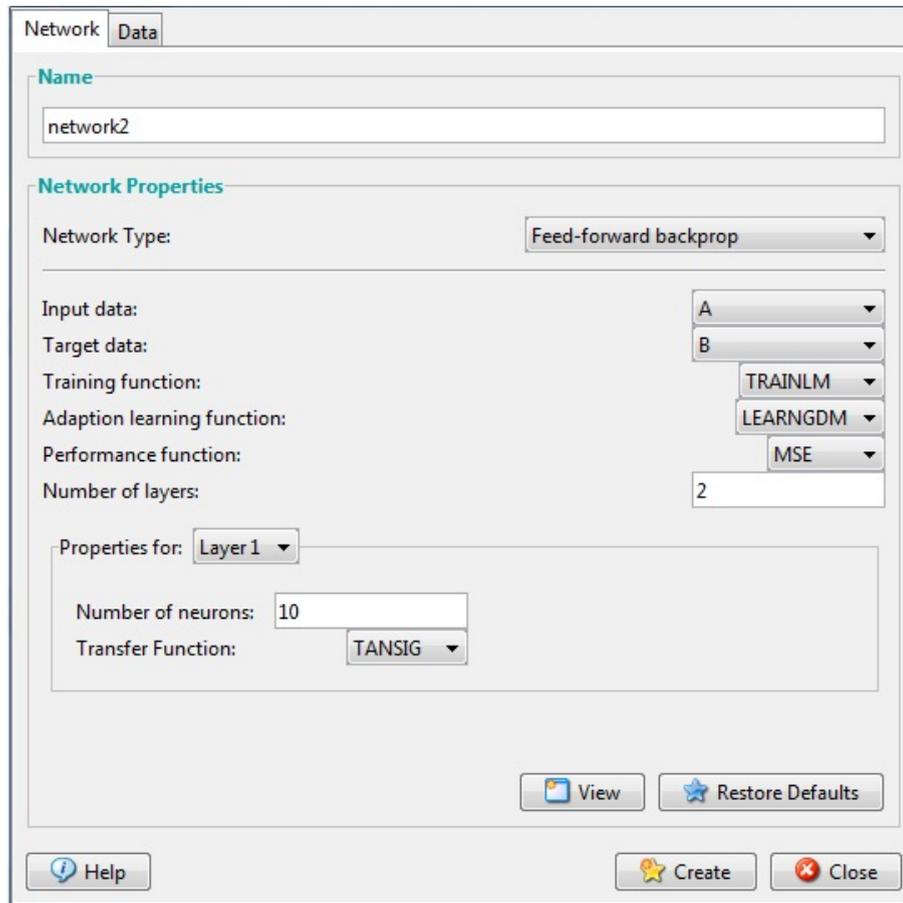


Figura 5.1: Configuração rede neural padrão nntool.

A Figura 5.2 mostra sua matriz de confusão, onde pode-se perceber a baixa taxa de acertos, sendo esta taxa de 34%.

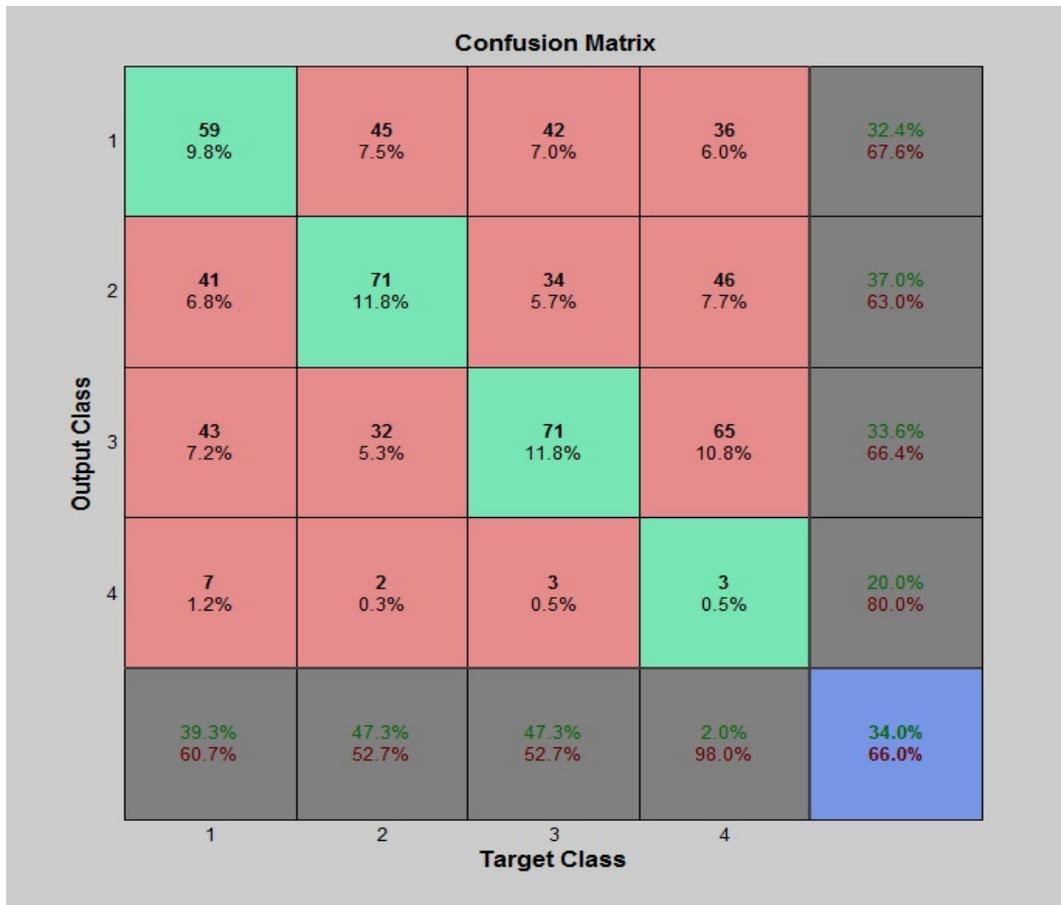


Figura 5.2: Matriz de confusão rede neural padrão nntool.

Radial basis function: Esta arquitetura apresentou um resultado incrível de 100% de acerto, mas a rede não tem capacidade de generalização, sendo então inútil para este trabalho. As configurações de redes utilizadas estão exibidas na Figura 5.3, e a Figura 5.4, mostra sua matriz de confusão.

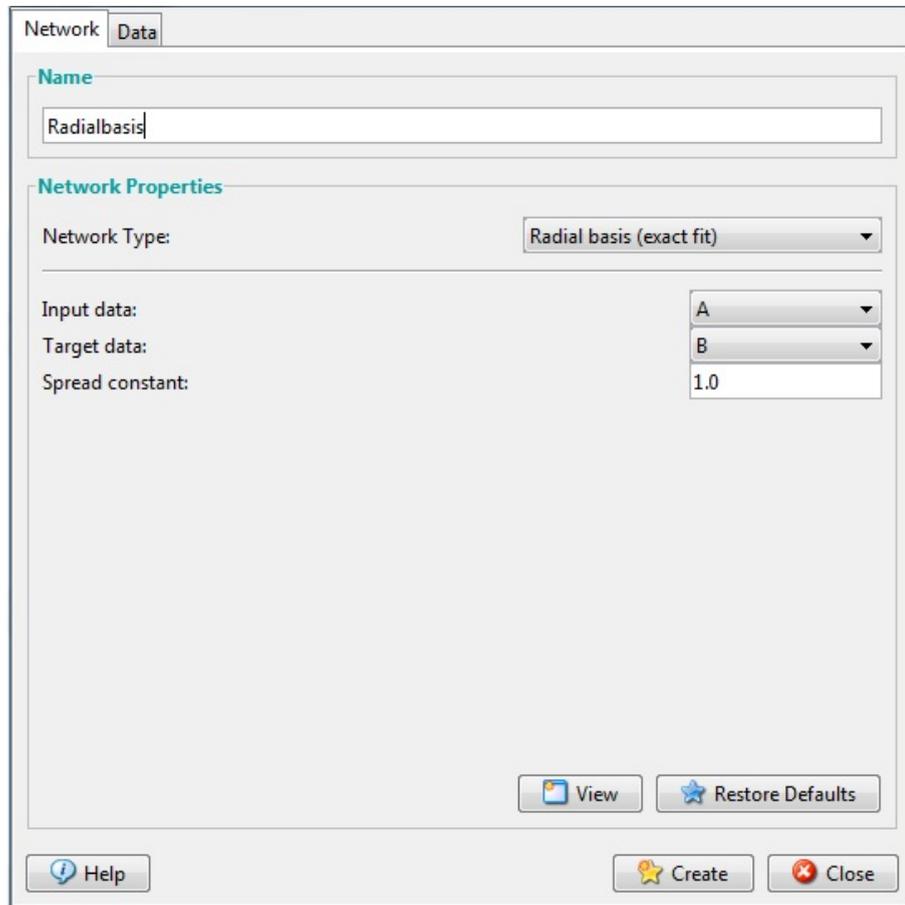


Figura 5.3: Configuração rede neural *radial basis (exact fit)*.

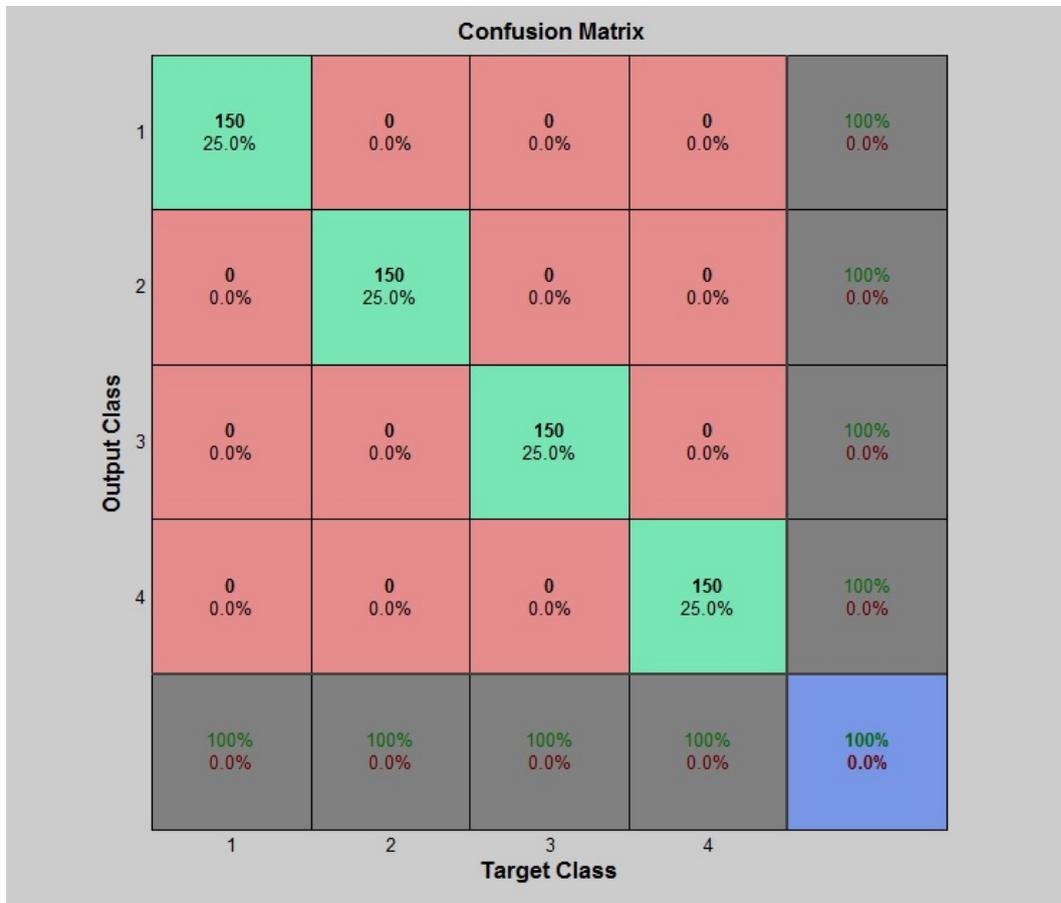


Figura 5.4: Matriz de confusão rede neural *radial basis*.

Feed-forward backprop: Chegando a níveis de acerto aceitáveis, esta rede tem diferenças apenas nas funções de ativação, sendo elas a LOGSIG e a TANSIG (ambas as funções de ativação são do tipo sigmóide), exibidas na Figura 5.5. A matriz de confusão é mostrada na Figura 5.6.

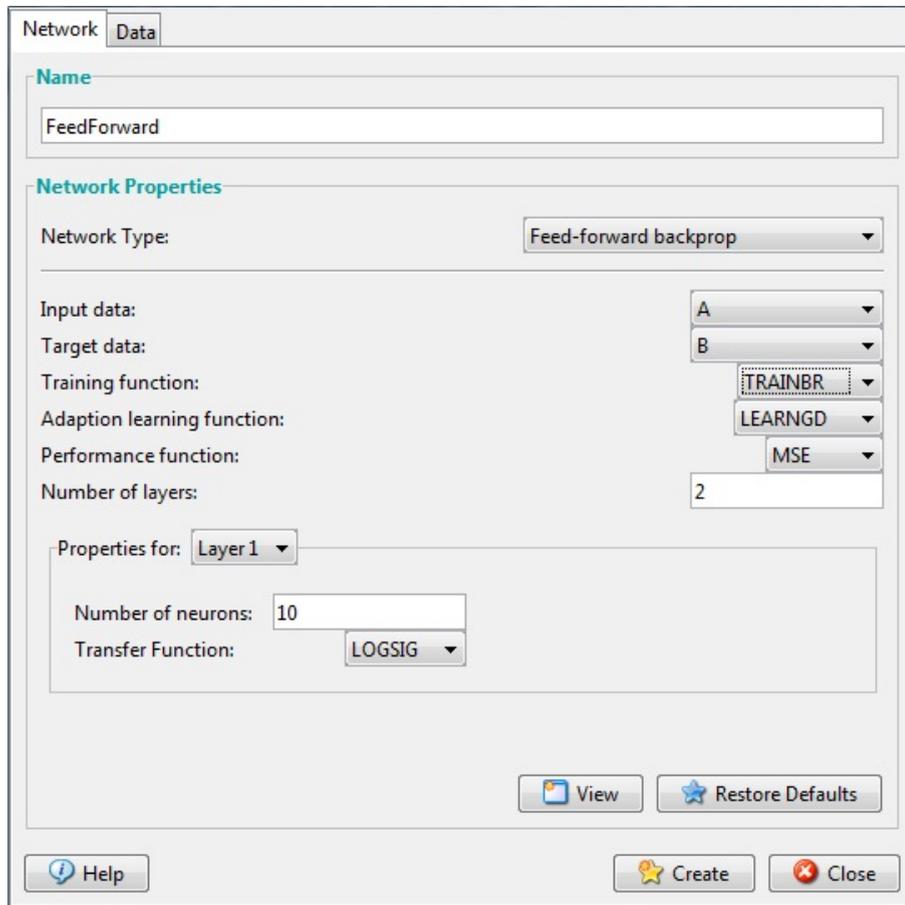


Figura 5.5: Configuração rede neural *feed-forward backpropagation* com função de ativação sigmóide.

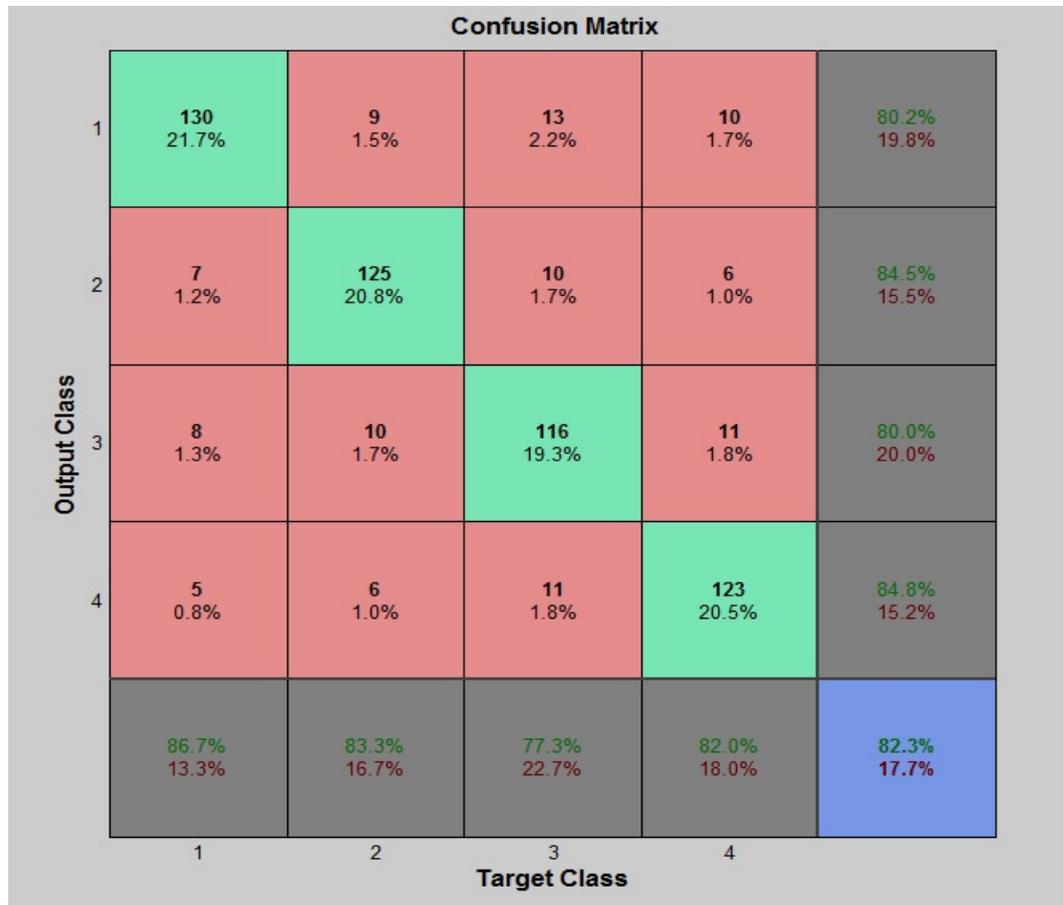


Figura 5.6: Matriz de confusão rede neural *feed-forward backpropagation* com função de ativação sigmóide.

Após estes testes chegamos a rede desejada, explicada no capítulo 4, a qual foi utilizada neste trabalho.

5.3 Trabalhos futuros

Como trabalhos futuros pode-se criar novas redes neurais, ou até mesmo ampliar a que já existe, inserir novos comandos, tanto para os motores, quanto para os sensores do robô LEGO Mindstorms, e realizar novas gravações analisando qual a influência destas gravações na rede neural, aumentando o tamanho da entrada, com pessoas com tonalidades de voz diferentes, com um maior número de gravações. Também é possível testar com outros robôs ou máquinas que possam se comunicar com um computador. Indo mais além pode-se implementar estas funcionalidades em dispositivos portáteis, aumentando assim sua mobilidade.

Referências Bibliográficas

ALENCAR, V. *Atributos e domínios de interpolação eficientes em reconhecimento de voz distribuído*. Tese (Doutorado) — Dissertação de mestrado. PUC-Rio, 2005.

AMARAL, M. do. *Identificador Neural de Comandos de Voz Embutido em um Robô Lego Mindstorms*. Dissertação (Mestrado) — Universidade Federal de Ouro Preto - UFOP, 2005.

BORGES, M. E. *Algoritmo Genético para Otimização de Arquiteturas de Redes MLP*. 2012. Acessado em 13 nov. 2014. Disponível em: <<http://www.borges-solutions.com/algoritmo-genetico-para-otimizacao-de-arquiteturas-de-redes-mlp/>>.

BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000.

BRAGA, P. de L. *Reconhecimento de voz dependente de locutor utilizando Redes Neurais Artificiais*. Dissertação (Mestrado) — Escola Politécnica de Pernambuco, 2006. Disponível em: <<http://tcc.ecomp.poli.br/20061/PetronioBraga.pdf>>.

BRESOLIN, A. de A. *Estudo do Reconhecimento de Voz para o Acionamento de Equipamentos Elétricos via Comandos em Português*. Dissertação (Mestrado) — Universidade de Santa Catarina - UDESC, 2003. Disponível em: <http://www.tede.udesc.br/tde/_busca/arquivo.php?codArquivo=2937>.

DECOM, J. da S. *Implementação de um Sistema Alternativo para Acionamento de Cadeira de Rodas Motorizada*. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina - UDESC, 2012. Disponível em: <<http://www.pergamum.udesc.br/dados-bu/000000/000000000015/000015DC.pdf>>.

HAYKIN, S. *Redes neurais artificiais: princípio e prática*. [S.l.: s.n.], 2000.

HUGO, M. *Uma interface de reconhecimento de voz para o sistema de gerenciamento de central de informação de fretes*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Centro Tecnológico., 1995.

KOHONEN, T. The 'neural' phonetic typewriter. *Computer*, IEEE, v. 21, n. 3, p. 11–22, 1988.

LEGO Mindstorms. 2014. Acessado em 13 out. 2014. Disponível em: <<http://www.lego.com/en-us/mindstorms>>.

- MARTINS, J. A. *Avaliação de Diferentes Técnicas para Reconhecimento de Fala*. Dissertação (Mestrado) — Universidade Estadual de Campinas, 1997. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?down=vtls000127015>>.
- MATHWORKS. *MATLAB Product Description*. 2014. Acessado em 13 out. 2014. Disponível em: <<http://www.mathworks.com/products/matlab/description1.html>>.
- MATHWORKS. *nntool*. 2014. Acessado em 13 out. 2014. Disponível em: <<http://www.mathworks.com/help/nnet/ref/nntool.html>>.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.
- NXTMotor. *Constructs an NXTMotor object*. 2014. Acessado em 13 out. 2014. Disponível em: <<http://www.mindstorms.rwth-aachen.de/documents/downloads/doc/version-4.03/help/NXTMotor.html>>.
- REIS, A. Pré-processamento de séries temporais para um previsor de carga elétrica a curto prazo via redes neurais artificiais. In: *XIII Congresso Brasileiro de Automática*. [S.l.: s.n.], 1999. p. 2036–2041.
- RICH, E.; KNIGHT, K. *Inteligência artificial*. [S.l.: s.n.], 1994.
- RODRIGUES, F. F. *Acionamento de um Robô Lego Mindstorms por Comandos Vocais Utilizando Redes Neurais Artificiais*. Dissertação (Mestrado) — Universidade Federal de Ouro Preto Escola de Minas, 2009. Disponível em: <<http://www.em.ufop.br/cecau/monografias/2009/FABIO\FONSECA.pdf>>.
- RWTH. *Mindstorms NXT Toolbox*. 2014. Acessado em 13 out. 2014. Disponível em: <<http://www.mindstorms.rwth-aachen.de>>.
- SILVA, A. B. D. *Programação de Lego MINDSTORMS NXT 2.0 utilizando matlab*. Dissertação (Mestrado) — UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO, 2013. Disponível em: <<http://pt.slideshare.net/Ciidiinhaa/programao-de-lego-mindstorms-nxt-20-utilizando-matlab>>.
- SILVA, A. G. da. *Reconhecimento de Voz para Palavras Isoladas*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009. Disponível em: <<http://www.cin.ufpe.br/~tg/2009-2/ags.pdf>>.
- SILVA, I. d.; SPATTI, D. H.; FLAUZINO, R. A. *Redes neurais artificiais para engenharia e ciências aplicadas*. [S.l.: s.n.], 2010. 33–111 p.
- VALIATI, J. F. *Reconhecimento de Voz para Comandos de Direcionamento por meio de Redes Neurais*. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, 2000.