

**MBA<sup>+</sup>**

# MBA em Artificial Intelligence e Machine Learning

**MBA<sup>+</sup>**

**Deep Learning**

**Profª Felipe Teodoro.**

[Proffelipe.teodoro@fiap.com.br](mailto:Proffelipe.teodoro@fiap.com.br)

2016



# Trabalho Final

## Considerações Gerais

Todos os exercícios devem ser entregues no prazo estabelecido, em caso de atraso será descontado 1 ponto por dia de atraso;

- Apenas os exercícios desta lista devem ser entregues;
- Justifique as suas escolhas de implementação;
- Não esqueça que é sempre importante normalizar os dados;
- Alguns exercícios levam um certo tempo para treinar os modelos solicitados, logo não deixe o trabalho para os últimos dias;
- Qualquer tipo de plágio previsto no regulamento da FIAP será penalizado com a anulação do exercício por completo. (Cópia de código estrutural trocando apenas “comentários” por exemplo / idêntico). É permitida a consulta de ideias parâmetros devidamente explicados, mas cópia idêntica de código não;
- Os exercícios 5.1 e 8 são obrigatórios, os alunos poderão selecionar os outros exercícios a serem desenvolvidos até completar 10 pontos;
- Todos os exercícios devem ser entregues em formato de notebook (ipynb);

## Redes Neurais Multicamadas – Aprendizado

### Exercício #3

Utilizando as bibliotecas do scikit-learn faça:

- Utilizando do dataset Iris (embarcado no scikit-learn), configure uma Rede Neural MLP para classificar as amostras de plantas nas 3 classes presentes no dataset.
- Utilizando o dataset Heart Disease UCI (disponível no repositório da disciplina e em <https://www.kaggle.com/ronitf/heart-disease-uci>) configure uma Rede Neural MLP que classifique as amostras em saudáveis (coluna target =0) ou não.
- Utilizando o dataset Pulsar Star (disponível no repositório da disciplina e em <https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star/>) construa uma Rede Neural MLP que classifique as amostras em estrelas de nêutrons ou não.

## Redes Neurais Multicamadas – Aprendizado

### Exercício #3

Utilizando as bibliotecas do scikit-learn faça:

- Dicas e recomendações:
  - Procure utilizar os conceitos de GridSearch vistos anteriormente.
  - É necessário calibrar o parâmetro alpha (regularização) neste momento.

## Exercício #4

Utilizando a biblioteca `elm` do pacote `sklearn_extensions.extreme_learning_machines` e o dataset Mushroom Classification (disponível no repositório da disciplina e em <https://www.kaggle.com/uciml/mushroom-classification>) encontre a configuração de ELM que atinge a melhor acurácia .

Dicas:

- Utilize o exemplo `plot_elm_comparison.py` como base;
- Não é necessário plotar as fronteiras de decisão do classificador;

## Exercício #5

Utilizando o dataset do Spotify (disponível no repositório da disciplina) construa um modelo sequencial ou funcional no Keras para fazer a classificação desse dataset.

Dicas:

Tente calibrar a quantidade de camadas e neurônios a fim de atingir uma taxa de acerto semelhante a um SVM otimizado;

Tente utilizar o dropout a fim de evitar eventual overfitting.

- Aplique os pré-processamentos necessários.
- Apresente a evolução do conjunto de treino e validação.
- Apresente a acurácia do conjunto de testes.
- Apresente a configuração (estrutura) da rede neural.

## Exercício #5.1

Utilizando o dataset do Exercício5.1.csv (disponível no repositório da disciplina) construa um modelo sequencial ou funcional no Keras para fazer a classificação desse dataset que é **multiclass**.

Tente calibrar a quantidade de camadas e neurônios a fim de atingir uma taxa de acerto superior a **75%** no conjunto de teste.

- Aplique os pré-processamentos necessários.
- Apresente a evolução do conjunto de treino e validação.
- Apresente a acurácia do conjunto de testes.
- Apresente a configuração (estrutura) da rede neural.
- O modelo não pode estar superajustado (“*overfitado*”) utilize as técnicas vistas em sala de aula para prevenir ou mitigar esse problema.



## Exercício #6.2

Utilizando o exemplo “Exemplo CNN Dataset CIFAR10.py” aumente a taxa de acerto atual (que em cerca de 60%) em ao menos 18% (totalizando 78-80% de acerto aproximadamente) calibrando a CNN.

Dicas:

- Tente alterar a quantidade de épocas;
  - Tente alterar o tamanho do batch\_size;
  - Tente alterar o tamanho dos filtros convolucionais;
    - Cuidado para não gerar um conjunto de filtros inválidos
  - Tente alterar a função de ativação na camada completamente conectada.
- 
- Aplique os pré-processamentos necessários.
  - Apresente a evolução do conjunto de treino e validação.
  - Apresente a acurácia do conjunto de testes.
  - Apresente a configuração (estrutura) da rede neural.

## Exercício #7

Utilizando o dataset Fashion MNIST, construa uma CNN para a classificação do dataset.

Dicas:

- Utilize a LetNet-5 como exemplo bem como o exemplo visto em sala de aula do dataset mnist.
- Utilize o comando `(x_train, y_train), (x_test, y_test) = tf.keras.datasets.fashion_mnist.load_data()` para carregar o dataset.
- Os labels da classe são os seguintes:

0 T-shirt/top	5 Sandal
1 Trouser	6 Shirt
2 Pullover	7 Sneaker
3 Dress	8 Bag
4 Coat	9 Ankle boot

## Exercício #8

Inicialmente os alunos deverão escolher entre 4 datasets:

#1 – Selecionar 15 frutas distintas do dataset fruits 360 (Fácil)

#2 – Selecionar todas as frutas do dataset fruits 360 (Médio)

Disponível em:

<https://www.dropbox.com/s/kzhuv6xim8ejxp3/fruits.zip?dl=0>

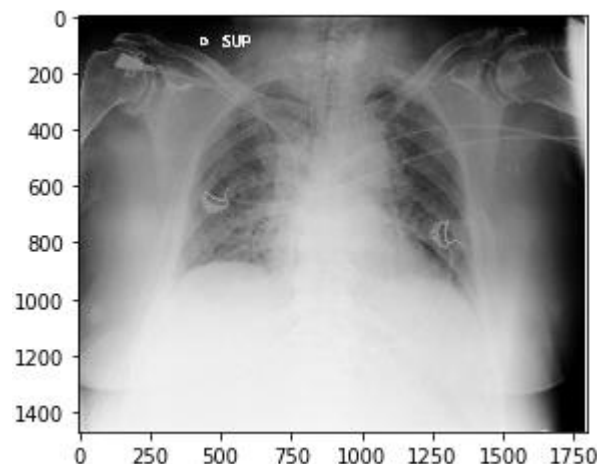
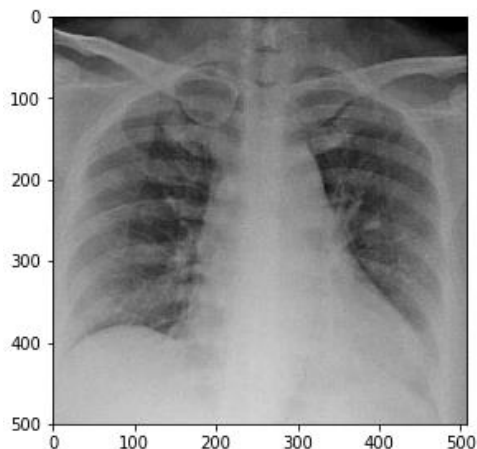


## Exercício #8

Inicialmente os alunos deverão escolher entre 4 datasets:

#3 –CoronaHack -Chest X-Ray-Dataset (Médio):

Classificar apenas se existe Pneumonia ou Não, **opcionalmente** classificar a existência de COVID-19:



Disponível em:

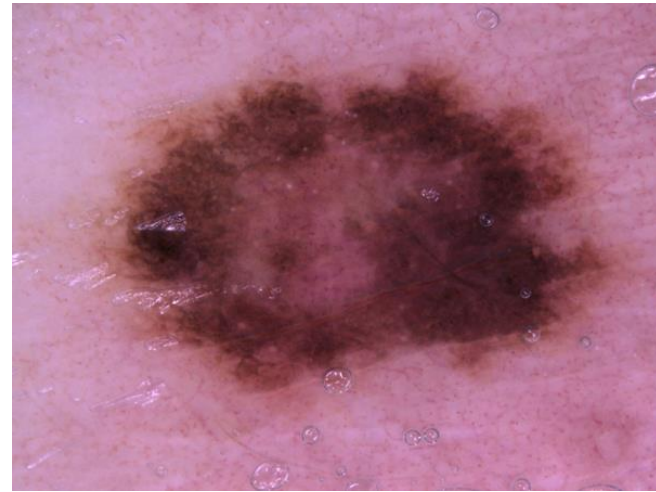
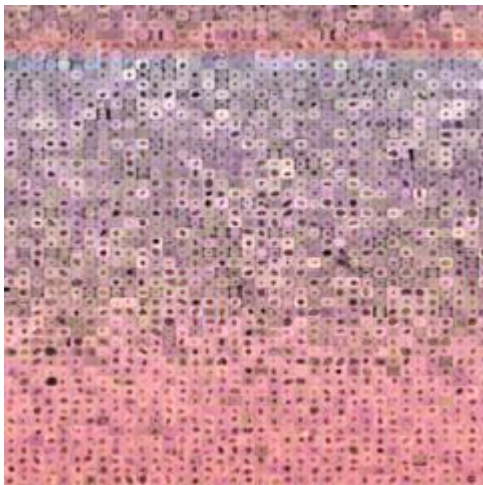
[https://www.dropbox.com/s/zmo4c6pzwcgd9c4/562468\\_1022626\\_bundle\\_archive.zip?dl=0](https://www.dropbox.com/s/zmo4c6pzwcgd9c4/562468_1022626_bundle_archive.zip?dl=0)

## Exercício #8

Inicialmente os alunos deverão escolher entre 4 datasets:

#4 – Dataset Skin Cancer MNIST: HAM10000 (difícil)

Disponível em: <https://www.dropbox.com/s/8o4ysiccspp46ov/skin-cancer-mnist-ham10000.zip?dl=0>



## Exercício #8

Com o dataset selecionado deve-se construir uma CNN para a classificação das imagens do dataset selecionado

Instruções Gerais:

- Utilize a LetNet-5 como exemplo inicial bem como o exemplo visto em sala de aula do dataset minst;
- Inicie o trabalho com parte do dataset vá adicionando gradativamente mais tipos distintos de frutos;
- É permitido usar outras abordagens e arquiteturas;
- É permitido o uso de Autoencoders;
- É obrigatório o uso de CNN;
- É permitido o uso de técnicas de processamento de imagens;
- Não é permitido o uso de modelos pré-treinados;
- Faça uma boa distribuição dos dados de teste e treinamento e validação;

## Exercício #8

Com o dataset selecionado deve-se construir uma CNN para a classificação das imagens do dataset selecionado

Instruções Gerais:

- Aplique os pré-processamentos necessários.
- Apresente a evolução do conjunto de treino e validação.
- Apresente a acurácia do conjunto de testes **(opcional)**.
  - O conjunto de teste deve variar entre 20% e 30% do conjunto total de treinamento e deve ter a mesma proporção da quantidade de amostras por classes de forma aleatório. Para o dataset fruits360 o conjunto precisa ser construído, em outras palavras é necessário fazer uma rotina que faça isso, selecionando imagens de forma aleatória e respeitando a proporcionalidade de amostras por classes (todo tipo de fruta deve ter a mesma quantidade).
- Apresente a configuração (estrutura) da rede neural.

## Exercício #8

Dicas:

Utilize as instruções de leituras de diretórios de imagens do próprio Keras (semelhante ao exemplo visto com o dataset de dígitos):

```
train_datagen =  
preprocessing.image.ImageDataGenerator(validation_split=0.20)  
train_generator = train_datagen.flow_from_directory(  
    'Dataset',  
    batch_size=200,  
    color_mode = 'grayscale',  
    target_size=(30, 30))
```



## Exercício #9

Utilizando o dataset BVMF3 (disponível em: <https://www.dropbox.com/s/lbdrugh5thzvvrk/BVMF3.SA.csv?dl=0>) faça uma previsão de série temporal utilizando LSTM para prever 30 dias do valor de fechamento a partir do registro #1500 do dataset.

### Instruções Gerais:

- É permitido usar outras abordagens e arquiteturas de LSTM;
- Não é permitido o uso de modelos pré-treinados;
- Faça uma boa distribuição dos dados de teste e treinamento;

# Trabalho Final

## Tabela de Avaliação:

Exercício 8 (obrigatório)	5 pontos
Exercícios 5.1 (obrigatório) e 9 (opcional)	2 pontos cada
Outros Exercícios	1 ponto cada

**MBA<sup>+</sup>**

Copyright © 2020 Prof. Msc. Felipe Teodoro

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).