



Aula Laboratório

# Redes Neurais Artificiais

*Aproximador de funções*

# Introdução

Inteligência Artificial e Robótica - CC 7711

Redes Neurais Artificiais

## Aplicações

MLP: Multi Layer Perceptron



Identificação de alvos militares:  
B-52, Boeing 747 e Space Shuttle



Reconhecimento de Faces



Autenticação de usuário.



Exploração de petróleo:  
Determinação e Litologia



Predição do Mercado Financeiro



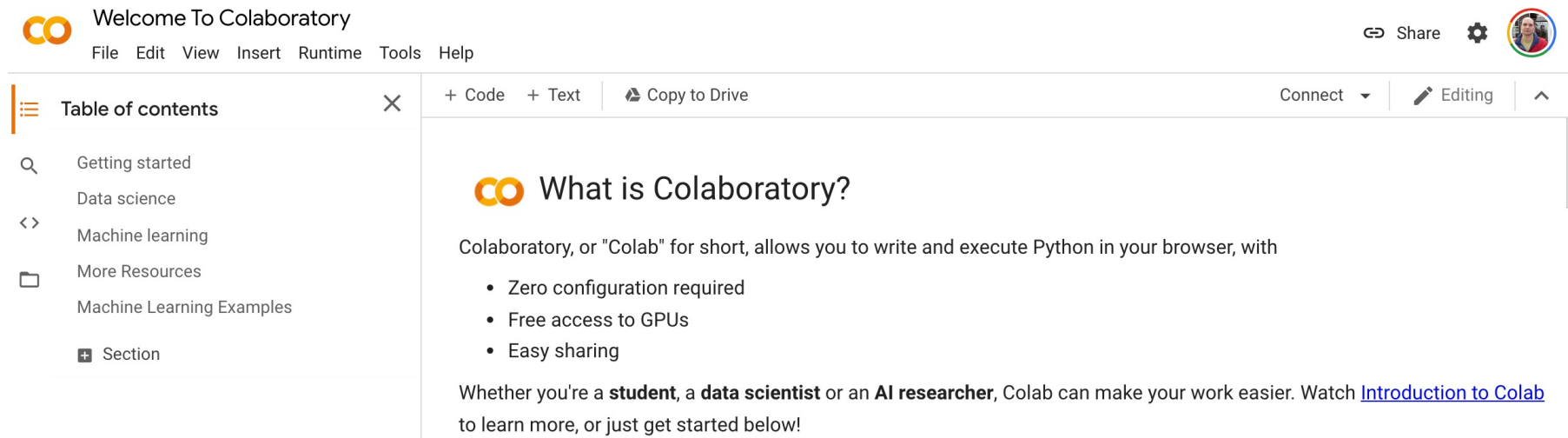
Controle de Navegação Autônoma de Veículos:  
ALVINN at CMU





# Python

*<https://colab.research.google.com/>*



The screenshot shows the Google Colaboratory (Colab) web interface. At the top, there's a header with the Colab logo, the text "Welcome To Colaboratory", and a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right of the header, there are icons for "Share", settings, and a user profile. Below the header, a sidebar on the left contains a "Table of contents" with links to "Getting started", "Data science", "Machine learning", "More Resources", "Machine Learning Examples", and a "Section" button. The main content area has a toolbar with "+ Code", "+ Text", and "Copy to Drive". The main content displays the "What is Colaboratory?" page, which includes a description of Colab, a bulleted list of features (Zero configuration required, Free access to GPUs, Easy sharing), and a paragraph encouraging users to watch an "Introduction to Colab" video.

co Welcome To Colaboratory  
File Edit View Insert Runtime Tools Help

Share ⚙️ 👤

☰ Table of contents ✕

- 🔍 Getting started
- Data science
- <> Machine learning
- 📁 More Resources
  - Machine Learning Examples
- ➕ Section

+ Code + Text 📄 Copy to Drive

Connect ▾ | ✎ Editing ^

## co What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

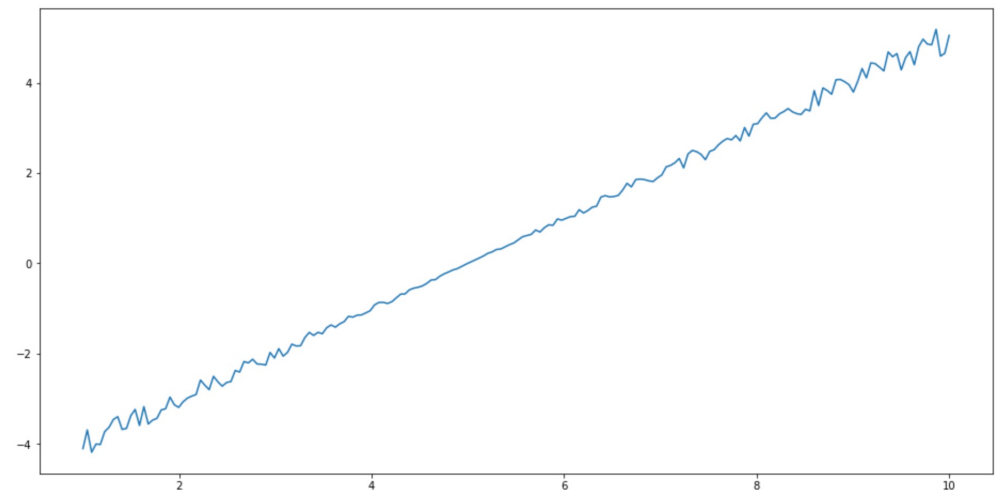
# Carregando as bases

```
▶ import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib inline
```

```
▶ arquivo = np.load('teste1.npy')
x = arquivo[0]
y = arquivo[1]

plt.figure(figsize=(16,8))
plt.plot(x,y)
plt.show()
```



# Carregando as bases



FEI - CC7711 - RNA AproximadorFuncoes.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Files



..



teste1.npy

teste2.npy

teste3.npy

teste4.npy

teste5.npy

+ Code + Text



0s

```
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


%matplotlib inline
```



0s

```
arquivo = np.load('teste1.npy')
x = arquivo[0]
y = arquivo[1]

plt.figure(figsize=(16,8))
plt.plot(x,y)
plt.show()
```



# Aproximação de funções usando MLP com Python

```
from sklearn.neural_network import MLPRegressor

regr = MLPRegressor(hidden_layer_sizes=(2,2),
                    max_iter=10000,
                    solver='adam',
                    learning_rate = 'adaptive',
                    n_iter_no_change=1000)

regr = regr.fit(x,y)

y_est = regr.predict(x)
```

[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html)



# Relatório

- Baseado no arquivo de exemplo, monte os aproximadores de função para os testes de 2 a 5 (disponíveis na atividade!)
  
- Faça pelo menos 3 simulações de arquitetura (Camadas e Qtdd e neurônios) para cada problema.
  - Cada simulação deve ser executada pelo menos 10x (apresente a media e o Desvio padrão do erro final)
  
- Incluir no relatório:
  - Link do Git com o Código-fonte
  - Gráficos com o melhor resultado das 3 simulações