



# Trabalho 6

Prof. Mateus Mendelson  
mendelson@unb.br

## 1. Introdução

Um dos grandes desafios na computação é a elaboração de métodos capazes de classificar elementos de acordo com certos padrões. Na maioria das vezes, essa é uma atividade que não oferece grandes dificuldades quando realizada por seres humanos. Entretanto, há situações que são mais complexas e que envolvem uma grande quantidade de elementos, podendo tornar o trabalho de classificação manual extremamente árduo. Uma solução para essas situações é o uso de computadores.

Há diversas abordagens voltadas à classificação. Dentre elas, destacam-se as redes neurais artificiais.

Redes neurais artificiais surgiram como uma tentativa de se imitar computacionalmente as redes neurais biológicas presentes no sistema nervoso central humano, estas consistem em um arranjo de neurônios interconectados que trocam informação entre si. Dessa forma, permitem a detecção, o aprendizado e a aplicação de padrões. Ademais, as redes neurais artificiais são muito utilizadas em áreas ligadas à inteligência artificial, principalmente para resolver problemas de classificação, de ajuste de função (regressão), de robótica, de controle etc.

O presente trabalho tem por objetivo o desenvolvimento de um novo método de classificação para as *features* já desenvolvidas no Projeto 02.

## 2. Realização do projeto

Leia em [1] a Seção 2.1.7 para adquirir o embasamento teórico necessário para esse projeto.

Implemente uma rede neural *feed-forward*. A função de ativação a ser utilizada deve ser a função logística, não a função degrau.

A quantidade de neurônios na camada de entrada deve ser igual a quantidade de *features* resultante de cada imagem (536). A camada de saída deve ser composta por apenas um único neurônio, que fornecerá valores entre 0 e 1.

Ainda, a rede neural deve apresentar uma única camada oculta. A quantidade de neurônios na camada oculta deve ser definida pelo próprio usuário via linha de comando no momento em que o programa for executado (ex.: \$ ./proj6 10).

A formação das *features* e base de dados devem ser reaproveitados do Projeto 02. Apesar de [1] citar que deve haver um conjunto de validação, utilizaremos apenas os mesmos conjuntos de treinamento e de teste.

O treinamento a ser realizado deve ser supervisionado. Ao treinar a rede neural, as saídas (*targets*) esperadas para o asfalto devem ter o valor 0 e, para a grama, o valor 1. Para que o treinamento seja realizado, utilize o algoritmo de treinamento Backpropagation. Há um exemplo numérico de como os cálculos nesse algoritmo são realizados em [1].

Ao apresentar o conjunto de teste, a rede neural irá gerar saídas entre 0 e 1. Caso a saída seja menor ou igual a 0.5, deve-se considerar que a imagem apresentada é de asfalto. Caso contrário, de grama.

Não é necessário utilizar o algoritmo de Levenberg-Marquardt e nem a técnica de Monte Carlo.

### 3. Métricas

Por fim, o sistema deve imprimir na tela as seguintes métricas:

- Taxa de acerto: a porcentagem de imagens do conjunto de teste que foram classificadas corretamente;
- Taxa de falsa aceitação: a porcentagem de imagens do conjunto de teste que pertencem à classe "asfalto", mas que foram classificadas como sendo "grama"; e
- Taxa de falsa rejeição: a porcentagem de imagens do conjunto de teste que pertencem à classe "grama", mas que foram classificadas como sendo "asfalto".

### Referências

- [1] SILVA, Mateus Mendelson Esteves da. "Using Artificial Neural Networks and Smartphone Accelerometers for Accident Prevention" (in portuguese). 2018. 69 f., il. Dissertation (Master in Mechatronics Systems)—University of Brasília, Brasília, 2018. – disponível em [mmendelson.com/publications](http://mmendelson.com/publications)