Ciência de Dados
(Big Data Processing and Analytics)

Big Data Analytics – Mineração e Análise de Dados

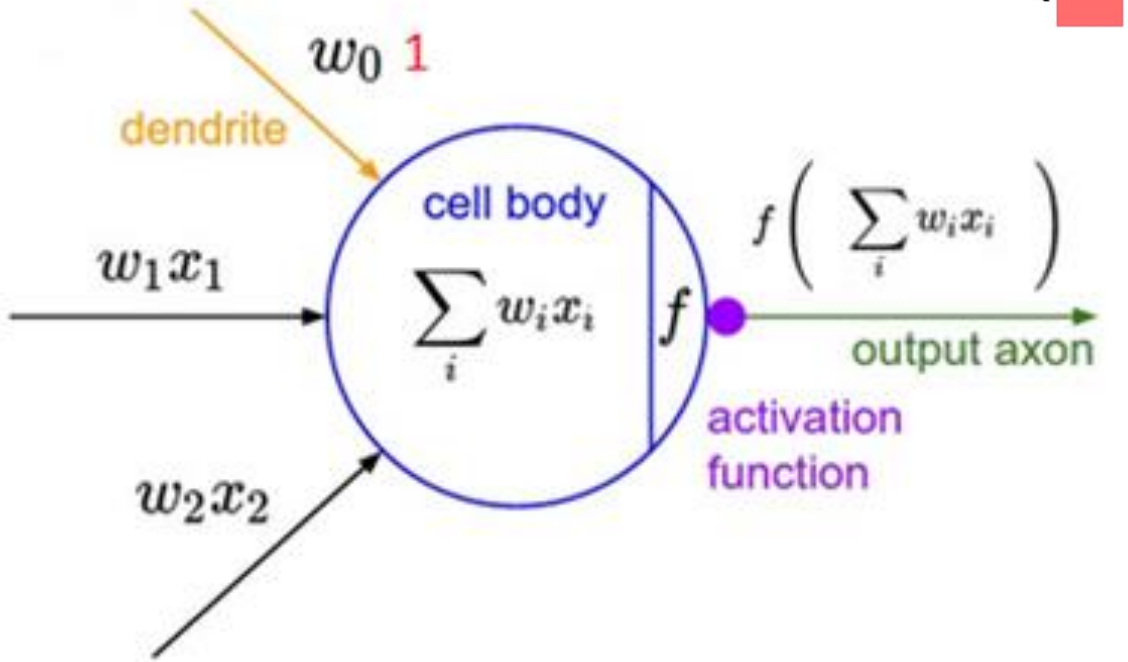
Professor curador
Prof. Dr. Rogério de Oliveira

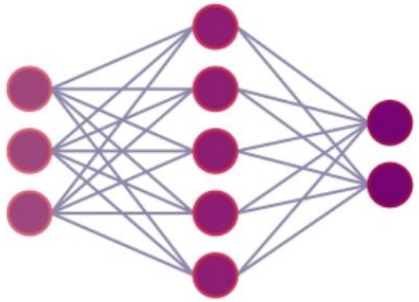
TRILHA 8
Redes Neurais e Deep Learning

Parte A

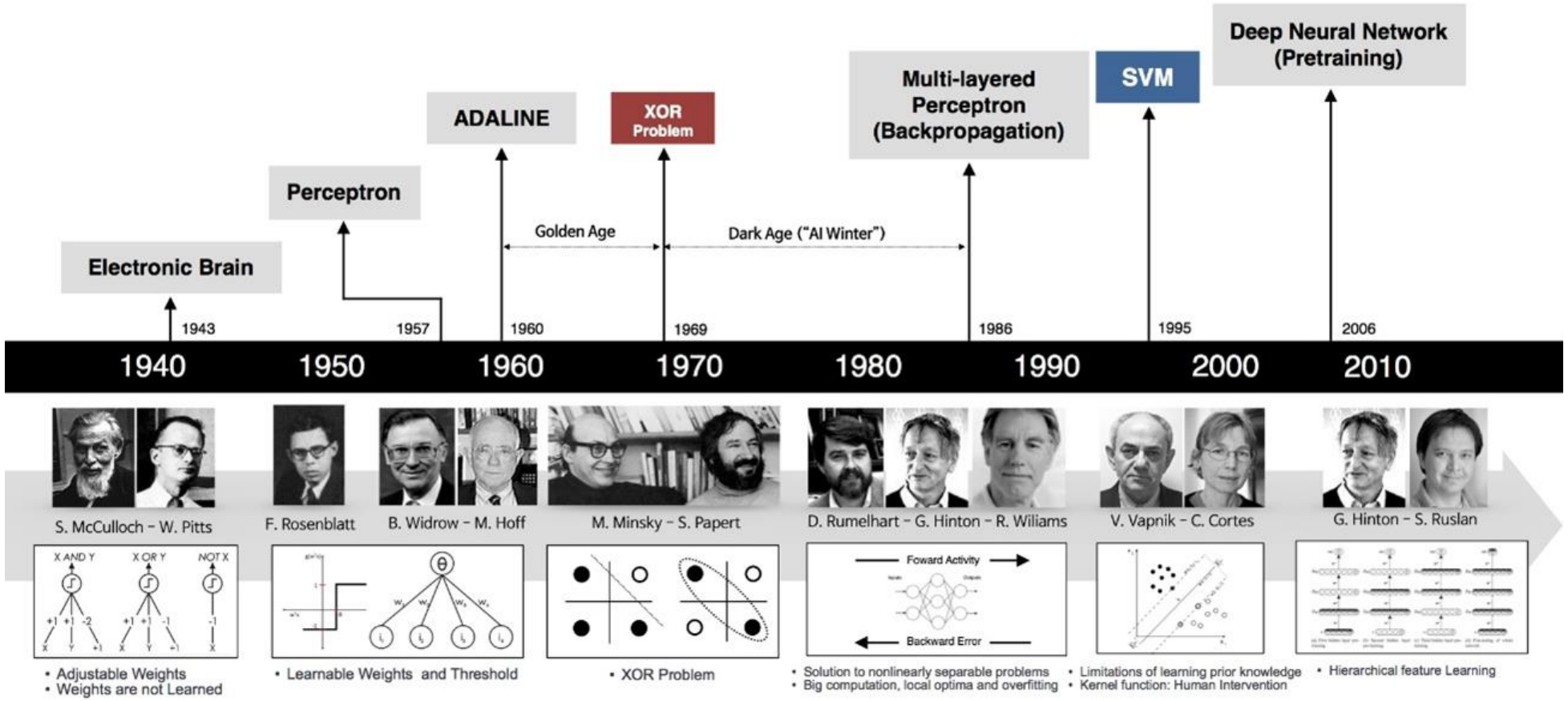
Neurônio Artificial

$$\sigma(\boxed{X} \times \boxed{W_{xh}}) \times \boxed{W_{hy}} = \boxed{Y}$$

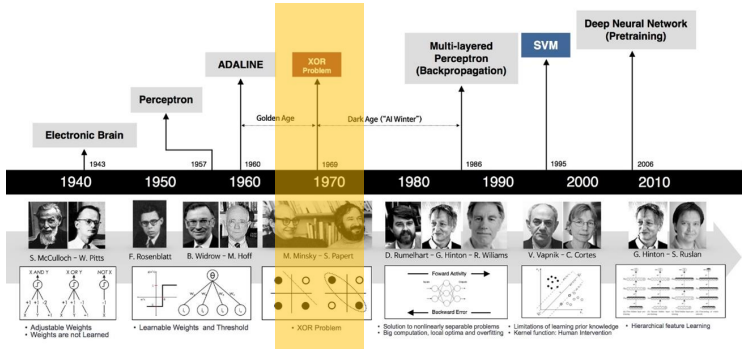


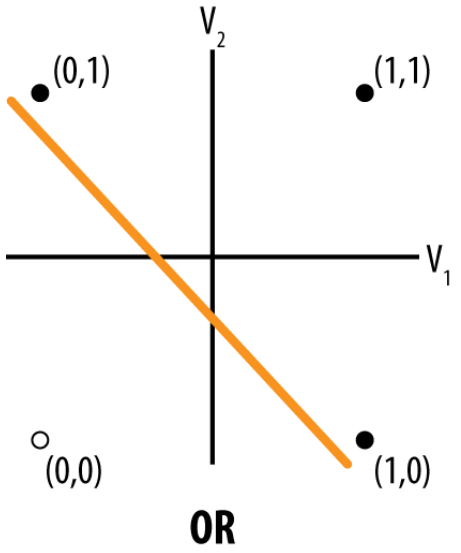


Redes Neurais: História

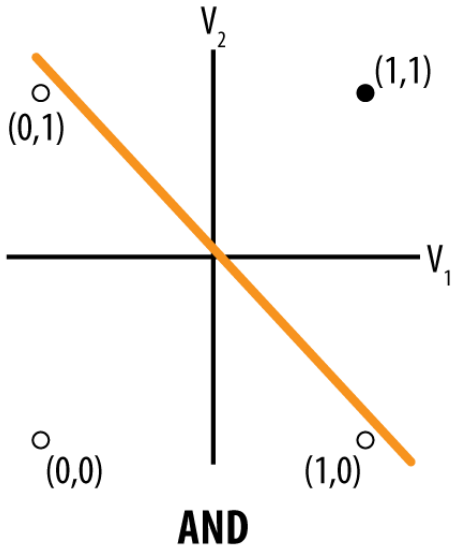


Redes Neurais: História

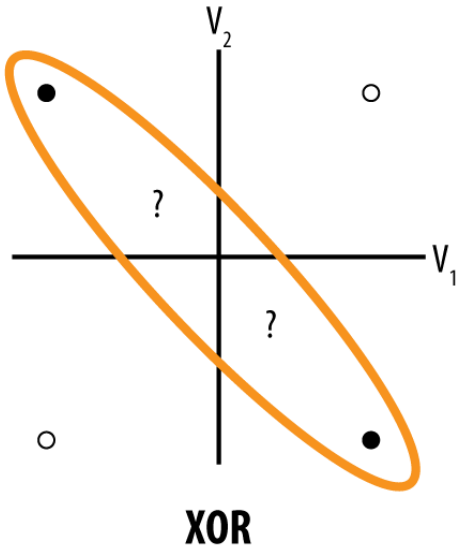




OR

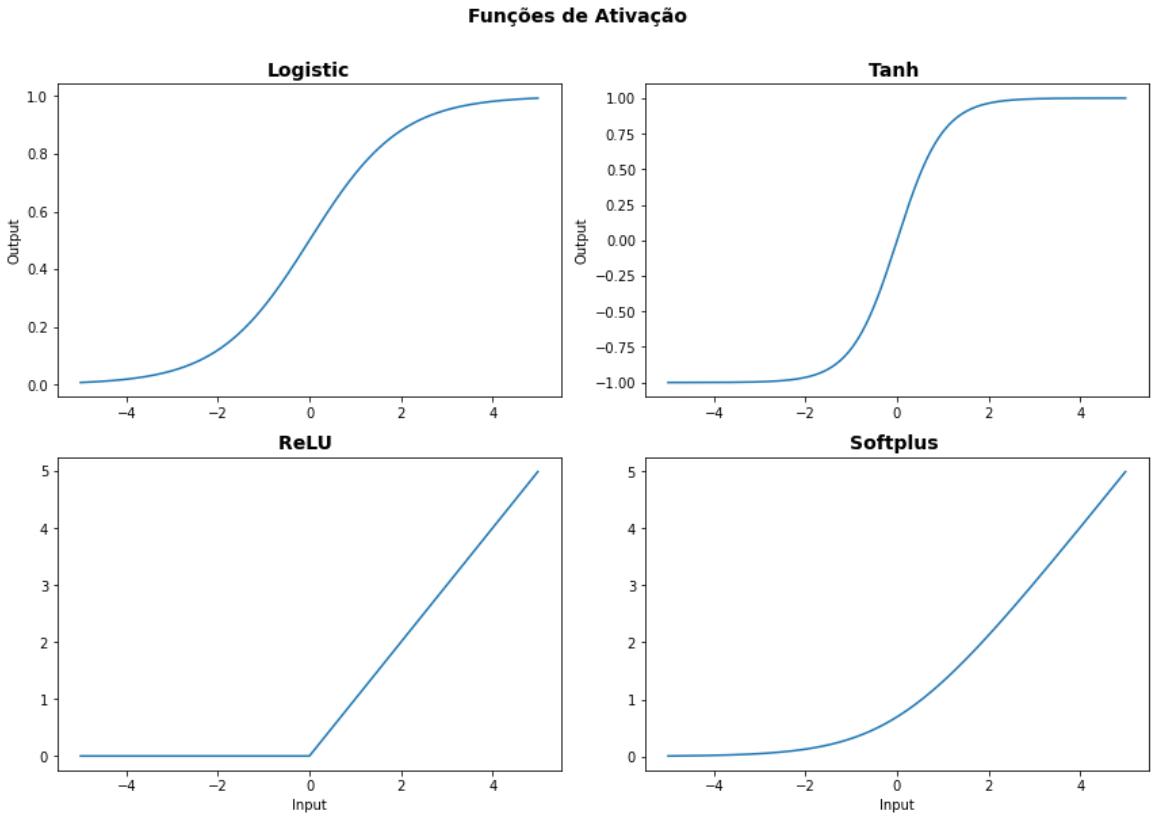


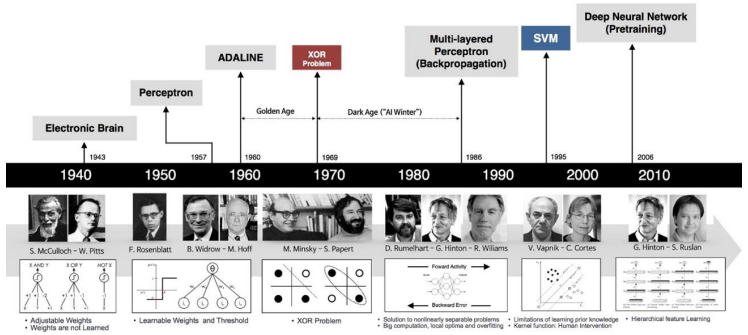
AND



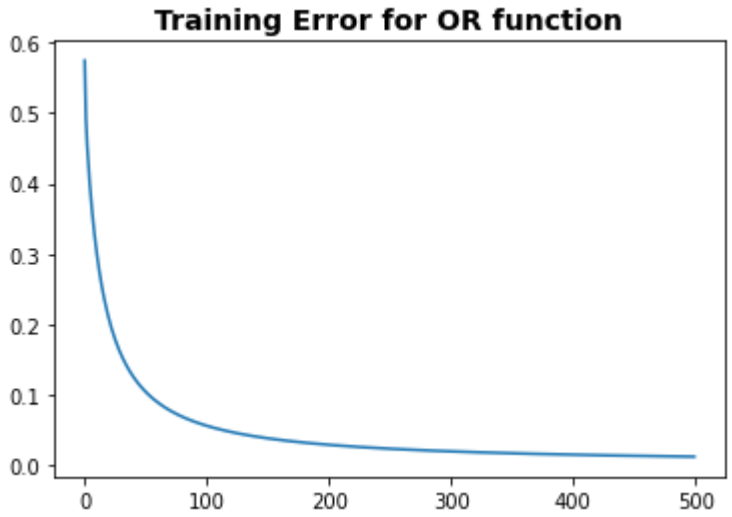
XOR

Funções de Ativação

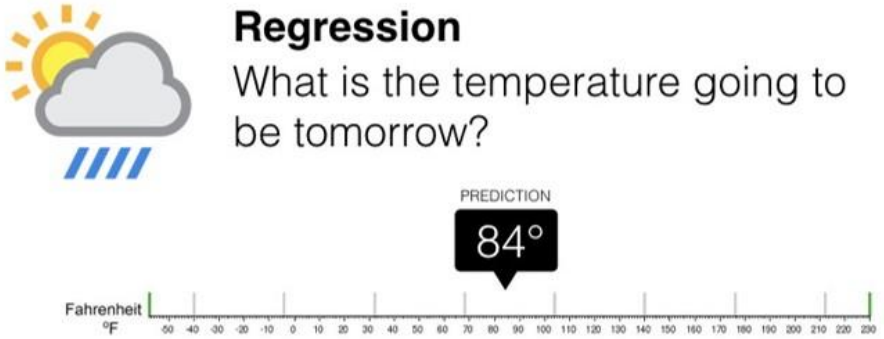


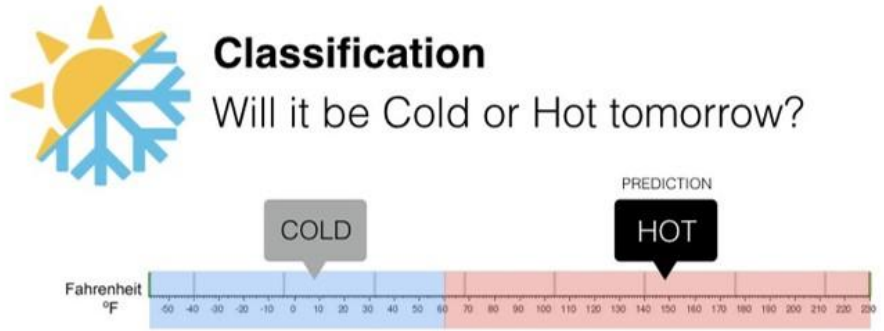


Aprendizado da Rede

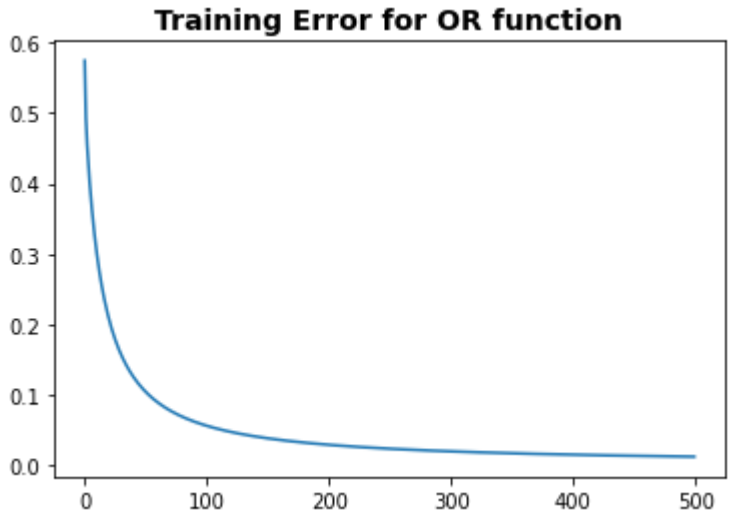


$$\sigma(\ x \times W_{xh}\ ) \times W_{hy} = Y$$

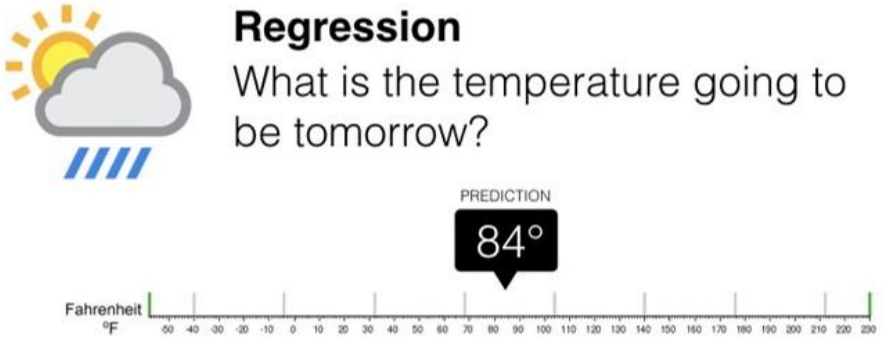


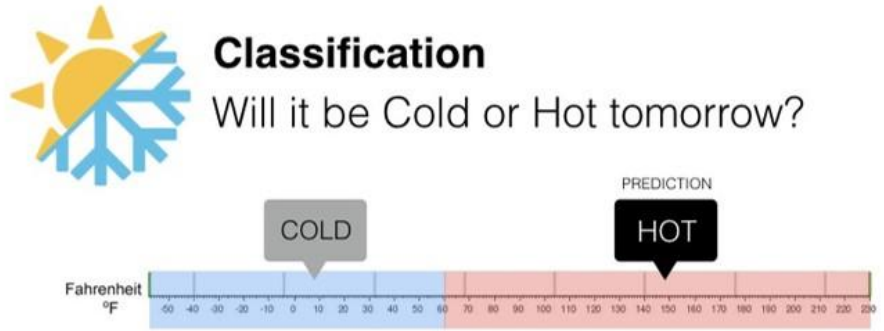


Aprendizado da Rede

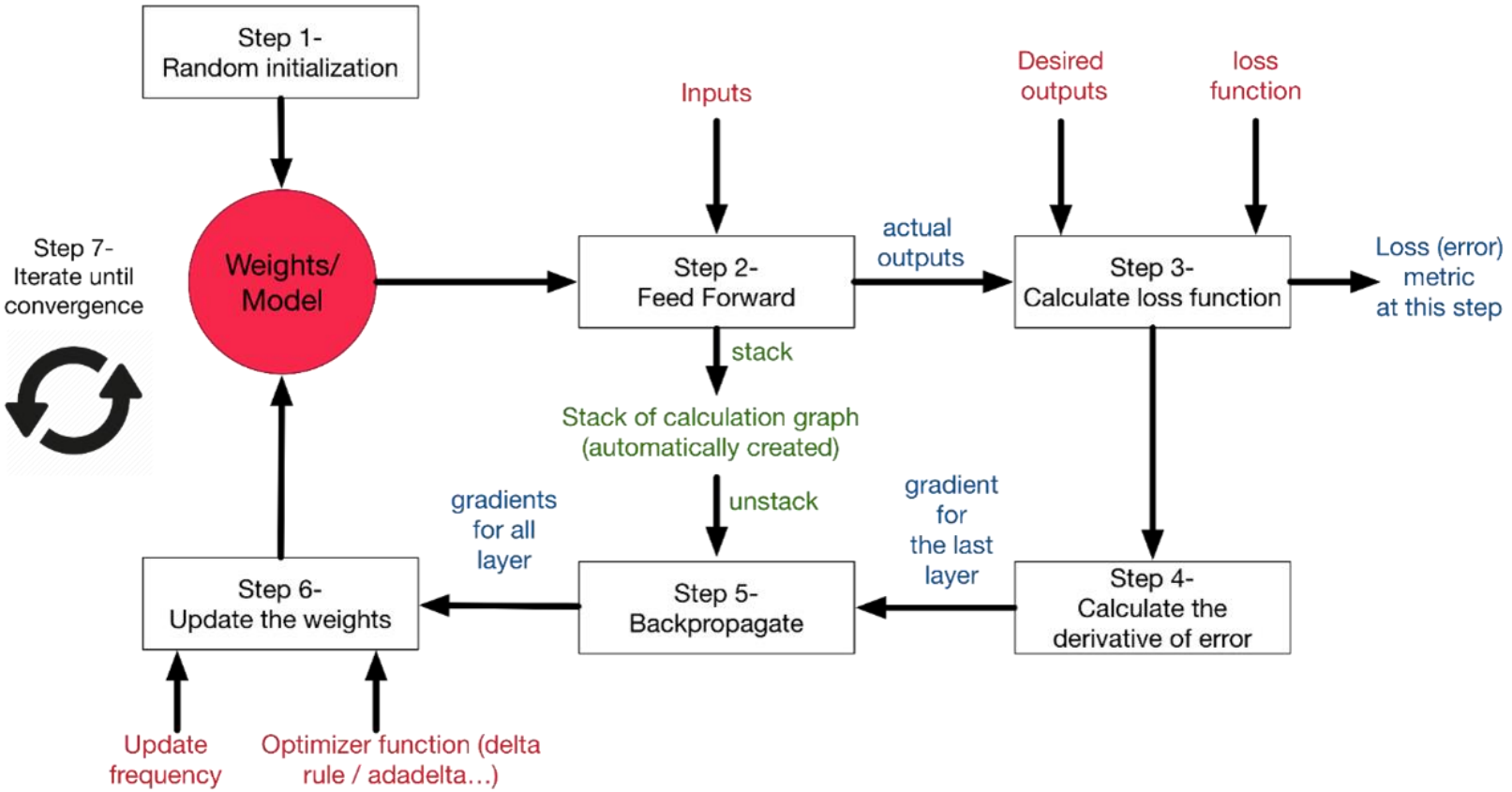


$$\sigma(\ x \times W_{xh}\ ) \times W_{hy} = Y$$



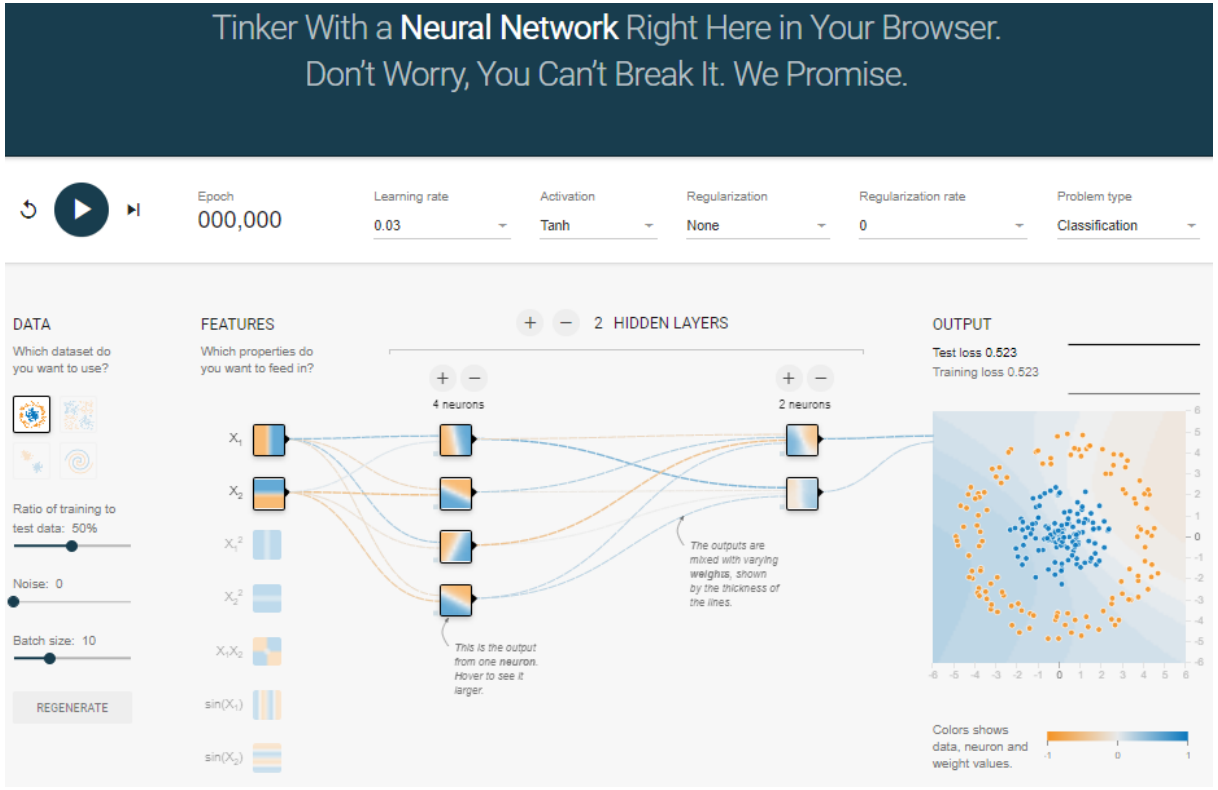


Backpropagation



Experimente

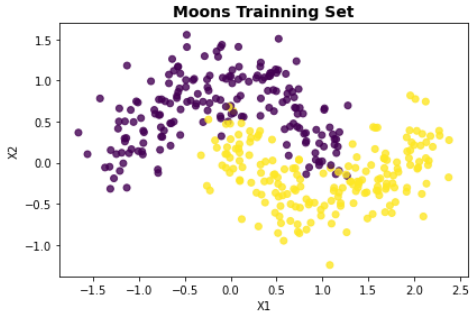
$$\sigma(\; x \times W_{xh} \;) \times W_{hy} = Y$$

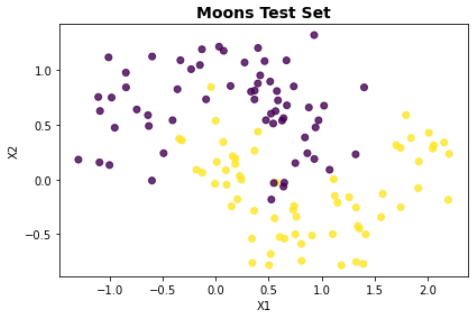


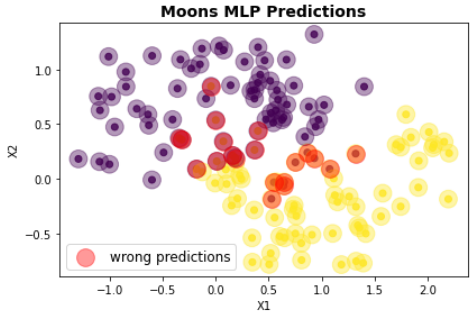
TRILHA 8
Redes Neurais e Deep Learning

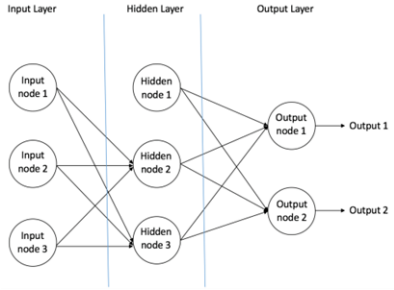
Parte B

MLP Com o scikit-learn










```python
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_moons

X, y = make_moons(n_samples=500, noise=0.25, random_state=1234)

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, random_state=1234)

mlp = MLPClassifier(activation='tanh', hidden_layer_sizes=(12, 12), random_state=1234)

mlp.fit(X_train, y_train)

y_pred = mlp.predict(X_test)

print("Accuracy on training set: {:.2f}".format(mlp.score(X_train, y_train)))
print("Accuracy on test set: {:.2f}".format(mlp.score(X_test, y_test)))
```

Frameworks de Deep Learning

	Keras	Pytorch	TensorFlow
API Level	High	Low	High and Low
Architecture	Simple, concise, readable	Complex, less readable	Not easy to use
Datasets	Smaller datasets	Large datasets, high performance	Large datasets, high performance
Debugging	Simple network, so debugging is not often needed	Good debugging capabilities	Difficult to conduct debugging
Does It Have Trained Models?	Yes	Yes	Yes
Popularity	Most popular	Third most popular	Second most popular
Speed	Slow, low performance	Fast, high-performance	Fast, high-performance
Written In	Python	Lua	C++, CUDA, Python

Keras Deep Learning

Modelo Sequencial

A construção do modelo segue basicamente os seguintes passos:

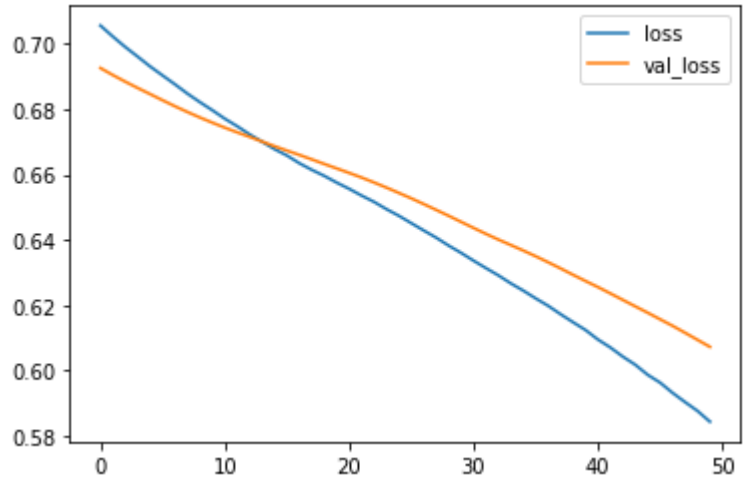
1. Defina o modelo.
2. Compile o modelo.
3. Treine o modelo.
4. Avalie o modelo.
5. Faça Predições.

Keras Deep Learning

```
# Define o Modelo
#--------------------------------------------------------------------------------

model = keras.Sequential(layers.Dense(X_train.shape[1], activation='relu', input_shape=[X_train.shape[1]])) # Entrada
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(y_train.shape[1], activation='sigmoid')) # Saída
```

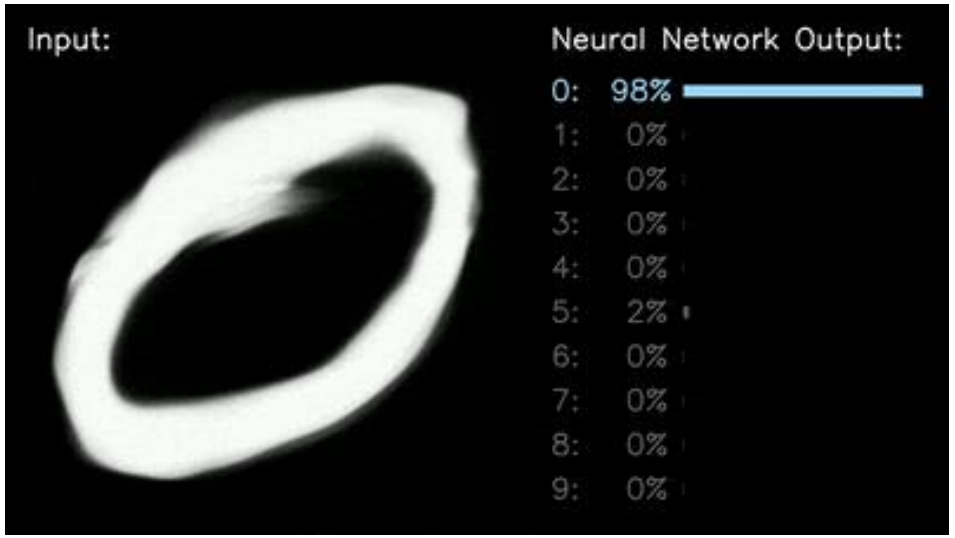
```
# Compila o Modelo
#--------------------------------------------------------------------------------

model.compile(loss='binary_crossentropy',  # Multiclass loss
              optimizer='adam',
              metrics=['binary_accuracy'])
```

```
# Treina o Modelo
#--------------------------------------------------------------------------------

history = model.fit(
    X_train, y_train,
    batch_size=32,
    validation_split=0.3,
    epochs=50,
    verbose=1,
)
```



Aplicações Redes Neurais

$$\sigma(X \times W_{xh}) \times W_{hy} = Y$$



Aplicações Deep Learning

$$\sigma(\ x \times W_{xh}\ ) \times W_{hy} = Y$$

1 Uma **câmera** instalada no veículo captura imagens do ambiente à sua volta. Essas imagens são registradas em uma malha bidimensional de sensores de luz

2 Cada um desses elementos armazena um **valor numérico binário** (010101, por exemplo), que corresponde à luminosidade incidente sobre eles

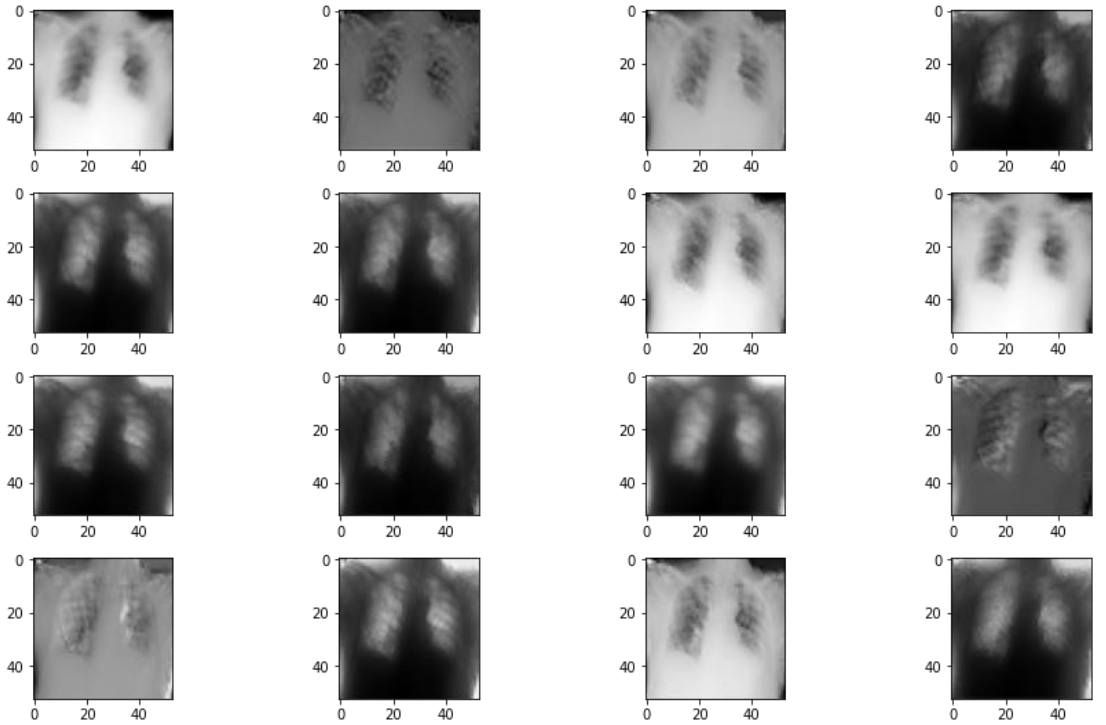
3 Essa imagem transformada em números é analisada para a identificação de **áreas de interesse**, eliminando elementos dispensáveis, como, por exemplo, o céu

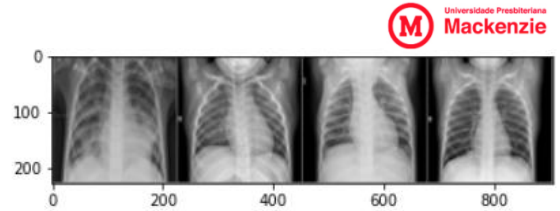
4 A última etapa envolve a análise, a classificação e o reconhecimento dos dados que importam, como **pessoas, outros carros, placas e sinais de trânsito**

5 O sistema faz isso com auxílio de **redes neurais** que reconhecem padrões a partir de um treinamento inicial e são capazes de evoluir a partir da experiência

Aplicações Deep Learning

$$\sigma(\ x \times W_{xh}\ ) \times W_{hy} = Y$$

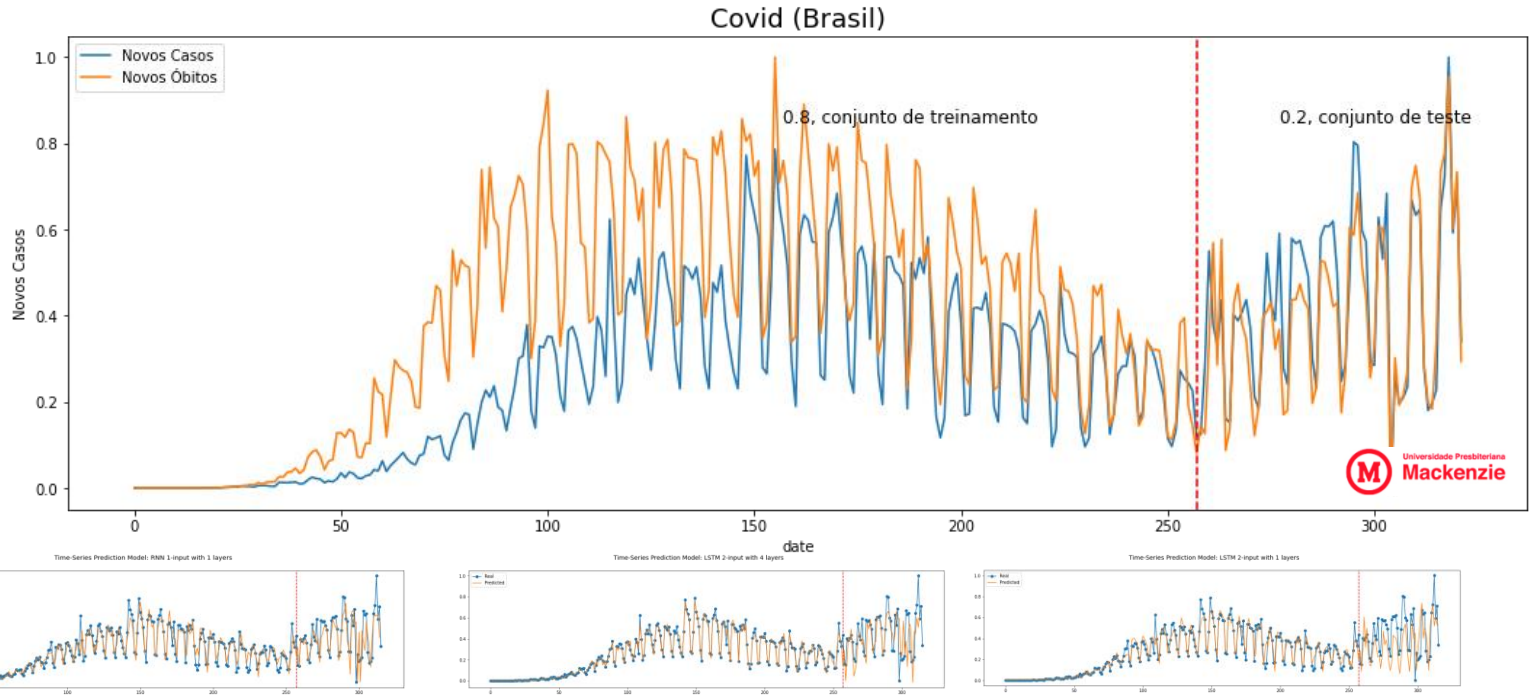




Real: VIRAL NORMAL NORMAL NORMAL

Aplicações Deep Learning

$$\sigma(x \times W_{xh}) \times W_{hy} = Y$$



Universidade Presbiteriana
Mackenzie