

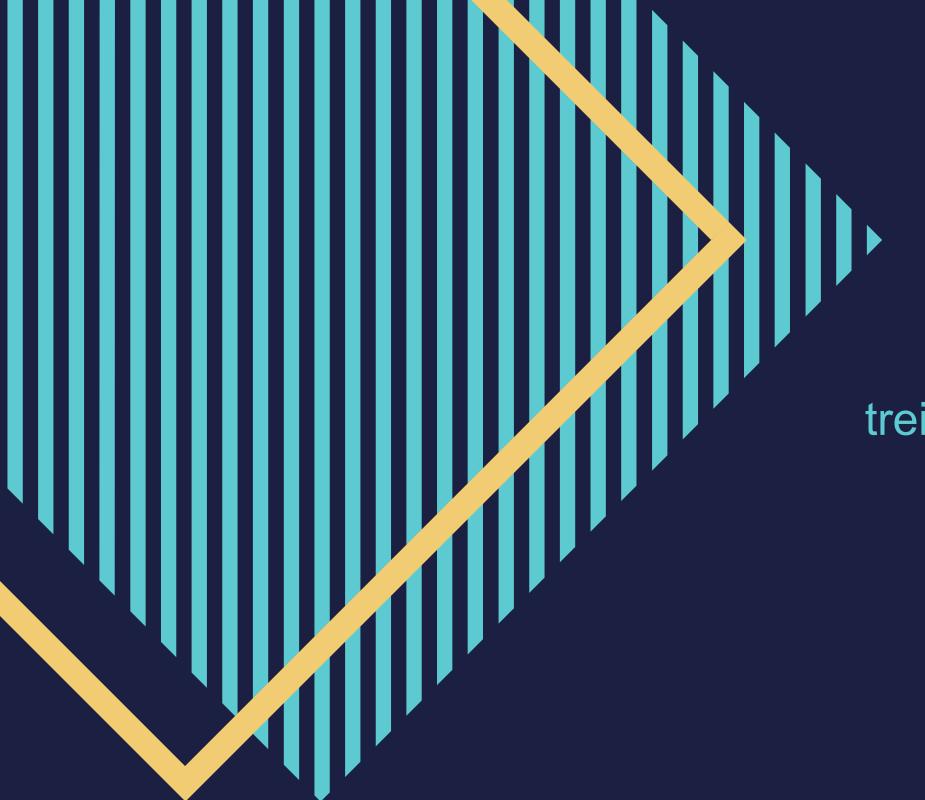
Universidade de São Paulo Escola de Artes, Ciências e Humanidades ACH2016 - Inteligência Artificial



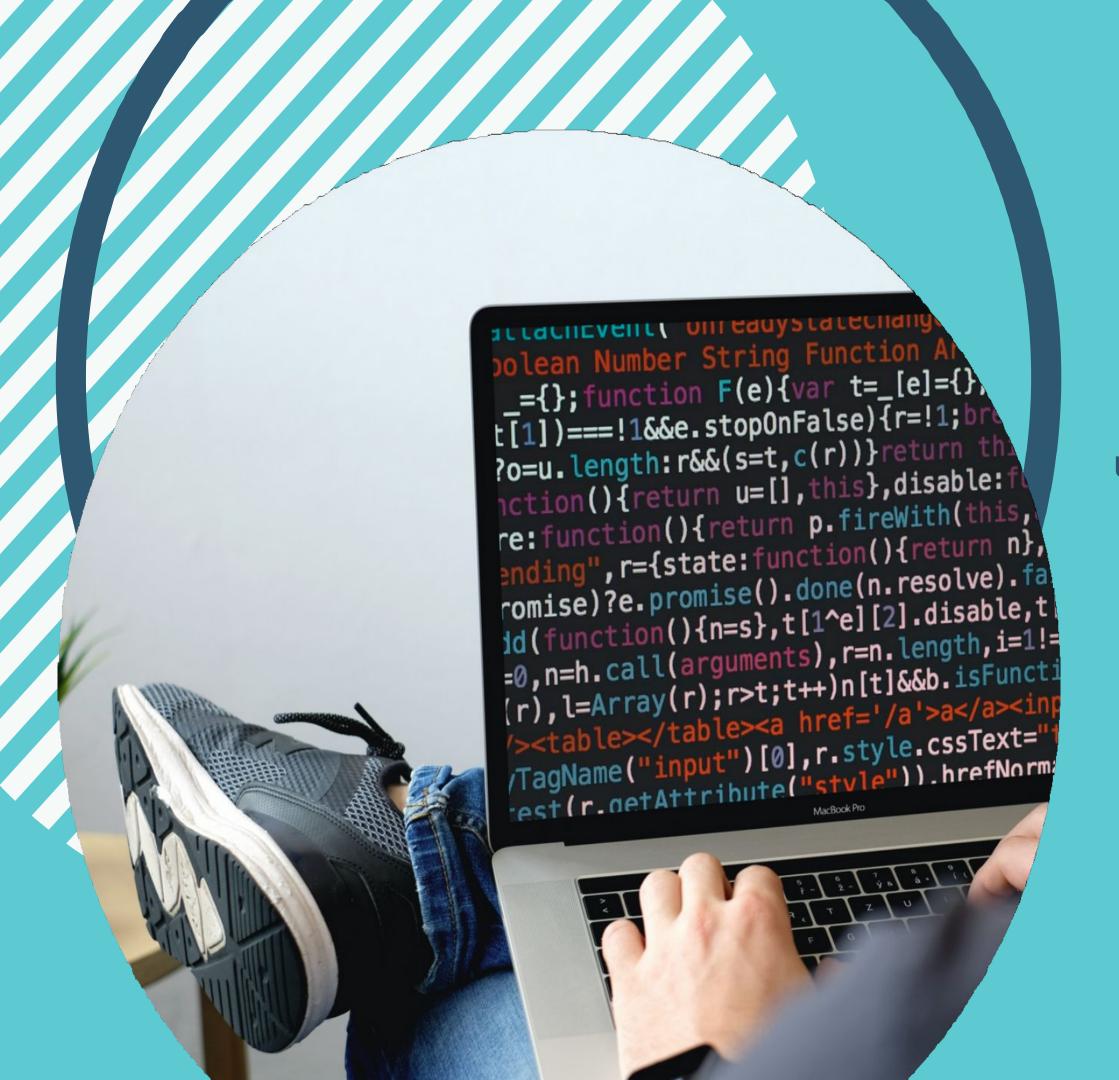


Fábio Heiji Yamada - 5690618 Heitor Lopes Bianchi - 10258730 Ivan Nascimento da Costa - 10257750 Mariana Silva Santana - 10258897





1- O Experimento
2- Desempenho da Rede
3- Erros obtidos pelo modelo
4- Curva de Aprendizado
5- Relação entre o conjunto de
treinamento e o desempenho do modelo
6- Análise do Cross Validation
7- Conclusão



O EXPERIMENTO

A escolha dos hiperparâmetros utilizados se deu baseado nos testes realizados considerando uma camada oculta. Desse modo, foi possível inferir que combinações com muitos nós e com um valor baixo do termo de erro resultava em modelos ruins. Por fim, com base nos testes, decidiu-se usar o termo momento e uma taxa de aprendizado baixa.



OS HIPERPARÂMETROS

Foram avaliados quatro hiperparâmetros:

- -ETA (taxa de aprendizado)
- -Alpha (termo momento)
- -Quantidade de nós da camada oculta
- -Épocas

Vale ressaltar que quanto mais nós aplicamos ao modelo, mais estamos tentando explicá-lo na base de treinamento, haja vista que treinando esse modelo em cima da base ele acaba ficando viciado, configurando um caso de overfitting.

Com base nos testes, chegamos a conclusão que a quantidade de **épocas** que trazia os **melhores** resultados, girava em torno de **120**. Além disso, a quantidade de **nós da camada oculta** girou em torno de **83** e o **Alpha** e o **ETA** por volta de **0.01**

Assim, é evidente que esses parâmetros interferem diretamente na acurácia da MLP.

PRINCIPAIS FERRAMENTAS E TECNOLOGIAS

- -Python
- -Jupyter Notebook
- -Google Colab

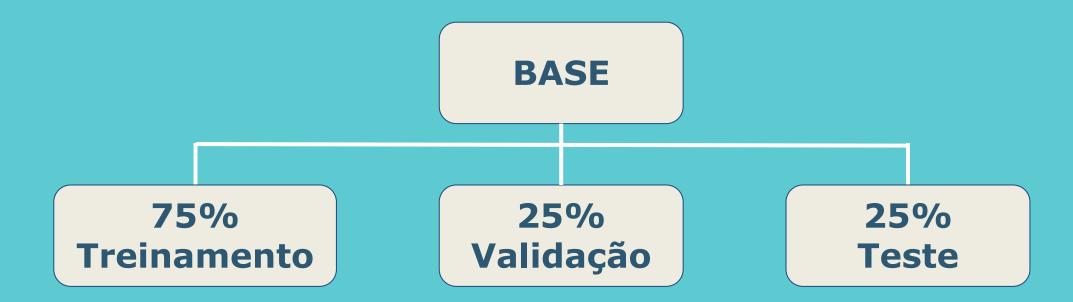
DIFICULDADES ENCONTRADAS

Inicialmente esquecemos de embaralhar os dados e, com isso não tivemos resultados satisfatórios no treinamento. Além disso, passamos bastante tempo buscando bons hiperparâmetros para o treinamento.

PREMISSAS DO MODELO

Dividimos a base da seguinte forma:

hold_out:



Utilizamos 1 camada oculta e a Função Sigmoide

Função de normalização:

(valor - média) / desvio_padrão

COMO EXECUTAR



TEMPO DE EXECUÇÃO COM MONO E MULTIPROCESSOS

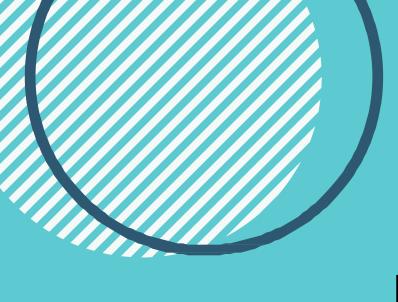
```
num_processos = 8
tempo = 6000ms
```

Multiprocessos

```
Tempo_Total_multi = 2000 * tempo / num_processos
Tempo_Total_multi = 1500s
```

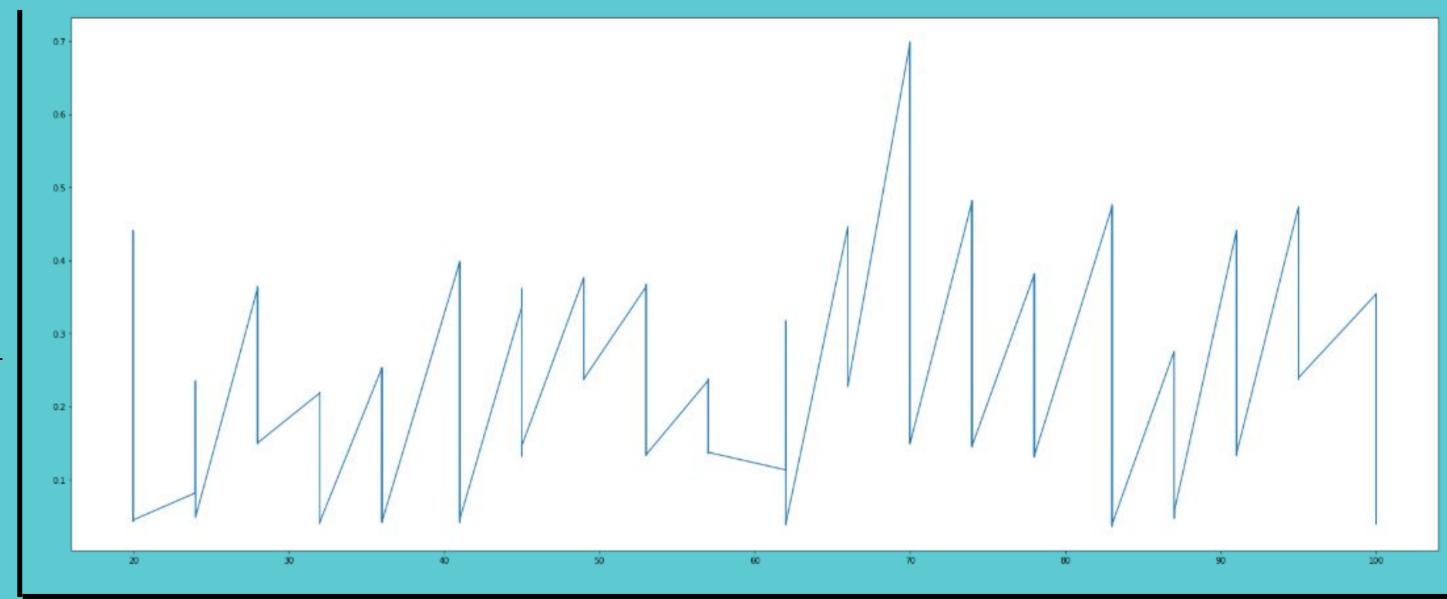
Mono

```
Tempo_Total_mono = 2000 * tempo
Tempo Total mono = 12000s
```



Quantidade de Nós da Camada Oculta x Erro Quadrático Médio

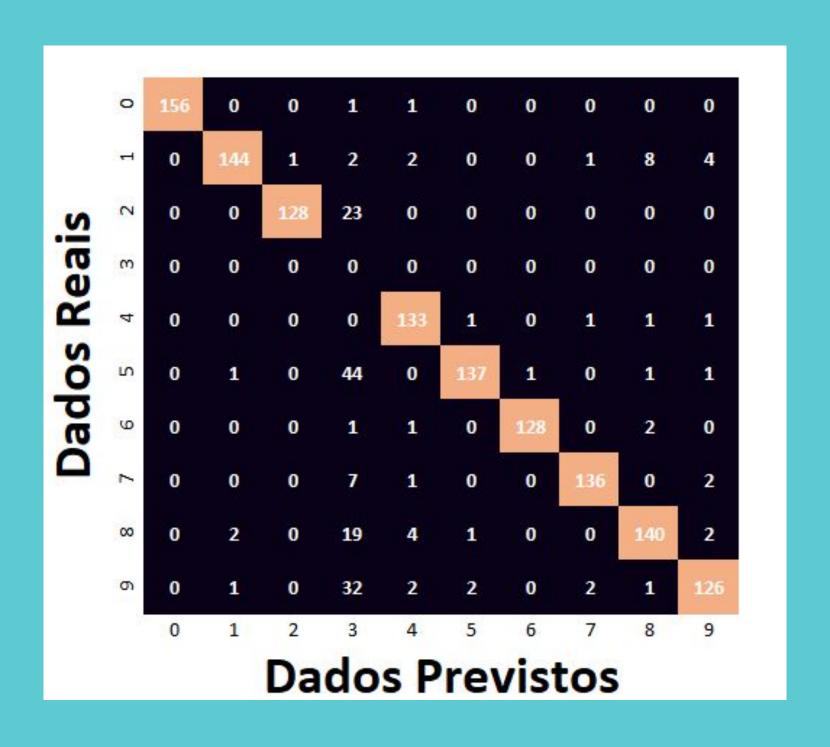




Quantidade de nós



MATRIZ DE CONFUSÃO DO HOLDOUT





SAÍDA DO HOLDOUT - ERRO VERDADEIRO

Nos da Camada Oculta Eta Alpha Acuracidade Treinamento Conjunto de Teste

83 0,02 0,017889 0,882562278 0,125978648

SAÍDA DO HOLDOUT ERROS X ÉPOCAS





MATRIZ DE CONFUSÃO DO CROSS VALIDATION



ANÁLISE DO CROSS VALIDATION

Nos da Camada Oculta Eta Alpha Acuracidade Treinamento Conjunto de Teste

83 0,02 0,017889 0,989991103 0,002197802

Index	Qtd.	Eta	Alpha	Erro_quadratico_medio_fold
0	83	0.02	0.017889	0,302895323
1	83	0.02	0.017889	0,113585746
2	83	0.02	0.017889	0,220489978
3	83	0.02	0.017889	0,202672606
4	83	0.02	0.017889	0,024498886
5	83	0.02	0.017889	0,013363029
6	83	0.02	0.017889	0,026726058
7	83	0.02	0.017889	0,008908686
8	83	0.02	0.017889	0,118040089
9	83	0.02	0.017889	0,098901099



CONCLUSÃO

Apesar estarmos tratando uma base bastante simples, o treinamento do modelo demorou bastante. Se não fosse base previamente tratada, não seria possível fazer tantos testes. Após alguns testes partimos para uma estratégia de paralelismo utilizando multiprocessos. Isso possibilitou que os testes fossem executados a tempo.

Ainda é possível debruçar-se sobre o experimento e desenrolar alguns desdobramentos, como testar com mais de uma camada, ou testar uma outra função de ativação, por exemplo.