

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Bruno Rafael Rodrigues Dias

Sistema Inteligente para Reconhecimento de Timbres

Picos
2013

Bruno Rafael Rodrigues Dias

Sistema Inteligente para Reconhecimento de Timbres

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte para obtenção do Grau de Bacharel em Sistemas de Informação. Área de concentração: Sistemas de Informação. Sob orientação do Professor MSc. Algeir Sampaio

Picos
2013

Bruno Rafael Rodrigues Dias

Sistema Inteligente para Reconhecimento de Timbres

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros da Universidade Federal do Piauí como parte para obtenção do Grau de Bacharel em Sistemas de Informação. Área de concentração: Sistemas de Informação. Sob orientação do Professor MSc. Algeir Sampaio

Data de Aprovação:

Professor Mestre Algeir Prazeres Sampaio_____ UFPI
Professor Especialista Ismael de Holanda Leal_____ UFPI
Professor Especialista Ivenilton Alexandre de Souza Moura_____ UFPI

Picos
2013

Eu, **Bruno Rafael Rodrigues Dias**, abaixo identificado(a) como autor(a), autorizo a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação abaixo discriminada, de minha autoria, em seu site, em formato PDF, para fins de leitura e/ou impressão, a partir da data de hoje.

Picos-PI, 18 de Abril de 2013.

FICHA CATALOGRÁFICA

Serviço de Processamento Técnico da Universidade Federal do Piauí Biblioteca José Albano de Macêdo

Dias, Bruno Rafael Rodrigues.

D541si Sistema inteligente para reconhecimento de timbres / Bruno Rafael Rodrigues Dias. - 2013.
CD-ROM : il. ; 4^{3/4} pol. (76 p.)

Monografia(Bacharelado em Sistemas de Informação) - Universidade Federal do Piauí.
Picos-PI, 2013.

Orientador: Prof. MSc. Algeir Prazeres Sampaio.

1 - Redes Neurais Artificiais. 2 - Inteligência Artificial.

3 - Reconhecimento de Instrumentos Musicais. I. Título.

CDD 005.1

Dedico esta monografia a toda minha família em especial a minha primi-nha Lyvia Beatryz, *in memoriam* e minha avó Teresa Rodrigues, *in memoriam* que me deram muito apoio nos momentos mais difíceis da minha vida, aos meus professores que me ensinaram que por mais que achamos que o nosso conhecimento já está bem profundo, estamos enganados pois o conhecimento é algo que está sempre se renovando. Obrigado por tudo!

“A imaginação é mais importante que o conhecimento.”

Albert Einstein

Agradeço primeiramente a Deus, pelo dom da vida, pela minha capacidade de pensar, ter me proporcionado, mas uma vitória e por me guiar nessa jornada de conquista.

Amar alguém profundamente lhe dá força. Ser amado profundamente lhe dá coragem. E toda força e coragem que obtive para ultrapassar as dificuldades foi minha família que me ofereceu: meus pais Hedilberto Rodrigues Dias e Maria Gissali de Sousa Dias pelo amor incondicional e compreensão e ao meu irmão Lucas Emanuel Rodrigues Dias pela cumplicidade.

Agradeço aos familiares e amigos pelo carinho, em especial aos meus Tios William Alves Nunes e Alessandra Rodrigues Celestino, Josevam Clementino de Moura Santos e Joana Darc Rodrigues Dias que me acolheu como filho.

Ao meu primo Paulo Egídio Ribeiro da Silva que esteve comigo nessa caminhada. A Paula Michele pelas palavras de motivação e perseverança.

Ao meu orientado Professor Msc. Algeir Sampaio pela orientação ao longo desta dissertação, pela prestividade, enorme paciência, pelo incentivo e confiança e, principalmente, pela amizade.

Aos demais professores do Curso de Sistemas de Informação, pelo apoio e pelos conhecimentos transmitidos.

Aos colegas de curso, pelo companheirismo. Especialmente ao Francisco Barroso por gravar e disponibilizar as notas musicais utilizadas neste trabalho.

Sou eternamente grato a DEUS por ter colocado vocês na minha vida.

“Seu trabalho vai ocupar uma grande parte da sua vida, e a única maneira de estar verdadeiramente satisfeito é fazendo aquilo que você acredita ser um ótimo trabalho. E a única maneira de fazer um ótimo trabalho é fazendo o que você ama fazer. Se você ainda não encontrou, continue procurando.”

Steve Jobs

Resumo

A Inteligência Artificial desenvolve-se no sentido de modelar o cérebro através da criação de Redes Neurais Artificiais (RNA), que se caracterizam por possuir as mesmas propriedades cognitivas e associativas do cérebro humano. É percebido na literatura que inúmeras redes neurais foram desenvolvidas para resolver problemas das mais diversas naturezas. O objetivo deste projeto é desenvolver um sistema inteligente capaz de reconhecer timbres de instrumentos musicais. Inicialmente, este projeto busca apresentar todo um conteúdo explicativo sobre o desenvolvimento de aplicações que utilizam Redes Neurais Artificiais (RNA) para classificação de diferentes tipos de instrumentos musicais. Terá destaque as suas potencialidades, especificando seus pontos positivos e negativos, reforçar a sua importância no âmbito de desenvolvimento de aplicações que utilizam redes neurais artificiais (RNA) na classificação de classes. Foi desenvolvido um sistema de reconhecimento de instrumentos musicais utilizando redes neurais, através do algoritmo *Backpropagation*, para o reconhecimento, usando as notas musicais Dó, Ré, Mi, Fá, Sol, Lá e Si, de cinco instrumentos musicais: Violão, Piano, Flauta e Trompete.

Palavras-chave: Redes Neurais Artificiais, Inteligência Artificial, Reconhecimento de Instrumentos Musicais.

Abstract

The Artificial Intelligence is developed in order to model the brain by creating artificial neural networks (ANN), which are characterized by possessing the same properties and cognitive associations of human brain. It is noticed in the literature that many neural networks have been developed to solve problems of diverse natures. The objective of this project is to develop an intelligent system capable of recognizing musical instrument tones. Initially, this project seeks to present an entire explanatory content on the development of applications using artificial neural networks (ANN) for classifying different types of musical instruments. Will highlight its potential, specifying their strengths and weaknesses, strengthen its importance in the development of applications using artificial neural networks (ANN) to classify classes. We developed a system for recognition of musical instruments using neural networks through the Backpropagation algorithm, for recognition, using musical notes C, D, E, F, G, A and B, five musical instruments: Guitar, Piano, Flute and Trumpet.

Keywords: Artificial neural networks, Artificial Intelligence, Recognition Musical Instruments.

Lista de Figuras

Figura 1 -	Onda Sonora de Maior Frequência	23
Figura 2 -	Onda Sonora de Menor Frequência	23
Figura 3 -	Operacao “Butterfly”	29
Figura 4 -	Neuronio Formal	36
Figura 5 -	Rede de Perceptrons	37
Figura 6 -	Rede Adaline e Madaline	38
Figura 7 -	Estrutura do Método Back-Propagation	39
Figura 8 -	Neurônio Biológico	40
Figura 9 -	Modelo de um Neurônio Artificial	40
Figura 10 -	Diagrama de Funcionamento de um Neurônio Artificial	41
Figura 11 -	Funções de Ativação de uma RNA	42
Figura 12 -	Modelo de Rede Neural com uma Camada	43
Figura 13 -	Modelo de Rede Neural com Duas Camadas	43
Figura 14 -	Modelo de Rede Neural com Três Camadas	44
Figura 15 -	Modelo de Rede Neural Recorrente	45
Figura 16 -	Arquitetura da Rede Neural	55
Figura 17 -	Reconhecedor de Timbres	56

Lista de Tabelas

Tabela 1 - Instrumento Trompete	57
Tabela 2 - Instrumento Violão	57
Tabela 3 - Instrumento Flauta	57
Tabela 4 - Instrumento Piano	58

Lista de abreviaturas e siglas

ADALINE	Adaptive Linear Network
CPU	Central Processing Unit
DFT	Transformada Discreta de Fourier
FFT	Transformada Rápida de Fourier
MADALINE	Many Adaline
MatLab	Matrix Laboratory
MPL	Multi-Layer Perceptron
RNA	Redes Neurais Artificiais

Lista de símbolos

Hz	Hertz
Log	logaritmo
α	Alfa
φ	Função de Ativação da Rede Neural
∞	Infinito

Sumário

1	Introdução	16
1.1	Motivação	16
1.2	Problema	16
1.3	Abordagem	17
1.4	Objetivos	18
1.4.1	Objetivos Gerais	18
1.4.2	Objetivos Específicos	18
1.5	Trabalhos Relacionados	18
1.5.1	RNAs e Reconhecimento de Padrões	18
1.5.2	RNAs e Reconhecimento de Voz	18
1.5.3	RNAs e Reconhecimento de Caracteres	19
1.5.4	RNAs e Previsão em Engenharia Química	19
1.5.5	RNAs e Controle de Processos	19
1.5.6	RNAs e Medicina	19
1.5.7	RNAs e Games	20
1.5.8	RNAs e Previsão de Séries Temporais	20
1.6	Organização do Texto	21
2	Fundamentação Teórica	22
2.1	Som	22
2.1.1	A Física e a Percepção do Som	22
2.1.2	Principais Características do Som	22

2.1.3	Conclusões	24
2.2	Transformada Rápida de Fourier	24
2.2.1	As Séries de Fourier	25
2.2.2	A Transformada Contínua de Fourier	26
2.2.3	A Transformada Discreta de Fourier	27
2.2.4	A Transformada Rápida de Fourier – FFT	27
2.2.5	Conclusões	29
2.3	Sistema Nervoso	30
2.3.1	O Neurônio	30
2.3.2	Corpo Celular	31
2.3.3	Dendritos	31
2.3.4	Axônio	31
2.3.5	Sinapses	32
2.3.6	Conclusão	32
2.4	Redes Neurais Artificiais	32
2.4.1	Histórico	36
2.4.2	O Neurônio Artificial	39
2.4.3	Função de Ativação de uma Rede Neural Artificial	41
2.4.4	Arquitetura de uma Rede Neural Artificial	42
2.4.5	Aprendizado	45
3	Concepção do Projeto	51
4	Implementação	53
4.1	Introdução	53
4.2	O MatLab	53
4.3	A Gravação das Notas Musicais	53
4.4	A Normalização das Notas Musicais	54

4.5	O Desenvolvimento da Rede Neural	54
4.5.1	Arquitetura da Rede Neural	55
4.6	O Programa Reconhecedor de Timbres	56
5	Testes Realizados	57
5.1	Introdução	57
5.2	Conclusão	58
6	Conclusão	59
6.1	Trabalhos Futuros	59
	Referências	61
	Anexo A - Algoritmo da Criação da Rede Neural	65
	Anexo B - Algoritmo de Treinamento da Rede Neural	71
	Anexo C - Algoritmo do Programa Reconhecedor de Timbres	72

1 Introdução

1.1 Motivação

O final da década de 80 marcou o ressurgimento da área de Redes Neurais Artificiais (RNA). Também conhecida como conexionismo ou sistemas de processamento paralelo e distribuído. Esta forma de computação não-algorítmica é caracterizada por sistemas que, em algum nível, relembram a estrutura do cérebro humano. Por não ser baseada em regras ou programas, a computação neural se constitui em uma alternativa à computação algorítmica convencional.

As redes neurais artificiais se fundamentam nos estudos sobre a estrutura do cérebro humano para tentar emular sua forma inteligente de processar informações. Apesar de se desconhecer a maneira pela qual o cérebro manipula informações complexas, sabe-se que a modelagem do conhecimento contido em um problema específico pode ser representada através de inter-conexões entre células nervosas.

A Inteligência Artificial desenvolve-se no sentido de modelar o cérebro através da criação de redes neurais artificiais (RNA), que se caracterizam por possuir as mesmas propriedades cognitivas e associativas do cérebro humano. É percebido na literatura que inúmeras redes neurais foram desenvolvidas para resolver problemas das mais diversas naturezas. Este trabalho, em questão, procura aplicar os conceitos de Redes Neurais Artificiais no reconhecimento de instrumentos musicais.

A solução de problema através de Redes Neurais Artificiais (RNA) é bastante atrativa, já que a forma como estes são representados internamente pela rede e o paralelismo natural inerente à arquitetura das Redes Neurais Artificiais (RNA) criam a possibilidade de um desempenho superior ao dos modelos convencionais. Em Redes Neurais Artificiais (RNA), o procedimento usual na solução de problema passa inicialmente por uma fase de aprendizagem, em que um conjunto de exemplos é representado para a rede, a qual extrai automaticamente as características necessárias para representar a informação fornecida.

1.2 Problema

As redes neurais são utilizadas em varias áreas para resoluções de problemas que a computação comum não resolveria de uma forma com que se possa chegar à obtenção de um objetivo no qual seria a inteligência paralela a do ser humano. Especificamente na área musical o uso de redes neurais vem crescendo, utilizando-se dessa área para aprimorar a evolução das

redes neurais, com objetivo de possibilitar aos computadores a capacidade de analisar e processar sinais, de forma similar a do cérebro humano ao mesmo tempo buscando soluções para vários aspectos da área musical. Como a utilização de computadores capazes de identificar instrumentos musicais a partir da captação dos sons por eles emitidos. Por meios de vários estudos soluções para essa problemática estão emergindo, com a evolução das redes neurais e a utilização de algoritmos bem mais precisos, assim então vários problemas estão obtendo soluções ou caminhando para uma possível solução.

O rápido avanço nas pesquisas e o desenvolvimento dos computadores levaram diversos pesquisadores, na área de inteligência artificial, a acreditar que a construção de computadores pensantes fosse uma tarefa relativamente simples. Obviamente, este era um pensamento prematuro, equivocado justamente por estar pautado apenas na euforia causada pela rápida evolução digital. Para os pioneiros nesta área, o cérebro era comumente separado da mente. Segundo esta visão, o cérebro se constituía apenas como o meio físico de atuação da mente, a qual deveria ser a verdadeira responsável pelo pensamento. Assim, a analogia entre cérebro/mente e *hardware/software* se fazia imediata. A mente poderia ser vista como um *software* instalado em um *hardware* chamado cérebro.

Considerava-se necessário, portanto, desenvolver softwares que representassem o comportamento da mente, enquanto que a modelagem do cérebro era visualizada como secundária neste processo. Acreditava-se que, assim como foi possível à construção de máquinas que voam sem, no entanto, bater as asas, seria também possível criar uma máquina de pensar sem a necessidade de criação de um cérebro artificial. Contudo, os fracassos sucessivos desta abordagem acabaram por mostrar a importância da modelagem do cérebro na criação de uma inteligência artificial. A partir de então, a área da inteligência artificial começou a desenvolver-se no sentido de modelar o cérebro através da criação de redes neurais artificiais (RNA), que se caracterizam por possuir as mesmas propriedades cognitivas e associativas do cérebro humano.

1.3 Abordagem

Neste trabalho realizamos uma investigação sobre as técnicas de Inteligência Artificial especificamente voltadas para o campo das redes neurais artificiais (RNA) utilizadas atualmente em várias áreas da ciência e tecnologia: análise e processamento de sinais; controle de processos; robótica; classificação de dados; reconhecimento de padrões em linhas de montagem ; filtros contra ruídos eletrônicos; análise de imagens; análise de voz; avaliação de crédito; análise de aroma e odor; diagnóstico de doenças em pacientes. Descrevendo suas arquiteturas e funcionalidades. Usando o software MatLab para desenvolvimento de aplicações utilizando as técnicas de redes neurais artificiais para o reconhecimento de padrões para sistemas operacionais como Windows, do qual esse trabalho busca realizar.

1.4 Objetivos

1.4.1 Objetivos Gerais

Este Projeto propõe o desenvolvimento de um sistema inteligente capaz de reconhecer timbres de instrumentos musicais, utilizando a tecnologia das redes neurais artificiais (RNAs).

1.4.2 Objetivos Específicos

- Realizar uma pequena abordagem estudo sobre o sistema nervoso humano;
- Processar sinais de instrumentos musicais;
- Desenvolver softwares para reconhecimento de padrões

1.5 Trabalhos Relacionados

1.5.1 RNAs e Reconhecimento de Padrões

Atualmente, as RNAs têm sido largamente utilizadas na tarefa de reconhecer padrões presentes em conjuntos de dados. O objetivo desse processo é apresentar à rede um conjunto de dados conhecidos de modo que, através de um processo de aprendizado, ela determine subconjuntos de dados com características semelhantes (padrões), e seja capaz de reconhecer tais características em um novo elemento qualquer, que a ela seja apresentado, de acordo com um dos padrões previamente fixados.

Uma referência clássica sobre o assunto é (BISHOP, 1995). Diversas técnicas e aplicações podem ser encontradas e analisadas. Para complementar a leitura básica, pode-se citar (RIPLEY; HJORT, 1995).

1.5.2 RNAs e Reconhecimento de Voz

Muitos podem ser os padrões assumidos, dependendo da aplicação. Um exemplo de padrão estudado é a voz e referências sobre o assunto podem ser encontradas em (WAIBEL; LEE, 1990). Neste livro, uma coletânea de artigos foi agrupada e disposta com várias aplicações do assunto. Além disso, uma revisão foi apresentada em (LIPPMANN, 1989). Tal revisão contém referências básicas e gerais sobre o assunto em questão.

Uma das metodologias clássicas utilizadas no tratamento de problemas de reconhecimento de voz são os modelos de Markov. Assim, algumas referências (BOURLAND, 1990) (MORGAN; BOURLARD, 1990) (RENALS et al., 1992) mostram a melhora dessa ferramenta

estatística através da aplicação de RNAs, ou ainda, uma melhor performance das redes sobre os métodos estatísticos clássicos.

1.5.3 RNAs e Reconhecimento de Caracteres

Em (GUYON, 1991) encontram-se informações gerais sobre aplicações de RNAs para o reconhecimento de caracteres.

Demais referências (JACKEL et al., 1990) (SIMARD et al., 1992) (LECUN et al., 1990) oferecem tentativas de melhora em problemas como reconhecimento de caracteres de dígitos manuscritos, de caracteres óticos e outros.

1.5.4 RNAs e Previsão em Engenharia Química

Como em qualquer área, a previsão é empregada em larga escala. No caso da engenharia química, as RNAs são aplicadas na previsão de diversas atividades químicas.

As mais estudadas incluem previsão de reatividade química (SIMON et al., 1993), de degradação de produtos (KOSTKA et al., 2001), de estrutura molecular (HEMMER; GASTEIGER, 2000), de reações diversas (SOUSA; GASTEIGER, 2001) (KOSTKA et al., 2001) (SOUSA et al., 2002) e previsão em aplicações de cromatografia (SOUSA; GASTEIGER, 2002).

1.5.5 RNAs e Controle de Processos

RNAs têm sido utilizadas com sucesso em muitas aplicações de controle de processos químicos como controladores adaptativos, uma vez que esta pode monitorar e controlar processos químicos complexos com a supervisão de um operador humano.

Assim, uma RNA pode ser utilizada para monitorar o processo, tomar decisões sobre a operação do sistema e ajustar controles apropriados para manter o processo operando com uma eficiência ótima. Além disso, tem a vantagem de poder ser atualizada com novas informações através de uma aprendizagem dinâmica.

Alguns exemplos de aplicações em diagnósticos de falta e temperatura podem ser encontrados, respectivamente, em (HOSKINS; HIMMELBLAU, 1991) (UNGER et al., 1990) e (KHALID; OMATU, 1992).

1.5.6 RNAs e Medicina

Pesquisadores da área médica têm utilizado redes neurais tipo *Perceptron* com múltiplas camadas utilizando o algoritmo de aprendizado *backpropagation* para Diagnóstico de Lesões Intersticiais pulmonares (AMBROSIO et al., 2000), Diagnóstico Diferencial de Anemias Carenciais (SANTOS et al., 2000), Classificação de Nódulos em Mamogramas digitalizados

(KINOSHITA et al., 2000), *Diagnosing the Cardiovascular System* (KELLER et al., 1995), *Diagnosing Coronary Artery Disease* (TURNER, 1994).

1.5.7 RNAs e Games

As RNAs são amplamente utilizadas para a criação de jogos inteligentes e estratégias de aprendizado. Alguns pesquisadores utilizam uma população de RNAs e vão evoluindo até encontrar a RNA mais adequada e adaptada ao jogo.

(CHELLAPILLA; FOGEL, 1999a) usaram uma RNA *feedforward* para avaliar posições de tábua alternativas usando uma estratégia de procura de minimax para jogos de xadrez. Após a evolução das RNAs o programa pôde derrotar dois jogadores de nível-perito, de acordo com a federação de Xadrez Norte Americana (CHELLAPILLA; FOGEL, 2000).

(CHELLAPILLA; FOGEL, 1999b) usaram um algoritmo híbrido para evoluir RNAs com o objetivo de encontrar a melhor rede para aprender as regras do jogo (jogos de soma zero).

1.5.8 RNAs e Previsão de Séries Temporais

A utilização de redes neurais para a previsão de séries temporais vem sendo proposta por vários autores, (LAPEDES; FARBER, 1987), (WEIGEND et al., 1990), (CONTRELL et al., 1995), (LACHTERMARCHER; FULLER, 1995), mostrando a viabilidade de utilizar estes modelos.

É notável que as redes neurais e técnicas de series temporais tradicionais vem sendo comparadas há algum tempo (SHARDA; PATIL, 1990), (TANG et al., 1991). Grande parte dos estudos comparativos realizados atualmente utilizam o algoritmo *back-propagation*, ou algumas de suas extensões. Tecnicamente este modelo já foi utilizado com grande sucesso em diversas áreas de conhecimento, dentre elas sistemas de potência (SRINIVASAN et al., 1991), (HWANG; MONN, 1991), (EL-SHARKAWI et al., 1991), (PENG et al., 1992), previsões econômicas (HOPTROFF et al., 1991) e análise de séries temporais (WEIGEND et al., 1991a), (WEIGEND et al., 1991b).

(LAPEDES; FARBER, 1987) aplicam redes neurais multicamadas (Multi-Layer Perceptron-MPL) para a previsão de duas espécies de séries temporais caóticas geradas por um processo determinístico não-linear. Os resultados indicaram que os modelos MLP podem “aprender”, simulando sistemas não-lineares, com excelentes características quando comparadas a métodos tradicionais.

1.6 Organização do Texto

Esta monografia está dividida em 6 capítulos.

No capítulo 1 é definida a área do projeto, o problema abordado, as possíveis contribuições do projeto e a organização da monografia.

O Capítulo 2 é dedicado ao referencial teórico. Será abordado na seção 2.1 estudo sobre o som apresentando seus principais pontos, pontos esses que foram de extrema necessidade para o desenvolvimento deste projeto. Na seção 2.2 será apresentado Transformada Rápida de Fourier (FFT), uma função matemática utilizada para realizar transferência de abordagem do sinal em função do tempo para sinal em função da frequência. A seção 2.3 descreve o Sistema Nervoso Humano, que visa um breve estudo, para que de tal modo possa obter um maior entendimento sobre o projeto realizado. A seção 2.4 é dedicado a uma visão geral das Redes Neurais Artificiais, onde será revisado os seus principais aspectos.

O Capítulo 3 será abordado a concepção deste trabalho falando um pouco sobre como emergiu a ideia de se trabalhar com redes neurais artificiais para o desenvolvimento deste projeto.

O Capítulo 4 descreve os passos realizados para o desenvolvimento da aplicação como a definição da arquitetura da rede neural desenvolvida e ilustração da aplicação.

O Capítulo 5 apresenta os resultados dos testes realizados com aplicação desenvolvida neste projeto.

O Capítulo 6 é dedicado a conclusão deste projeto abordando em modo geral os pontos positivos e negativos durante o desenvolvimento e seus objetivos alcançados com uma pequena parte dedicada aos trabalhos futuros que serão desenvolvidos nesta mesma linha de pesquisa.

2 Fundamentação Teórica

2.1 Som

Neste Capítulo será abordado estudo em relação à parte acústica do trabalho, onde descrevem os principais pontos de estudo sobre o som, pontos estes que são de grande relevância para o desenvolvimento deste trabalho.

A Seção 2.1.1 define a natureza física do som e como se pode descrevê-lo. Na Seção 2.1.2 revisam-se as características principais que compõem o som, características essas que ao estudá-las tem-se um maior entendimento sobre a composição do som. Na Seção 2.1.3 concluirá a abordagem teórica sobre o som.

2.1.1 A Física e a Percepção do Som

O som é uma onda mecânica que produz zonas de compressão e de descompressão no meio em que se propaga. Para este fenômeno ocorrer há a necessidade de três elementos relacionados em um sistema:

Emissor - Meio - Receptor

O emissor tem a função de produzir um distúrbio no meio, que será percebido pelo receptor. É importante notar que o meio tem influência na qualidade do distúrbio, pois afeta a maneira como este se propaga. Estes distúrbios de natureza mecânicos são pequenas e rápidas variações de pressão do meio, causadas pelo movimento das moléculas, caracterizados por compressões e rarefações (descompressões, expansões).

A audição é um sentido que o ser humano utiliza para perceber o mundo que o cerca. Indivíduos com boa audição podem, por meio dela, identificar vozes conhecidas, tipos de objetos que caem no chão, canto de pássaros, instrumentos musicais, ruídos de carros, enfim captar inúmeras informações do ambiente à sua volta.

2.1.2 Principais Características do Som

Frequência

O número de vezes que as oscilações ocorrem, quantos ciclos acontecem por unidade de tempo, é chamado de frequência e a sua percepção é conhecida por tom. Para descrever a frequência é utilizada a unidade *Hertz* (Hz), onde 1Hz corresponde a um ciclo de vibração por segundo.

Altura

A altura está relacionada com a frequência da onda sonora e recebe a qualidade de som grave ou agudo. A voz de um ser humano homem é considerada grave e a de uma mulher, aguda.

- Quanto maior a frequência, mais agudo é o som.

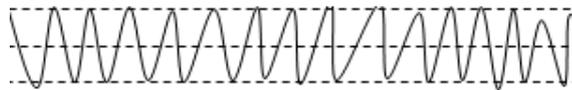


Figura 1 – Onda Sonora de Maior Frequência

- Quanto menor a frequência, mais grave é o som.

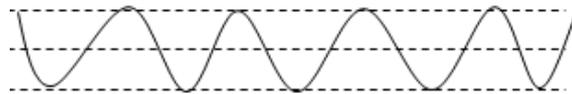


Figura 2 – Onda Sonora de Menor Frequência

As notas de uma escala musical estão classificadas desde a mais grave até a mais aguda.

Intensidade

A intensidade de uma onda sonora pode ser mais bem entendida se usarmos como exemplo a explosão dos fogos de artifício. Normalmente, quando estamos próximos do local da explosão, dizemos que o som é forte e, quando estamos afastados, dizemos que o som é fraco.

Harmônico e a Série Harmônica

Um som musical produzido por qualquer instrumento não é constituído apenas por uma nota, ou seja, por apenas uma frequência, e sim por uma série de outras notas resultantes de vibrações secundárias. A estas outras frequências, deu-se o nome de harmônicos.

O conjunto de harmônicos que acompanha uma nota fundamental forma a Série Harmônica. Toda nota gera uma série harmônica identicamente proporcional. Ao tocar a nota dó de um violão, por exemplo, a corda vibrará inteira e vibrará, também, de forma a dividir a corda pela metade, logo depois, vibrará em três partes iguais, quatro partes iguais e assim por diante, tudo simultaneamente.

Timbre

Cada tipo de instrumento musical tem uma espécie de assinatura, um conjunto de características sonoras associadas a ele. Embora possam parecer subjetivas, elas têm uma descrição matemática extremamente precisa que torna cada tipo de instrumento único. A mesma frequência emitida por um violão soa diferente quando produzida por um piano. Isto acontece porque, embora a frequência fundamental seja a mesma em ambos os instrumentos, o som de instrumentos musicais não é perfeitamente periódico.

Portanto, as amplitudes e o tempo de duração de cada um dos harmônicos presentes no som resultante podem variar independentemente. A combinação dessas duas propriedades tem o nome de timbre.

2.1.3 Conclusões

O objetivo deste tópico era agrupar os principais conceitos a respeito da parte acústica que será aplicada neste projeto. Desta forma, visava-se a oferecer o conhecimento mínimo necessário para a compreensão do sistema. Portanto, foi apresentado o embasamento teórico necessário para entender as principais características do som, tais como, frequência, altura, intensidade, harmônicos, serie harmônica e timbre.

2.2 Transformada Rápida de Fourier

A quase totalidade dos fenômenos físicos de natureza periódica tem sua origem em ondas (sonoras, elétricas, hidráulicas e mecânicas) que apresentam formas bem definidas.

Em meados do século XVII, Euler, Bernoulli e Lagrange já discutiam a representação de sinais periódicos através de séries trigonométricas.

Fourier (1768-1830) desenvolveu em seu estudo sobre a condução de calor pelos corpos a formulação de que um fenômeno de natureza periódica pode ser decomposto em uma série de funções senoidais, com amplitudes variáveis e frequências múltiplas da frequência fundamental, com fases coincidentes ou inversas. O estudo sobre a Teoria Analítica do Calor foi publicado em 1822.

Os microcomputadores encontraram aplicação imediata para as séries de Fourier dada a grande quantidade de operações necessárias para ser obtida uma confiabilidade razoável na análise de um evento de natureza periódica.

O ouvido humano responde à intensidade das frequências individuais presentes no som, por este motivo é preciso trabalhar com informações sobre este sinal de som no domínio da frequência.

Neste capítulo será apresentada uma avaliação da base matemática para converter qual-

quer função periódica nas suas componentes de frequência, a Transformada de Fourier.

Posteriormente será explanada a FFT (Transformada Rápida de Fourier) que é um algoritmo que otimiza o cálculo da Transformada Discreta de Fourier – DFT.

2.2.1 As Séries de Fourier

Os sons naturais são compostos de múltiplas componentes de frequência. Para iniciar a análise em frequência de um determinado som, deve-se supor que apenas uma curta seção deste som é utilizada. Nesta análise, pode-se ignorar o restante do sinal do som.

Supondo também que a seção curta do som a ser analisada é periódica, pode-se então aplicar os princípios matemáticos que foram desenvolvidos por Fourier, Bernoulli, e Euler.

Para uma função periódica (forma de onda) dada por $x(t)$, na qual o período é dado por T_0 , então:

1. $x(t)$ pode ser representado por um somatório de ondas senoidais.
2. Na forma de onda senoidal, a frequência fundamental é:

$$f_0 = 1/T_0$$

3. O somatório pode ser escrito como:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(2\pi f_0 n t) + b_n \sin(2\pi f_0 n t))$$

4. $\omega_0 = \frac{2\pi}{T_0} = 2\pi f_0$ é conhecido como a primeira harmônica ou frequência angular fundamental.

$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(\omega_0 n t) + b_n \sin(\omega_0 n t))$ para um conjunto de valores de a_n e b_n .

5. O somatório é chamado de “Série de Fourier” e os valores de a_n e b_n são chamados de “coeficiente da série”.

6. Os valores de a_n e b_n podem ser encontrados por:

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cos(\omega n t) dt \text{ e } b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sin(\omega n t) dt$$

Estes resultados podem ser aplicados a funções discretas (funções que só estão definidas em pontos específicos), como também a funções contínuas. Se tivermos uma taxa de amostragem

“ F_s ” e uma série de “ N ” amostras por período, então:

1. A menor frequência presente é $f_0 = \frac{F_s}{N}$ e $\omega_0 = 2\pi f_0$
2. A série com N amostra por período, pode ser representado por:

$$x(nT_s) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(2k\pi \frac{n}{N}) + b_k \sin(2k\pi \frac{n}{N}))$$

3. a_0 corresponde ao valor da componente contínua.
4. A série de Fourier pode ser reescrita como:

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(k\omega_0 t + \theta_k))$$

onde,

$$X_k = A_k e^{j\theta_k} = a_k - j b_k$$

$$X_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cdot e^{-jk\omega_0 t} dt$$

$$X_0 = A_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) dt$$

5. O coeficiente X_k pode ser calculado numericamente através da seguinte aproximação:

$$X_K \approx \frac{2}{N} \sum_{m=1}^N x(mT_s) \cdot \exp[-j2k\pi \frac{m}{N}]$$

$$X_0 \approx \frac{1}{N} \sum_{m=1}^N x(mT_s) \text{ onde } T_s = T/N$$

2.2.2 A Transformada Contínua de Fourier

A integral de Fourier é definida pela expressão:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j2\pi f t} dt$$

Se a integral definida na equação acima existe em todo f , $X(f)$ pode ser definida como a transformada de Fourier do sinal no tempo $x(t)$.

2.2.3 A Transformada Discreta de Fourier

A equação abaixo mostra a Transformada Discreta de Fourier, para N amostras de um sinal $x(k)$.

$$X[n] = \frac{1}{N} \sum_{k=0}^{n-1} x(k) \cdot e^{-j2\pi nk/N} \quad n = 0, 1, 2 \dots N-1$$

Vale salientar que tanto para transformada continua quanto para a discreta existem as transformadas inversas que retornam o sinal do domínio da frequência para o domínio do tempo.

2.2.4 A Transformada Rápida de Fourier – FFT

A Transformada Discreta de Fourier apresentada pela equação acima, leva ao cálculo N^2 operações de multiplicação para se obter o espectro de frequência de um sinal.

Com o algoritmo da FFT (Transformada Rápida de Fourier), desenvolvido a seguir, o número de multiplicações necessárias cai para $N \cdot \log_2 N$.

A operação “*Butterfly*”

Se fizermos na equação anterior, $W_n = e^{-\frac{i2\pi}{n}}$,

$$\text{então: } y_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}} \text{ torna-se } y_k = \sum_{n=0}^{N-1} x_n W_N^{kn}$$

Se a série é dividida em elementos pares e ímpares temos:

$$y_k = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{(2n+1)k}$$

$$y_k = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{2nk}$$

Devido a $W_N^{2nk} = W_{N/2}^{nk}$, isto permite escrevermos

$$y_k = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_{N/2}^{nk}$$

Se fizermos:

$$x_1(m) = \text{AMOSTRAS PARES} = x(2n) \text{ e } x_2(m) = \text{AMOSTRAS IMPARES} = x(2n+1)$$

$$y_k = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{nk}$$

Assim, se existir um número par de amostra, o problema computacional de y_k pode ser reduzido pela metade. Então fazendo $y_{k+\frac{N}{2}}$ da forma,

$$y_{k+\frac{N}{2}} = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{n(k+N/2)} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{n(k+N/2)}$$

$$y_{k+\frac{N}{2}} = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{nk} - W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{nk}$$

Devido a $W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk}$ e $W_N^{N/2} = -1$ [prova a seguir]

$W_N^{N/2} = -1$	$W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk}$
$W_N^{N/2} = (e^{-i2\pi})^{N/2}$	$W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk} W_{N/2}^{nN/2}$
$W_N^{N/2} = (e^{i\pi})$	$W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk} W_{N/2}^{2nN/4}$
$W_N^{N/2} = -1$	$W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk} - 1^{2n}$ porque $W_N^{N/2} = -1$
	$W_{N/2}^{n(k+N/2)} = W_{N/2}^{nk}$

Com estes resultados

$$y_{k+\frac{N}{2}} = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{nk}$$

e

$$y_{k+\frac{N}{2}} = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{nk} - W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{nk}$$

A **Operação Butterfly** é construída conforme a figura 3.

$$X(k) = X_1(k) + W_N^k X_2(k) \text{ e } X(k + N/2) = X_1(k) - W_N^k X_2(k)$$

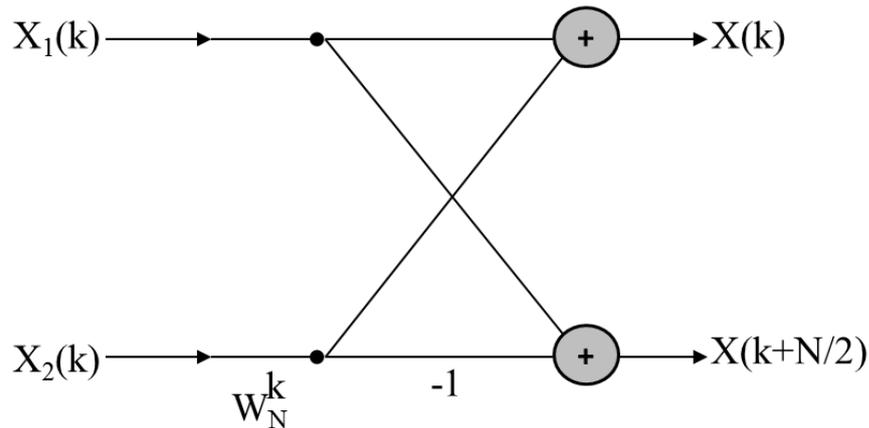


Figura 3 – Operação “Butterfly”

A operação “*butterfly*” permite que a computação da Transformada Discreta de Fourier seja dividida ao meio, ou seja, se existem 2^n amostras, a computação da Transformada Discreta de Fourier pode ser repetida exigindo somente metade do cálculo.

A Transformada Rápida de Fourier – FFT é um método computacional (algoritmo) que calcula, de forma extremamente rápida, a DFT. Para comparar, suponha um número de amostras igual a 1.024, ou seja, $N = 2^{10}$. A DFT consome tempo de CPU (processamento de computador) na ordem $O(N^2)$ o que resulta em 1.048.576 operações, enquanto a FFT tem a ordem $O(N \log_2^N)$ resultando em 10.240 operações. Em segundos, com $N = 106$, se a FFT, utilizando o processador hipotético X gasta 30 segundos, a DFT gastaria duas semanas no mesmo processador (SMITH, 1999).

2.2.5 Conclusões

Este tópico teve como objetivo descrever as fundamentações matemáticas utilizadas para o desenvolvimento deste trabalho, também comparando duas funções matemáticas cuja comparação visando a demonstração de qual desta é a de melhor utilização.

2.3 Sistema Nervoso

O neurônio é a principal unidade sinalizadora do sistema nervoso e exerce as suas funções com a participação dos gliócitos. É uma célula cuja morfologia está adaptada para as funções de transmissão e processamento de sinais: tem muitos prolongamentos próximos ao corpo celular (os dendritos), que funcionam como antenas para os sinais de outros neurônios, e um prolongamento longo que leva as mensagens do neurônio para sítios distantes (o axônio).

O neurônio comunicam-se através de estruturas chamadas sinapses, que consistem cada uma delas em uma zona de contato entre dois neurônios, ou entre um neurônio e uma célula muscular. A sinapse é o chip do sistema nervoso; é capaz não só de transmitir mensagens entre duas células, mas também de bloqueá-las ou modifica-las inteiramente: realizar um verdadeiro processamento de informação.

O impulso nervoso é o principal sinal de comunicação do neurônio, um pulso elétrico gerado pela membrana, rápido e invariável, que se propaga com enorme velocidade ao longo do axônio. Ao chegar á extremidade do axônio, o impulso nervoso provoca a emissão de uma mensagem química que leva a informação – intacta ou modificada – para célula seguinte.

2.3.1 O Neurônio

Classicamente se considera o neurônio como a unidade morfofuncional fundamental do sistema nervoso, e o gliócito como unidade de apoio. Isso porque se verificou que a célula nervosa produz e veicula diminutos sinais elétricos que são verdadeiros bits de informação, capazes de codificar tudo o que percebemos a partir do mundo exterior e do interior do nosso corpo (como os músculos e as glândulas), e tudo o que sentimos e pensamos a partir de nossa atividade mental. Por essa visão clássica, o gliócito seria encarregado apenas de alimentar e garantir a saúde do neurônio.

Entretanto, a importância dessas células de apoio cresceu muito em tempos recentes, depois que se constatou que elas lidam com sinais também, embora de tipos diferentes – sinais químicos de orientação do crescimento e da migração dos neurônios durante o desenvolvimento, de comunicação entre eles durante a vida adulta, de defesa e reconhecimento na vigência de situações patológicas e outras funções. Constatou-se mesmo que os gliócitos interferem na comunicação entre os neurônios, podendo assim modificar o conteúdo da informação transmitida.

Sendo unidades funcionais de informação, os neurônios operam em grandes conjuntos, e não isoladamente. Quanto maior o numero de neurônios e gliócitos, maior vai ser o grau de comportamento e adaptação desse ser aos diferentes ambientes que encontram na Terra.

Esses conjuntos de neurônios associados formam os chamados circuitos ou redes neurais. Por exemplo, as células nervosas da retina, que captam as imagens formadas pela luz do ambiente, só se tornam capazes de propiciar a visão se veicularem os sinais elétricos que geram

em resposta à luz, para outros neurônios localizados na própria retina e depois no cérebro. Cada um deles realiza uma pequena parte do trabalho cooperativo que ao final nos possibilitará ler um livro, ver um filme ou admirar uma tela de pintura.

2.3.2 Corpo Celular

O corpo celular é o centro metabólico do neurônio, responsável pela síntese de todas as proteínas neuronais, bem como pela maioria dos processos de degradação e renovação de constituintes celulares, inclusive de membranas. O corpo celular é composto por prolongamento. Os prolongamentos neuronais ou neuritos são o axônio, geralmente único, e um ou mais dendritos. Esses dois tipos de prolongamento conferem ao neurônio uma polaridade que é essencial à sua função. O axônio veicula os sinais de saída do neurônio, isto é, as informações que esse neurônio gera e conduz a outras células.

Os dendritos, por sua vez, recebem as informações que chegam, provenientes de outros neurônios. É no corpo celular onde são processadas as informações, obtidas a partir dos dendritos.

2.3.3 Dendritos

Geralmente são curtos (de alguns micrômetros a alguns milímetros de comprimento) e ramificam-se profusamente, à maneira de galhos de uma árvore, em ângulo agudo, originando dendritos de menor diâmetro. Apresentam contorno irregular. Os dendritos são especializados em receber estímulos, traduzindo-os em alterações do potencial de repouso da membrana.

Tais alterações envolvem entrada ou saída de determinados íons e podem expressar-se por pequena despolarização ou hiperpolarização. A despolarização é excitatória e significa redução da carga negativa do lado citoplasmático da membrana. A hiperpolarização é inibitória e significa aumento da carga negativa do lado de dentro da célula ou, então, aumento da positiva do lado de fora.

Os distúrbios elétricos que ocorrem ao nível dos dendritos e do corpo celular constituem potenciais graduáveis (podem somar-se), também chamados eletrotônicos, de pequena amplitude, e que percorrem pequenas distâncias até se extinguirem. Esses potenciais propagam-se em direção ao corpo e, neste, em direção ao cone de implantação do axônio.

2.3.4 Axônio

A grande maioria dos neurônios possui um axônio, prolongamento longo e fino que se origina do corpo ou de um dendrito principal, em região denominada cone de implantação, praticamente desprovida de substância cromidial. O axônio apresenta comprimento muito variável, dependendo do tipo de neurônio, podendo ter, na espécie humana, de alguns milímetros a mais

de um metro. É cilíndrico e, quando se ramifica, o faz em ângulo obtuso, originando colaterais de mesmo diâmetro do inicial.

O axônio é capaz de gerar, em seu segmento inicial, alteração do potencial de membrana, denominada potencial de ação ou impulso nervoso, ou seja, despolarização da membrana de grande amplitude, do tipo “tudo ou nada”, capaz de repetir-se ao longo do axônio, conservando sua amplitude até atingir a terminação axônica. Portanto, o axônio é especializado em gerar e conduzir o potencial de ação. O local onde o primeiro potencial de ação é gerado denomina-se também zona gatilho.

Tal especialização da membrana plasmática se deve à presença de canais de sódio e potássio sensíveis à voltagem, isto é, canais iônicos que ficam fechados no potencial de repouso da membrana e se abrem quando despolarizações de pequena amplitude (os potenciais graduáveis referidos acima) os atingem. O potencial de ação originado na zona gatilho repete-se ao longo do axônio porque ele próprio origina distúrbio local eletrônico que se propaga ate novos locais ricos em canais de sódio e potássio sensíveis à voltagem.

2.3.5 Sinapses

Os neurônios, principalmente através de suas terminações axônicas, entram em contato com outros neurônios, passando-lhes informações. Os locais de tais contatos são denominados sinapses, ou, mais precisamente, sinapses iterneurais. Nessas sinapses, as membranas plasmáticas dos neurônios envolvidos entram em contato, conservando espaço entre elas de apenas 2-3nm.

No entanto, há acoplamento iônico, isto é, ocorre comunicação entre os dois neurônios, através de canais iônicos concentrados em cada uma das membranas em contato. Esses canais projetam-se no espaço intercelular, justapondo-se de modo a estabelecer comunicações intercelulares, que permitem a passagem direta de pequenas moléculas, como íons, do citoplasma de uma das células para o da outra.

2.3.6 Conclusão

Neste tópico foi abordado os principais aspectos do sistema nervoso, apontando cada parte que compõem esse sistema, apresentando seu funcionamento para um melhor entendimento do trabalho produzido.

2.4 Redes Neurais Artificiais

O trabalho em redes neurais artificiais, usualmente denominadas “redes neurais”, tem sido motivado desde o começo pelo reconhecimento de que o cérebro humano processa infor-

mações de uma forma inteiramente diferente do computador digital convencional. O cérebro é um computador (sistema de processamento de informação) altamente complexo, não-linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos (p.ex., reconhecimento de padrões, percepção e controle motor) muito mais rapidamente que o mais rápido computador digital hoje existente. Considere, por exemplo, a visão humana, que é uma tarefa de processamento de informação.

A função do sistema visual é fornecer uma representação ambiente à nossa volta e, mais importante que isso, fornecer a informação de que necessitamos para interagir com o ambiente. Para sermos específicos, o cérebro realiza rotineiramente tarefas de reconhecimento perceptivo (p.ex., reconhecendo um rosto familiar inserido em uma cena não-familiar) em aproximadamente 100-200ms, ao passo que tarefas de complexidade muito menor podem levar dias para serem executadas em um computador convencional.

Um neurônio em “desenvolvimento” é sinônimo de um cérebro plástico: a plasticidade permite que o sistema nervoso em desenvolvimento se adapte ao seu meio ambiente. Assim como a plasticidade parece ser essencial para o funcionamento dos neurônios como unidade de processamento de informação do cérebro humano, também ela o é com relação às redes neurais construídas com neurônios artificiais. Na sua forma mais geral, uma rede neural é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital. Nosso interesse nesse trabalho está restrito a uma classe importante de redes neurais que realizam computação útil através de um processo de aprendizagem.

Para alcançarem bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples denominadas “neurônios” ou “unidades de processamento”. Nós podemos então oferecer a seguinte definição de uma rede neural vista como uma máquina adaptativa:

(HAYKIN, 2001) Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torna-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido.

As RNA são inspiradas na estrutura neural humana e utilizam técnicas computacionais que simulam o seu funcionamento, aprendendo através da experiência.

São compostas de muitos elementos simples, inspirados pelo sistema nervoso biológico, que operam em paralelo. A função da rede é determinada pelas conexões entre os seus elementos (STEINER et al., 2005).

Segundo (BRAGA et al., 2000) as RNAs tentam reproduzir as funções das redes biológicas, buscando implementar seu comportamento básico e sua dinâmica. No entanto, do ponto de vista físico, no momento as redes artificiais se diferem bastante das redes biológicas. É importante, contudo, observar as similaridades entre estes dois tipos de sistemas, tanto para que se possa entender melhor o sistema nervoso quanto para buscar ideias e inspirações para a pesquisa em neurocomputação. Como características comuns, pode-se citar que os dois sistemas são baseados em unidades de computação paralela e distribuídos que se comunicam por meio de conexões sinápticas, possuem detentores de características, redundância e modularização das conexões. Apesar da pouca similaridade, do ponto de vista biológico, entre os dois sistemas, estas características comuns permitem às RNAs reproduzir com fidelidade várias funções somente encontradas nos seres humanos. Acredita-se, portanto, que o futuro da neurocomputação possa se beneficiar ainda mais do desenvolvimento de modelos que tenham apelo biológico. Estruturas encontradas nos sistemas biológicos podem inspirar o desenvolvimento de novas arquiteturas para modelos de RNAs. Similarmente, espera-se que a Biologia e as Ciências Cognitivas possam se beneficiar do desenvolvimento dos modelos da neurocomputação.

As RNAs foram criadas, então, não para imitar o funcionamento do cérebro, mas como um modelo de resolução de problemas não algorítmicos inspirado no funcionamento do cérebro humano, com a capacidade de aprender e se adaptar continuamente ao ambiente.

A tecnologia das Redes Neurais Artificiais (RNAs) visa solucionar problemas de reconhecimento de padrões que geralmente são baseados em um conjunto de informações previamente conhecido. Geralmente os conjuntos de dados são divididos em conjunto de treinamento e conjunto de teste. Atualmente, pesquisadores em RNA's estão buscando uma compreensão das capacidades da natureza humana, as quais possibilitam que as pessoas construam soluções para problemas que não sejam resolvidos através de métodos tradicionais.

As Redes Neurais Artificiais (RNAs) visam na sua maioria solucionar problemas de inteligência artificial, modelando sistemas através de circuitos (conexões) que possam simular o sistema nervoso humano, abrangendo a capacidade que o mesmo possui de aprender e agir perante as mais adversas situações apresentadas, bem como adquirir conhecimento através da experiência e da observação.

A complexidade das estruturas elementares das Redes Neurais Biológicas é muito maior

do que a dos modelos matemáticos usados nas Redes Neurais Artificiais (RNA's), demonstrando as dificuldades encontradas para se tentar imitar o funcionamento do sistema nervoso humano. O sistema nervoso é formado por bilhões de células nervosas, enquanto que uma rede neural artificial possui de dezenas a no máximo milhares de unidades de processamento (neurônio).

Uma rede neural artificial pode ser vista como um conjunto de varias unidades interconectadas (similar à estrutura do cérebro), denominadas de neurônios artificiais, cada qual contendo uma pequena porção local de memória. Estes conceitos foram baseados e fundamentados nos estudos realizados nas células nervosas naturais. Portanto, busca-se aproximar ao máximo o funcionamento das redes neurais artificiais das redes neurais biológicas, na tentativa de buscar a desenvoltura com que o cérebro humano desempenha suas funções.

O procedimento utilizado para realizar o processo de aprendizagem é chamado de *algoritmo de aprendizagem*, cuja função é modificar os pesos sinápticos da rede de uma forma ordenada para alcançar um objetivo de projeto desejado.

A modificação dos pesos sinápticos é o método tradicional para o projeto de redes neurais. Esta abordagem é bastante próxima da teoria dos filtros adaptativos lineares, que já está bem estabelecida e foi aplicada com sucesso em diversas áreas (WINDDROW; STEARNS, 1985; HAYKIN, 2001). Entretanto, é possível também para uma rede neural modificar sua própria topologia, o que é motivado pelo fato de os neurônios no cérebro humano poderem morrer e que novas conexões sinápticas possam crescer.

As redes neurais são também referidas na literatura como *neurocomputadores*, *redes connexionistas*, *processadores paralelamente distribuídos*.

Alguns modelos de redes neurais artificiais possuem muitos neurônios conectados numa estrutura de pesos de conexão e com facilidade de adaptação, proporcionando uma estrutura paralela. A estrutura paralela é desejável pois se algum(s) neurônio(s) falhar(em), os efeitos na rede como um todo não será significativo para o desempenho do sistema se outro caminho de conexão entre os neurônios puder burlar a falha, surgindo então a tolerância á falha.

É evidente que uma rede neural extrai seu poder computacional através, primeiro, de sua estrutura maciçamente paralelamente distribuída e segundo de sua habilidade de aprender e portanto de generalizar. A *generalização* se refere ao fato de a rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento (aprendizagem). Estas duas capacidades de processamento de informação tornam possível para as redes neurais resolver problemas complexos (de grande escala) que são atualmente intratáveis. Na prática, contudo, as redes neurais não podem fornecer uma solução trabalhando individualmente. Em vez disso, elas precisam ser integradas em uma abordagem consistente de engenharia de sistemas. Especificamente, um problema complexo de interesse é *decomposto* em um número de tarefas relativamente simples, e atribui-se a redes neurais um subconjunto de tarefas que *coinci-*

dem com as suas capacidades inerentes. Entretanto, é importante reconhecer que nós temos um longo caminho a percorrer antes de construirmos (se porventura conseguirmos) uma arquitetura computacional que mimetize um cérebro humano.

Principais características das RNA's (BARONE, 1999):

- Capacidade de “aprender” através de exemplos e de generalizar este aprendizado de forma a reconhecer elementos similares, que não foram apresentados no conjunto de exemplo (treinamento);
- Bom desempenho em tarefas pouco ou mal definidas, onde falta o conhecimento explícito de como resolvê-las, o aprendizado se dá através de exemplos;
- Robustez à presença de informações falsas ou ausentes, escolha dos elementos no próprio conjunto de treinamento (integridade do conjunto de treinamento);
- No contexto de classificação de padrões, uma rede neural pode fornecer informações sobre quais padrões selecionar em função do grau de confiança apresentado (confiabilidade de conjunto de treinamento);
- Tolerância à falha.

2.4.1 Histórico

Data de 1943 o primeiro modelo artificial de um neurônio biológico, que foi denominado pelos seus criadores Warren McCulloch (neurofisiologista) e Walter Pitts (matemático) de neurônio formal ou neurônio artificial. Em seu trabalho "*A Logical Calculus of the Ideas Immanent in Nervous Activity*", que unificou conceitos de neurofisiologia e lógica matemática, eles mostraram que uma coleção de neurônios formais era capaz de realizar alguns cálculos lógicos. O princípio de funcionamento do neurônio formal é apresentado na figura 4.

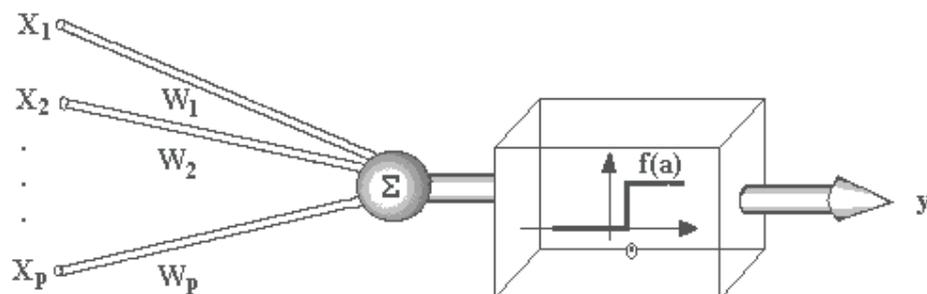


Figura 4 – Neurônio Formal

Fonte: Disponível em: <<http://www.din.uem.br/ia/neurais/>>

Por um longo período de tempo, poucos resultados foram obtidos. Em 1949, Donald Hebb, em seu livro intitulado "*The Organization of Behavior*", mostrou novas idéias, sendo o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios, de maneira que as mudanças nas forças das sinapses fossem proporcionais às ativações. Em seguida, Widrow e Hoff mostraram uma regra de aprendizado baseada na minimização do erro através do método do gradiente, que ficou conhecida como regra de *Widrow-Hoff* ou regra delta.

Em 1959, a partir das idéias desenvolvidas por Pitts e McCulloch, Frank Rosenblatt descreveu o perceptron, primeiro modelo de rede neural que acrescenta sinapses aos nodos de Pitts e McCulloch. Rosenblatt descreveu uma topologia onde os neurônios eram organizados em camadas de entrada e saída, conforme se observa na Figura 5. Os pesos das conexões eram adaptados levando a rede a aprender certas funções lógicas e classificar vários tipos de padrões.

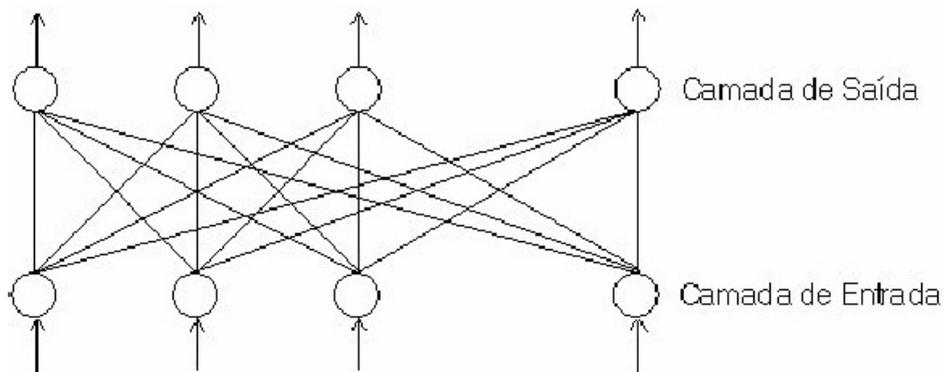


Figura 5 – Rede de Perceptrons

Fonte: Disponível em: <<http://www.din.uem.br/ia/neurais/>>

Em 1962, Widrow e Hoff desenvolveram as redes *ADALINE* (*ADaptive Linear Network*) e o *MADALINE* (*MAny ADALINE*) *perceptron*, que apresentam uma arquitetura de três camadas, conforme se pode observar na figura 6.

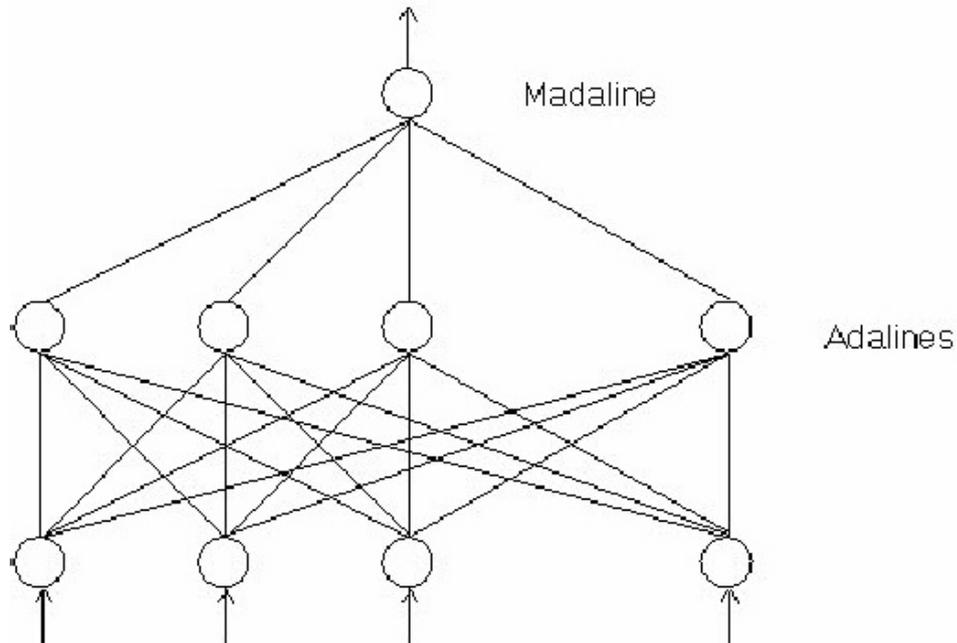


Figura 6 – Rede Adaline e Madaline

Fonte: Disponível em: <<http://www.din.uem.br/ia/neurais/>>

A partir de 1969, a pesquisa sobre redes neurais foi praticamente abandonada devido à repercussão do trabalho de Minsky e Papert, que expunha as limitações do *Perceptron*. Eles diziam que tal modelo se restringia à resolução de problemas linearmente separáveis, ou seja, problemas cuja solução pode ser obtida dividindo o espaço de entrada em duas regiões através de uma reta.

Foi em 1974 que Werbos lançou as bases do algoritmo *back-propagation* (retro-propagação), cuja estrutura se apresenta na figura 7, o que possibilitou que redes neurais de múltiplas camadas apresentassem capacidade de aprendizado para resolver problemas não-lineares. A partir de meados da década de 80 foram publicados os resultados obtidos pela técnica de treinamento por retro-propagação, fato que gerou uma explosão de interesse pelas RNA na comunidade internacional (HAYKIN, 2001).

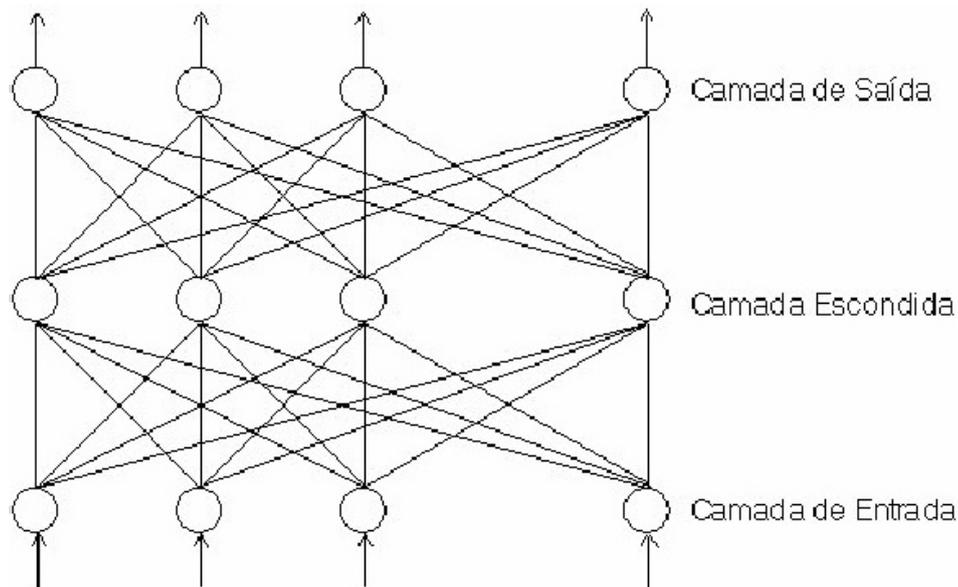


Figura 7 – Estrutura do Método Back-Propagation

Fonte: Disponível em: <<http://www.din.uem.br/ia/neurais/>>

2.4.2 O Neurônio Artificial

O funcionamento do cérebro humano é muito complexo e dinâmico. Ele é capaz, através de uma constante interação com o meio em que está inserido, de receber, processar e responder a uma enorme quantidade de informações em um minúsculo intervalo de tempo. Apesar da complexidade geral do seu funcionamento, o cérebro é constituído basicamente de neurônios, que nada mais são do que simples unidades de processamento. Tais neurônios, ilustrados na figura 8, são constituídos basicamente pelas seguintes partes:

- Dendritos: constituem as entradas dos impulsos elétricos;
- Corpo celular: região onde são processadas as informações, obtidas a partir dos dendritos;
- Axônio: saída das informações que foram processadas no corpo celular.

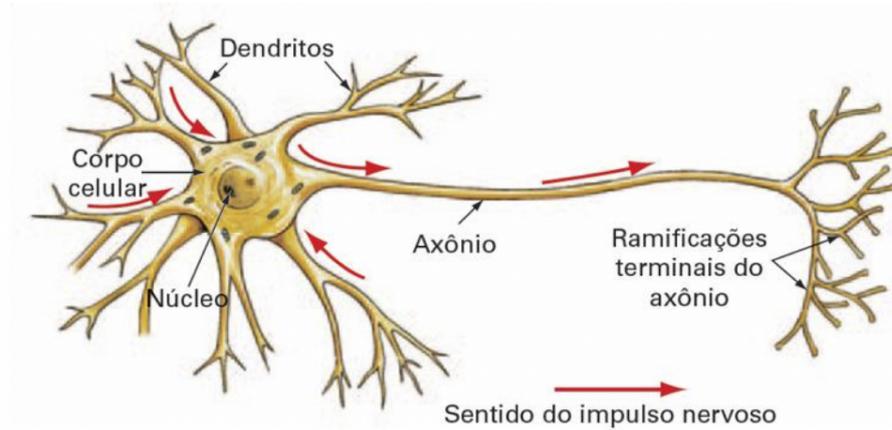


Figura 8 – Neurônio Biológico

Fonte: (WASSERMAN, 1989)

O neurônio artificial propõe um funcionamento análogo ao neurônio biológico, e é a unidade de processamento de informação fundamental para operação de uma RNA. A figura 9 mostra um modelo de neurônio artificial em que se pode verificar os cinco elementos básicos que o compõem que serão, em seguida, descritos: um conjunto de sinapses ou elos de conexão, um somador, o bias, uma função de ativação e uma saída.

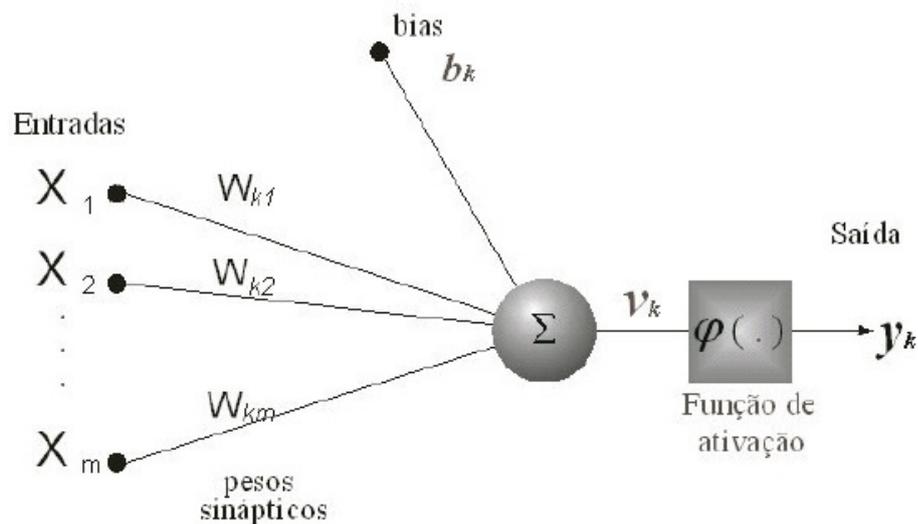


Figura 9 – Modelo de um Neurônio Artificial

Fonte: (HAYKIN, 2001)

- sinapses ou elos de conexão: são caracterizados por um peso ou força própria (W_{ki});
- somador (V_k): tem a função de somar os sinais de entrada, ponderado pelas respectivas sinapses ou pesos do neurônio. A equação a seguir, representa este somatório:

$$V_k = \sum_{i=-1}^m X_i W_{ki}$$

onde: W_{ki} = pesos do neurônio k ; X_i = sinais de entrada; m = número de entradas.

- função de ativação (φ): tem a função de limitar o sinal de saída a um valor finito.
- bias (b_k): trata-se de uma entrada adicional acrescida ao neurônio artificial. Seu valor de entrada é fixado em "1", e seu peso é ajustável como os demais pesos da rede.
- saída (V_k): é expressa pela equação a seguir.

$$Y_k = \varphi(V_k + b_k)$$

Resumidamente, um neurônio artificial opera de acordo com o diagrama apresentado na figura 10.

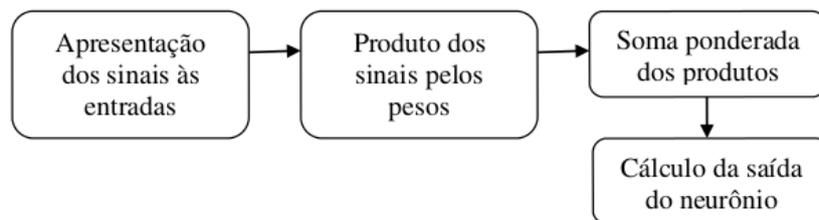


Figura 10 – Diagrama de Funcionamento de um Neurônio Artificial

2.4.3 Função de Ativação de uma Rede Neural Artificial

Segundo (STEINER et al., 2005), a função de ativação é muito importante para o comportamento de uma RNA, porque é ela que define a saída do neurônio artificial e, portanto, o caminho pelo qual a informação é conduzida.

Existem vários tipos de funções de ativação, dentre os quais se destacam como principais: a função degrau que produz uma saída binária; a função linear; a função sigmoideal, que é não-linear crescente e diferenciável. A figura 11, traz as representações gráficas destas três funções de ativação.

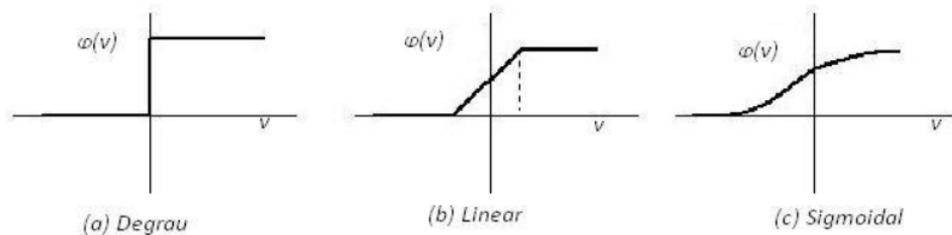


Figura 11 – Funções de Ativação de uma RNA

Fonte: (HAYKIN, 2001)

2.4.4 Arquitetura de uma Rede Neural Artificial

Outro detalhe importante a ser considerado é a maneira como os neurônios artificiais podem ser agrupados. Este agrupamento se sucede no cérebro humano de maneira que as informações possam ser processadas de forma dinâmica ou interativa. Biologicamente, as redes neurais são organizadas e construídas de forma tridimensional por componentes microscópicos. Há uma forte restrição no número de camadas que a rede pode conter, limitando consideravelmente o tipo e o escopo da implementação da mesma em silício.

Os neurônios que constituem uma RNA podem ser organizados em camadas e de diversas maneiras. A este arranjo entre os neurônios e suas respectivas conexões dá-se o nome de topologia ou arquitetura da rede. Em função do sentido do fluxo de dados, as redes podem ser classificadas em: *feed-forward* ("alimentadas"adiante) em que os dados só se propagam para frente e *feedback* (recorrentes), se os dados podem se propagar nos dois sentidos. De acordo com (HAYKIN, 2001), existem três diferentes tipos de arquiteturas:

1. Rede neural com uma única camada alimentada adiante (*feed-forward*): a figura 12 mostra o modelo desta arquitetura. É importante notar que a denominação de camada única se dá pelo fato de não ocorrer nenhum processamento na camada de entrada nem na camada de pesos. Tal processamento ocorre exclusivamente na camada de saída, daí a sua denominação.

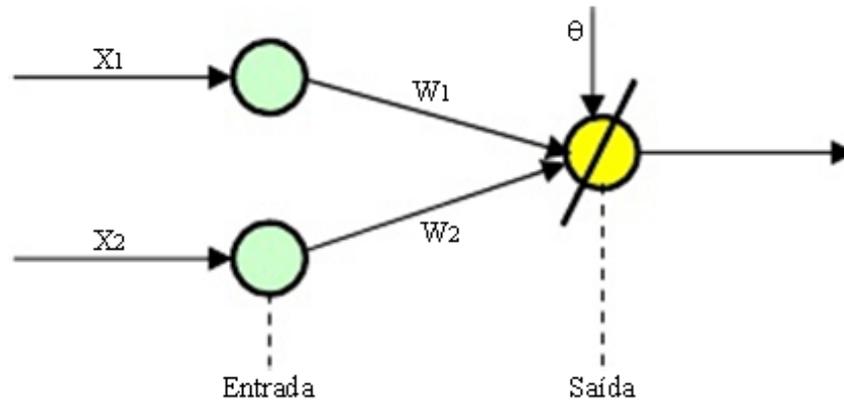


Figura 12 – Modelo de Rede Neural com uma Camada

2. Rede neural com múltiplas camadas alimentadas adiante (*feed-forward*): as figuras 13 e 14 ilustram este tipo de rede. Na primeira, tem-se uma rede com duas camadas – uma oculta outra de saída. A segunda ilustra uma rede com três camadas – duas ocultas e uma de saída. Mais uma vez cabe ressaltar que, para denominar as redes, só se levam em conta as camadas onde ocorre processamento; logo, desconsideram-se a camada de entrada e as de pesos.

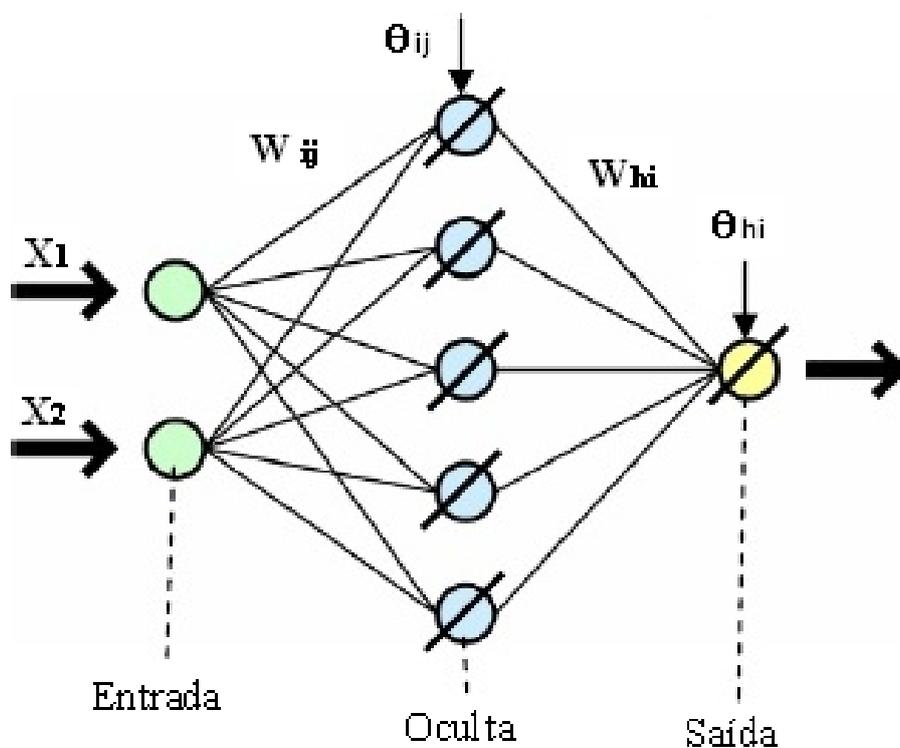


Figura 13 – Modelo de Rede Neural com Duas Camadas

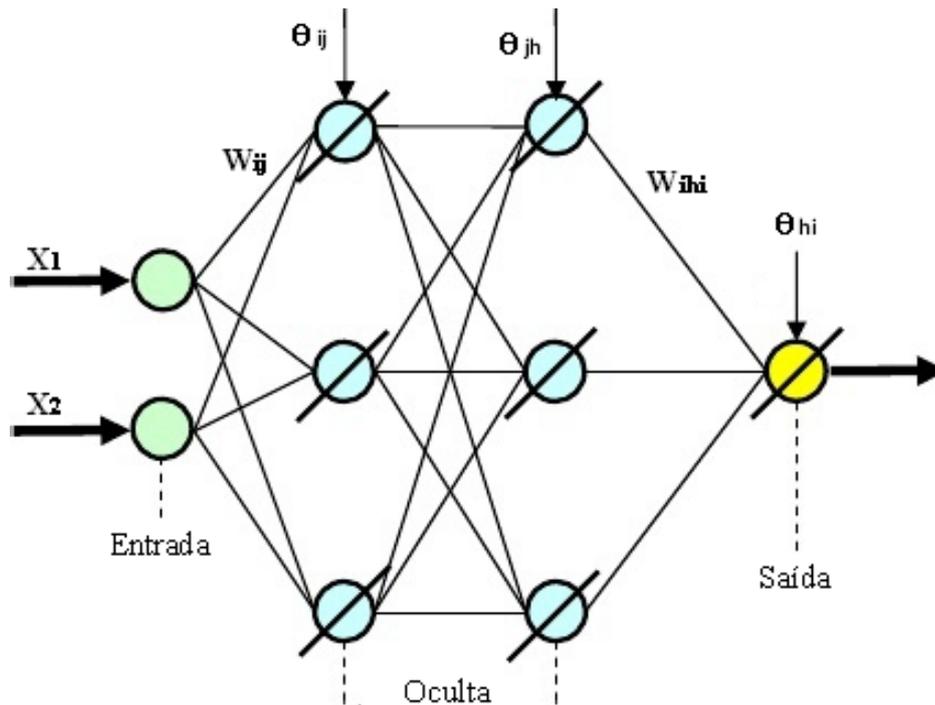


Figura 14 – Modelo de Rede Neural com Três Camadas

3. Rede neural recorrente (*feedback*): caracteriza-se pela apresentação de pelo menos uma laço de realimentação. Este tipo de rede é ilustrado na figura 15, a seguir.

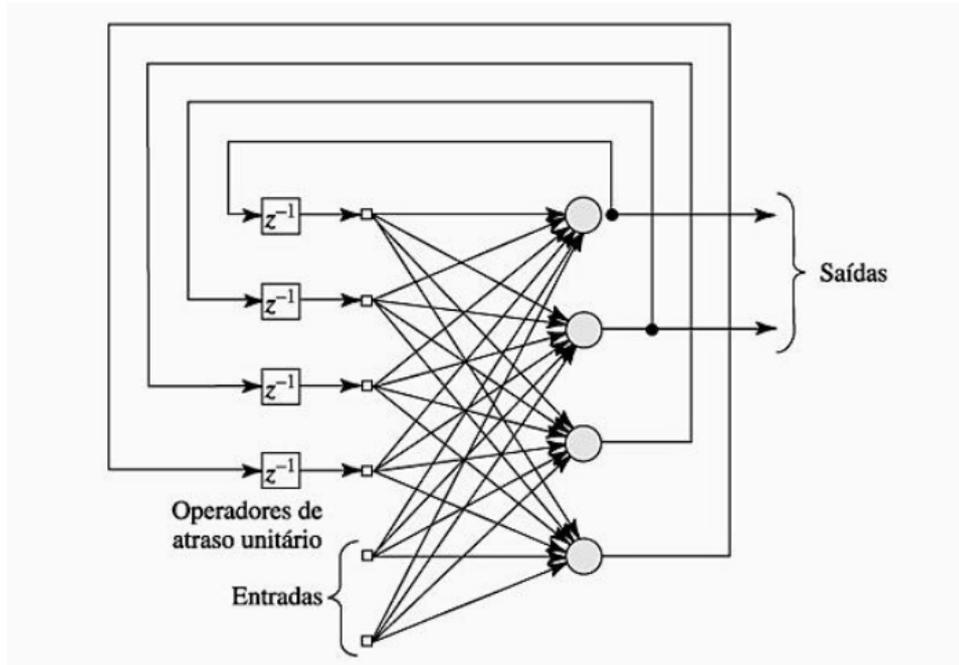


Figura 15 – Modelo de Rede Neural Recorrente

Fonte: (HAYKIN, 2001)

2.4.5 Aprendizado

Para o aprendizado das redes neurais, foram propostos diversos métodos de treinamento, sendo estes subdivididos em dois paradigmas principais: o aprendizado supervisionado e o não supervisionado. Para estes modelos existem vantagens e desvantagens que serão expostas a seguir. As RNA's possuem a capacidade de aprender por exemplo, determinando a intensidade de conexões entre os neurônios pertencentes à rede. Logo, um conjunto de procedimentos definidos para ajustar os parâmetros de uma RNA, a fim que a mesma possa aprender uma determinada função, é chamado de algoritmo de aprendizado. A designação de uma RNA, na resolução de um determinado problema, passa inicialmente por um processo de aprendizagem, onde a rede procura extrair informações relevantes de padrões de informação apresentados a ela, modelando uma representação própria.

Redes Neurais Artificiais possuem a capacidade de aprender por exemplos e fazer interpolações e extrapolações do que aprenderam. No aprendizado conexionista, não se procura obter regras como na abordagem simbólica da Inteligência Artificial (IA), mas sim determinar a intensidade de conexões en-tre neurônios.(BRAGA et al., 2000)

Supervisionado

A vasta maioria das redes neurais artificiais tem utilizado o treinamento supervisionado. Deste modo, a saída atual da rede neural é comparada com a saída desejada. Os pesos

terão os seus valores iniciais setados aleatoriamente, e serão ajustados, através do algoritmo de aprendizagem, pela rede na próxima iteração ou ciclo.

O ajuste sináptico é dependente do valor esperado e do sinal atual de saída. Desta maneira, o método de aprendizado tenta minimizar o fluxo corrente de erros de todos os elementos em processamento. Esta redução global de erros trabalha modificando continuamente os pesos até que a rede alcance uma certa precisão.

Com o aprendizado supervisionado, as redes neurais artificiais devem ser treinadas antes de serem usadas. O treinamento consiste da apresentação dos sinais de entrada e saída à rede. Estes dados são frequentemente referenciados ao conjunto de treinamento. A fase de treinamento pode consumir uma grande fatia de tempo. Em alguns sistemas protótipos, com um inadequado poder de processamento, o aprendizado pode levar semanas. O treinamento é considerado completo quando a rede neural alcança um certo nível de performance. Este nível significa que a rede alcançou uma precisão estatística conforme as produções de saída necessárias para uma dada sequência de entradas. Quando não há mais a necessidade de aprendizado, os pesos são praticamente “congelados” para aplicação. Alguns tipos de redes neurais permitem um treinamento contínuo, com uma taxa muito baixa de aprendizado, enquanto a mesma está em operação. Este processo ajuda a rede a adaptar-se gradualmente as condições de mudança.

O conjunto de treinamento precisa ser suficientemente grande para conter informações necessárias para que a rede aprenda os moldes e as relações importantes. Se a rede é treinada somente com um exemplo e um determinado tempo, todos os pesos serão setados meticulosamente para este fato, os quais poderiam sofrer alterações drásticas no aprendizado de um próximo fato. Conforme um resultado, o sistema precisa aprender com todos os fatos em conjunto, provendo posteriormente o melhor ajuste dos pesos para todo o conjunto de fatos.

A maneira com que os sinais de entrada são representados, ou codificados, determina o maior componente constituinte para o sucesso de instruções da rede. Normalmente, as redes neurais artificiais somente manipulam, ou trabalham, com dados numéricos como entrada. Por este motivo, os dados do mundo exterior, devem ser tratados e convertidos para que se possa alimentar a rede. Esta captura de estímulos do mundo real pode ser feita através de vários tipos de dispositivos, tais como: câmeras de vídeo, diversos de sensores, microfones, etc.

Várias técnicas de condicionamento já estão disponíveis para serem aplicadas a implementações de redes neurais artificiais, viabilizando e principalmente facilitando para que o desenvolvedor da rede encontre o melhor formato para os dados, e uma arquitetura adequada para a rede objetivando uma determinada aplicação.

Após o treinamento supervisionado, é importante analisar o que a rede pode realizar com os dados que ainda não foram apresentados à mesma. Se o resultado de saída do sistema não for razoável para este novo conjunto de dados (chamado conjunto de teste), presume-se que o

treinamento da rede ainda não foi suficiente.

Esta avaliação é crítica para assegurar que a rede simplesmente não memorizou um dado conjunto de dados, mas sim aprendeu os modelos/padrões gerais envolvidos na aplicação (generalização). É importante ressaltar que às vezes o problema da generalização é dividido á má qualidade dos dados usados para o treinamento e não um problema da rede.

Não Supervisionado

O aprendizado não supervisionado é a grande promessa para o futuro, visto que implica que a rede aprenda sem a necessidade de um conjunto de treinamento.

Estas redes não suportam influências externas para ajustar os seus pesos sinápticos, pois há um monitoramento de desempenho interno da mesma, analisando as regularidades e/ou tendências dos sinais de entrada, e conseqüentemente adaptando-se automaticamente as necessidades da rede.

Possuindo características de auto nivelção, sem um suposto auxílio para determinar se o aprendizado converge ou não para o caminho certo, a rede possui mecanismos, mais precisamente, informações, de como se organizar. Esta propriedade e percepção da rede devem-se a topologia e as regras de aprendizado adotado pela rede neural artificial.

Uma rede com o algoritmo de aprendizado não supervisionado deve ter enfatizado a cooperação entre as camadas de unidades de processamento. A competição entre estas unidade é base de aprendizado da rede. Normalmente, quando a competição pelo aprendizado ocorre de fato, somente os pesos pertencentes à unidade de processamento vencedora são ajustados.

Velocidade de Aprendizado

A velocidade com que uma rede neural artificial aprende, depende de vários fatores. A baixa taxa de aprendizado resulta num tempo relativamente grande para a realização do aprendizado da rede, a fim de produzir um resultado adequado para o sistema em questão. Já com uma alta taxa de aprendizado, a rede pode não ser capaz de fazer uma possível discriminação fina em relação aos sistemas que aprendem de forma mais lenta.

Geralmente, vários fatores, além do tempo de aprendizado, precisam ser considerados quando se aborda a tarefa de aprendizado e treinamento da rede.

Alguns fatores que estão fortemente acoplados ao processo de aprendizado de uma RNA (DATA... , 2012):

- A complexidade da rede;
- O número de camadas (tamanho);

- O paradigma de seleção;
- A arquitetura adotada;
- A arquitetura de aprendizado;
- Algoritmo de aprendizado;
- As regras empregadas;
- A precisão desejada;

Todos estes fatores contribuem e alteram o tempo de treinamento da rede. A mudança de qualquer um destes fatores pode estender o tempo de treinamento para uma razão não muito significativa, ou resultando em uma precisão não satisfatória.

A maioria dos algoritmos de aprendizado possuem alguma provisão para a taxa de aprendizado ou em muitos casos, uma constante de tempo. Normalmente, este valor está compreendido num intervalo entre zero e um $[0, 1]$. Se a taxa de aprendizado exceder o valor máximo, o algoritmo de aprendizado irá corrigir os pesos da rede.

Pequenos valores da taxa de aprendizado não retificarão os erros tão rapidamente, mas se pequenos passos são tomados na correção de erros, há uma grande possibilidade de se alcançar uma boa convergência de aprendizado da rede.

Algoritmos de aprendizado

Muitas das leis de aprendizado estão em uso, e a maioria delas são apenas variações da mais difundida lei, que é a lei de Hebb. As pesquisas em torno das funções de aprendizado continuam, e busca-se aproximar cada vez mais estes modelos dos tão sonhados e perfeitos padrões biológicos.

Embora o homem esteja muito além de descobrir realmente como ocorre o processamento biológico, e o aprendizado seja algo extremamente complexo, simplificações e morfologias matemáticas continuam em desenvolvimento.

Abaixo, são apresentadas algumas das mais conhecidas e difundidas leis de aprendizado (DATA. . . , 2012):

- Princípio de aprendizado de Hebb – a primeira e indiscutivelmente a mais conhecida regra de aprendizado, foi apresentada pelo biólogo e psicólogo Donald Hebb. A descrição da mesma foi exposta em seu famoso livro *The Organization of Behavior* em 1949. A regra parte do seguinte pressuposto. Se um neurônio recebe uma entrada, proveniente de outro neurônio, e ambos estão ativos, isto é, possuem o mesmo sinal, os pesos entre os neurônios precisam ser excitados;

- A Lei de *Hospfield* – é praticamente similar ao princípio de aprendizado de Hebb com apenas uma exceção: a lei de Hopfield especifica a magnitude da excitação ou inibição. Se a saída desejada e o sinal de entrada estão ambos ativos ou inativos, os pesos são incrementados pela taxa de aprendizado, caso contrário, decrementados;
- Regra Delta de Widrow – esta regra é uma variação um pouco além do princípio de aprendizado proposto por Hebb. A regra delta apresentada por Widrow é a mais comumente usada nos dias atuais. Esta regra se baseia na simples ideia da contínua modificação da intensidade e a importância das conexões de entrada; reduzindo consideravelmente a diferença entre o valor de saída desejado e o atual valor de saída da unidade de processamento, no caso o neurônio artificial. A regra delta altera os pesos sinápticos de modo que minimize o erro quadrático da rede, trabalhando da seguinte forma: o erro calculado na saída é transformado pela derivação da função de transferência e conseqüentemente usado para ajustar os pesos de entrada da camada previa da rede, ou seja, o erro é retro propagado às camadas anteriores, sendo uma de cada vez. O processo de retro propagação dos erros da rede continua até que a primeira camada da rede seja alcançada. Os tipos de redes chamadas de *feedforward* e *backpropagation* derivam seus nomes dos métodos adotados pelo processamento do erro. Quando se usa a regra delta, é importante assegurar que o conjunto de dados de entrada está disposto de forma aleatória ou gerado de forma randômica. Uma vez mal organizado, este conjunto de treinamento pode conduzir a rede a não convergência da precisão desejada, impossibilitando o aprendizado do problema em questão.
- A Lei de aprendizado de *Teuvo Kohonen* – desenvolvida por Teuvo Kohonen, a mesma foi inspirada nos sistemas biológicos, onde os elementos competem entre si por uma oportunidade de aprender, ou atualizar/ajustar seus respectivos pesos. A unidade de processamento que possui o melhor sinal de saída será considerada o mais apto, e conseqüentemente passarão a ter capacidade e privilégio de inibir os ajustes sinápticos de seus respectivos vizinhos terão permissão para ajustar seus pesos. A abrangência e a possível área que uma unidade vizinha pertence está relacionada ao período de treinamento da rede. O paradigma atual usa o seguinte procedimento: é formada uma grande área de vizinhança e a medida com que ocorre o processo de treinamento, há uma seleção e conseqüentemente um estreitamento da mesma.

Redes de Múltiplas Camadas ou Redes *Feed-forward*

Em toda rede *feed-forward*, conforme ilustrado nas figuras 13 e 14 anteriores, podem ser observadas as seguintes unidades básicas:

- Entradas: encarregam-se de receber as informações do meio externo. Uma rede neural deve possuir o número de entradas igual ao número de coordenadas do vetor de entrada de dados. Este número é obtido a partir da análise dos atributos do problema estudado. Nesta unidade básica não ocorre o processamento de informações.
- Camadas intermediárias ou camadas escondidas: não interagem diretamente com o ambiente, mas têm a importante função de processar as informações vindas da camada anterior e enviar o resultado deste processamento à camada seguinte, auxiliando no ajuste dos pesos sinápticos da rede.
- Camada de saída: recebe informações da camada anterior, processa-as e as envia, ao meio externo, o resultado do processamento.

As redes *feed-forward* podem ser treinadas com o algoritmo *back-propagation* a fim de encontrar um erro mínimo. Este procedimento, porém, pode conduzir a um erro mínimo local e não global como o desejado. Para se evitar este impasse, deve-se variar o conjunto de pesos iniciais, o número de neurônios na camada escondida ou até mesmo o número de camadas escondidas buscando-se, assim, diferentes maneiras de encontrar a melhor solução para o problema.

Estas redes normalmente utilizam a função de ativação logística sigmoideal, que fornece saídas entre no intervalo $(0, 1)$ para entradas no intervalo $(-\infty, \infty)$.

3 Concepção do Projeto

A seguir será apresentada os principais pontos que levou este trabalho a ser realizado utilizando as técnicas de redes neurais artificiais, buscando um maior entendimento em relação à utilização desta técnica para o desenvolvimento de sistemas especialistas com a capacidade de um processamento de informações inerentemente ao processamento do cérebro humano buscando o aprendizado com um maior índice de resultados positivos.

De acordo com diversas estruturas neurais e algoritmos de aprendizagem propostos por vários pesquisadores, redes neurais possuem certas características exclusivas de sistemas biológicos. Tais características entram em conflito com os tradicionais métodos computacionais. Sistemas de computação baseados em redes neurais tem a capacidade de receber ao mesmo tempo várias entradas e distribuí-las de maneira organizada. Geralmente, as informações armazenadas por uma rede neural é compartilhada por todas as suas unidades de processamento, característica que contrasta com os atuais esquemas de memória, onde a informação fica confinada em um determinado endereço.

Em um sistema de rede neural, a informação pode parecer ter representação redundante, porém, o fato de que ela se encontre distribuída por todos os elementos da rede significa que mesmo que parte da rede seja destruída, a informação contida nesta parte ainda estará presente na rede, e poderá ser recuperada. Portanto, a redundância na representação de informações em uma rede neural, diferente de outros sistemas, transforma-se em uma vantagem, que torna o sistema tolerante a falhas. Os atributos de uma rede neural, tais como aprender através de exemplos, generalizações redundantes, e tolerância a falhas, proporcionam fortes incentivos para a escolha de redes neurais como uma escolha apropriada para aproximação para a modelagem de sistemas biológicos.

O modelo de rede neural tem muitos neurônios conectados por pesos com capacidade de adaptação que podem ser arranjados em uma estrutura paralela. Por causa deste paralelismo, a falha de alguns neurônios não causam efeitos significativos para a performance de todo o sistema, o que é chamado de tolerância a falhas.

A principal força na estrutura de redes neurais reside em sua habilidades de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada de maneira razoável tem a habilidade de generalizar quando é apresentada à entradas que não estão presentes em dados já conhecidos por ela.

A característica mais significativa de redes neurais está em sua habilidade de aproximar

qualquer função contínua não linear de um grau de correção desejado. Esta habilidade das redes neurais as têm tornado útil para modelar sistemas não lineares na combinação de controladores não lineares.

Redes Neurais podem ter várias entradas e várias saídas, eles são facilmente aplicáveis à sistemas com muitas variáveis.

Com o avanço em tecnologias de *hardware*, existem componentes com funções voltadas à sistemas com implementações voltadas para redes neurais, o que traz uma velocidade adicional à computação neural.

Levando todos esse aspectos descritos nos paragrafos acima, foi escolhido como tecnologia para desenvolvimento deste sistema as redes neurais artificiais como um meio bem mais interessante para ser empregado na criação deste projeto.

4 Implementação

4.1 Introdução

Nesta capítulo são descritos os processos de desenvolvimento do trabalho. Apresentando detalhadamente a criação da aplicação de reconhecimento de instrumentos musicais, desde a concepção até a versão final da aplicação desenvolvida.

4.2 O MatLab

Para realização deste trabalho foi utilizado o software MatLab que é um programa de computador especializado e otimizado para cálculos científicos e de engenharia. Inicialmente, era projetado para cálculos com matrizes; ao longo dos anos, transformou-se em um sistema computacional flexível, capaz de resolver essencialmente qualquer problema técnico.

O programa MatLab implementa a linguagem de programação MatLab, juntamente com uma grande biblioteca de funções predefinidas que tornam as tarefas de programação técnica mais fáceis e eficientes (CHAPMAN, 2006).

MatLab é um programa muito grande, com uma rica variedade de funções. Até mesmo a versão básica do MatLab, sem ferramentas adicionais, é muito mais rica que outras linguagens de programação técnica. Existem mais de 1.000 funções no produto MatLab, e as ferramentas adicionais ampliam esses recursos com muito mais funções em diferentes especialidades (CHAPMAN, 2006).

4.3 A Gravação das Notas Musicais

Para gravação das notas musicais foi utilizado o software “*Fl Studio 10*”, que tem a capacidade de reproduzir fielmente o som sem ruído de diferentes instrumentos musicais e suas notas, para a continuação desse projeto foram gravadas as notas musicais Dó, Ré, Mi, Fá e Sol, dos cinco instrumentos musicais: violão, piano, flauta e trompete, em formato wav com uma taxa de amostragem de 44.100 Hz, por ser uma taxa de amostragem alta as mesmas foram reduzidas posteriormente utilizando o *software “FormatFactory”* para uma taxa de amostragem de 4.400 Hz. Com intuito de automatizar a rede neural na fase de processamento.

4.4 A Normalização das Notas Musicais

Para o treinamento da rede neural é necessário transformar os dados de entrada da rede em padrões de fato reconhecíveis e processáveis por ela. O processo descrito abaixo é realizado para cada nota musical carregada no programa de reconhecimento de timbres.

```
1. nota = nota/max(abs(nota));
2. nota = abs(fft(nota));
```

Na primeira linha são normalizados os valores de entradas das notas musicais.

Na segunda linha a função *fft* calcula a Transformada Rápida de Fourier.

4.5 O Desenvolvimento da Rede Neural

Para a construção da aplicação foi utilizado o *software MatLab versão 7.0* a linguagem de programação utilizada é a do próprio *MatLab*. Abaixo está descrito previamente o código de criação da rede neural implementada no *MatLab*.

```
net = newff([0 17; 0 3; 0 17; 0 16; 0 8; 0 9; 0 16], (Limites dos padrões de entrada - mín. e máx.)
([18,1]), (Número de neurônios de cada camada)
'tansig','purelin', (Função de ativação de cada camada)
'trainlm'); (Algoritmo de treinamento)
net.trainParam.epochs = handles.metricdata.numCiclo;
net.trainParam.goal = 0;
net.trainParam.lr = handles.metricdata.taxaApre;
```

A função *newff* cria uma rede do tipo *Backpropagation*.

As funções de ativação utilizadas na aplicação foram a *tansig* que é a função Tangente Hiperbólica e a *purelin* que é a função Linear.

O algoritmo de treinamento utilizado na aplicação foi o *trainlm* que é o algoritmo *Backpropagation Levenberg-Marquardt*.

A variável *net.trainParam.goal* é responsável por armazenar o erro desejável que é utilizado no treinamento como uma condição de parada da rede neural, em outras palavras quando

a rede tiver treinada o erro tem que ser igual a zero.

A variável *net.trainParam.epochs* é responsável por armazenar o numero máximo de ciclos ou interações que a rede neural ira realizar.

A variável *net.trainParam.lr* é responsável por armazenar a taxa de aprendizagem da rede neural fundamental para a aprendizagem da rede.

Abaixo está descrito a função *train* que é utilizada para treinar a rede neural de acordo com seus parâmetros de entrada que é a rede neural criada anteriormente e as entradas da rede neural que são as notas musicais dos instrumentos e a amostra que é o padrão utilizado para que a rede neural consiga reconhecer os diferentes tipos de instrumentos musicais utilizado nesse projeto.

```
net = train(net,entradas,amostra);
```

Abaixo está descrito a função *sim* que é utilizada para simular a rede neural já treinada. Onde serão passadas como parâmetros a rede neural treinada e as entradas que são as notas musicais.

```
a = sim(net,entradas);
```

4.5.1 Arquitetura da Rede Neural

Conforme pode ser visto na figura abaixo, a rede possui 7 neurônios na camada de entrada, pois foram usadas as 7 notas musicais no padrão de entrada, 18 neurônios na camada escondida e 1 neurônio na camada de saída.

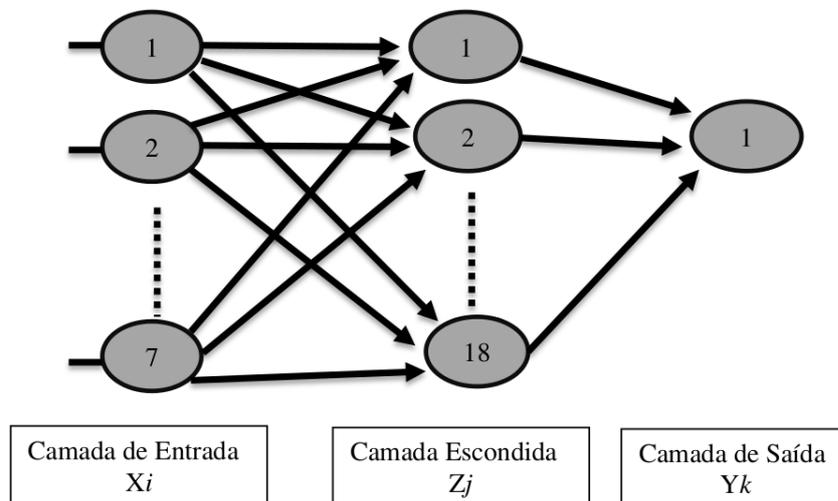


Figura 16 – Arquitetura da Rede Neural

4.6 O Programa Reconhecedor de Timbres

Procurou-se desenvolver uma interface simples e de fácil utilização. Na janela principal do Reconhecedor de Timbres (Figura 4), podem ser preenchidas duas caixas de edição: taxa de aprendizagem (α - alfa) e número de ciclos (épocas), também sendo carregadas as notas musicas do instrumento que queira reconhecer.



Figura 17 – Reconhecedor de Timbres

Após escolhidos a taxa de aprendizagem, número de ciclos e carregadas todas as notas musicais do instrumento que queira reconhecer, o usuário deve clicar no botão “Treinar”. Após o devido treinamento da rede neural então o usuário devera clicar no botão “Reconhecer” para que a figura do instrumento musical reconhecido pela rede neural apareça sob o campo resultados.

5 Testes Realizados

5.1 Introdução

Para verificar a eficiência do reconhecedor de timbres de instrumentos musicais, foram realizados treinamentos diversos variando os valores da taxa de aprendizagem e número de épocas, tendo como parâmetro de resposta a imagem do respectivo instrumento musical reconhecido.

Tabela 1 – Instrumento Trompete

Testes	Taxa de Aprendizagem	Numero de Épocas	Reconheceu
1	0.1	20	SIM
2	0.01	30	SIM
3	0.2	50	SIM
4	0.02	150	SIM

Como vimos na tabela 1 o Trompete foi devidamente reconhecido pela rede neural em todos os testes realizados.

Tabela 2 – Instrumento Violão

Testes	Taxa de Aprendizagem	Numero de Épocas	Reconheceu
1	0.1	20	NÃO
2	0.01	30	NÃO
3	0.2	50	NÃO
4	0.02	150	SIM

Como vimos na tabela 2 o Violão foi devidamente reconhecido pela rede neural no teste 4.

Tabela 3 – Instrumento Flauta

Testes	Taxa de Aprendizagem	Numero de Épocas	Reconheceu
1	0.1	20	SIM
2	0.01	30	SIM
3	0.2	50	SIM
4	0.02	150	SIM

Como vimos na tabela 3 a Flauta foi devidamente reconhecido pela rede neural em todos os testes realizados.

Tabela 4 – Instrumento Piano

Testes	Taxa de Aprendizagem	Numero de Épocas	Reconheceu
1	0.1	20	SIM
2	0.01	30	SIM
3	0.2	50	SIM
4	0.02	150	SIM

Como vimos na tabela 4 o Piano foi devidamente reconhecido pela rede neural em todos os testes realizados.

5.2 Conclusão

Com isso conclui-se que a aplicação desenvolvida atingiu com sucesso o seu objetivo, o reconhecimento dos quatro instrumentos musicais (Flauta, Piano, Violão e Trompete), utilizando as notas musicais (Dó, Ré, Mi, Fá, Sol, Lá e Si).

6 Conclusão

O desafio de integrar técnicas de processamento digital de sinais às tecnologias da informação e física da música foi a motivação deste trabalho, que teve resultado o desenvolvimento de um Sistema de Reconhecimento de Instrumentos Musicais.

Neste trabalho foram abordados de forma sucinta, a teoria musical necessária para a compreensão do funcionamento do Sistema. Adicionalmente, foram explicados os conceitos da física da música e aplicação da transformada rápida de Fourier (FFT) ao problema proposto. E por fim toda parte que integra as redes neurais artificiais desde seus pontos históricos às suas arquiteturas e suas utilizações para resoluções de diferentes problemas em diferentes áreas.

Depois do embasamento teórico estar fundamentado, foi explicado o funcionamento do Sistema e os detalhes da sua implementação. Em seguida, todos os testes realizados foram descritos e analisados.

A transformada rápida de Fourier, utilizada como ferramenta na transformação das amostras de áudio do domínio do tempo para a frequência, atendeu às expectativas, pois se mostrou bastante rápida e eficiente na execução desta tarefa.

A interação das técnicas de redes neurais artificiais, da tecnologia da informação e da física da música amplia enormemente as possibilidades de novas experiências e *softwares* que certamente poderão contribuir para o desenvolvimento de novos trabalhos acadêmicos.

Conforme esperado, com a simplicidade do teste efetuado, a rede neural artificial conseguiu reconhecer 100% dos padrões treinados a partir de uma quantidade específica de ciclos de treinamento – neste caso, a partir de 150 ciclos.

Acredita-se que a normalização, executada na atividade de pré-processamento, desempenhou um papel importante no reconhecimento dos padrões, contribuindo para eficácia do processo de treinamento.

Diante dos resultados obtidos concluiu-se que é possível realizar o reconhecimento de instrumentos musicais através dos seus respectivos timbres musicais, utilizando técnicas de redes neurais artificiais, através de um Sistema Especialista totalmente desenvolvido em ambiente *MatLab*.

6.1 Trabalhos Futuros

Propõem-se, como trabalhos futuros:

- O desenvolvimento de uma rede neural artificial capaz de reconhecer instrumentos musi-

cais a partir de qualquer nota musical;

- O desenvolvimento de uma rede neural capaz de reconhecer instrumentos distintos em um único arquivo de som;
- O desenvolvimento de uma rede neural artificial capaz de reconhecer em tempo real qualquer instrumento musical apresentado para a mesma através de uma nota específica ou mesmo de um acorde.

Muitos esforços ainda serão dispensados em testes e pesquisas sobre o reconhecimento de áudio e timbre musical. Este trabalho é uma humilde contribuição, apenas para sinalizar um caminho ou mostrar outros caminhos que podem ser percorridos para o sucesso completo do projeto.

Referências

- AMBROSIO, P. E.; RODRIGUES, J. A. H.; MARQUES, P. M. A. Sistema computacional de apoio ao diagnóstico de lesões intersticiais pulmonares baseado em redes neurais artificiais. *Anais do CBEB*, 2000.
- BARONE, Dantas Augusto Couto. Projeto revox. *Versão eletrônica em <http://www.usc.tche.br/revox>*, 1999.
- BISHOP, C. M. *Neural networks for pattern recognition*. 1995.
- BOURLAND, H. A. *How Connectionist Models Could Improve Markov Models for Speech Recognition*. Amsterdam: North-Holland: Advanced Neural Computers, 1990. 247-254 p.
- BRAGA, Antônio de Pádua; CARVALHO, André Ponce Ferreira; LUDERMIR, Teresa Bernada. *Redes Neurais Artificiais Teoria e Aplicações*. Rio de Janeiro: Editora LTC – Livros Técnicos e Científicos: trad. Paulo Martins Engel., 2000.
- CHAPMAN, Stephen J. *Programação em MATLAB para engenheiro*. São Paulo: Thomson Learning: tradução técnica Flávio Soares Correa da Silva, 2006.
- CHELLAPILLA, K.; FOGEL, D. Evolution, neural networks, games, and intelligence. *Proc. IEEE*, v. 87, p. 1471–1496, 1999.
- CHELLAPILLA, K.; FOGEL, D. Evolving neural networks to play checkers without expert knowledge. *IEEE Trans. Neural Networks*, v. 10, p. 1382–1391, 1999.
- CHELLAPILLA, K.; FOGEL, D. Anaconda defeats hoyle 6-0: “a case study competing an evolved checkers program against commercially available software”. *Congress on Evolutionary Computation 2000 (CEC’00)*, p. 857–863, 2000.
- CONTRELL, M.; GIRARD, B.; MULLER, C. Neural modeling for time series: A statistical stepwise method for weight elimination. *IEE Transaction on Neural Networks*, v. 6, p. 1355–1364, 1995.
- DATA and Analysis Center for Software. *Artificial Neural Networks Technology*, Disponível em <<http://www.dacs.dtic.mil>> Acesso 01 de Março de 2012, 2012.
- EL-SHARKAWI, M. A.; DAMBORG, M. J.; BRACE, C. M. Short term electric load forecasting using an adaptive trained layered perceptron. *In Proc. Of the Frist Forum on Application of Neural Networks to Powers Systems*, p. 3–6, 1991.
- GUYON, I. Applications of neural networks to character recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 5, p. 353–382, 1991.
- HAYKIN, Simon. *Redes Neurais: Princípios e prática / Simon Haykin*. 2. ed. Porto Alegre: Bookman: trad. Paulo Martins Engel., 2001.

HEMMER, M. C.; GASTEIGER, J. Prediction of three-dimensional molecular structures using information from infrared spectra. *Anal. Chim. Acta*, p. 145–154, 2000.

HOPTROFF, R. G.; BRAMSON, M. J.; HAL, T. J. Forecasting economic turning points with neural network. *in Proc. Of the IEEE/International Joint Conference of Neural Networks*, v. 1, p. 347–352, 1991.

HOSKINS, J. C.; HIMMELBLAU, D. M. Fault diagnosis in complex chemical plants using artificial neural networks. *American Institute of Chemical Engineering Journal*, v. 37, p. 137–141, 1991.

HWANG, J. L.; MONN, S. Temporal difference method for multi-step prediction: Application to power load forecasting. *In Proc. Of the First Forum on Application of Neural Networks to Powers Systems*, p. 41–45, 1991.

JACKEL, L. D.; BOSER, B.; DENKER, J. S. Hardware requirements for neural net optical character recognition. *International Joint Conference on Neural Networks*, San Diego, CA, v. 2, p. 855–861, 1990.

KELLER, P. E.; KANGAS, L. J.; ALLEN, P. A. A novel approach to modeling and diagnosing the cardiovascular system. *World Congress on Neural Networks (WCNN'95)*, Washington EUA, p. 17–21, 1995.

KHALID, M.; OMATU, S. A neural network controller for a temperature control system. *IEEE Control Systems*, p. 58–64, 1992.

KINOSHITA, S. K.; MARQUES, P. M. A.; FRERE, A. F. Utilização da análise de componentes principais e redes neurais artificiais para a classificação de nódulos em mamogramas digitalizados. *Anais do CBEB*, 2000.

KOSTKA, T.; SELZER, P.; GASTEIGER, J. A combined application of reaction prediction and infrared spectra simulation for the identification of degradation products of s-triazine herbicides. *Chem. European Journal*, v. 7, p. 2254–2260, 2001.

LACHTERMARCHER, G.; FULLER, J. D. Backpropagation in time-series forecasting. *Journal of Forecasting*, v. 14, p. 381–393, 1995.

LAPEDES, A.; FARBER, R. Nonlinear signal processing using neural network: Prediction and system modelling. *Technical Report LA-Ur-and-2662*, Los Alamos National National Laboratory, 1987.

LECUN, Y.; BOSER, B.; JACKEL, L. D. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, San Mateo, CA, Morgan Kaufmann, v. 2, p. 598–605, 1990.

LIPPMANN, R. P. Review of neural networks for speech recognition. *Neural Computation* 1, p. 1–38, 1989.

MORGAN, N.; BOURLARD, H. *Continuous Speech Recognition Using Multilayer Perceptrons With Hidden Markov Models*. Albuquerque, NM: International Conference on Acoustics Speech, and Signal Processing, 1990. 413-416 p.

- PENG, T. M.; HUBELE, H. F.; KARADY, G. G. Advancement in the application of neural network for short-term load forecasting. *in Proc of IEEE/ Power Engineering Society Summer Meeting, 91 SM 451-5 PWRS, and tin Transactions on Power Systems*, v. 7, 1992.
- RENALS, S.; MORGAN, N.; BOULARD, H. Improving statistical speech recognition. *International Joint Conference on Neural Networks Baltimore, MD*, v. 2, p. 302–307, 1992.
- RIPLEY, B. D.; HJORT, N. L. Pattern recognition neural networks. 1995.
- SANTOS, P. P.; ALMEIDA, R. M. V. R.; GISMONDI, R. C. Redes neurais artificiais e algoritmos genéticos no diagnóstico diferencial de anemias carenciais. *Anais do CBEB*, 2000.
- SHARDA, R.; PATIL, R. Neural network as forecasting experts: A empirical test. *in International Joint Conference on Neural Networks Washington*, v. 1, p. 441–494, 1990.
- SIMARD, P.; VICTORRI, B.; DENKER, J. Tangent prop-a formalism for specifying selected invariances in an adaptive network. *Advances in Neural Information Processing Systems*, San Mateo, CA, Morgan Kaufmann, v. 4, p. 895–903, 1992.
- SIMON, V.; GASTEIGER, J.; ZUPAN, J. A combined application of two different neural network types for the prediction of chemical reactivity. *J. Am. Chem. Soc.*, p. 9148–9159, 1993.
- SMITH, Steve W. *RThe scientist and engineer's guide to digital signal processing*. 2. ed. California Technical Publishing, California: [s.n.], 1999.
- SOUSA, J. A.; GASTEIGER, J. New description of molecular chirality and its application to the prediction of the preferred enantiomer in stereoselective reactions. *J. Chem. Inf. Comput. Sci*, p. 369–375, 2001.
- SOUSA, J. A.; GASTEIGER, J. Prediction of enantiomeric selectivityin chromatography. application of conformation-dependent and conformation-independent descriptors of molecular chirality. *Journal Mol. Graphics and Model*, v. 20, p. 373–388, 2002.
- SOUSA, J. A.; HEMMER, M.; GASTEIGER, J. Prediction of 1h nmr chemical shifts using neural networks. *Anal. Chem*, p. 80–90, 2002.
- SRINIVASAN, D.; LIEW, A. C.; CHEN, J. S. P. Short term forecasting using neural network approach. *In Proc. Of the Frist Forum on Application of Neural Networks to Powers Systems*, p. 12–19, 1991.
- STEINER, M. T. A.; LEMOS, E. P.; NIEVOLA, J. C. Análise de crédito bancário por meio de redes neurais e Árvores de decisão: uma aplicação simples de data mining. *RAUSP - Revista de Administração da Universidade de São Paulo*, v. 40, 2005.
- TANG, Z.; ALMEIDA, C.; FISHIWICK, P. A. Time series forescasting using neural networks vs box-jenkins methodology. *Simulation* 57, v. 5, p. 303–310, 1991.
- TURNER, D. D. Diagnosing coronary artery disease with a backpropagation neural network. *Masters of Science Thesis, Eastern Washington University Cheney*, Washington EUA, 1994.
- UNGER, L. H.; POWELL, B. A.; KAMENS, S. N. Adaptive networks for fault diagnosis and process control. *Computers in Chemical Engineering*, v. 24, p. 561–572, 1990.
- WAIBEL, A.; LEE, K. Readings in speech recognition. *Morgan Kaufmann*, 1990.

WASSERMAN, P. D. Neural computing: Theory and practice. *Van Nostrand Reinhold*, New York EUA, 1989.

WEIGEND, A. S.; RUMERLHART, D.; HUBERMAN, B. A. Predicting the future: A connectionist approach. *International Journal Neural System*, 1990.

WEIGEND, A. S.; RUMERLHART, D.; HUBERMAN, B. A. Generalization by weight-elimination application to forecasting. *Avance in Neural information Processing Systems*, v. 3, p. 875–882, 1991.

WEIGEND, A. S.; RUMERLHART, D.; HUBERMAN, B. A. Generalization by weight-elimination applied to currency exchange rate prediction. *Proceedings of the IEEE/International Joint Conferenre of Neural Networks*, v. 1, p. 837–841, 1991.

WINDDROW, B.; STEARNS, S. D. Adaptive signal processing. *Prentecce Hall, Upper Saddle River, NJ*, 1985.

ANEXO A – Algoritmo da Criação da Rede Neural

```

function out1 = newff(varargin)
persistent INFO;
if (nargin < 1), nnerr.throw(); end
in1 = varargin{1};
if ischar(in1)
    switch in1
        case 'info',
            if isempty(INFO), INFO = get_info; end
            out1 = INFO;
    end
else
    out1 = create_network(varargin{:});
end

function info = get_info
info.function = mfilename;
info.name = 'Feed-Forward';
info.description = nnfcn.get_mhelp_title(mfilename);
info.type = 'nntype.network_fcn';
info.version = 6.0;

function net = create_network(varargin)
if nargin < 2, nnerr.throw(), end
v1 = varargin{1};
if isa(v1,'cell'), v1 = cell2mat(v1); end
v2 = varargin{2};
if nargin > 2, v3 = varargin{3}; end

if (nargin<= 6) && (size(v1,2)==2) && (~iscell(v2)) && (size(v2,1)==1)
    nnerr.obs_use(mfilename, [upper(mfilename)]);
    net = new_5p0(varargin{:});

```

```

else
    net = new_5p1(varargin{:});
end

%=====
function net = new_5p1(p,t,s,tf,btf,blf,pf,ipf,tpf,ddf)
if nargin < 2, nnerr.throw(), end
if (nargin < 3), s = []; end
if (nargin < 4), tf = {}; end
if (nargin < 5), btf = 'trainlm'; end
if (nargin < 6), blf = 'learngdm'; end
if (nargin < 7), pf = 'mse'; end
if (nargin < 8), ipf = {'fixunknowns','removeconstantrows','mapminmax'}
end
if (nargin < 9), tpf = {'removeconstantrows','mapminmax'}; end
if (nargin < 10), ddf = 'dividerand'; end

if isa(p,'cell'), p = cell2mat(p); end
if isa(t,'cell'), t = cell2mat(t); end

if ~(isa(p,'double') || isreal(p) || islogical(t))
    nnerr.throw()
end
if ~(isa(t,'double') || isreal(t) || islogical(t))
    nnerr.throw()
end
if isa(s,'cell')
    if (size(s,1) ~= 1)
        nnerr.throw()
    end
    for i=1:length(s)
        si = s{i};
        if ~isa(si,'double') || ~isreal(si) || any(size(si) ~= 1)
            || any(si<1) || any(round(si) ~= si)
                nnerr.throw()
        end
    end
end

```

```

    end
    s = cell2mat(s);
end
if (~isa(s,'double')) || ~isreal(s) || (size(s,1) > 1)
|| any(s<1) || any(round(s) ~= s)
    nnerr.throw()
end

Nl = length(s)+1;
net = network;
net.numInputs = 1;
net.numLayers = Nl;
net.biasConnect = ones(Nl,1);
net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
net.layerConnect = (j == (i-1));
net.outputConnect(Nl) = 1;

net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        net.layers{i}.size = s(i);
        if (Nl == 2)
            net.layers{i}.name = 'Hidden Layer';
        else
            net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        net.layers{i}.name = 'Output Layer';
    end
    if (length(tf) < i) || all(isnan(tf{i}))
        if (i<Nl)
            net.layers{i}.transferFcn = 'tansig';
        else
            net.layers{i}.transferFcn = 'purelin';
        end
    end
end

```

```

    else
        net.layers{i}.transferFcn = tf{i};
    end
end
net.outputs{Nl}.processFcns = tpf;

net.adaptFcn = 'adaptwb';
net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    net.biases{i}.learnFcn = blf;
    net.layerWeights{i,:}.learnFcn = blf;
end

net.trainFcn = btf;
net.divideFcn = ddf;
net.performFcn = pf;

net.initFcn = 'initlay';
for i=1:Nl
    net.layers{i}.initFcn = 'initnw';
end

net.inputs{1}.exampleInput = p;
net.outputs{Nl}.exampleOutput = t;
net = init(net);
net.plotFcns = {'plotperform','plottrainstate','plotregression'};
%=====
function net = new_5p0(p,s,tf,btf,blf,pf)
if nargin < 2, nnerr.throw(), end
Nl = length(s);
if nargin < 3, tf = {'tansig'}; tf = tf(ones(1,Nl)); end
if nargin < 4, btf = 'trainlm'; end
if nargin < 5, blf = 'learngdm'; end
if nargin < 6, pf = 'mse'; end

if isa(p,'cell') && all(size(p)==[1 1]), p = p{1,1}; end

```

```

if (~isa(p,'double')) || ~isreal(p)
    nnerr.throw()
end
if isa(s,'cell')
    if (size(s,1) ~= 1)
        nnerr.throw()
    end
    for i=1:length(s)
        si = s{i};
        if ~isa(si,'double') || ~isreal(si) || any(size(si) ~= 1)
            || any(si<1) || any(round(si) ~= si)
                nnerr.throw()
        end
    end
end
s = cell2mat(s);
end
if (~isa(s,'double')) || ~isreal(s) || (size(s,1) ~= 1)
|| any(s<1) || any(round(s) ~= s)
    nnerr.throw()
end

net = network(1,Nl);
net.biasConnect = ones(Nl,1);
net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
net.layerConnect = (j == (i-1));
net.outputConnect(Nl) = 1;

for i=1:Nl
    net.layers{i}.size = s(i);
    net.layers{i}.transferFcn = tf{i};
end
net.performFcn = pf;
net.adaptFcn = 'adaptwb';
net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl

```

```
net.biases{i}.learnFcn = blf;
net.layerWeights{i,:}.learnFcn = blf;
end

net.trainFcn = btf;
net.initFcn = 'initlay';
for i=1:Nl
    net.layers{i}.initFcn = 'initnw';
end
net.inputs{1}.exampleInput = p;
net = init(net);
net.plotFcns = {'plotperform','plottrainstate','plotregression'};
```

ANEXO B – Algoritmo de Treinamento da Rede Neural

```

function [net,tr,out3,out4,out5,out6]=train(net,X,T,Xi,Ai,EW,arg7)

if nargin < 1, nnerr.throw('Not enough input arguments.');
```

end

```

[net,err] = nntype.network('format',net);
if ~isempty(err), nnerr.throw(nnerr.value(err,'NET'));
```

end

```

if isempty(net.trainFcn), nnerr.throw('NET.trainFcn is undefined.');
```

end

```

if nargin < 2, X = {}; end
if nargin < 3, T = {}; end
if nargin < 4, Xi = {}; end
if nargin < 5, Ai = {}; end
if nargin < 6, EW = 1; end

if (nargin == 6) && (isstruct(EW))
    [net,tr,out3,out4,out5,out6] = v51_train_arg6(net,X,T,Xi,Ai,EW);
    return
elseif (nargin == 7) && (isstruct(EW) || isstruct(arg7))
    [net,tr,out3,out4,out5,out6] = v51_train_arg7(net,X,T,Xi,Ai,EW,arg7);
    return
end

[net,tr] = feval(net.trainFcn,net,X,T,Xi,Ai,EW,net.trainParam);

if nargout > 2
    [out3,out5,out6] = sim(net,X,Xi,Ai,T);
    out4 = gsubtract(T,out3);
end
```

ANEXO C – Algoritmo do Programa

Reconhecedor de Timbres

```

function varargout = reconhecedor(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @reconhecedor_OpeningFcn, ...
                  'gui_OutputFcn',  @reconhecedor_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function reconhecedor_OpeningFcn(hObject, eventdata, handles,
varargin)
    axes(handles.imagResul);
    imagem = imread('resultado.jpg');
    axis off;
    imshow(imagem);

handles.output = hObject;

guidata(hObject, handles);

```

```

function varargout = reconhecedor_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function treinar_Callback(hObject, eventdata, handles)
global t;
global nota_do;
global nota_re;
global nota_mi;
global nota_fa;
global nota_sol;
global nota_la;
global nota_si;

t = 0;

net=newff([0 17; 0 3; 0 17; 0 16; 0 8; 0 9; 0 16],[18,1],{'tansig',
'purelin'},
'trainlm');
net.trainParam.epochs = handles.metricdata.numCiclo;
net.trainParam.goal = 0;
net.trainParam.lr = handles.metricdata.taxaApre;

entradas = [nota_do;nota_re;nota_mi;nota_fa;nota_sol;nota_la;nota_si];
amostra = [1 1 2 3 3 0 5 2 0 0 0 0 1 1 0 0];

net = train(net,entradas,amostra);

a = sim(net,entradas);
b = round(a);

for i = 1:16
    if amostra(i) == b(i)
        t = t+1;
    else

```

```
end
end

function reconhecer_Callback(hObject, eventdata, handles)

global t;

if t == 16
    axes(handles.imagResul);
    imagem = imread('flauta.jpg');
    axis off;
    imshow(imagem);

else
    if t == 12
        axes(handles.imagResul);
        imagem = imread('violao.jpg');
        axis off;
        imshow(imagem);

    else
        if t == 6
            axes(handles.imagResul);
            imagem = imread('trompete.jpg');
            axis off;
            imshow(imagem);
        else
            if t == 11
                axes(handles.imagResul);
                imagem = imread('piano.jpg');
                axis off;
                imshow(imagem);
            else
                axes(handles.imagResul);
                imagem = imread('interrogacao.jpg');
```

```

        axis off;
        imshow(imagem);
        end
    end
end

function bntLimpar_Callback(hObject, eventdata, handles)
set(handles.taxaApre,'String','');
set(handles.numCiclo,'String','');

axes(handles.imagResul);
    imagem = imread('resultado.jpg');
    axis off;
    imshow(imagem);

function bntSair_Callback(hObject, eventdata, handles)
close;

function numCiclo_Callback(hObject, eventdata, handles)
numCiclo = str2double(get(hObject,'String'));
handles.metricdata.numCiclo = numCiclo;
guidata(hObject,handles);

function numCiclo_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function taxaApre_Callback(hObject, eventdata, handles)
taxaApre = str2double(get(hObject,'String'));
handles.metricdata.taxaApre = taxaApre;
guidata(hObject,handles);

```

```
function taxaApre_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function teste_Callback(hObject, eventdata, handles)
```

```
function teste_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Car_Re_Callback(hObject, eventdata, handles)
global nota_re;
```

```
[filename, pathname] = uigetfile('*.wav','Select Data File');
[nota_re, fs] = wavread([pathname filename]);
nota_re = nota_re/max(abs(nota_re));
nota_re = abs(fft(nota_re));
nota_re = round(nota_re);
nota_re = nota_re(1:16);
nota_re = nota_re';
```

```
function Car_Mi_Callback(hObject, eventdata, handles)
global nota_mi;
```

```
[filename, pathname] = uigetfile('*.wav','Select Data File');
[nota_mi, fs] = wavread([pathname filename]);
nota_mi = nota_mi/max(abs(nota_mi));
nota_mi = abs(fft(nota_mi));
nota_mi = round(nota_mi);
```

```
nota_mi = nota_mi(1:16);  
nota_mi = nota_mi';
```

```
function Car_Do_Callback(hObject, eventdata, handles)  
global nota_do;  
  
[filename, pathname] = uigetfile('*.wav','Select Data File');  
[nota_do, fs] = wavread([pathname filename]);  
nota_do = nota_do/max(abs(nota_do));  
nota_do = abs(fft(nota_do));  
nota_do = round(nota_do);  
nota_do = nota_do(1:16);  
nota_do = nota_do';
```

```
function Car_Fa_Callback(hObject, eventdata, handles)  
global nota_fa;  
  
[filename, pathname] = uigetfile('*.wav','Select Data File');  
[nota_fa, fs] = wavread([pathname filename]);  
nota_fa = nota_fa/max(abs(nota_fa));  
nota_fa = abs(fft(nota_fa));  
nota_fa = round(nota_fa);  
nota_fa = nota_fa(1:16);  
nota_fa = nota_fa';
```

```
function Car_Sol_Callback(hObject, eventdata, handles)  
global nota_sol;  
  
[filename, pathname] = uigetfile('*.wav','Select Data File');  
[nota_sol, fs] = wavread([pathname filename]);  
nota_sol = nota_sol/max(abs(nota_sol));  
nota_sol = abs(fft(nota_sol));  
nota_sol = round(nota_sol);  
nota_sol = nota_sol(1:16);
```

```
nota_sol = nota_sol';
```

```
function Car_La_Callback(hObject, eventdata, handles)
```

```
global nota_la;
```

```
[filename, pathname] = uigetfile('*.wav','Select Data File');
```

```
[nota_la, fs] = wavread([pathname filename]);
```

```
nota_la = nota_la/max(abs(nota_la));
```

```
nota_la = abs(fft(nota_la));
```

```
nota_la = round(nota_la);
```

```
nota_la = nota_la(1:16);
```

```
nota_la = nota_la';
```

```
function Car_Si_Callback(hObject, eventdata, handles)
```

```
global nota_si;
```

```
[filename, pathname] = uigetfile('*.wav','Select Data File');
```

```
[nota_si, fs] = wavread([pathname filename]);
```

```
nota_si = nota_si/max(abs(nota_si));
```

```
nota_si = abs(fft(nota_si));
```

```
nota_si = round(nota_si);
```

```
nota_si = nota_si(1:16);
```

```
nota_si = nota_si';
```