

# Relatório - Trabalho 2

## Inteligência Artificial

**Brendon Henrique**  
bhps17@inf.ufpr.br

GRR20170203

Universidade Federal do Paraná  
Bacharelado em Ciência da Computação

## 1 Implementação cálculo de atualização dos pesos: Parte 1

Na primeira parte da implementação do trabalho foi realizado o cálculo para atualização dos pesos no perceptron, para fazer isso foi executado o produto interno no do peso atual com os valores da instância, após isso era verificado se o valor é menor ou igual a 0 para definir se a label é 0 ou 1, caso a condição seja verdadeira a label obtida é 0, caso contrário é 1. No penúltimo passo fazemos a checagem para ver se obtemos um erro na respectiva época vendo se a label obtida é diferente da label esperada, caso seja verificamos o valor do erro como a variação de label esperado com a label obtida e recalculamos os pesos com a função também implementada sendo esse o último passo do algoritmo.

## 2 Método de comparação: Parte 2

Classificadores utilizados:

- Perceptron Classifier
- MLP Classifier

Foi implementado a classificação dos 3 datasets disponibilizados (iris, artificial e xor) utilizando ambos classificadores com uma variação entre os parâmetros utilizados para instanciar os classificadores. Com a finalidade de testar para tentar obter a parametrização que obtinha os melhores resultados foi utilizado um código que está comentado no arquivo ml.py que basicamente refazia o aprendizado sobre o conjunto de teste e salvava os resultados obtidos (acurácia do modelo e parâmetros utilizados em um array de resultados) para auxiliar na escolha da melhor parametrização. Para os parâmetros escolhidos que serão informados a diante foram obtidos 5220 execuções de classificação.

Parâmetros validos em ambos classificadores:

- Learning rate (Taxa de aprendizado): Com os seguintes valores [0.001, 0.01, 0.1, 1.0].
- Número de iterações: Com os seguintes valores [100, 200, 400].

Parâmetros utilizados somente no classificador MLP:

- Solver: Tipo de solução utilizada para otimização dos pesos com os seguintes valores [lbfgs, sgd, adam]
- Frequência de learning rate: Forma como a taxa de aprendizagem se comporta com os seguintes valores [constant, invscaling, adaptive]
- Activation: Função ativadora para outra camada com os seguintes valores [identity, logistic, tanh, relu]
- Variação de combinações de número de neurônios e número de camadas: [1 camada e 5 neurônios, 1 camada e 100 neurônios, 2 camadas e 10 neurônios em cada, 3 camadas e 20 neurônios em cada]

### 3 Resultados obtidos

Todos os resultados obtidos podem ser reproduzidos descomentando e executando o código entregue e executando o `ml.py`, foi utilizada essa opção devido a instrução da forma de entrega do trabalho.

Segue respectivamente os melhores parâmetros com as melhores acuracias obtidas em ambos classificadores por dataset:

Dataset Iris: Foi obtido 0.96 de acurácia tanto com ambos os classificadores utilizando learning rate igual a 1 com número de iterações igual a 400. Além disso os parâmetros utilizados no classificador MLP foram o solver `lbfgs`, frequência de learning rate constante e função de ativação logística e 3 camadas com 20 neurônios em cada camada.

Dataset Artificial: Foi obtido 0.8 de acurácia utilizando o MLP e 0.66 utilizando o Perceptron utilizando uma learning rate igual a 1 e um número de iterações igual a 100. Além disso no classificador MLP o solver utilizado foi o `lbfgs`, frequência de learning rate constante e função de ativação `relu` e 3 camadas com 20 neurônios em cada camada.

Dataset XOR: Foi obtido 1.0 de acurácia utilizando o MLP e somente 0.5 utilizando o Perceptron. Os parâmetros utilizados no `MLPClassifier` que geraram essa acurácia foram diversos. Um exemplo é o learning rate igual a 1.0, com 100 iterações, solver `lbfgs`, uma frequência de learning rate constante e função de ativação logística e 1 camada com 5 neurônios. No Classificador Perceptron que obteve acurácia de 50% foi utilizado learning rate igual a 1.0 e 200 iterações.

### 4 Conclusão

É possível notar que a maior acurácia obtida foi encontrada no dataset com menos features (XOR) e maior conjunto de dados utilizando MLP, o que indica que quanto maior o conjunto de dados "verticalmente" e menor "horizontalmente" no sentido de linhas por colunas mais fácil a otimização de problema utilizando esse classificador. Em contrapartida não foram obtidos bom resultados com o Perceptron nesse mesmo dataset acredito eu pela dificuldade de otimizar o problema mesmo estressando o mesmo com mais iterações e uma learning rate diferente. Nos outros dois datasets as acurácias máximas obtidas não são tão diferentes, no dataset Iris obtemos 0.96 de acurácia chegando próximo ao 100% enquanto no dataset artificial vemos um limite máximo no MLP classificador de 0.8 de acurácia e 0.66 para o Perceptron, acredito eu que por conta do alto número de features e um conjunto de dados não muito volumoso.