



Universidade do Porto  
Faculdade de Engenharia  
**FEUP**

## **Redes Neurais para a predição da origem geográfica de música**

Relatório Final

Inteligência Artificial  
3º ano

Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Duarte Brandão - ei10060 - ei10060@fe.up.pt

Hugo Matos - ei11134 - ei11134@fe.up.pt

Jéssica Namora - ei11045 - ei11045@fe.up.pt

29 de Maio de 2015

## 1. Objetivo

Este trabalho tem como objetivo a aplicação de Redes Neurais artificiais na predição da origem geográfica de música. Para este efeito, iremos treinar a nossa rede neuronal através do algoritmo "Back-Propagation", com recurso a um conjunto de dados que nos foram fornecidos aquando da escolha do trabalho.

Após a entrega intermédia do trabalho, depois de termos escolhido a API a ser utilizada e de conhecermos bem o nosso problema o objetivo é treinar a nossa rede dentro das várias configurações possíveis (nº de camadas, nº de células nas diferentes camadas, variáveis de entrada, parâmetros do algoritmo de aprendizagem) de forma a obter o maior número possível de classificações corretas na fase de teste.

Depois de concluído o trabalho, espera-se que o mesmo consiga classificar novos casos, que não coincidam com os dados fornecidos para o estudo da rede, através da aplicação do algoritmo solicitado.

## 2. Especificação

Este projeto foca-se na criação de um programa que consiga treinar adequadamente uma Rede Neuronal artificial usando o algoritmo "Back-Propagation". Para este efeito, decidimos recorrer à biblioteca WEKA, que tem este algoritmo representado na classe *Multilayer Perceptron*.

Para a criação da Rede Neuronal disponibilizaram-nos dois conjuntos de dados contendo uma coleção pessoal de 1059 *audio tracks* cobrindo 33 países. Um dos ficheiros de dados continha 68 atributos denominados *audio features* e o outro continha 116. Uma vez que o tratamento de redes neuronais dá melhor resultado para redes com menos nós de entrada, optamos por usar o ficheiro que continha 68 *audio features* para seguir com o nosso trabalho.

A cada linha do *dataset* corresponde uma música (portanto 1059 músicas). As duas ultimas colunas de cada linha correspondem a latitude e a longitude que foram convertidas pelos respectivos países para uma interpretação mais intuitiva dos dados.

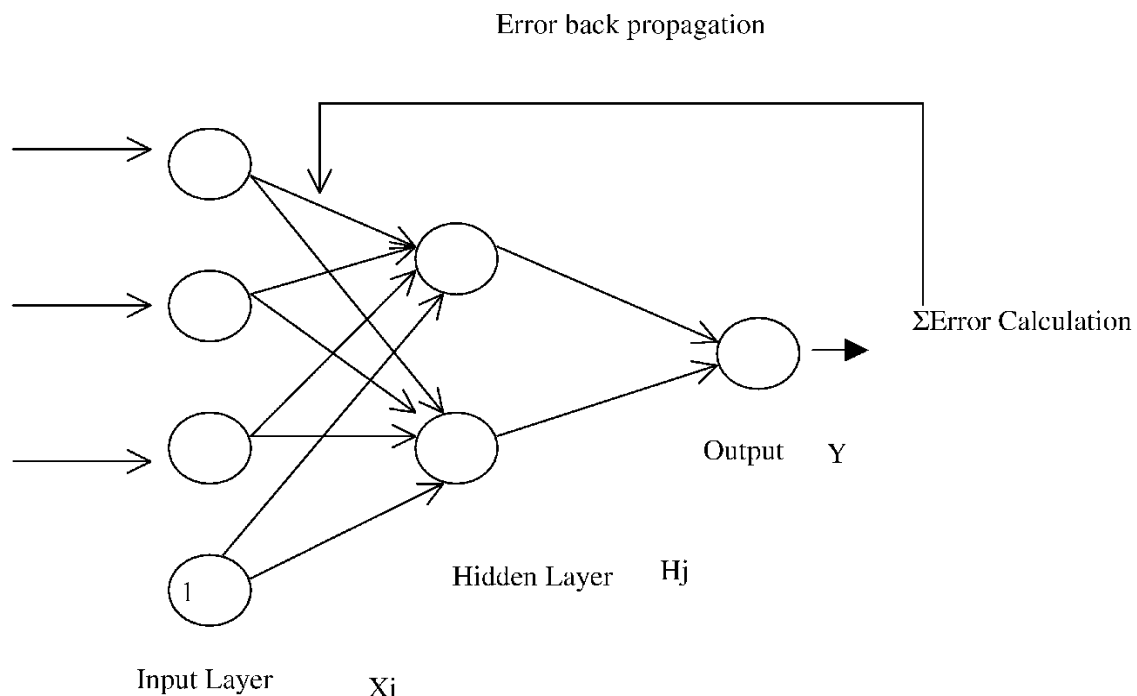
Para a construção da rede neuronal foram usados como *input* os 68 *audio features* presentes por cada linha e como *output* o país de onde essa música é originária.

## 2.1. Algoritmo Back-Propagation

Num primeiro passo este algoritmo irá propagar os valores das camadas de entrada pelas várias camadas até ser gerado um padrão nas camadas de saída.

Posteriormente, o padrão que se encontra em cada saída é comparado com o valor que se pretendia obter como resposta e traduzido num valor numérico denominado de erro.

Por fim, os pesos das ligações da rede são reajustados com base nos erros obtidos na etapa anterior.

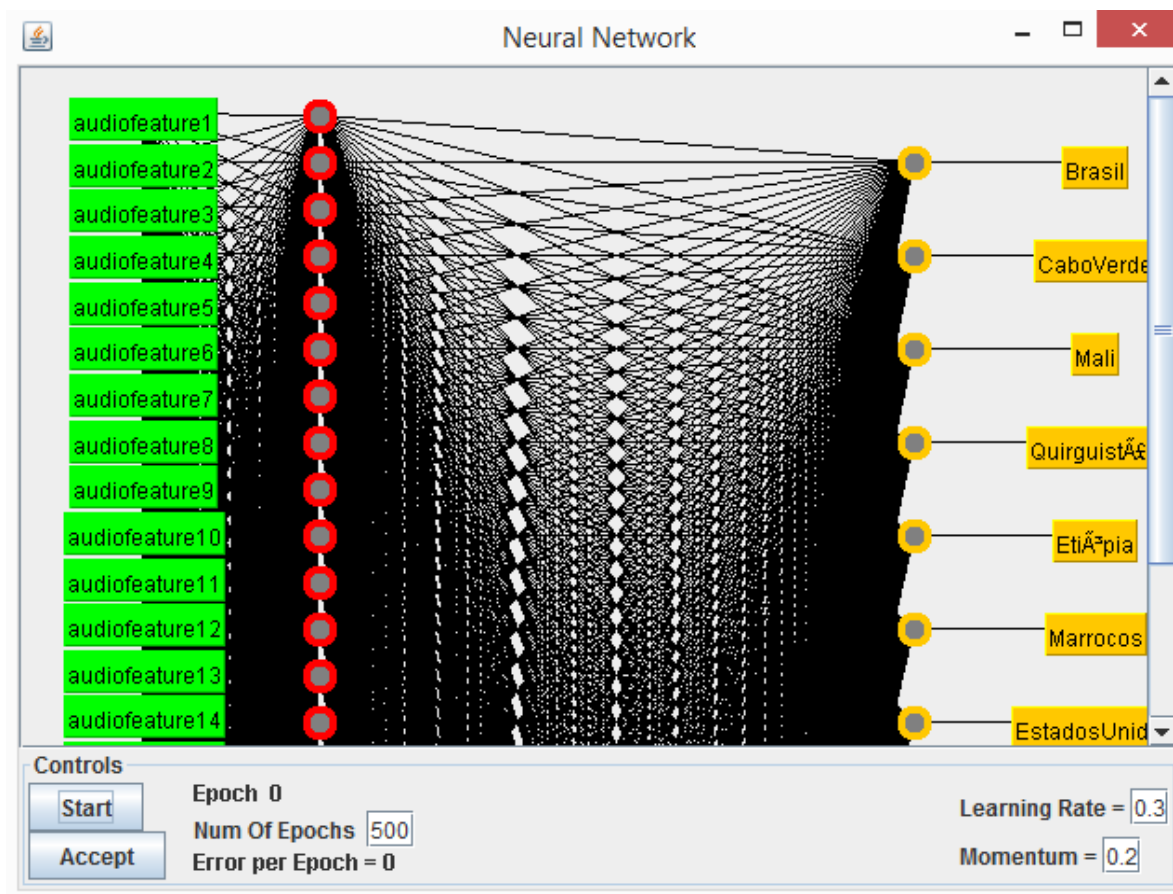


1. Exemplo duma rede neuronal e das suas camadas

Uma vez que nos foi pedido que usássemos este algoritmo no enunciado do trabalho, fizemos uma pesquisa da biblioteca do *Weka* sobre que *classifiers* utilizariam este algoritmo. Posto isto, realizamos o nosso trabalho recorrendo ao *classifier Multilayer Perceptron*.

## 2.2 Esquemas de Representação

O nosso projeto dá a possibilidade ao utilizador de visualizar os dados de entrada e de saída sobre a forma de um grafo dirigido. Na figura a baixo, podemos verificar, tal como foi dito na secção 2-Especificação, que os dados de entrada correspondem a *audio features* e os dados de saída correspondem a países de onde esses *audio features* são originários(imagem 2).



2. GUI da rede

[illegible]

Em cima podemos concluir que quase 75% das instâncias foram corretamente identificadas. Ao passo que 25% das mesmas foram incorretamente classificadas. Na diagonal principal da matriz de confusão podemos observar as músicas corretamente classificadas enquanto que os números não nulos fora da diagonal principal representam as músicas mal classificadas. Escolhendo um erro da matriz, o x representa o país de onde a música é originária, e o y representa o país com que a música foi confundida(fazendo corresponder o x e o y com os mostrados na GUI).

### 3. Desenvolvimento

O projeto foi implementado na linguagem de programação Java, com o uso da API da aplicação *Weka* (*Waikato Environment for Knowledge Analysis*) através do eclipse versão Juno.

Para a utilização do algoritmo backpropagation é utilizada a implementação própria do *Weka*, designada de *Multilayer Perceptron*.

Na implementação em Java foi apenas usada a classe *test* que contém a função *main* responsável por toda a execução do programa e da impressão dos resultados num ficheiro *log.txt*.

### 4. Experiências

- Diferente número de nós na camada intermédia

Fizemos uma pesquisa sobre o número de nós a adotar na camada intermédia e percebemos que as opiniões são muito dispersas. Fomos seguindo algumas delas e experimentando números distintos descobrindo qual o número que mais beneficiava os nossos testes. Sendo assim, optamos por utilizar 67 nós na camada intermédia.

*"To calculate the number of hidden nodes we use a general rule of: (Number of inputs + outputs) \* (2/3)" (from the FAQ for a commercial neural network software company).*

- Diferente número de camadas

Uma vez que recorremos a API *Weka* foi-nos fácil testar a possibilidade de ter diferente número de camadas intermédias e concluímos que essa mudança não traria qualquer tipo de benefício nos resultados finais. Posto isto, optamos por utilizar apenas uma camada intermédia uma vez que tornava a fase de teste mais rápida.

- Diferente número de variáveis de entrada

No início do nosso trabalho foi-nos fornecido dois ficheiros de texto que apenas diferenciavam no número de variáveis a entrar na rede. Através dos dados fornecidos juntamente com os mesmos, podemos verificar que para os dados em estudo era indiferente o uso de qualquer um deles e por isso optamos pelo que tinha menos dados, tornando o estudo da rede mais rápido.

- Parâmetros do algoritmo de aprendizagem

Para o treino da nossa rede neuronal tivemos de escolher duas variáveis denominadas *Momentum* e *Learning Rate*. Estas duas variáveis são utilizadas para o cálculo do peso. O cálculo do peso é aprendido através do set de treino. As mudanças no peso são calculadas multiplicando o gradiente pelo *learning rate* e somando a mudança de peso anterior multiplicada pelo *momentum*:

$$W_{\text{next}} = W + \Delta W$$

$$\Delta W = - \text{learning\_rate} \times \text{gradient} + \text{momentum} \times \Delta W_{\text{previous}}$$

O treino da rede neuronal passou por 500 iterações para criar o seu modelo.

Após todas estas experiências, optamos por usar o valor de *Learning Rate* = 0.3, *Momentum* = 0.2, 90% das músicas fornecidas para o caso de treino e 100% para o caso de teste. Apenas uma camada intermédia e 67 nós nessa camada. Tendo em conta estas alterações, conseguimos classificar as instâncias com ~74,98% de sucesso.

## 5. Conclusões

Com este projeto foi-nos possível compreender o conceito de Redes Neurais e as suas aplicações no mundo real. Apesar do projeto final ter sido desenvolvido no eclipse, para conseguirmos usar a biblioteca *Weka* tivemos de trabalhar diretamente com a mesma, o que

nos proporcionará a oportunidade de trabalhar mais facilmente para outros trabalhos e sobre outros algoritmos.

O facto do tema do nosso trabalho ter sido leccionado apenas no fim do ano fez com que trabalhássemos as nossas capacidades de pesquisa e esse foi um dos motivos da escolha deste tema.

## 6. Melhoramentos

Para o futuro tentaríamos implementar uma interface gráfica em *swing* em vez da interface devolvida pelo *Weka*.

Para uma melhor classificação das músicas pela sua origem geográfica poderíamos acrescentar mais casos à nossa classe de treino para tentarmos chegar a uma percentagem mais elevada de sucessos.

## 7. Recursos

- [http://www.cs.waikato.ac.nz/~ml/weka/gui\\_explorer.html](http://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html)
- <http://archive.ics.uci.edu/ml/datasets/Geographical+Original+of+Music>
- <https://web.fe.up.pt/~eol/IA/dokuwiki/doku.php?id=weka>
- <http://weka.wikispaces.com/ARFF+%28book+version%29>
- <http://www.heatonresearch.com/node/707>
- <http://grepcode.com/file/repo1.maven.org/maven2/nz.ac.waikato.cms.weka/weka-stable/3.6.7/weka/classifiers/functions/MultilayerPerceptron.java?av=f>
- <http://www.cs.waikato.ac.nz/ml/weka/mooc/moredataminingwithweka/slides/Class5-MoreDataMiningWithWeka-2014.pdf>

Uma vez que a realização deste trabalho teve mais parte de pesquisa teórica do que trabalho prático, cada um trabalhou separadamente ao longo das primeiras semanas. Na fase da implementação o trabalho foi feito sempre em conjunto tendo todos os intervenientes o mesmo nível de empenho.