

# INTELIGÊNCIA ARTIFICIAL

## *Detalhes do desenvolvimento do trabalho sobre MLP - Multilayer Perceptron*

Gabriel Folegatti Santana nº USP: 9911506

Hugo Shiroma nº USP: 9390378

Isabel Pereira Boroni nº USP: 10284480

Larissa Marques nº USP: 9894515

Victor Planas nº USP: 9778578

Para esta segunda entrega do trabalho da disciplina de Inteligência Artificial, nosso desafio foi codificar uma rede neural do tipo Multilayer Perceptron utilizando frameworks que tenham esse tipo de função. Pelo fato de estarmos seguindo com o desenvolvimento deste exercício em Python, procuramos frameworks na linguagem e descobrimos o **SciKit Learn** e o **TensorFlow**.

As principais diferenças entre os dois: o TensorFlow é um framework que permite desenvolvermos uma rede de forma mais baixo nível, onde podemos ver exatamente qual etapa da rede estamos construindo. O SciKit Learn possibilita a construção de uma rede de forma mais abstraída, se atentando apenas aos parâmetros iniciais e não se preocupando com os muitos detalhes do que ocorre no processo de treinamento e teste. Pela facilidade de manipulação, iniciamos o desenvolvimento dessa entrega com o SciKit Learn, porém no decorrer desse processo, por conta das inconstâncias que encontrávamos na acurácia do algoritmo feito, decidimos dividir o grupo em dois e enquanto uma parte buscava compreender melhor o funcionamento do SciKit, a outra buscou entender o Tensorflow. Como o TensorFlow possibilita esse detalhamento maior dos passos, pudemos reconhecer no código exatamente onde se iniciava o feedforward e o backpropagation, por exemplo, mas ainda assim haviam inconstâncias na acurácia do treinamento da rede. Com um mesmo grupo de hiperparâmetros e testando o mesmo conjunto de treino a acurácia variava entre 80% a 100%, o que nos estranhava já que deveria variar mas não de modo tão aleatório por se tratar de conjuntos iguais. Por conta disso, não continuamos os testes para o conjunto de caracteres com ruídos no TensorFlow.

Paralelamente, pudemos compreender os pontos que faltavam na implementação com SciKit Learn, para estabilizar a acurácia do nosso algoritmo. Este framework necessita dividir o nosso dataframe em conjuntos de treino e teste para assim começar a apresentar uma melhor constância de resultados quando colocamos o parâmetro stratify. Esse parâmetro garante que todos os targets vão ser checados na hora de treinar. O split favorece nosso treino, potencializando nossa MLP e isso foi importante para encontrarmos os melhores parâmetros. Outro ponto que exploramos para tentar estabilizar a rede foi utilizando a classe GridSearch, que testa nossa rede exaustivamente procurando a melhor configuração dos hyperametros.

Ao encontrarmos a melhor configuração para a rede, adaptamos nosso código para treinar com o conjunto de dados de caracteres limpos e testar com o conjunto de caracteres com ruídos, além de treinar e testar o AND, OR e XOR. O SciKit Learn gera os pesos iniciais aleatoriamente. Um ponto interessante do SciKit é que ele permite que alteremos as classes

prontas do framework, e com isso pudemos abrir a classe `MLPClassifier` para encontrar em qual momento o algoritmo calcula seus pesos, assim pudemos printar e registrar no nosso log os pesos iniciais, aleatórios, e os finais.