



UNIVERSIDADE FEDERAL DO PARÁ

CAMPUS UNIVERSITÁRIO DE CASTANHAL

FACULDADE DE COMPUTAÇÃO

ANTONIO RAIMUNDO ALVES ANDRADE

CLASSIFICAÇÃO DE GÊNEROS MÚSICAIS UTILIZANDO APRENDIZADO DE MÁQUINA

CASTANHAL

2018

ANTONIO RAIMUNDO ALVES ANDRADE

**CLASSIFICAÇÃO DE GÊNEROS MUSICAIS UTILIZANDO APRENDIZADO DE
MÁQUINA**

Trabalho de conclusão de curso apresentado à
Faculdade de Sistemas de Informação da
Universidade Federal do Pará como requisito
parcial para obtenção de grau de graduado

Orientador: Prof. Igor Ruiz

CASTANHAL

2018

*“Um espírito nobre engrandece o
menor dos homens.”*

Jebediah Springfield

AGRADECIMENTOS

Chegar aqui não foi fácil, um jovem de família humilde que um dia sonhou em cursar e concluir um curso superior em uma instituição federal, as chances de alcançar esse sonho não eram boas principalmente por ter estudado a vida toda em escola pública que na maioria das vezes não dá um ensino adequado preparando-o para o ensino superior, mas acima de tudo gostaria de agradecer a Deus que em meio as dificuldades me susteve, me deu a força e sabedoria para poder continuar acreditando nos meus sonhos. Agradeço também aos meus pais que não só me educaram desde criança como também me ensinaram a ter caráter, me ensinaram a ser quem eu sou e como eu sou, me deram força, motivação e estiveram sempre ao meu lado me ajudando no que fosse preciso para que eu pudesse priorizar os meus estudos e alcançar esta conquista. Agradeço aos meus familiares em geral que de alguma forma contribuíram para que esse momento fosse possível. Agradeço a cada um dos meus amigos da faculdade, amigos de escola e amigos do trabalho, todos aqueles que estiveram do meu lado, em especial eu gostaria de agradecer a equipe “JEDI” que se formou de uma forma inesperada depois de poucos dias de curso, que era composta por Cledson, Samuel, Neilson e por mim. Esse grupo de quatro pessoas foi muito importante durante todo o decorrer do curso, me ajudaram diversas vezes, e o mais importante é que eramos unidos suprindo as necessidades ou deficiência em alguma disciplina um do outro, e por isso gostaria muito de agradecer a essas pessoas que foram muito importantes não só na faculdade como também na minha vida. Quero agradecer a cada um dos professores que me ajudaram nessa reta final na conclusão do meu TCC, ao meu orientador Prof^o Igor Ruiz, a Prof^a Yomara Pires que me ajudou bastante na construção do TCC, e ao Prof^o Tassio que sempre se disponibilizou a esclarecer minhas dúvidas e a ajudar todos os alunos do curso. E por fim quero agradecer a Instituição UFPA como um todo e dizer que nunca devemos desistir de nossos sonhos.

RESUMO

A classificação automática de gêneros musicais é uma tarefa de crescente interesse, sobretudo devido à enorme quantidade de músicas disponíveis. Com gêneros devidamente classificados, torna-se possível a busca por gênero, facilitando a manutenção de acervos musicais. Entretanto, é consenso na literatura que muito ainda pode e deve ser feito, pois, por exemplo, ainda não se chegou a um conjunto de parâmetros que extraídos de uma faixa musical possibilite a melhor classificação possível. Por isso, neste trabalho aplicamos as Redes Neurais Artificiais ao problema de detecção automática de gêneros musicais. Os resultados encontrados acompanham os resultados de outros trabalhos disponíveis na literatura, mostrando que a técnica se adapta ao problema, mesmo diante da sua alta dimensionalidade.

Palavras-chave: Gêneros Musicais, Aprendizado de Máquina, Redes Neurais Artificiais

LISTA DE FIGURAS

Figura 1 - Matriz de Confusão e Acurácia.....	4
Figura 2 – Estrutura genérica de rede neurônios.....	6
Figura 3– Modelo não-linear de um neurônio.....	9
Figura 4– Representação do efeito bias no controle sobre o sinal de entrada líquida na função de ativação.....	10
Figura 5 – Algumas das funções de ativação mais utilizadas.....	11
Figura 6 – Representação de uma rede monocamada <i>feedforward</i>	13
Figura 7– Rede MLP com duas camadas intermediárias.....	14
Figura 8 – Representação de um modelo simples de aprendizagem.....	16
Figura 9 – Operação do algoritmo Backpropagation.....	18
Figura 10 – Operação do algoritmo Backpropagation em modo progressivo (<i>feedforward</i>).	18
Figura 11 - Arquitetura das Redes Neurais utilizadas.....	21
Figura 12 - Matriz de confusão de uma RNA usando 10-fold <i>Cross validation</i>	23
Figura 13 - Curva de desempenho de uma RNA.....	24
Figura 14 - Histograma de erro quadrático médio.....	24
Figura 15 - Quantidade de Erros em treino e teste dos modelos.....	26

LISTA DE TABELAS

Tabela 1 - Eficiência de Classificação.....	4
Tabela 2 - Exemplo de Matriz de Confusão.....	22
Tabela 3 - Taxa de Erro em Porcentagem.....	25
Tabela 4 - Matriz de Confusão das Redes Neurais Artificiais.....	26

LISTA DE SÍMBOLOS E SIGLAS

B – matriz susceptância.

b_k – bias.

ΔW_{ki} – variação dos pesos sinápticos.

ε – tolerância de erro.

φ(.) ou **fa** – função de ativação.

H, M, N e P – submatrizes da matriz Jacobiana.

IA – inteligência artificial.

IEEE – instituto de engenheiros eletrônicos e eletricitistas
(*instituteofelectricalandelectronicsengineers*).

I_{km} – corrente no ramo k-m.

J – matriz Jacobiana.

mse – erro médio quadrático (*meansquarederror*).

RNA – rede neural artificial.

σ – vetor raiz.

u_k – resultante de um combinador linear.

v_k – potencial de ativação.

W_{ki} – pesos sinápticos.

x_i – entrada do neurônio.

Y – matriz admitância.

y_k – saída do neurônio.km

SUMÁRIO

RESUMO.....	I
LISTA DE FIGURAS.....	II
LISTA DE TABELAS.....	III
LISTA DE SÍMBOLOS E SIGLAS.....	IV
1 INTRODUÇÃO.....	1
1.1 – JUSTIFICATIVA.....	2
1.2 – OBJETIVO GERAL.....	3
1.3 – OBJETIVOS ESPECÍFICOS.....	3
2 TRABALHOS CORRELATOS.....	4
2.1 – <i>MUSIC INFORMATION RETRIEVAL IN MUSIC REPOSITORIES</i>	4
2.2 – <i>CONTEST: MUSIC INFORMATION RETRIEVAL</i>	5
3 REDES NEURAIS ARTIFICIAIS.....	6
3.1 – INTRODUÇÃO.....	6
3.2 – MODELO MATEMÁTICO DE UM NEURÔNIO.....	9
3.3 – ARQUITETURA DE UMA REDE.....	12
3.4 – APRENDIZAGEM.....	15
3.4.1 – Algoritmo <i>Backpropation</i>	17
4 RESULTADOS OBTIDOS.....	19
4.1 – CONJUNTO DE DADOS.....	19
4.2 – EXPERIMENTOS REALIZADOS.....	20
4.3 – TÉCNICAS DE AVALIAÇÃO DOS RESULTADOS.....	21
4.4 – RESULTADOS OBTIDOS.....	22
5 CONCLUSÃO.....	28
REFERÊNCIAS.....	29

1 Introdução

Para o ser humano é trivial o reconhecimento de sons devido a milênios de evolução do aparelho auditivo e do cérebro. Treinamos nossos ouvidos para reconhecer sons que dependendo da ocasião seriam determinantes para nossa sobrevivência.

A partir do momento em que o ser humano começou a se estabelecer em tribos pós-período neolítico, nossos sentidos passaram a ser usados para tarefas mais construtivas do que simplesmente fugir e caçar. Criamos costumes, crenças e culturas que nos diferenciavam ou destacavam entre outras tribos transferindo esses axiomas para as gerações que se seguiram até hoje. Esta consciência habilitou a complexidade das culturas e da comunicação, que foram aprimoradas ramificando-se para novos estilos de expressão de opiniões e crenças. Dentre elas a forma de expressão ritmo artística denominada música.

A música faz parte do cotidiano da maioria das pessoas. Cada indivíduo possui um gosto particular, ainda que seja parecido com o gosto de outras pessoas. Esse gosto normalmente é definido pelo ritmo da música, também chamado de gênero ou estilo musical. Essa definição de estilo musical parece simples, mas na realidade dependem de vários atributos, condições e características que aprendemos a distinguir de forma sutil, ainda que muitos estudiosos se dediquem a estudar profundamente os diversos gêneros musicais.

Aprender a diferenciar gêneros musicais é trivial para o ser humano seja ele especialista ou não. Entretanto, ensinar o computador a distinguir gêneros musicais é uma tarefa extremamente complexa, mesmo diante dos avanços tecnológicos mais recentes. Nesse sentido, este trabalho se propõe a utilizar as redes neurais artificiais para detectar automaticamente (classificar) gêneros musicais, a partir de estudos realizados previamente e disponibilizados pela comunidade científica.

Na sessão 2 deste trabalho apresenta-se um breve estudo sobre o problema e o estado da arte em classificação de gêneros musicais, em seguida na sessão 3, abordaremos o conceito de Redes Neurais Artificiais e o algoritmo de treinamento *Backpropagation*. Na sessão 4 será descrito os experimentos realizados e é detalhado o conjunto de dados elaborado pelo Departamento de Sistemas Multimídia da Universidade Técnica de Gdańsk. Também se apresenta o ambiente em que ocorreram os experimentos, por fim temos as conclusões alcançadas e sugestões de trabalhos futuros que são apresentados na sessão 5.

1.1 – Justificativa

Hoje comercial, a música existe há muito como o resultado de nossas opiniões, crenças e gostos. Muitas ideias foram transcritas melodicamente ao longo dos séculos.

As músicas são classificadas em gêneros, conforme o seu ritmo. Essa classificação parece simples, mas na realidade ela depende de vários atributos. Sabemos que aprender a diferenciar gêneros musicais é trivial para o ser humano seja ele especialista ou não.

Entretanto, ensinar uma máquina a "ouvir" uma trilha e determinar o gênero musical, com taxa de acerto semelhante, ou melhor, a de um ser humano, é uma tarefa complexa, e por isso têm sido amplamente investigada pela comunidade científica. Daí a necessidade de técnicas mais elaboradas para permitir que o computador realize essa tarefa.

Dentre as dificuldades, pode-se listar:

- Não existe um conjunto de atributos ideal: os trabalhos que investigam esta área de pesquisa relatam que ainda não se alcançou um conjunto de atributos ideal para representar trechos das músicas, assim, cada autor utiliza variações dos dados de trabalhos anteriores;

- Alta dimensionalidade dos dados: são utilizadas técnicas de extração de características para obter um conjunto de dados menor e com menos atributos, visando diminuir a complexidade dos modelos gerados;

- Não linearidade dos dados: as amostras geradas que pertencem ao mesmo gênero apresentam alta variação em seus atributos, dificultando a tarefa de classificação;

- Soluções proprietárias do tipo caixa pretas: diversas empresas possuem soluções para classificação de gêneros musicais, porém, são soluções proprietárias e pouco se conhece sobre elas;

- As melhores soluções atuais são complexas, utilizando *ensembles* de classificadores: trabalhos encontrados na revisão da literatura, os que alcançam melhores taxas de acerto são os que utilizam esquemas de classificação mais complexos, inclusive utilizando vários classificadores em paralelo e realizando votação para classificar as amostras;

As redes neurais artificiais do tipo Perceptron de Múltiplas Camadas (MLP) foram escolhidas para realizar a classificação de gêneros musicais pelos motivos abaixo:

- Este modelo é considerado um aproximador universal de funções, portanto pode ser utilizado para realizar o mapeamento entre trecho da música e o seu gênero;

- É utilizado amplamente na literatura como um modelo para classificação, porém, não foi encontrado um trabalho que utilize estas redes na tarefa de classificar gêneros musicais;

- As MLP são ideais para classificar conjuntos de dados com atributos numéricos, como é o caso dos dados de trechos de músicas utilizado neste trabalho e conforme encontrado na revisão da literatura;

Trabalhos recentes recomendam o treino de redes MLP com a variante de Levenberg-Marquardt do algoritmo *Backpropagation* (retropropagação do erro), pois é a variante que otimiza o tempo de convergência para ótimos globais (Haykin).

1.2 – Objetivos Gerais

O objetivo geral deste trabalho é aplicar Redes Neurais Artificiais ao problema de classificação automática de gêneros musicais.

1.3 – Objetivos Específicos

Dividir um conjunto de dados em seis gêneros musicais: *Blues*, *Clássica*, *Jazz*, *Metal*, *Pop* e *Rock*. Em seguida treinar uma rede neural com esse conjunto de dados (dados esses que foram disponibilizados pelo Departamento de Sistemas Multimídia da Universidade de Tecnologia de Gdansk) utilizando o Matlab e avaliar o desempenho da rede.

2 Trabalhos Correlatos

2.1 – Music Information Retrieval in Music Repositories

Este trabalho foi desenvolvido por Bozema Kostek e publicado em 2013 (KOSTEK, 2013), apresenta uma revisão de conceitos chaves associados à recuperação de informações musicais de forma automatizada. Utiliza o conjunto de dados criado e disponibilizado pelo Departamento de Sistemas Multimídia da Universidade Técnica de Gdansk.

Foram utilizadas 50% das amostras para construção do modelo e o restante para teste. Utilizou também os algoritmos Máquinas de Vetores de Suporte (SVM), árvores de decisão J48 e florestas aleatórias, ambos no WEKA, e *rough set-based analysis*. A Tabela 1 apresenta a eficiência de classificação obtida e a Figura 1 apresenta a matriz de confusão obtida para os dados de teste pelo trabalho em questão.

Tabela 1 - Eficiência de Classificação

Classificador	Eficiência (%)
SVM	90.87
J48	77.40
Florestas Aleatórias	84.72
<i>Rough Sets</i>	78.80

Fonte: Adaptada de KOSTEK, 2013

Figura 1 - Matriz de Confusão e Acurácia

	Classical	Jazz	Rock	Blues	Metal	Pop	No. of objects	Accuracy
Classical	2169	159	5	39	0	8	2380	0.911
Jazz	572	2110	19	149	15	95	2960	0.713
Rock	18	106	1097	49	129	141	1540	0.712
Blues	5	21	27	726	33	148	960	0.756
Metal	3	13	63	40	545	56	720	0.757
Pop	6	34	73	15	20	1012	1160	0.872
True positive rate	0.78	0.78	0.78	0.78	0.78	0.78		

Fonte: KOSTEK, 2013

2.2 - Contest: Music Information Retrieval

Este trabalho apresenta um relatório obtido a partir dos resultados da competição ocorrida no XIX Simpósio de Metodologias para Sistemas Inteligentes, realizado entre 10 de janeiro e 21 de março no ano de 2011 (KOSTEK et al., 2011). Um dos temas abordados nesta competição foi reconhecimento automático de gêneros musicais. Também utilizou o conjunto de dados e vetores de características preparados e disponibilizados pelo Departamento de Sistemas Multimídia da Universidade de Tecnologia de Gdansk. A equipe vencedora alcançou resultados semelhantes ao obtido em Kostek et al., 2013.

Este trabalho utiliza o mesmo conjunto de dados, porém aplica a técnica de Redes Neurais Artificiais, do tipo Perceptron de múltiplas camadas ao problema de classificação de gêneros musicais, buscando obter resultados melhores.

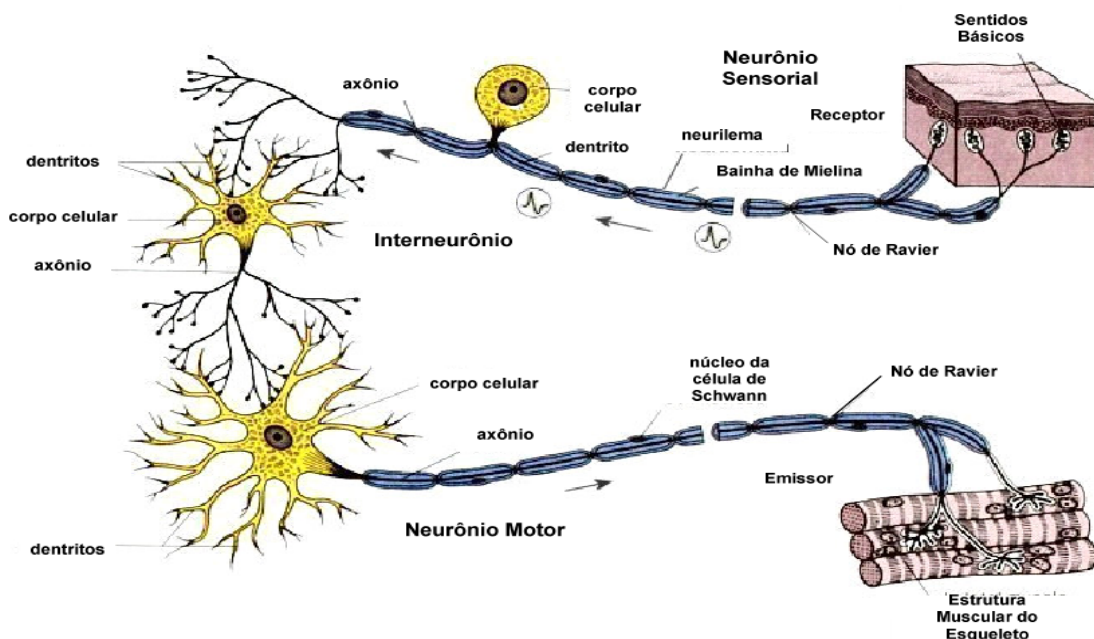
3 Redes Neurais Artificiais

3.1 – Introdução

O comportamento do ser humano apresenta, em meio às complexidades de seu comportamento cerebral, a característica de aprender a lidar diariamente com novas tarefas que necessitam de decisões rápidas e inteligentes para alcançar o melhor rendimento possível em novos desafios. Estas decisões são tomadas a partir de um processo de comunicação paralelas e distribuídas entre os neurônios: sinapses cerebrais. Estas permitem que a atividade neural auxilie o homem em suas habilidades não só quanto ao processo de decidir, mas também quanto ao de aprender e armazenar as informações necessárias para o mesmo.

A rede de neurônios, exemplificada na Figura 2, presente na camada externa do cérebro (córtex), viabiliza a capacidade cognitiva que o ser humano recorre a cada novo conhecimento ou a cada nova experiência que, por sua vez, pode implicar em novas informações e estruturas cerebrais. Estes rearranjos, na estrutura da rede de neurônios, acontecem por meio de uma possível necessidade de intensificação ou inibição de algumas sinapses a fim de melhorar os critérios da atividade neural para se atingir seus objetivos (HAYKIN, 2001).

Figura 2 – Estrutura genérica de rede neurônios.



Fonte: MÜLLER, 2008.

Nesta comunicação entre neurônios, a sinapse é, em geral, considerada como um processo químico de envio e recepção de informações, obtido a partir de impulsos elétricos sob determinadas condições nas quais, basicamente, os dendritos são responsáveis pela transmissão de “dados” entre neurônios. O corpo celular se encarrega do “tratamento do sinal recebido”, classificando-o em seu correspondente nível de excitação. E, de acordo com esta classificação, uma nova resposta será transmitida por meio do axônio que, por sua vez, faz a comunicação com a estrutura muscular (emissor) ou com a estrutura sensorial-cerebral (receptor) (HAYKIN, 2001).

Assim, a partir desta capacidade humana, várias modelagens computacionais foram estudadas para que se alcançasse também maior eficiência nos processos de tomadas de decisões em grande escala (BRAGA et al. 2000). Além disso, as pesquisas também proporcionaram, cada vez mais, novos testes para estas conexões de alta capacidade de aprendizagem por meio de redes, surgindo, assim, as redes neurais artificiais, inspiradas por mecanismos da inteligência biológica para se beneficiar da utilização da inteligência artificial (IA) nas diversas áreas de tecnologia.

Os estudos começaram a ser desenvolvidos a partir do primeiro modelo de neurônio proposto por McCulloch e Pitts em 1943. Outro modelo foi desenvolvido em 1949 por Hebb, enquanto que Frank Rosenblatt avançou ainda mais as pesquisas com um novo modelo “Perceptron”, proposto em 1958. Entretanto, após uma publicação de Minsky e Papert, que criticavam este modelo, muitos autores atenuaram suas pesquisas, enfraquecendo novos avanços no ramo. Após esse período, o interesse na área de IA só retornou (e de forma decisiva) após pesquisas relacionadas ao campo de aprendizado com *Backpropagation* publicadas por Hopfield, Rumelhart, Hinton e Williams em 1982 (MINSKY e PAPERT, 1969).

Assim, foi incentivada uma série de novas pesquisas, reduzindo falhas na aprendizagem, possíveis ruídos nas conexões das RNA's, além ampliar o campo de implementação desta inteligência avançada. Tudo isso foi motivado pela ideia de que o cérebro humano é limitado diante de processos que requerem grande quantidade de informações a serem tratadas em um tempo cada vez mais curto. Isso, então, passa a ser usado pelas RNA's a fim de obter maior organização e eficiência neste sentido, permitindo que uma estrutura neural aprenda novas operações e tarefas, como reconhecer novos padrões, percepções e controle de um sistema mais rapidamente, aproveitando-se também da complexidade de multitarefas que o computador é apto a operar simultaneamente (HAYKIN, 2001).

Uma destas áreas que recebeu influências desse desenvolvimento computacional de RNA's foram os Sistemas Elétricos de Potência (SEP's). Isto ocorreu a partir do avanço de pesquisas desenvolvidas com sucesso nessa área, constatadas por meio de novas publicações de pesquisas que, ao longo dos anos, favoreceram a confiabilidade deste recurso em diversos sistemas de distribuição de energia. Este avanço possibilitou, então, cada vez mais, o uso de RNA's na solução de problemas complexos, como o cálculo de fluxo de carga (HAYKIN, 2001 e MINSKY e PAPERT, 1969). Esta área representará o foco deste trabalho, realizando-se testes de implementação de RNA's neste ramo.

As soluções para os cálculos de fluxo de carga via RNA's devem ser obtidas de forma a minimizar a divergências de valores esperados, admitindo-se, assim, a necessidade de um treinamento da rede a partir de um método padrão capaz de realizar ajustes a cada tentativa de treinamento da rede. Além disso, também se deve adotar um critério para a classificação das respostas geradas como aceitáveis ou esperadas, beneficiando a qualidade da solução final. Esse treinamento pode obter uma aprendizagem de modo supervisionado, quando são conhecidos todos os dados de entradas e os de saída, ou não supervisionado, na qual só se conhecem os dados de entrada, enquanto que os de saída são estimados pela rede. De qualquer modo, o processo de tentativas de solução continua até que o sistema atinja um bom desempenho, influenciando de forma positiva na qualidade de uma saída de rede neural implementada na obtenção de um cálculo de fluxo de carga (MINSKY e PAPERT, 1969).

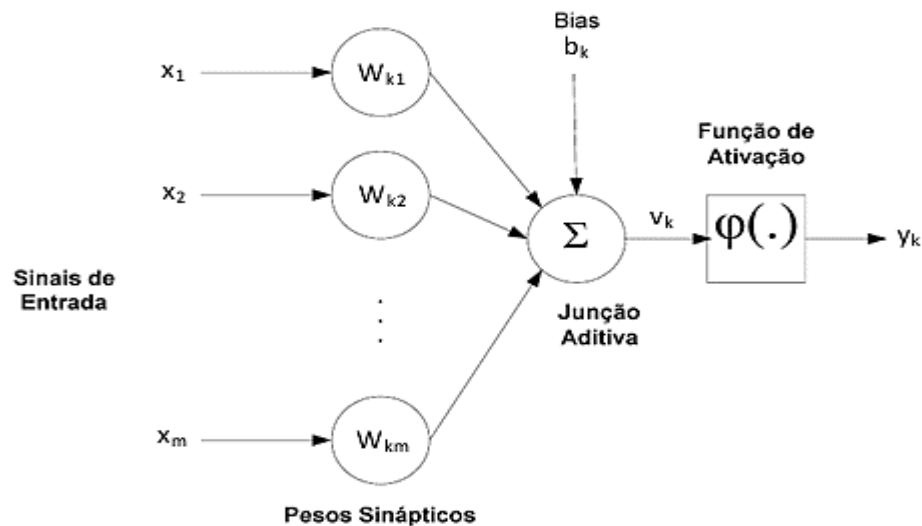
As tentativas que a rede executa para encontrar um resultado considerado aceitável são realizadas de acordo com um algoritmo de treinamento capaz de ordenar as iterações necessárias até que se encontre uma resposta satisfatória.

Neste trabalho, o algoritmo utilizado como referência para a realização de testes foi o Backpropagation, que busca o aprendizado de modo supervisionado em redes multicamadas, baseando-se no método da tentativa e erro (MINSKY e PAPERT, 1969). Além disso, é capaz de conduzir a rede ao modo como se deve ajustar os pesos e bias a cada *feedback*, que acontece em caso de a resposta da rede não for satisfatória ao se comparar com a resposta desejada. Assim, o erro é calculado e retro propagado para as camadas ocultas, e, então, são realizados os devidos ajustes de cada neurônio de acordo com o nível de erro recebido, a fim de que a próxima iteração demonstre uma minimização deste erro. Este processo, portanto, persiste até que a resposta seja aceitável. E para manter estas características desta rede treinada, a rede assume, então, somente o modo de alimentação progressiva (*feedforward*), para que próximos dados de entrada (dentro do limite de treino) sigam as mesmas especificações anteriormente ajustadas para esta RNA.

3.2 – Modelo Matemático de Um Neurônio

Assim como um neurônio biológico, um neurônio artificial também é constituído de estruturas que organizam matematicamente seu funcionamento (Figura 5), a fim de estabelecer a comunicação entre entradas e saídas de sinais, beneficiando a base do projeto a ser implementado por redes neurais.

Figura 3– Modelo não-linear de um neurônio.



Fonte: Adaptado de HAYKIN, 2001.

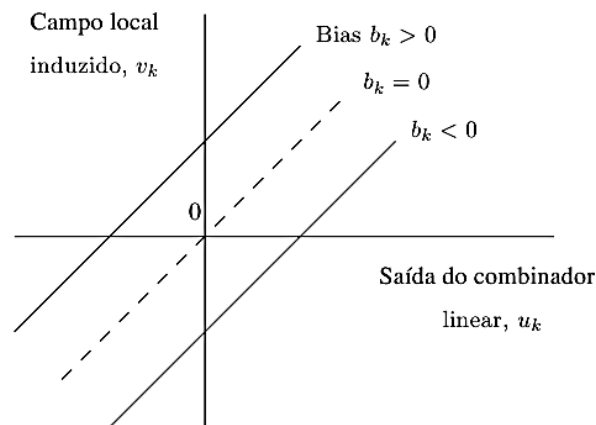
Neste modelo, pode-se perceber que há um conjunto de sinapses representadas pelas entradas de sinais x_i que são ponderados por meio de seus respectivos pesos sinápticos w_{ki} . Estes, por sua vez, tem a capacidade de fazer adaptações nas operações sinápticas a partir dos treinamentos durante a aprendizagem de tal forma que, se esses pesos forem positivos, a sinapse será excitatória, e quando negativos, inibitória (HAYKIN, 2001 e MINSKY e PAPERT, 1969).

Além disso, a estrutura conta ainda com um somador que reúne as informações dos sinais de entrada já ponderados pelas suas correspondentes sinapses neuronais, constituindo um *combinador linear*, cujo sinal resultante será u_k , como mostrado na equação abaixo:

$$u_k = \sum_{i=1}^N (x_i * w_{ki}) \quad (3-00)$$

Assim, chega-se a um potencial de ativação v_k ao somar todos os sinais do terminal do somador como representado na equação (3-02), incluindo o bias b_k , que é capaz de aumentar a entrada líquida de v_k quando apresenta valores positivos e diminuí-la, quando negativos, (como representado na Figura 4). O objetivo deste fator é de limitar e controlar o valor de sinal da saída y_k a uma amplitude normalizada com intervalos variantes de $[0, 1]$ ou $[-1, 1]$ por meio de uma função de ativação $\varphi(v_k)$.

Figura 4– Representação do efeito bias no controle sobre o sinal de entrada líquida na função de ativação.



Fonte: CARVALHO, 2000.

Matematicamente, temos:

$$v_k = u_k + b_k \quad (3-00)$$

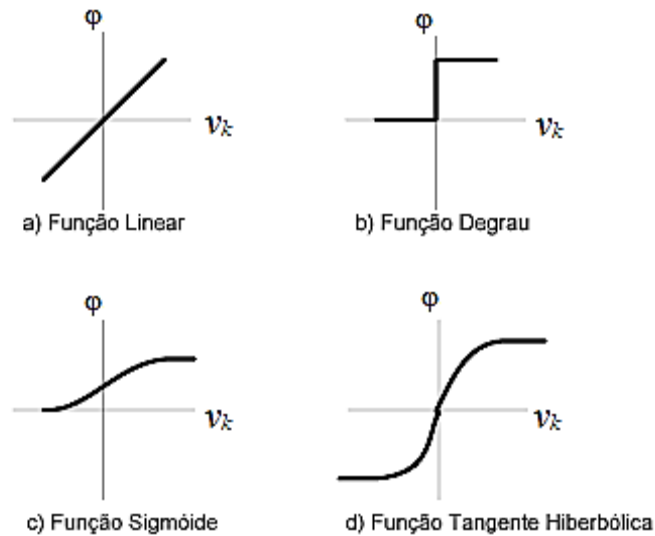
E, portanto, a saída y_k será:

$$y_k = \varphi(v_k) \quad (3-00)$$

Estas funções de ativação são importantes no controle do sinal de saída do neurônio, podendo ser representada por diversas funções que são escolhidas de acordo com os ajustes desejados para cada tipo de aprendizagem da rede neural. Algumas destas funções são mais conhecidas e comumente usadas no processo de correção dos pesos, limitando as respostas às devidas faixas de valores desejados. Entre elas, encontram-se, por exemplo, as funções lineares, degrau (muito usadas para a limitação da camada de saída), tangente hiperbólica e sigmoide. Estas duas últimas, além de muito utilizadas no controle das camadas ocultas, na prática, são as mais utilizadas no aprendizado de uma RNA.

Destaca-se, ainda, que todas estas funções de ativação (Figura 5), influenciam no tempo de convergência e na estabilidade da rede, que, em geral, dispõe estes dois fatores em posições inversamente proporcionais, ou seja, quanto maior o tempo de convergência, menor será a estabilidade da rede.

Figura 5 – Algumas das funções de ativação mais utilizadas.



Fonte: Adaptado de MÜLLER, 2008.

Algumas das características destas funções de ativação podem ser, assim descritas:

a) **Função Linear**: função comumente usada em camadas de saídas que não necessitam de efeito de saturação, ou seja, há presença de uma função diferencial no controle das respostas. Assim, tem-se:

$$\varphi(v_k) = v_k \quad (3-04)$$

b) **Função Degrau**: é uma função definida pelo controle de saídas também sem efeito de saturação, a partir de uma mudança limiar que alterna a mudança de estado do neurônio, como formulado a seguir:

$$\varphi(v_k) = \begin{cases} 1, & \text{se } v_k \geq 0. \\ 0, & \text{se } v_k < 0. \end{cases} \quad (3-05)$$

c) Função Sigmoide: É uma das funções mais utilizadas em RNA's é definida como uma função crescente, que apresenta um balanço entre o comportamento linear e não-linear, ou seja, observa-se, assim, uma necessidade de uma função diferencial. Assim, tem-se:

$$\varphi(v_k) = \frac{1}{1 + \exp(-av_k)} \quad (3-06)$$

Onde a é um parâmetro de inclinação crescente da função sigmoide, ou seja, quanto maior o valor de a , maior será a inclinação da função.

d) Função Tangente Hiperbólica: é uma variação de uma função sigmoide que, no entanto, podem apresentar limites compreendidos entre -1 e 1, assumindo, portanto, até mesmo valores negativos. Assim, temos:

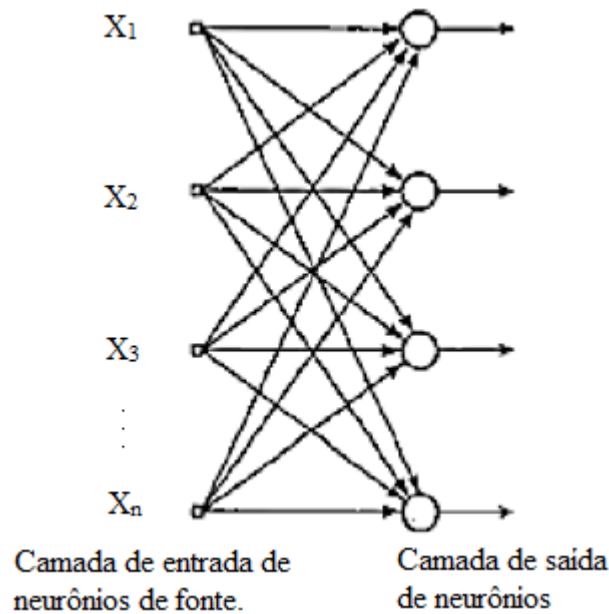
$$\varphi(v_k) = \tanh(v_k) \quad (3-07)$$

3.3 – Arquitetura de uma rede

As RNA's são moldadas para seguir um padrão de solução de acordo com os tipos de problema a serem tratados. A arquitetura da rede, portanto, é um aspecto importante a ser estudado durante o processo de determinação do comportamento das conexões. Nesta estrutura, analisa-se tanto o número de neurônios quanto o número de camadas envolvidas na saída e na seção intermediária — chamada também de camada escondida ou oculta. Assim, uma rede neural pode ser monocamada ou multicamada.

Uma RNA monocamada é assim chamada para caracterizar a única camada, a de saída, que reúne o conjunto de nós computacionais (neurônios) que operam na rede. Um exemplo de como estas redes operam é o modo progressivo, na qual as redes são alimentadas sempre adiante (*feedforward*), realizando tarefas mais restritas em seu desenvolvimento. Esta operação, considerada mais simples, só consegue resolver problemas em que os dados, em questão, sejam linearmente separáveis, não existindo retro propagação da camada de saída para as camadas ocultas. Na Figura 6, segue uma esquematização simples deste tipo de rede de monocamadas (BRAGA et al. 2000 e MINSKY e PAPERT, 1969):

Figura 6 – Representação de uma rede monocamada *feedforward*.



Fonte: HAYKIN, 2001.

Os conjuntos de nós de fonte ($x_1, x_2, x_3, \dots, x_n$) representam a camada de entrada que recebem os sinais por meio de sinapses e, em seguida, permitem que esses sinais sejam ponderados por meio dos pesos, chegando, assim, à única camada de saída, representada pelo conjunto de nós de saída (HAYKIN, 2001 e MINSKY e PAPERT, 1969.).

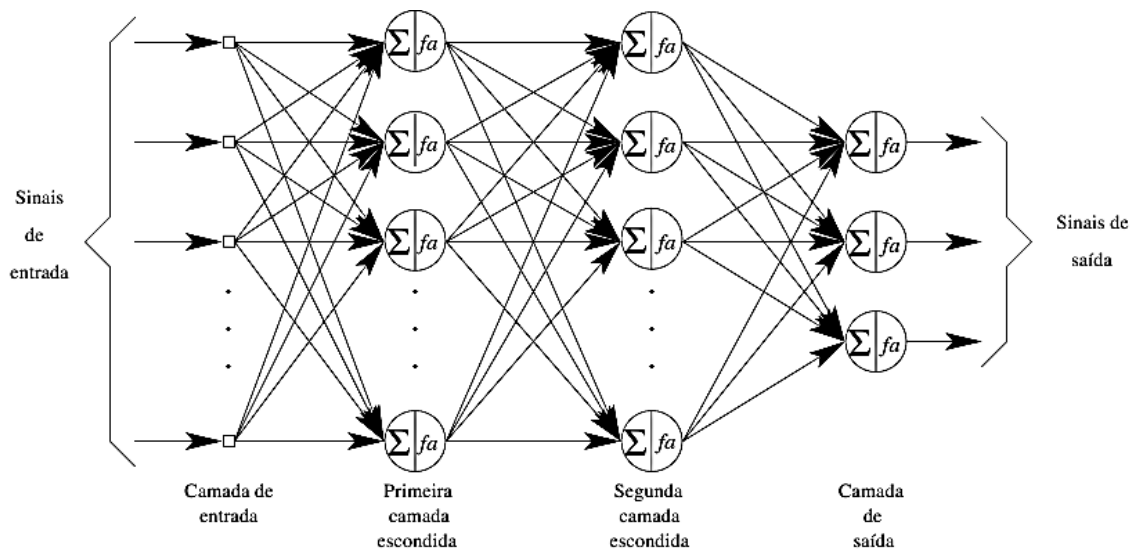
As arquiteturas que abrangem estas redes monocamadas em uma estrutura *feedforward* são as redes Perceptrons, Adaline e os mapa auto-organizáveis de Kohonen (HAYKIN, 2001).

Entretanto, as RNA's cujas estruturas apresentam pelo menos uma camada oculta entre as camadas de entrada e de saída são chamadas de redes multicamadas. Estas camadas extras de conexões permitem que a rede melhore a sua capacidade de aprendizagem, armazenando mais informações (conjunto de dados), possibilitando a extração de mais conhecimento. Este tipo de rede é especialmente útil quando se tem uma grande quantidade de sinapses na camada de entrada (HAYKIN, 2001 e BRAGA et al. 2000). Entre elas está a Rede Perceptron de Múltiplas Camadas MLP (*Multi Layer Perceptron*) como uma das mais amplamente conhecidas e usadas em diversos ramos da inteligência artificial, podendo ser esquematizada basicamente como mostrado na Figura 9. Esta rede, portanto, será a escolhida como referência para a realização de testes executados neste trabalho.

Neste tipo de rede, percebe-se também a característica de que cada camada é alimentada pelo sinal de saída da camada precedente, ou seja, assim como as monocamadas,

as MLP também apresentam características *feedforward*. A Figura 7 apresenta uma rede e as setas indicam esta propagação do sinal para frente. Este modo de operação é mantido após o erro calculado entre saídas obtidas pela rede e saídas desejadas atingir um valor satisfatório. No entanto, enquanto este erro calculado precisar ser minimizado, opera também no sentido de retropropagação dos sinais de saída para as camadas ocultas, como será analisado posteriormente no algoritmo Backpropagation (NIED, 2007).

Figura 7– Rede MLP com duas camadas intermediárias.



Fonte: NIED 2007.

Apesar das várias classificações de estruturas para uma rede neural, todas são estimuladas pelo ambiente de aprendizado a desenvolver tarefas básicas de modo eficiente, e de acordo com os tipos de problemas a serem resolvidos por meio da aplicação das RNA's. Em geral, entre as tarefas básicas de um mecanismo de IA, destaca-se a capacitação em armazenar conhecimento, aplicar o conhecimento armazenado para resolver o problema em questão e assumir um novo conhecimento por meio da experiência obtidas em meio as iterações (BECKER 1991).

Este conhecimento, portanto, para que seja reconhecido pela estrutura de uma rede, deve ser adquirida pela mesma por um processo chamado de aprendizagem, beneficiando o processo de desenvolvimento de uma rede neural a partir de seu ambiente (HAYKIN, 2001).

3.4 – Aprendizagem

Do mesmo modo como a mente humana precisa aprender a realizar uma tarefa antes que se possa, de fato, fazê-la, a rede neural também necessita aprender a identificar e reconhecer as características do ambiente em que se deseja que a mesma opere. Este processo, chamado de aprendizagem, é crucial para a eficiência da operação de uma rede, seja de forma supervisionada, na qual a rede é instruída (treinada) integralmente por meio de dados conhecidos do ambiente, seja de forma não supervisionada, na qual a rede se auto-organiza por meio de uma medição independente que estime valores para uma melhor representação da saída da rede (HAYKIN, 2001 e BECKER 1991). Independente destas classificações, a rede deve ser capaz de aprender os padrões do ambiente envolvido, proporcionando, ao final do treinamento da rede, o desenvolvimento de novas tarefas de acordo com a aprendizagem recebida.

A aprendizagem, de uma forma geral, permite que a rede possa melhorar seu desempenho por meio de iterações que viabilizam constantes reconhecimentos de padrões adaptados pelos parâmetros livres do ambiente por meio de um treinamento. Estes parâmetros são adaptados através de um processo de estimulação (algoritmo de aprendizagem) para que haja as modificações necessárias, tornando a rede apta a representar as características do ambiente em que está inserida (SILVA et al. 2010). Para cada tipo de rede, os processos de aprendizagem se adaptam às suas correspondentes necessidades para a solução do problema, seguindo basicamente a seguinte sequência de eventos (HAYKIN, 2001):

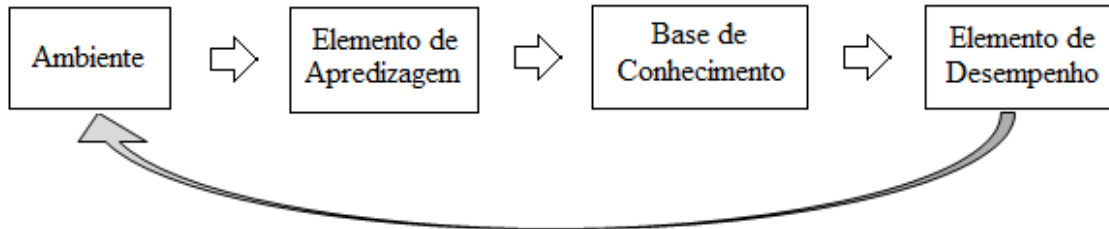
- a) A RNA é estimulada por um ambiente.
- b) Como resultado da estimulação, a RNA sofre alterações nos seus parâmetros livres.
- c) Como consequência das alterações, a RNA responde de uma nova maneira ao ambiente, buscando-se uma resposta com o mínimo de erro desejado.

Nesta aprendizagem, tem-se como uma das principais buscas a obtenção do valor mais adequado do incremento ΔW_k aplicado como ajuste sobre os pesos a serem modificados, buscando-se o aprimoramento de técnicas que permitam que a rede alcance a solução desejada por meio de um algoritmo de treinamento (BRAGA et al. 2000). Assim, para cada valor de peso atualizado após uma iteração, tem-se:

$$W_k(t+1) = W_k(t) + \Delta W_k \quad (3-08)$$

Em um modelo simples de aprendizagem, como representado na Figura 8, pode se resumir este processo, de uma forma geral, observando-se os seguintes conceitos:

Figura 8 – Representação de um modelo simples de aprendizagem.



Fonte: Adaptado de CARVALHO, 2000

Na aprendizagem de uma rede MLP, utilizada como referência para este trabalho, também se aplicam estas mesmas regras: o elemento de aprendizagem começa a ser adquirido, portanto, a partir das informações do ambiente (dados de entrada e saída) em que a rede está inserida para aperfeiçoar a base de conhecimento (dados de treino) a ser treinada pela rede de modo supervisionado. A partir disso, o elemento de desempenho (neurônios regidos por um algoritmo de treinamento) pode desenvolver o treinamento da rede para gerar novas respostas, utilizando-se dos dados da base de conhecimento. Estas novas respostas obtidas, então, são submetidas à comparação com as respostas desejadas (por meio de uma realimentação a partir do elemento de desempenho).

O método de comparação em meio ao processo de aprendizado é necessário, pois as informações cedidas pelo ambiente podem ser “imperfeitas” para que se dê início ao treino. Isto implica, portanto, em realizar possíveis tratamentos de dados, haja vista o elemento de desempenho não conseguir inicialmente realizá-los. Uma destas possibilidades, por exemplo, é a normalização dos dados, na qual se optou, neste trabalho, pela estratégia do teorema de Tales, a fim de normalizar as amostras tanto de treino como de validação dentro de uma mesma escala, visando a melhoria do treinamento da rede (HAYKIN, 2001. e SILVA et al. 2010).

Entre outras regras de aprendizagem de rede MLP, cita-se que cada vetor de entrada é treinado para obter sua correspondente faixa de saída normalizada que pode variar entre 0 e 1 (ou alternativamente, entre -1 e 1). E, por meio de estudos, comprovou-se que, se uma solução existe, a regra de aprendizagem convergirá para uma solução em um dado tempo finito (HAYKIN 2008 e RUSSELL e NORVIG 1995).

Assim, para que estas regras e necessidades das redes sejam atendidas de modo correto, deve-se considerar as devidas especificações do ambiente de rede, destacando-se a importância de se escolher um algoritmo de treinamento adequado para cada tipo de problema. Na rede MLP utilizada nos testes, por exemplo, o algoritmo aplicado foi o Backpropagation (retropropagação), amplamente empregado em treinamentos de diversas redes neurais.

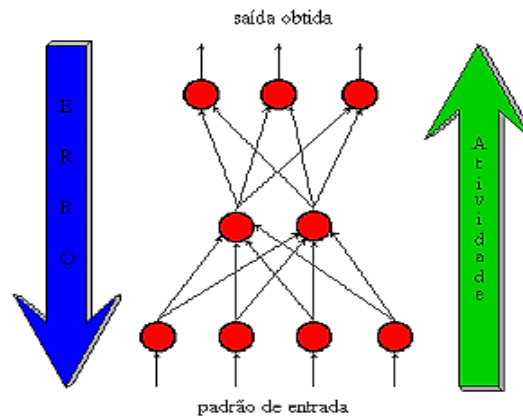
3.4.1 – Algoritmo Backpropagation

O algoritmo Backpropagation é um algoritmo de aprendizado que abrange um conjunto de regras bem definidas para solucionar um problema de rede neural. Neste trabalho, foi utilizado uma variante deste algoritmo chamado de Backpropagation de Levenberg-Marquardt. Neste processo, destaca-se principalmente o modo de como se ajustar adequadamente os pesos da rede a fim de melhorar a execução de tarefas de modo supervisionado (BRAGA et al. 2000 e RUSSELL e NORVIG 1995).

Uma das características do algoritmo é a realização da aprendizagem com base na tentativa e erro, o que torna o tempo de treinamento um fator relevante. Um aspecto que interfere neste tempo é a taxa de atualização dos pesos, pois a rede pode consumir um tempo de treinamento muito grande se a taxa for muito baixa. Entretanto, caso a taxa seja muito alta, a rede converge em um tempo curto, sendo mais suscetível, porém, a suspeita de resultados menos confiáveis devido à instabilidade da rede para gerar sinais de saída para novas entradas (MINSKY e PAPERT, 1969).

Na aplicação deste algoritmo no treinamento da rede, nota-se uma sequência de dois passos (Figura 9). Inicialmente, a camada de entrada da rede recebe um padrão de dados. De camada em camada, a atividade é estimulada pelos dados de entrada (vetor de entrada), se propagando até a camada de saída, produzindo, por meio desta, uma resposta. No segundo passo, ocorre a comparação entre esta resposta obtida e a desejada pela mesma para esse padrão particular. Feita a comparação, o erro é calculado em caso de divergências de respostas. Este erro, então, retropropaga até a camada de entrada, resultando nos devidos ajustes de pesos das unidades das camadas ocultas de acordo com o erro registrado a fim de minimizá-lo a cada iteração (CARVALHO 2000).

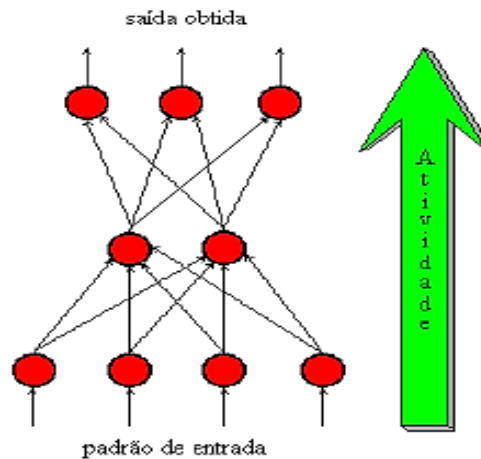
Figura 9 – Operação do algoritmo Backpropagation.



Fonte: CARVALHO 2000.

Quando a resposta obtida atingir um nível de erro satisfatório, a rede, então, estará apta à classificação de novos dados de entrada, mantendo-se o mesmo padrão por ela aprendido inicialmente (Figura 10). Para manter este padrão, o modo progressivo (*feedforward*) passa a ser utilizado pela rede treinada. Nesta configuração, os novos dados de entrada são recebidos pela camada de entrada, processadas nas camadas ocultas e, então, recebidas pela camada de saída para a obtenção de novas respostas baseadas no treinamento. Neste procedimento, no entanto, excetua-se a etapa de retropropagação do erro. Assim, as respostas a serem geradas para novos dados contidos dentro deste padrão continuam conservando a interpretação da rede instruída pelo treinamento durante o aprendizado (CARVALHO 2000).

Figura 10 – Operação do algoritmo Backpropagation em modo progressivo (*feedforward*).



Fonte: CARVALHO 2000.

4 Resultados Obtidos

4.1 – Conjunto de dados

O conjunto de dados utilizado no sistema foram disponibilizados pelo Departamento de Sistemas Multimídia da Universidade Técnica de Gdańsk [1]. Que em seu acervo conta com 60 peças musicais de compositores e artistas. De álbuns com os mesmos estilos, 15 a 20 peças musicais do compositor foram escolhidas que tinham qualidade de CD 16-bits de resolução padrão e 44.1kHz de taxa de amostragem em formato PCM estéreo.

Os gêneros musicais da base de dados criada foram limitados aos estilos básicos, como música clássica, jazz, blues, rock, metal e pop. Para cada peça, 20 seguimentos de 25 segundos foram extraídos e parametrizados, obtendo 20 vetores de características contendo 171 atributos para cada peça musical. A parametrização envolveu a extração dos seguintes parâmetros: 127 descritores do padrão MPEG-7, *Mel-Frequency Cepstral Coefficients* (descritores MFCC-20), bem como parâmetros “dedicados” relacionados ao tempo (24 descritores). Para os descritores MPEG-7, a janela *hamming* de comprimento 8.1920 com sobreposição de 50%. Por exemplo, para um trecho de áudio de 5 minutos, o número de quadros era de 155 (em cada segmento de áudio de 15 segundos com uma frequência de 44.100 Hz com sobreposição, resulta em 661.500 amostras, portanto, isso retorna 155 quadros).

No caso de *Descriptors* MPEG-7 a média e os valores de variância foram calculados dentro de cada segmento de áudio para todos os quadros contendo 8.192 amostras sobre todas as bandas de frequência. Para os parâmetros MFCC, foi utilizado o comprimento 512 com sobreposição de 50%. O vetor de parâmetros inclui 171 descritores, e o número total de vetores de parâmetros na banda base de dados completa é de 21.680. Complementarmente deve ser adicionado que os parâmetros foram normalizados para estar no intervalo de (-1, +1). Uma vez que as características MPEG-7 e *MEL-Frequency Cepstral-Coefficients* são apresentados ricamente na literatura relacionada com o assunto. Abaixo, estão listados estes parâmetros, conforme [1].

- 1 - Centróide Temporal (definido como o tempo médio sobre a energia);
- 2 - Centróide Espectral (calculado como potência média ponderada da frequência dos escaninhos no espectro de potência) valor médio calculado para todos os quadros;
- 3 - Variância do Centróide Espectral, calculada para todos os quadros;

- 4-37 - *Audio Spectrum Envelope* (ASE) descreve o espectro do áudio de acordo com uma escala de frequência logarítmica; valores médios em 34 bandas de frequência (calculadas em intervalos de uma quarta-oitava na gama de 62,5 Hz a 16kHz);
- 38: valor médio do ASE calculado para 34 bandas de frequência;
- 39-72 - Valores da variância de ASE nas 34 bandas de frequência (as mesmas das bandas de frequência referidas acima);
- 73 - Média da Variância de ASE;
- 74 e 75 - Centróide do Espectro de áudio. (definido como o centróide da potência ponderada frequência logarítmica) média e variância;
- 76 e 77: Propagação do espectro de áudio (descreve o segundo momento do espectro de potência da frequência logarítmica) valores médios e de variância.
- 78-101 - Média de Planalidade Espectral (SFM) média de valores para 24 bandas de frequência, calculados a $\frac{1}{4} 8$ (um quarto de oitava) sobre a faixa de 250 Hz a 16 kHz; (SFM descreve a propriedade plana do espectro de qualquer sinal de áudio dentro de determinado número de bandas de frequência);
- 102 - Valor Médio de SFM (média para 24 banda de frequências).
- 103-126 - Média de Planicidade Espectral (SFM) - (variância para 24 bandas de frequência);
- Parâmetro 127 - Média da Variância de parâmetros SFM (valor de variância para 24 frequências de banda);
- Parâmetro 128-147: 20 primeiros MFCC (valores médios);
- Parâmetros 148-171: Parâmetros dedicados no domínio do tempo com base na análise da distribuição de envelopes em relação ao valor RMS (média quadrática).
-

4.2 – Experimentos Realizados

Os experimentos foram realizados no ambiente do MATLAB 2017a, em um computador com configuração de processador Intel i3, 1 TB de disco rígido, 4 GB de RAM e placa de vídeo AMD Radeon com 2 GB de memória dedicada.

Para a técnica utilizada (Redes Neurais) foram executadas 20 simulações em conjunto. Em cada simulação, o conjunto de dados foi dividido aleatoriamente em 2 subconjuntos, com a mesma quantidade de amostras. Com o objetivo de minimizar as diferenças nos resultados. Logo em seguida foi treinada a rede neural. Em seguida, os modelos foram avaliados com o subconjunto de teste e os dados de desempenho foram salvos adequadamente. Para a técnica, foi utilizado o método de treinamento *k-fold cross validation*, com valor de k igual a 10. Segue o pseudocódigo que ilustra o processo de treino e teste:

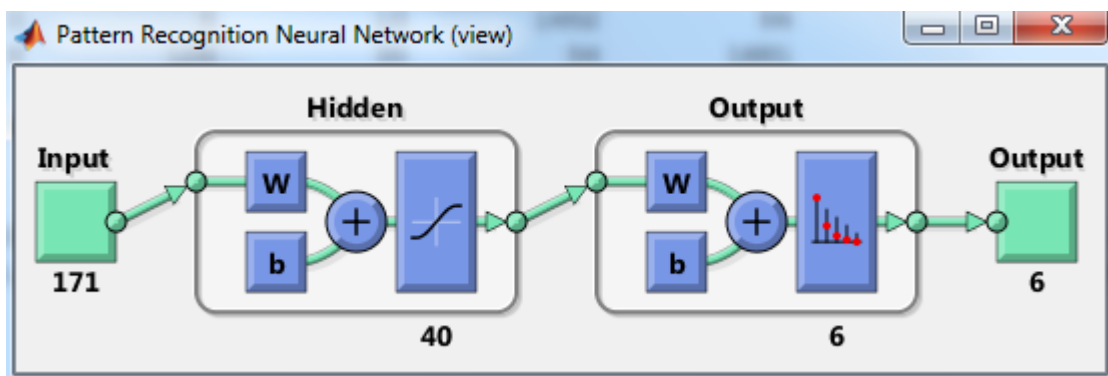
```

para i = 1:20
    divida aleatoriamente o conjunto de dados em 2;
    inicialize a rede neural;
    treine a rede neural;
    teste a rede neural;
    armazene os resultados de treino e de teste;
fim-para

```

Testes iniciais foram realizados para encontrar a quantidade ideal de neurônios na camada escondida. Visando equilibrar desempenho computacional e qualidade nos resultados, a topologia da rede neural ficou com 40 neurônios na camada escondida, 171 neurônios na camada de entrada (um para cada atributo) e 6 neurônios competitivos na camada de saída. Essa configuração de neurônios competitivos significa que cada neurônio representa uma classe, sendo o primeiro neurônio o gênero Blues, segundo Música Clássica, terceiro Jazz, quarto Heavy Metal, quinto é Pop e o sexto é Rock. O critério de parada da rede neural foi a quantidade máxima de épocas (1000) ou a quantidade máxima de épocas sem melhora (10). A Figura 12 mostra a arquitetura das redes neurais utilizadas.

Figura 11 - Arquitetura das Redes Neurais utilizadas.



4.3 – Técnicas de Avaliação dos Resultados

Como forma de avaliar os resultados obtidos pelo modelo, deve-se utilizar técnicas que permitam calcular o quão próximo do esperado foi o resultado obtido. Assim, duas técnicas amplamente utilizadas na literatura foram adotadas: a Matriz de Confusão e a Acurácia.

A Matriz de Confusão Uma matriz de confusão é criada classificando-se todas as amostras em classes, determinando se o valor previsto correspondeu ou não ao valor real. Todas as amostras em cada classe são somadas e os totais são exibidos na matriz, tanto as classificadas corretamente quanto as erradas. Esta matriz é uma ferramenta padrão para avaliação de modelos estatísticos e de Aprendizado de Máquina Supervisionado.

As linhas na matriz representam os valores reais e as colunas representam os valores previstos pelo modelo. Para problemas de classificação multiclasse, cada linha/coluna corresponde a uma das classes possíveis. Dessa forma, a diagonal principal da matriz de confusão apresenta os acertos do modelo e as demais células apresentam a quantidade de erros. A Tabela 2 apresenta um exemplo de matriz de confusão para um conjunto de dados de 30 animais, sendo 10 gatos, 10 cachorros e 10 ratos.

Tabela 2 - Exemplo de Matriz de Confusão

Real \ Obtido	Gato	Cachorro	Rato
Gato	9	0	1
Cachorro	2	7	1
Rato	0	2	9

A Acurácia mede a proporção de predições corretas geradas pelo modelo de classificação, também é chamada de taxa de acerto, e pode ser calculado a partir da Matriz de Confusão. É determinada pela comparação dos dados preditos com os observados, conforme a expressão abaixo, e o valor calculado está entre 0 e 1, sendo facilmente transformado em porcentagem:

$$\text{ACURÁCIA} = (\text{TOTAL_DE_ACERTOS}) / (\text{TOTAL_DE_AMOSTRAS})$$

Para o exemplo anterior, a Acurácia é 0,8, portanto gerando uma taxa de acerto de 80%.

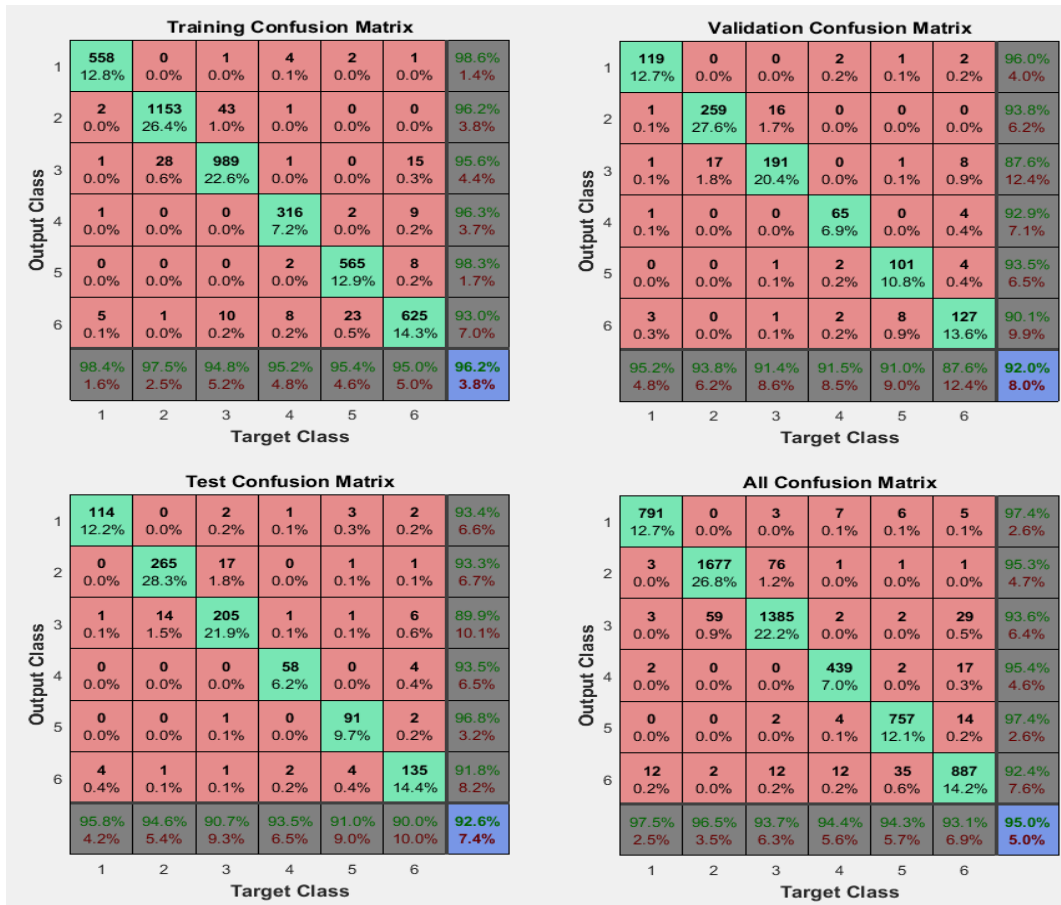
Esta medida é altamente afetada por desbalanceamentos do conjunto de dados, podendo gerar conclusões equivocadas sobre o desempenho do modelo. Entretanto, o conjunto de dados utilizado neste trabalho não apresenta grande desbalanceamento, portanto, o uso da acurácia não está comprometido.

4.4 – Resultados Obtidos

A Figura 12 mostra a matriz de confusão gerada por uma das RNAs, durante o treinamento com validação cruzada. Pode-se observar que esta RNA alcançou 96,2% de

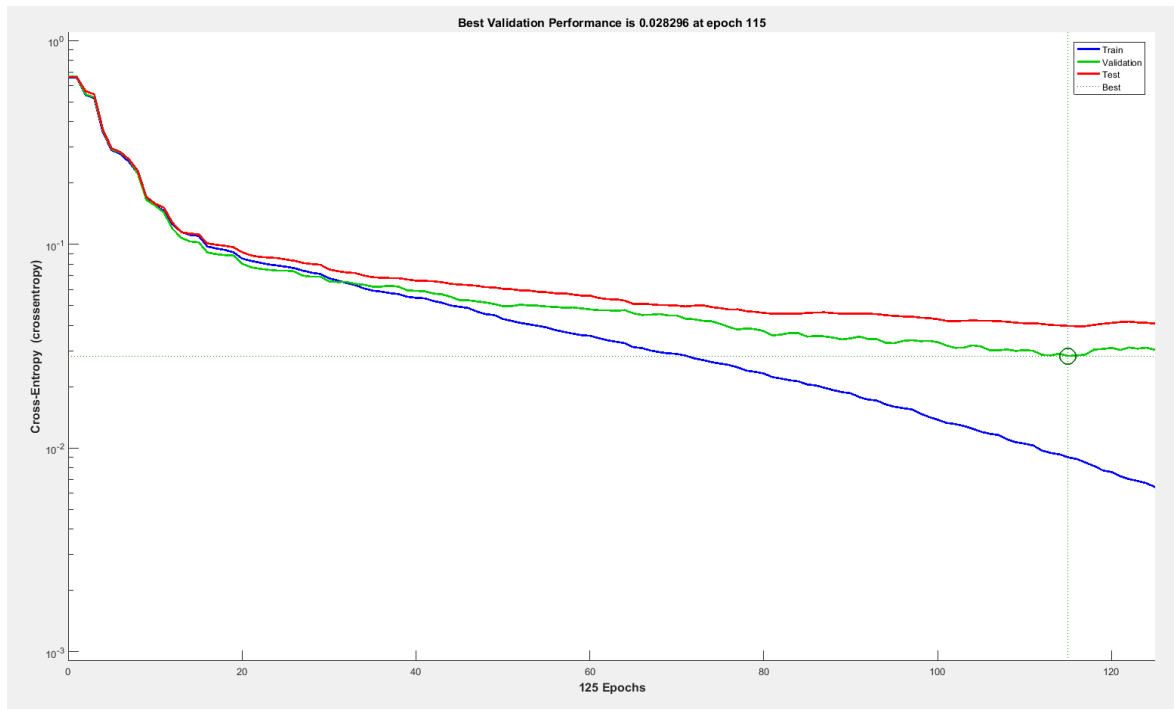
acerto no subconjunto de treino, 92% de acerto no conjunto de validação e 92,6% de acerto no conjunto de teste.

Figura 12 - Matriz de confusão de uma RNA usando 10-fold Cross validation.



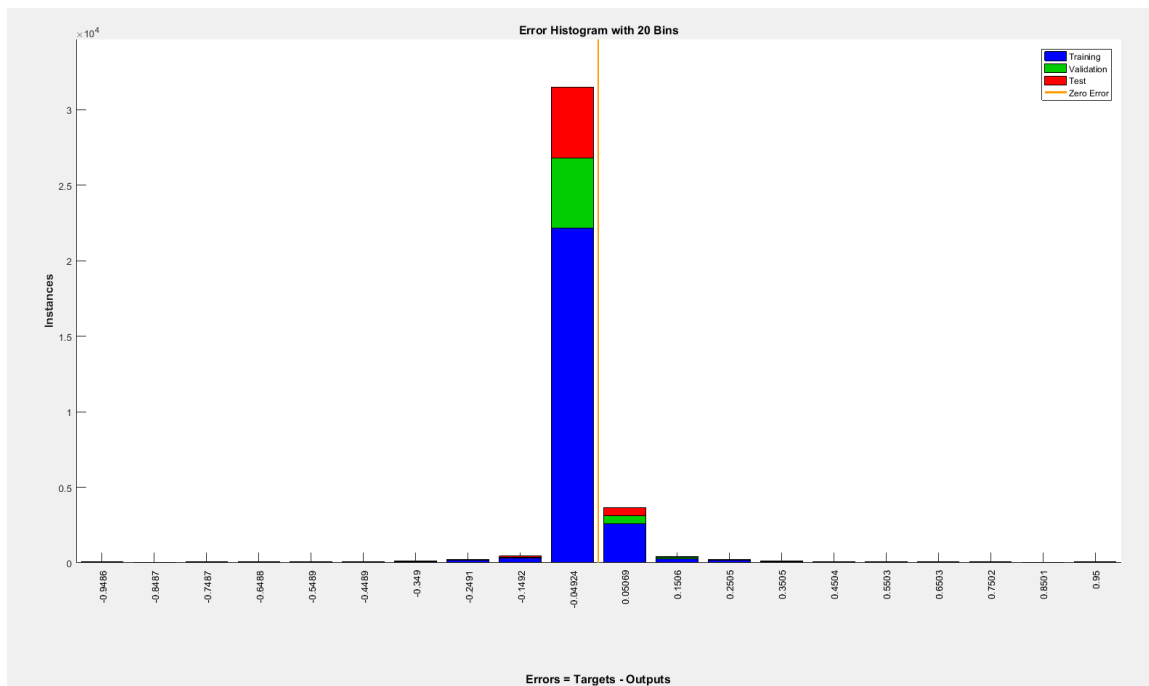
A Figura 13 mostra a curva de desempenho da RNA, que convergiu após 125 épocas. Percebe-se que após 125 épocas a rede continua melhorando a classificação para o conjunto de treino, porém aumenta o erro para o conjunto de validação, caracterizando o *overfitting*, ou seja, a rede estava "*decorando*" os dados de treino, prejudicando sua capacidade de generalização.

Figura 13 - Curva de desempenho de uma RNA



Abaixo temos a Figura 14 com o histograma de erro quadrático médio:

Figura 14 - Histograma de erro quadrático médio



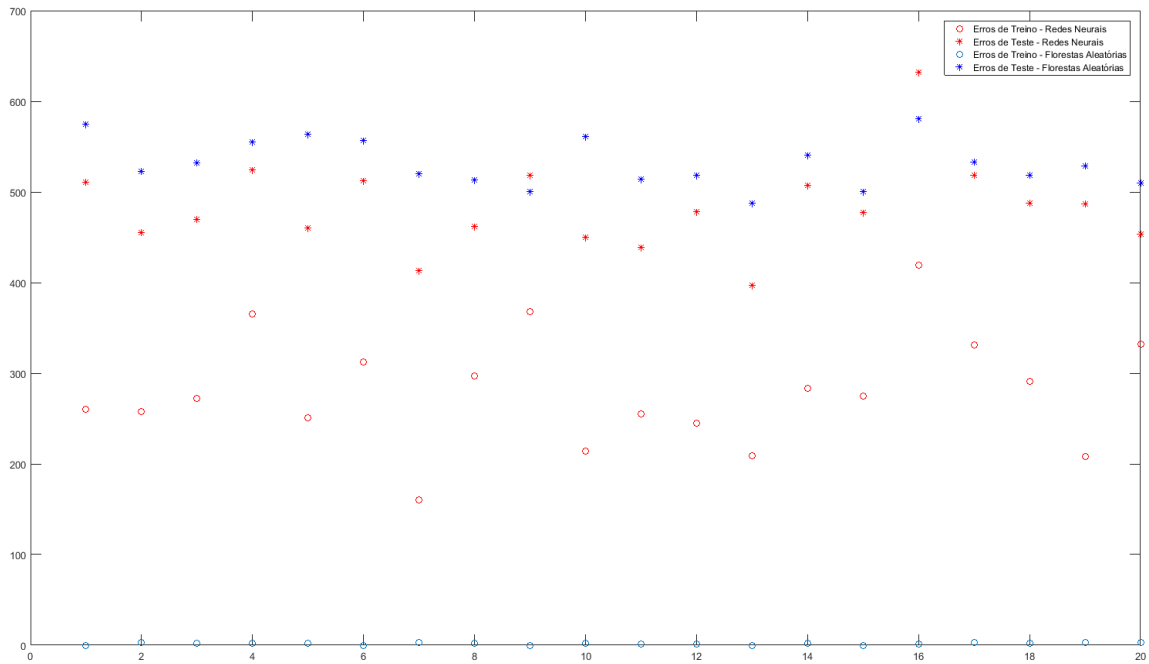
A Tabela 2 mostra as taxas de erro obtidas nas vinte simulações, para os dados de treino e de teste. E a Figura 15 é representação gráfica dos valores absolutos. Na sétima

execução a rede MLP gerada alcançou aproximadamente 93,6% de acerto para os dados de teste, sendo este o melhor resultado obtido.

Tabela 3 - Taxa de Erro em Porcentagem

Simulação	RNA - Treino	RNA-Teste
1	4,1613	8,1799
2	4,1293	7,2835
3	4,3534	7,5236
4	5,8419	8,388
5	4,0173	7,3635
6	4,9936	8,1959
7	2,5608	6,6112
8	4,7535	7,3955
9	5,8899	8,292
10	3,4251	7,2035
11	4,0813	7,0274
12	3,9213	7,6517
13	3,3451	6,3551
14	4,5294	8,1159
15	4,4014	7,6357
16	6,7061	10,1169
17	5,2977	8,292
18	4,6575	7,8117
19	3,3291	7,7957
20	5,3137	7,2515

Figura 15 - Quantidade de Erros em treino e teste dos modelos.



Da análise da Figura 15 e da Tabela 2, percebe-se que as redes neurais conseguem generalizar melhor o aprendizado, resultando em menores taxas de erro para os dados de teste.

E por fim como forma de avaliar o resultado obtido, foi gerada a matriz de confusão para a classificação realizada por uma das redes neurais implementadas conforme a Tabela 3. Nesta matriz, as linhas representam a Classe real das amostras e as colunas representam as classes preditas pelo modelo.

Tabela 4 - Matriz de Confusão das Redes Neurais Artificiais

RNA	Blues	Clássica	Jazz	Metal	Pop	Rock
Blues	759	5	3	8	5	1
Clássica	0	1712	73	0	0	3
Jazz	5	115	1349	2	2	12
Metal	16	2	7	392	13	23
Pop	13	0	6	9	724	27
Rock	8	4	44	18	26	861

Com a Matriz de Confusão gerada, foi possível verificar as classes mais semelhantes, segundo a RNA gerada. Percebe-se que a maior proporção de erros encontra-se entre as classes Música Clássica e Jazz. De fato, estes gêneros possuem algumas semelhanças e dependendo das músicas escolhidas para criar o conjunto de dados essa semelhança pode ter sido acentuada. Deve-se esclarecer que o conjunto de dados não determina a qual música cada amostra corresponde.

5 Conclusão

Neste trabalho pudemos entender como funciona as redes neurais, como surgiu, e seus usos mais frequentes, este trabalho também apresentou o desempenho das Redes Neurais Perceptron de Múltiplas Camadas na tarefa de Classificação Automática de Gêneros Musicais. A rede Perceptron se mostrou muito eficiente no reconhecimento de padrões, após alguns testes alcançando bons resultados, conforme demonstrado nas tabelas de confusão e sempre utilizando o mesmo conjunto de dados. É inegável que num futuro bem próximo as máquinas farão cada vez mais o que antes só se imaginava ser possível de ser realizado pelo ser humano, ainda há uma gama enorme de estudos que podem ser realizados seguindo essa linha de pesquisa, principalmente no reconhecimento de voz e de seus padrões.

Em trabalhos futuros pretende-se comparar os resultados obtidos nos testes realizados com outros classificadores, como Florestas Aleatórias, Máquinas de Vetores de Suporte com *Kernel* RBF e Naive Bayes, podendo assim verificar quais as vantagens e desvantagens de utilizar cada um, porém é necessária uma infraestrutura computacional mais robusta e um tempo maior de desenvolvimento dos testes. Métodos baseados em Grafos probabilísticos também podem alcançar bons resultados neste tipo de reconhecimento de padrões (classificação).

Referências

- [1] KOSTEK, BOZENA; *Music Information Retrieval in Music Repositories*; In: SKOWRON, A. e SURAJ, Z.; *Intelligent Systems Reference Library: Rough Sets and Intelligent Systems*. Berlin: Springer-Verlag, 2013, 463-489.
- [2] KOSTEK, B., KUPRYJANOW, A., ZWAN, P., JIANG, W., RA'S, Z., WOJNARSKI, M. e SWIETLICKA, J.; *Report of the ISMIS 2011 Contest: Music Information Retrieval*; In: KRYSZKIEWICZ, M., RYBINSKI, H., SKOWRON, A. e RA'S, Z. . *Lecture Notes in Artificial Intelligence: Foundations of Intelligent Systems*. Berlin: Springer, 2011, 715–724.
- [3] HAYKIN, S. *Redes Neurais: princípios e prática*. Porto Alegre: Bookman, 2001.
- [4] MÜLLER, H. H. *Redes Neurais Artificiais Aplicadas a Sistemas de Potência*. 2008. 106 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas-SP, 2008.
- [5] HAYKIN, S. *Neural Networks: a comprehensive foundation*, 2nd.ed., Prentice Hall, 1998.
- [6] BRAGA, A. de P.; CARVALHO, A. C. P. de L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: teoria e aplicações*. São Paulo: LTC, 2000.
- [7] MINSKY, M.; PAPERT, S. *Perceptrons: an introduction to computational geometry*. MIT Press, Massachusetts, 1969.
- [8] NIED, A. *Treinamento de Redes Neurais Artificiais Baseado em Sistemas de Estrutura Variável com Taxa de Aprendizado Adaptativa*. 2007. X f. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte-MG, 2007.
- [9] BECKER, S. Unsupervised learning procedures for neural networks. *International Journal of Neural Systems*, vol. 2, pp. 17–33, 1991.
- [10] SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A., *Redes Neurais Artificiais para Engenharia e ciências aplicas*. São Paulo: Artliber, 2010. 399 f.
- [11] HAYKIN, S. *Neural Networks and Learning Machines*, 3rd ed., Prentice Hall, 2008.
- [12] RUSSELL, S.; NORVIG, P. *Artificial Intelligence - A Modern Approach*. Prentice-Hall, New Jersey, 1995, p.563-597.
- [13] CARVALHO, A. P. de L. F., Home Page “Perceptron Multicamadas”. Departamento de Ciência da Computação. USP – SP. <http://www.icmc.sc.usp.br/~andre/neural2.html>, consulta em 10/05/2000.