

## PÓS GRADUAÇÃO EM CIÊNCIA DE DADOS - FUNDAMENTOS DE INTELIGÊNCIA ARTIFICIAL

Trabalho final

Questão 1: Otimização Bayesiana

NOMES - MATRÍCULAS:

Silvio Cesar de Santana - 2017103163

Marcel Tinoco Ribeiro - 2020101920

Ruan Braz de Araujo - 2019101232

Luis Claudio Simões Baptista - 2015100108

### Rede Neural - Otimização Bayesiana

#### Optuna – Ajuste de hiper parâmetros

**\*\*Treino / Validação\*\***

- Camada Neurônios
- Camada Invisível (Oculta)
- Camada Saída

#### Testando Ativação camada saída e Otimizadores

10 EPOCHS 50 TRIALS

Dados para validação : 100

**\*\*Camada de saída Ativação ( linear )**

Otimizador ( adam )\*\* Resultado 0.0005820

**\*\*Camada de saída Ativação ( linear )**

Otimizador ( sgd )\*\* Resultado 0.50194870

**\*\*Camada de saída Ativação ( linear )**

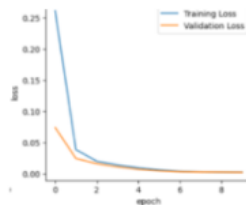
Otimizador ( RMSprop )\*\* Resultado 0.0004972

#### No retreino

##### No retreino

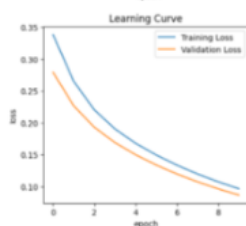
##### Utilizando linear e adam

```
Epoch 6/10
25/25 [=====] - 0s 4ms/step - loss: 0.0055 - val_loss: 0.0040
Epoch 7/10
25/25 [=====] - 0s 4ms/step - loss: 0.0034 - val_loss: 0.0027
Epoch 8/10
25/25 [=====] - 0s 3ms/step - loss: 0.0024 - val_loss: 0.0022
Epoch 9/10
25/25 [=====] - 0s 3ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 10/10
25/25 [=====] - 0s 3ms/step - loss: 0.0018 - val_loss: 0.0017
```



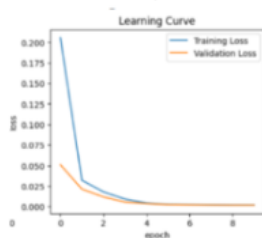
##### Utilizando linear e sgd

```
Epoch 6/10
25/25 [=====] - 0s 3ms/step - loss: 0.1402 - val_loss: 0.1336
Epoch 7/10
25/25 [=====] - 0s 3ms/step - loss: 0.1334 - val_loss: 0.1195
Epoch 8/10
25/25 [=====] - 0s 10ms/step - loss: 0.1194 - val_loss: 0.1071
Epoch 9/10
25/25 [=====] - 0s 4ms/step - loss: 0.1071 - val_loss: 0.0961
Epoch 10/10
25/25 [=====] - 0s 3ms/step - loss: 0.0962 - val_loss: 0.0864
```



##### Utilizando linear e RMSprop

```
Epoch 6/10
25/25 [=====] - 0s 4ms/step - loss: 0.0028 - val_loss: 0.0025
Epoch 7/10
25/25 [=====] - 0s 4ms/step - loss: 0.0025 - val_loss: 0.0022
Epoch 8/10
25/25 [=====] - 0s 4ms/step - loss: 0.0023 - val_loss: 0.0020
Epoch 9/10
25/25 [=====] - 0s 4ms/step - loss: 0.0020 - val_loss: 0.0018
Epoch 10/10
25/25 [=====] - 0s 4ms/step - loss: 0.0019 - val_loss: 0.0018
```



**\*\*Linear e Adam\*\*** - loss: 0.0018 - val\_loss: 0.0017

**\*\*Linear e SGD\*\*** - loss: 0.0962 - val\_loss: 0.0864

**\*\*Linear e RMSprop\*\*** - loss: 0.0019 - val\_loss: 0.0018

Melhor resultado : **\*\*Linear e Adam\*\***

#### Testando outros ajustes de Ativação saída e Otimizadores

Camada de saída Ativação ( sigmoid )                      Otimizador ( adam ) Resultado 0.10225773  
Camada de saída Ativação ( softmax )                      Otimizador ( adam ) Resultado 0.1675887

## Alterando parâmetros

Se apenas ajustar no parametro da segunda camada densa de 100 para 200, obteremos o seguinte resultado:

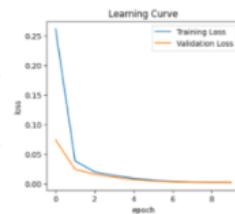
```
def objective(trial):  
    model = Sequential()  
    model.add(Dense(trial.suggest_int('n_units1', 4, 128),  
                    activation='softmax'))  
    model.add(Dense(trial.suggest_int('n_hidden1', 2, 200),  
                    activation='sigmoid'))
```

Resultado 0.000319764  
Best params: {'n\_units1': 113, 'n\_hidden1': 189}

Houve uma redução consideravel no valor final que era de **0.000582063** e foi para **0.000319764**

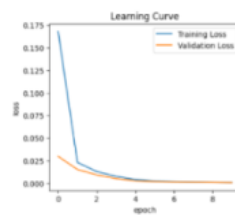
### Anterior com intervalo até 100

```
Epoch 6/10  
25/25 [=====] - 0s 4ms/step - loss: 0.0055 - val_loss: 0.0040  
Epoch 7/10  
25/25 [=====] - 0s 4ms/step - loss: 0.0034 - val_loss: 0.0027  
Epoch 8/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0034 - val_loss: 0.0022  
Epoch 9/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0019 - val_loss: 0.0020  
Epoch 10/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0018 - val_loss: 0.0017
```



### Aumentando intervalo p/ 200

```
Epoch 6/10  
25/25 [=====] - 0s 4ms/step - loss: 0.0027 - val_loss: 0.0020  
Epoch 7/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0018 - val_loss: 0.0013  
Epoch 8/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0014 - val_loss: 0.0013  
Epoch 9/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0012 - val_loss: 0.0011  
Epoch 10/10  
25/25 [=====] - 0s 3ms/step - loss: 0.0009e-04 - val_loss: 7.0024e-04
```



E nos valores de retreino

- loss: 8.8660e-04 - val\_loss: 7.0824e-04 que eram:  
- loss: 0.0018 - val\_loss: 0.0017

## Ajustando para 30 EPOCHS e 10 TRIALS

Quando utilizamos os ajustes de 30 EPOCHS 10 TRIALS e a alteração de 100 p/200 na segunda camada, temos os valores:

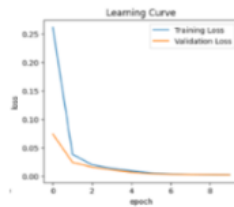
Trial 6 finished with value: 5.4340678616426885e-05 and parameters: {'n\_units1': 75, 'n\_hidden1': 113}. Best is trial 2 with value: 5.4216776334214956e-05.  
Trial 7 finished with value: 6.456566188717261e-05 and parameters: {'n\_units1': 86, 'n\_hidden1': 88}. Best is trial 2 with value: 5.4216776334214956e-05.  
Trial 8 finished with value: 4.575316415866837e-05 and parameters: {'n\_units1': 128, 'n\_hidden1': 86}. Best is trial 8 with value: 4.575316415866837e-05.  
Trial 9 finished with value: 0.0003734398924279958 and parameters: {'n\_units1': 61, 'n\_hidden1': 25}. Best is trial 8 with value: 4.575316415866837e-05.

Best params: {'n\_units1': 128, 'n\_hidden1': 86}

## E no retreino

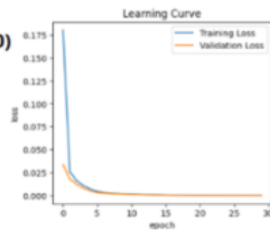
## Utilizando linear e adam 10 EPOCHS 50 TRIALS (Intervalo 2ª camada até 100)

```
Epoch 6/10
25/25 [=====] - 0s 4ms/step - loss: 0.0055 - val_loss: 0.0040
Epoch 7/10
25/25 [=====] - 0s 4ms/step - loss: 0.0034 - val_loss: 0.0027
Epoch 8/10
25/25 [=====] - 0s 3ms/step - loss: 0.0024 - val_loss: 0.0022
Epoch 9/10
25/25 [=====] - 0s 3ms/step - loss: 0.0019 - val_loss: 0.0020
Epoch 10/10
25/25 [=====] - 0s 3ms/step - loss: 0.0018 - val_loss: 0.0017
```



## Utilizando linear e adam 30 EPOCHS 10 TRIALS (Intervalo 2ª camada até 200)

```
Epoch 25/30
25/25 [=====] - 0s 5ms/step - loss: 1.3201e-04 - val_loss: 1.3470e-04
Epoch 26/30
25/25 [=====] - 0s 5ms/step - loss: 1.4510e-04 - val_loss: 1.4340e-04
Epoch 27/30
25/25 [=====] - 0s 4ms/step - loss: 6.6571e-05 - val_loss: 6.4210e-05
Epoch 28/30
25/25 [=====] - 0s 5ms/step - loss: 6.8123e-05 - val_loss: 6.2586e-05
Epoch 29/30
25/25 [=====] - 0s 5ms/step - loss: 6.1021e-05 - val_loss: 6.2076e-05
Epoch 30/30
25/25 [=====] - 0s 5ms/step - loss: 7.8692e-05 - val_loss: 7.8103e-05
```



Obtemos os resultados:- loss: 7.8692e-05 - val\_loss: 7.8103e-05

Que antes era - loss: 0.0018 - val\_loss: 0.0017

Uma melhora, nos resultados obtidos.

## Conclusão

Alterando de 10 EPOCHS e 50 TRIALS para 30 EPOCHS e 10 TRIALS tem-se o resultado:

Camada de saída Ativação ( linear ) Otimizador ( adam )

TREINO (loss)

**\*\*Otimizadores - Resultados com 10 EPOCHS e 50 TRIALS\*\*** (após retreino)

**\*\*ADAM\*\*** loss: 0.0018 - val\_loss: 0.0017

**\*\*SGD\*\*** loss: 0.0962 - val\_loss: 0.0864

**\*\*RMSprop\*\*** loss: 0.0019 - val\_loss: 0.0018

Obs:

Utilizando otimizador sgd, a partir do trial 3 o erro estabiliza até o trial 32, onde ocorre um ajuste no trial 33 até o final.

Nesse teste o adam se mostrou mais eficiente, com resultado 0.0018 no treino e 0.0017, na validação. Quanto a ativação na camada de saída, o mais apropriado foi utilizar linear, pois utilizando softmax, os resultados mostraram que a partir do trial 3, não há melhora dos valores.

Nesses testes, utilizando 5 retreinos, o **\*\*ADAM\*\*** obteve um melhor resultado final. O RMSprop obteve melhor resultado apenas no primeiro retreino, porém nas demais o desempenho ficou instável.

**\*\*Comparando funções de ativação das camadas\*\***

Resultados com 30 EPOCHS e 10 TRIALS

**\*\*Relu\*\*** Best is trial 4 with value: 3.298554656794295e-05

Best params: {'n\_units1': 117, 'n\_hidden1': 188}

**\*\*Elu\*\*** Best is trial 9 with value: 0.00043812114745378494

Best params: {'n\_units1': 128, 'n\_hidden1': 161}

**\*\*LeakyReLU\*\*** Best is trial 3 with value: 9.78622047114186e-05

Best params: {'n\_units1': 100, 'n\_hidden1': 128}

**\*\*silu\*\*** Best is trial 6 with value: 3.4275915822945535e-05

Best params: {'n\_units1': 117, 'n\_hidden1': 173}

**\*\*No retreino\*\***

**\*\*Relu\*\*** loss: 5.6859e-05 - val\_loss: 6.2403e-05

**\*\*Elu\*\*** loss: 5.6140e-04 - val\_loss: 5.1519e-04

**\*\*LeakyReLU\*\*** loss: 3.1283e-04 - val\_loss: 2.6427e-04

Utilizando **\*\*silu\*\*** no retreino, foi verificado inconsistências, obtendo resultados não satisfatórios.

Sendo necessárias outras combinações de ajustes.

**Em todos os testes foram feitos retreinos no mínimo 5 vezes, obtendo resultados satisfatórios sem muitas alterações nos resultados finais.**

**Foram ajustados somente alguns parâmetros, como, número de camadas escondidas, Otimizadores, Ativação de entrada e saída, tendo ainda muitos outros ajustes possíveis, como exemplo, combinar varias ativações de entrada, adicionar mais camadas densas, ...**

