

# Python e predição de dados usando redes neurais multicamadas

Jonathan R. de Sousa<sup>1</sup>, Juliana F. Antunes<sup>1</sup>, Ícaro A. Alencar<sup>1</sup>, Júnior C. R. Leite<sup>1</sup>,  
Thayná M. S. Martins<sup>1</sup>, Thays G. S. Santos<sup>1</sup>

Departamento de Computação – Instituto Federal de Educação, Ciência e Tecnologia de  
Mato Grosso – Cuiabá ,MT – Brasil

jonathan.caminho@outlook.com, juliana.antunes@cba.ifmt.edu.br,  
icaroalencar@live.com, {juniorcrleite, thay0229}@gmail.com, thaysgss@hotmail.com

**Abstract.** *This article describe a possible approach of prediction using multilayer perceptron(MLP) that are networks based on artificial neurons in graphs, that receive inputs to generate outputs based on previous deference, process called learning or training, suggesting that the artificial intelligence will be improved by these processes until it becomes perfect, inerrant for the initial environment. To demonstrate, will be used a truth table of the logic gate XOR for 3 inputs and their predictions.*

**Resumo.** *Este artigo descrever uma abordagem de predição usando redes neurais multicamadas (MLP) que são grafos em redes de neurônios artificiais, onde recebem entradas para gerar saídas baseadas nas deferências anteriores, processo conhecido como aprendizado ou treinamento, sugerindo que a Inteligência Artificial seja aperfeiçoada nesses processos, até que se torne perfeita, inerrável, para aquele mesmo ambiente. Para simulação foi utilizada a tabela verdade da porta lógica XOR (Ou Exclusiva) para 3 entradas e suas respectivas predições implementadas na linguagem Python.*

## 1. Introdução

Por muito tempo as ferramentas matemáticas de probabilidade e estatística foram utilizadas para tentar fazer predições e entender melhor o comportamento das informações (BARBETTA, 2004), porém nos dias atuais, com o advento da informatização, a inteligência artificial surgiu como meio de se chegar a esse mesmo resultado de forma mais simples e menos dependente de intervenção humana.

A abordagem apresentada utiliza a linguagem Python e conceitos de programação estruturada com a biblioteca conhecida como *NumPy* para o trabalho com números de forma mais precisa; além da implementação da rede perceptron multicamadas nessa linguagem.

## 3. Python

A Linguagem de Programação *Python* foi criada por Guido van Rossum (1991), focada em produtividade e legibilidade, ou seja, produz um código fácil para programador entender, dentre os itens que cooperam para este aspectos estão: baixo uso de caracteres

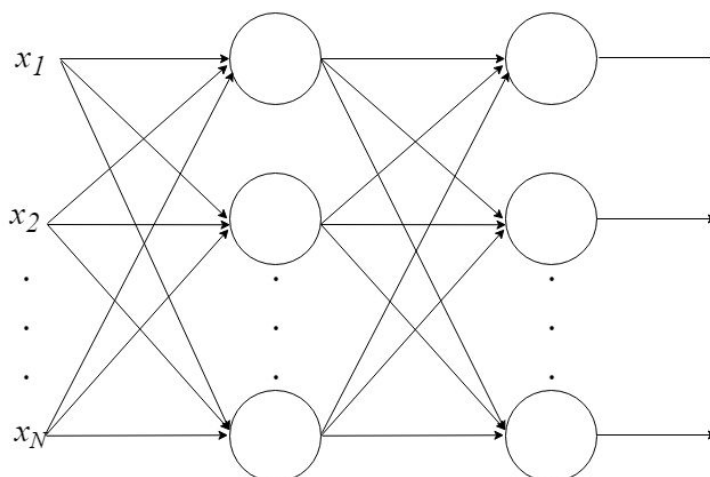
especiais; baixo índice de palavras-chave; gerenciador de memória eficiente que evita vazamento de memória.

Além do mais a biblioteca padrão é muito extensa, de tal forma a ter uma diversidade de abordagens prontas para serem utilizadas, seja acesso a banco de dados, trabalho com números complexos, interfaces gráficas etc.

O Python também é multiplataforma, e caso apareça alguma que ainda ele não suporte, os desenvolvedores podem modificar o código para que seja possível compilar para aquela linguagem, pois o Python também é uma linguagem livre. Assim, todos esses fatores fazem com que o Python seja uma das preferidas linguagens para o trabalho com inteligência artificial, *data mining* e *machine learning*.

#### 4. MLP – Multilayered Perceptron

Conforme mostra a Figura 1, a rede MLP (*Multilayer Perceptron*) é um tipo de rede artificial retroalimentada que gera uma série de saídas a partir de uma série de entradas, assim se diferenciando da rede Perceptron comum que gera uma saída apenas. Geralmente a rede MLP é utilizada em aplicações *deep learning*. (MARS LAND, 2014)



**Figura 1 - MLP com entrada, processamento de dados e sentido de saídas**

A rede Perceptron multicamadas é constituída de várias camadas em grafos, de forma que o sinal parte para um único sentido, cada nó tem uma função de ativação não linear. Porém a MLP precisa ser supervisionada no momento da aprendizagem, dentre suas aplicações está: Reconhecimento de fala, Reconhecimento de imagens e Aprendizado de Máquina.

#### 5. Aplicação

Na simulação foi utilizada a *IDE PyCharm*, da JetBrains na versão para estudos, com todo o pacote *NumPy*, para o trabalho com Arrays multidimensionais e cálculos numéricos, utilizadas para *machine learning*.

Inicialmente foi implementada as camadas de neurônios, definindo o número de neurônios, de entradas por neurônios e inicialização dos pesos como randômicos, posteriormente inicializada as camadas e definidas as funções de teste dos dados para

aprendizagem. A rede neural usa tentativa e erro para ajustar os pesos, denominado como aprendizado, desta forma ela pode prever informações com certa margem de erro, quanto maior o tempo de treinamento maior será a acurácia das predições.

A Tabela 1 mostra os dados fornecidos para treinar a rede, a última linha da tabela é a nova entrada para a previsão. Após o treinamento da rede *MLP*, foi realizada a entrada do valor [1,1,0], tendo como a saída prevista o valor de 0,0078876, ou seja, muito próximo do correto, que é zero.

**Tabela 1 - Dados de treinamento e a nova situação para o teste**

X1	X2	X3	Saída
0	0	1	0
0	1	1	1
1	0	1	1
0	1	0	1
1	0	0	1
1	1	1	0
0	0	0	0
1	1	0	?

## 7. Conclusão

Tendo em vista as poucas informações tratadas pela rede MLP e com um único treinamento a predição do valor foi muito próxima do valor real, isto é, o erro foi de praticamente 0%, a proposta de se conhecer a ferramenta Python e programar uma rede neural nessa linguagem teve sucesso através do treinamento e previsão do exemplo clássico de porta lógica. O estudo inicial possibilitou conhecer novas tecnologias que podem ser usadas para predição utilizando base de dados maiores, como predição de ataques hacker, de terremotos, de queimadas ou até jogos de loteria, que está proposto em estudos futuros.

## Referências

BARBETTA, Pedro Alberto; REIS, Marcelo Menezes; BORNIA, Antônio Cezar **Estatística para Cursos de Engenharia e Informática**. 3Ed. São Paulo: Editora Atlas, 2010.

Marsland, (2014) "Machine Learning", Chapman and Hall/CRC, Estados Unidos