



# Formação Inteligência Artificial



# Deep Learning Frameworks





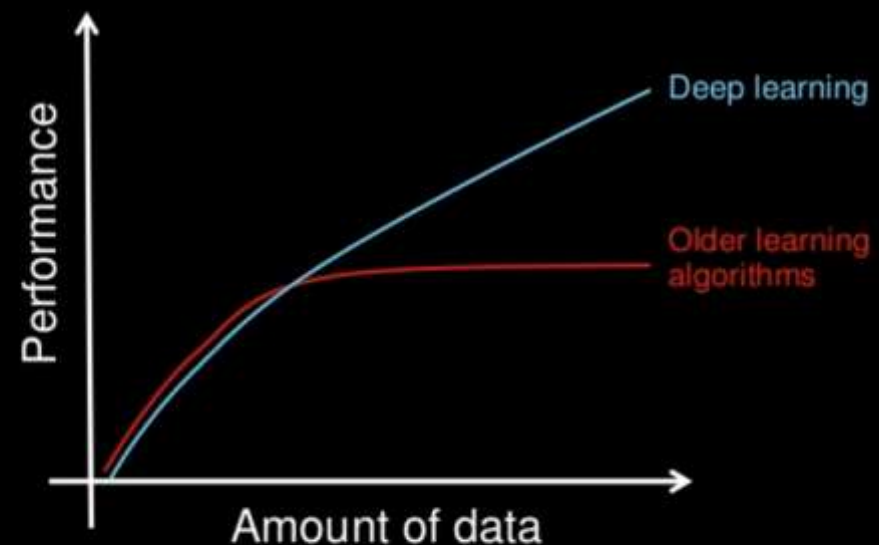
# Deep Learning



# Deep Learning

Deep Learning é uma área de pesquisa em Aprendizagem de Máquina, que foi introduzida com o objetivo de aproximar Machine Learning de um de seus objetivos originais: **Inteligência Artificial.**

## Why deep learning



How do data science techniques scale with amount of data?





# Deep Learning

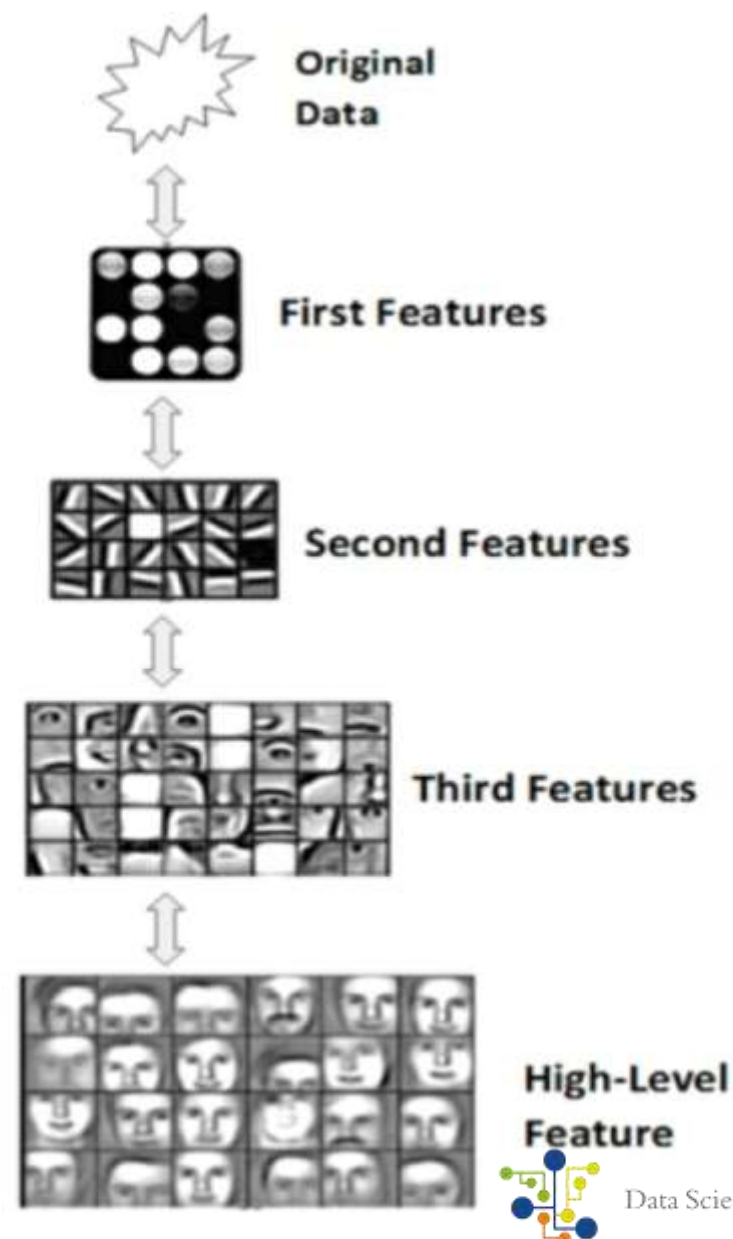
## Rocket engines: Deep Learning driven by scale





# Deep Learning

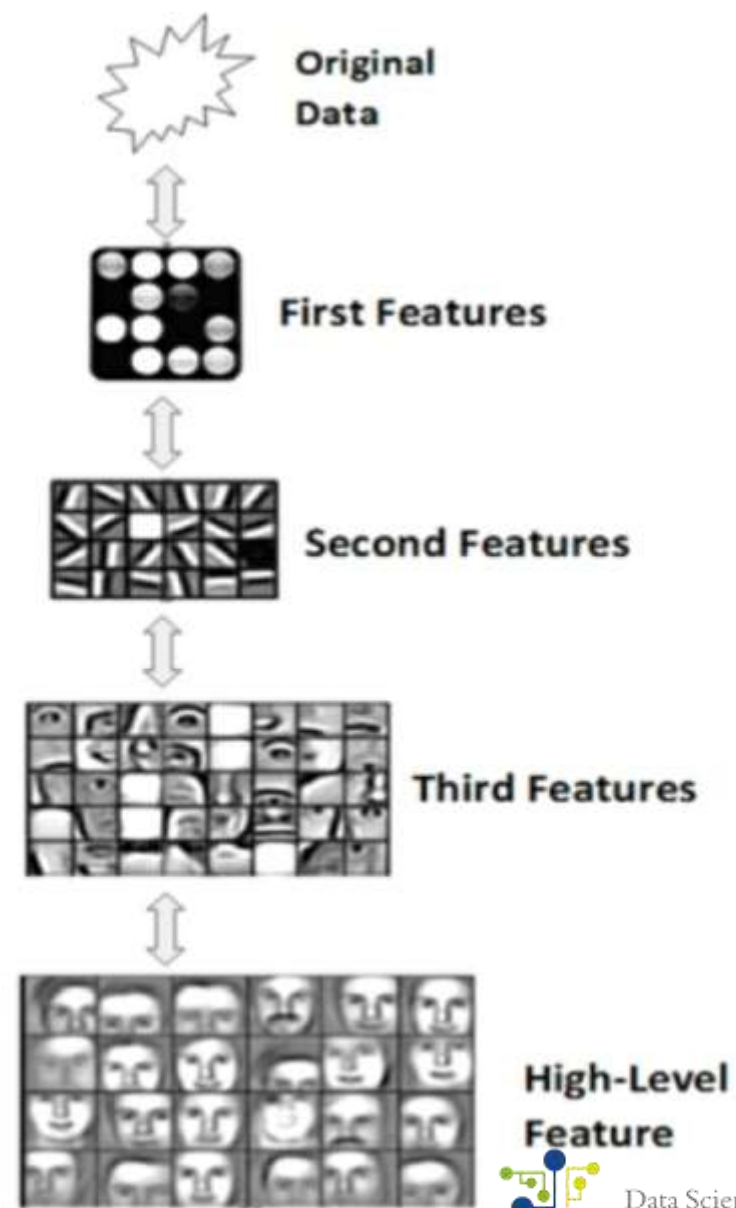
Outra razão é certamente a crescente facilidade de encontrar conjuntos de dados cada vez mais numerosos sobre os quais treinar um sistema.





# Deep Learning

pixel --> edge --> texture --> motif --> part --> object

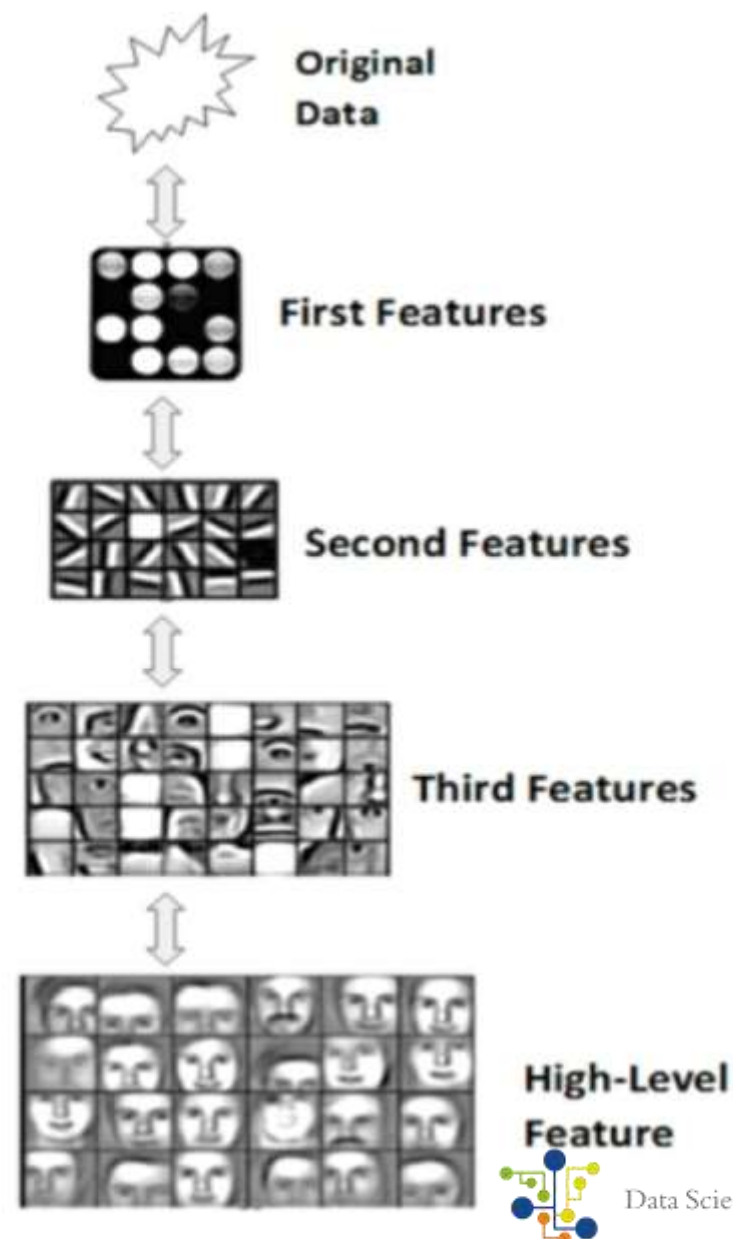






# Deep Learning

character --> word --> word group --> clause --> sentence --> story

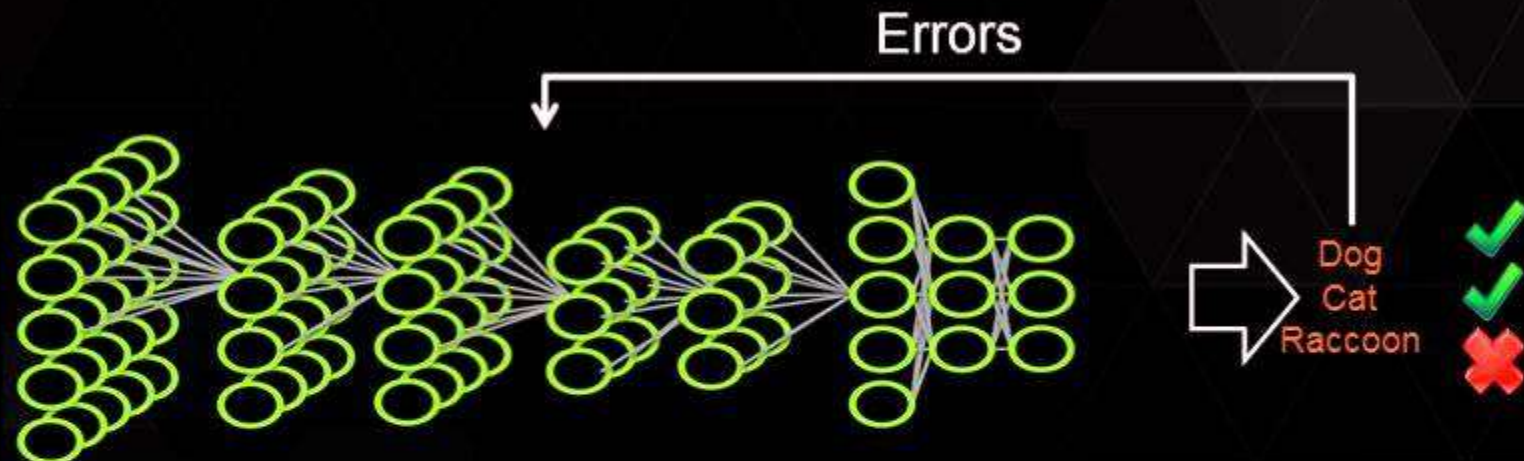




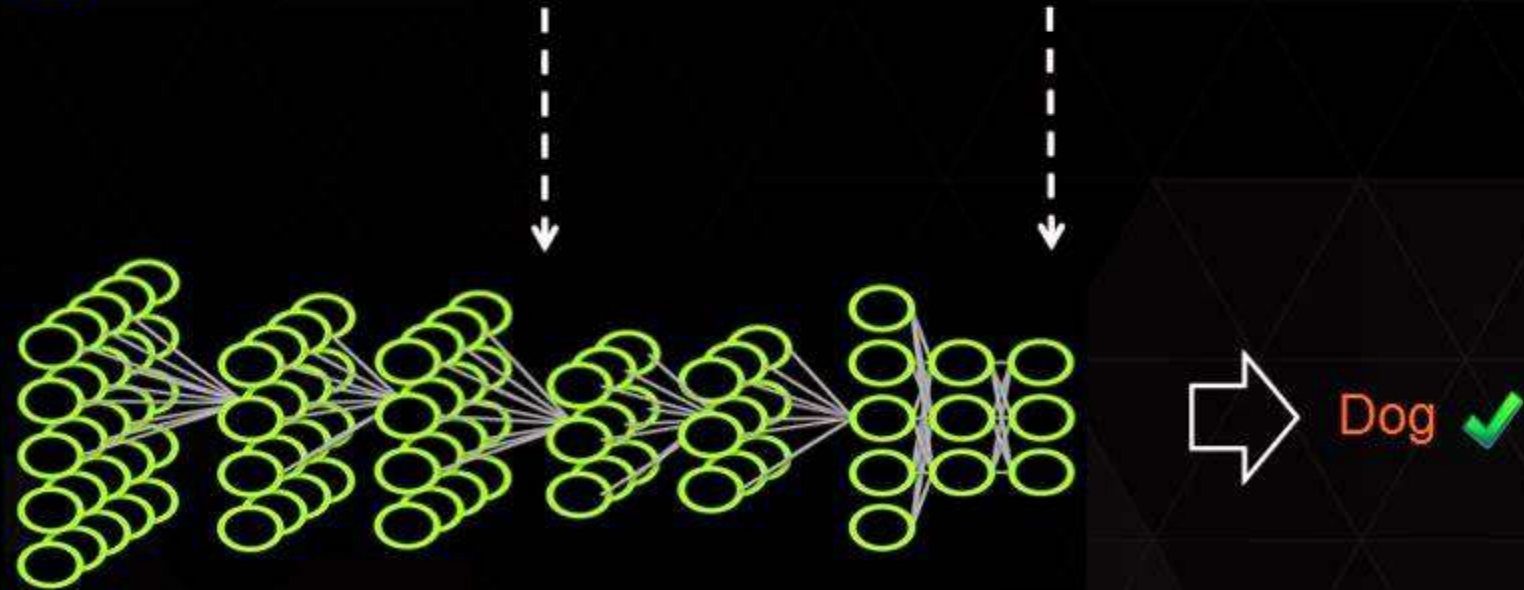
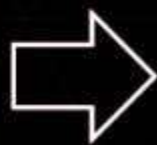


# DEEP LEARNING APPROACH

**Train:**

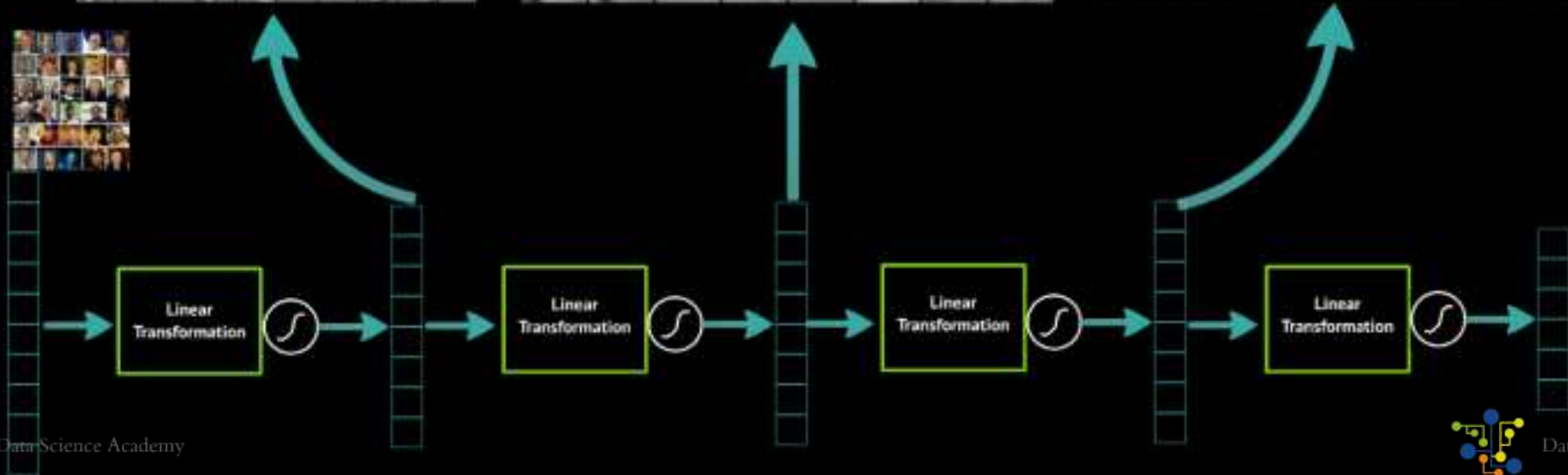
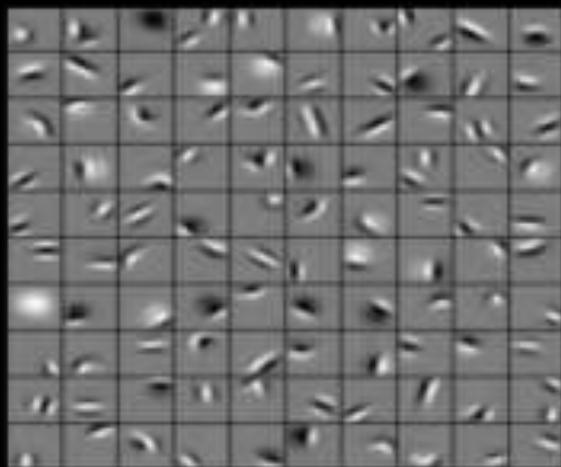


**Deploy:**





# Deep Learning learns layers of features





# Deep Learning

An abstract background diagram consisting of several colored circles (blue, yellow, green, orange) connected by lines of the same color, forming a network-like structure. Two large blue rounded rectangles are positioned in the center, containing the text 'Convolutional Neural Networks' and 'Recurrent Neural Networks' respectively.

Convolutional  
Neural Networks

Recurrent Neural  
Networks



Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# Deep Neural Networks



Data Science Academy

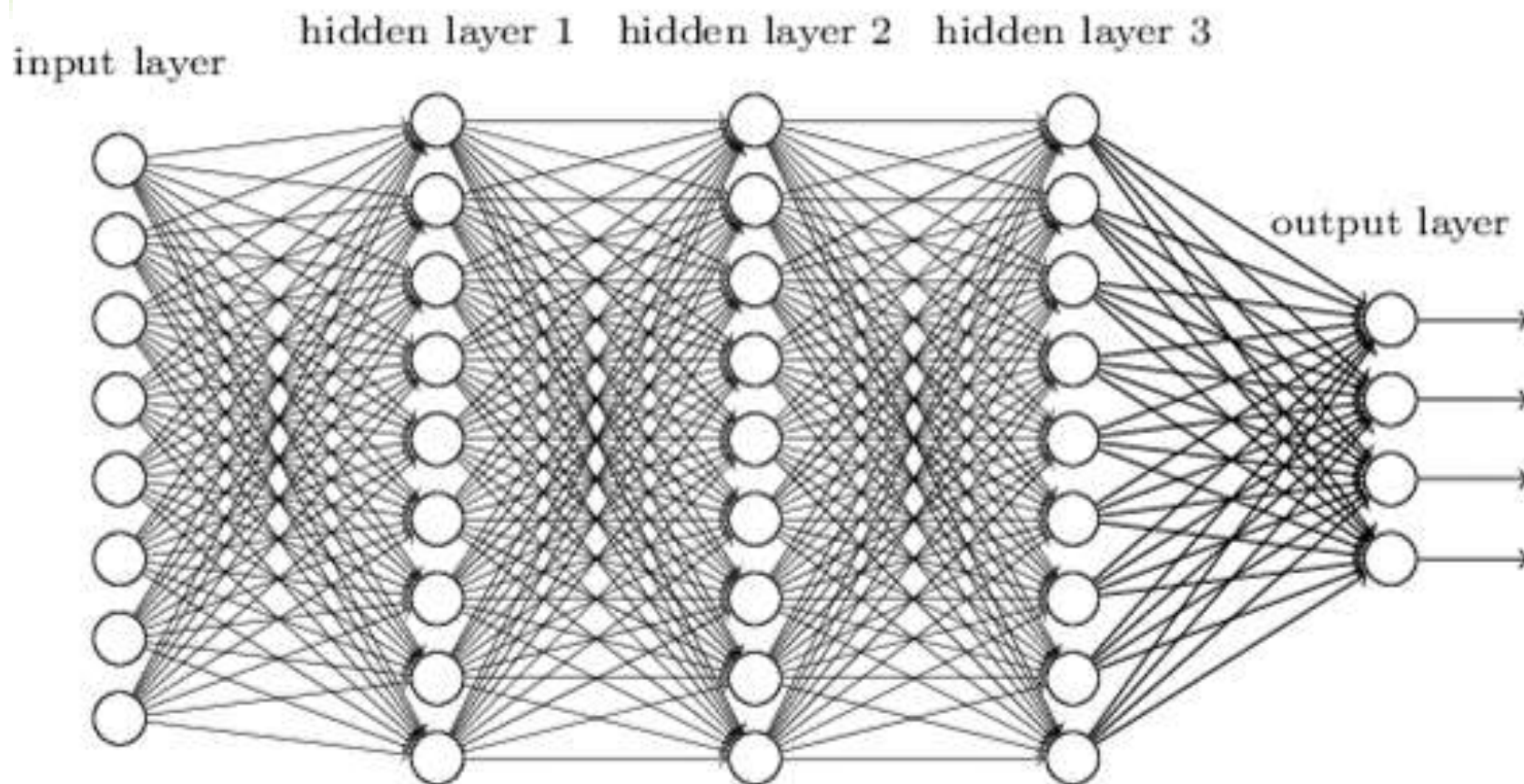


Data Science Academy





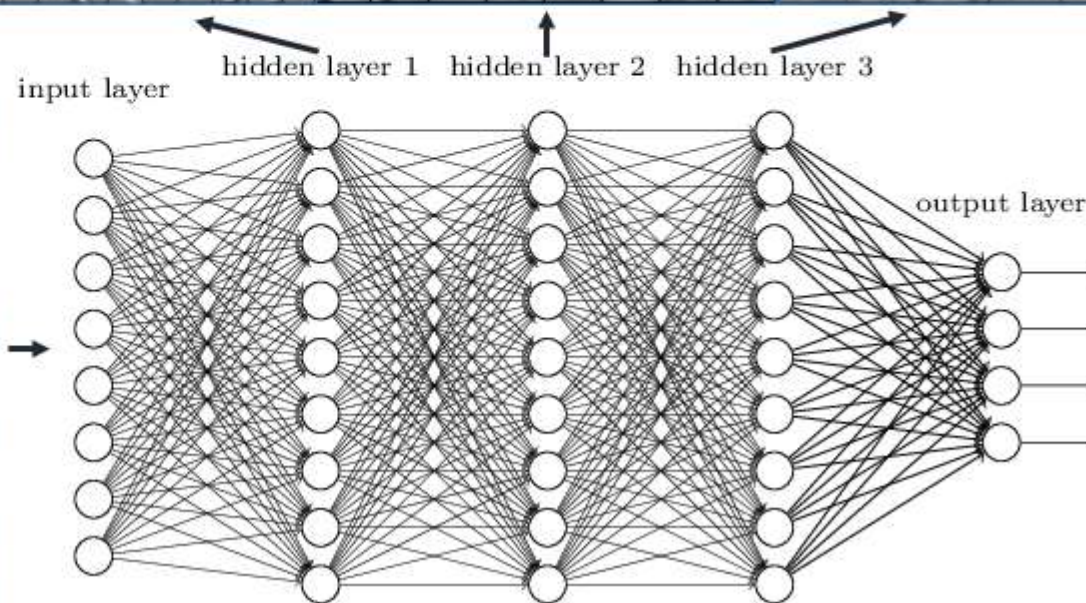
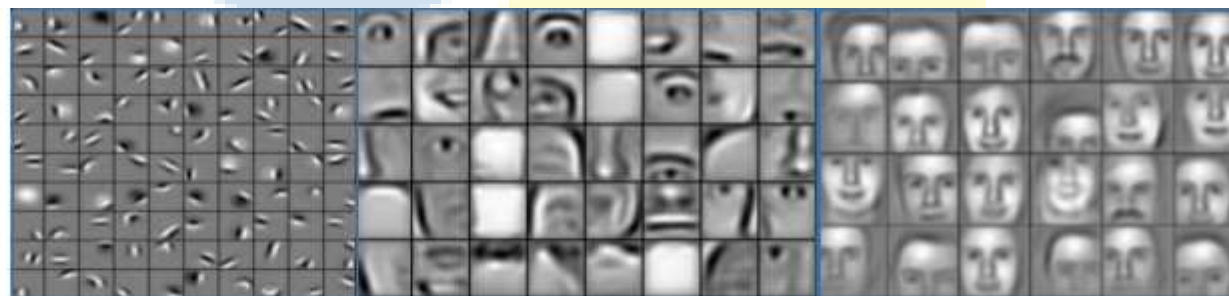
# Deep Neural Networks





# Deep Neural Networks

Deep neural  
networks learn  
hierarchical feature  
representations





# Input Video



⋮



Data Science Academy

# Convolutional Net



Data Science Academy



⋮



mean pooling

# Recurrent Net



# Output

*A*

*boy*

*is*

*playing*

*a*

*ball*



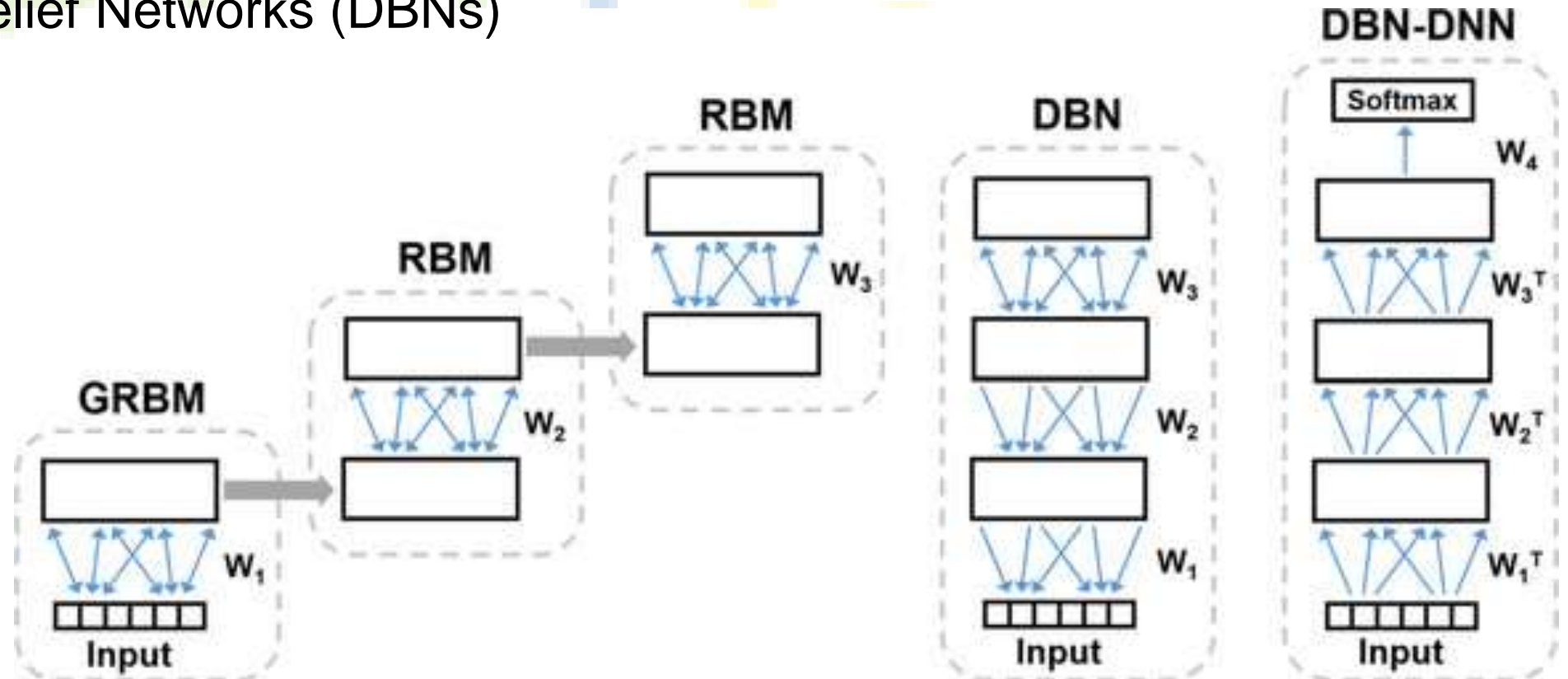
Data Science Academy





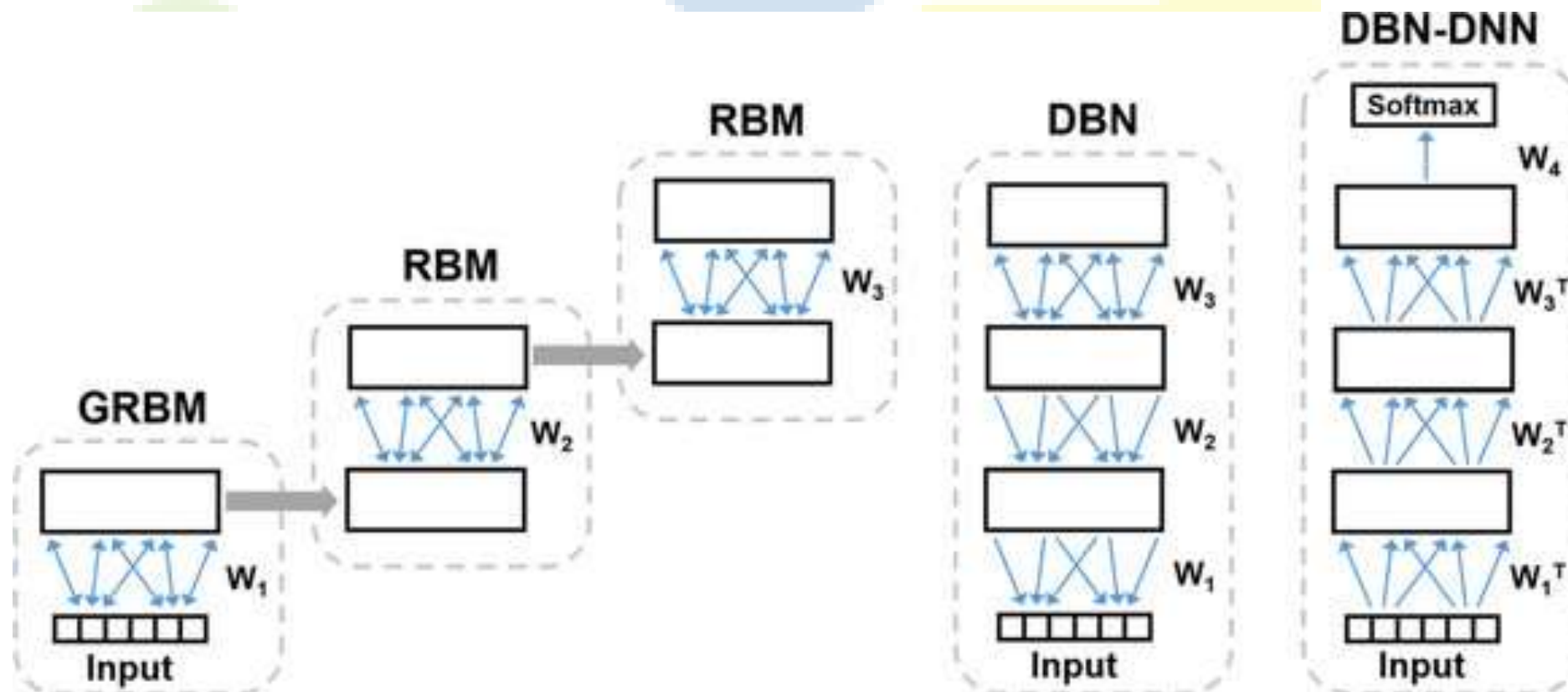
# Deep Neural Networks

## Deep Belief Networks (DBNs)





# Deep Neural Networks





Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# Convolutional Neural Networks



Data Science Academy



Data Science Academy



# Convolutional Neural Networks



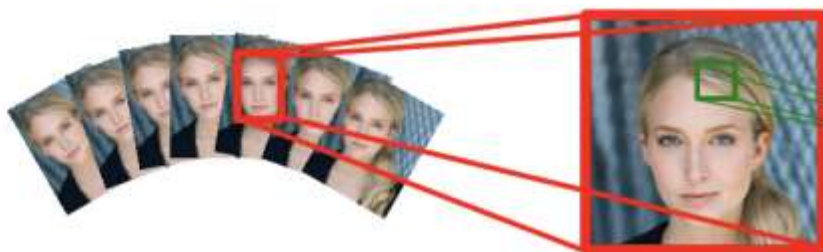
1. Input image

2. Face detection

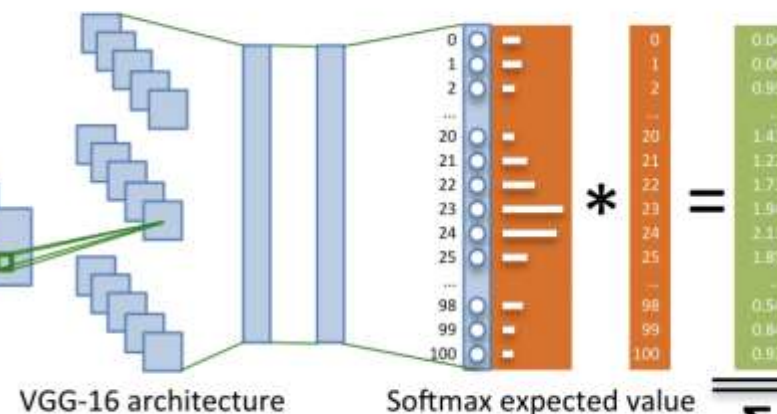
3. Cropped face

4. Feature extraction

5. Prediction



+ 40% margin



\*

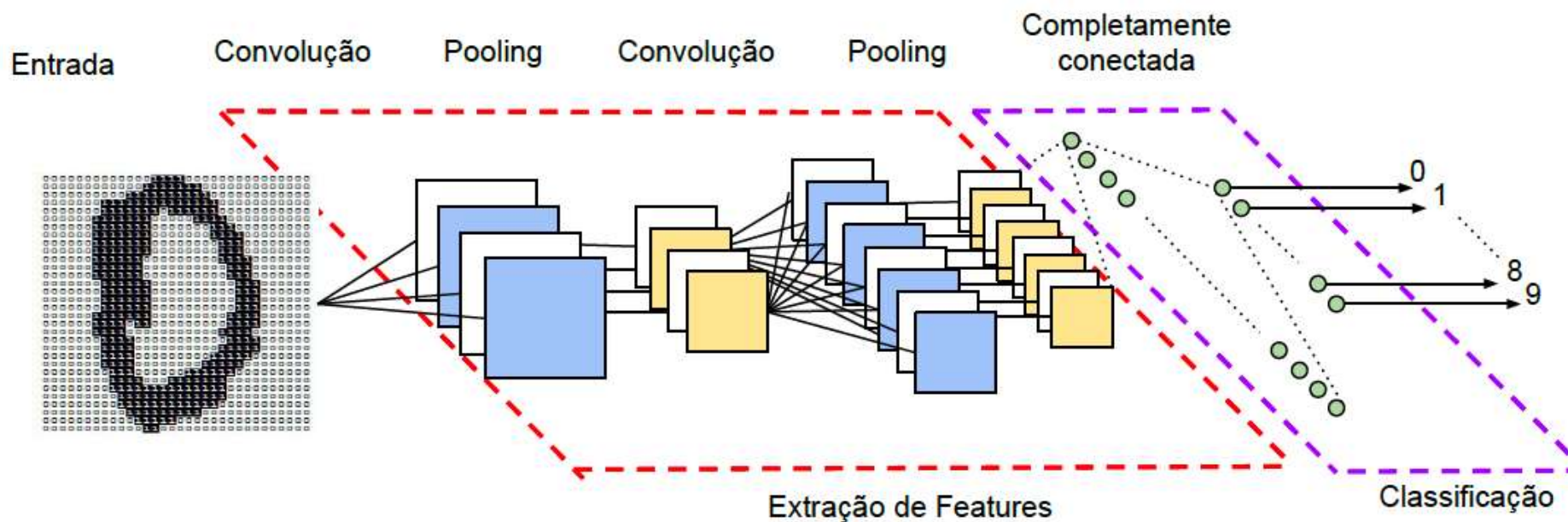
=

$\Sigma = 23.4 \text{ years}$





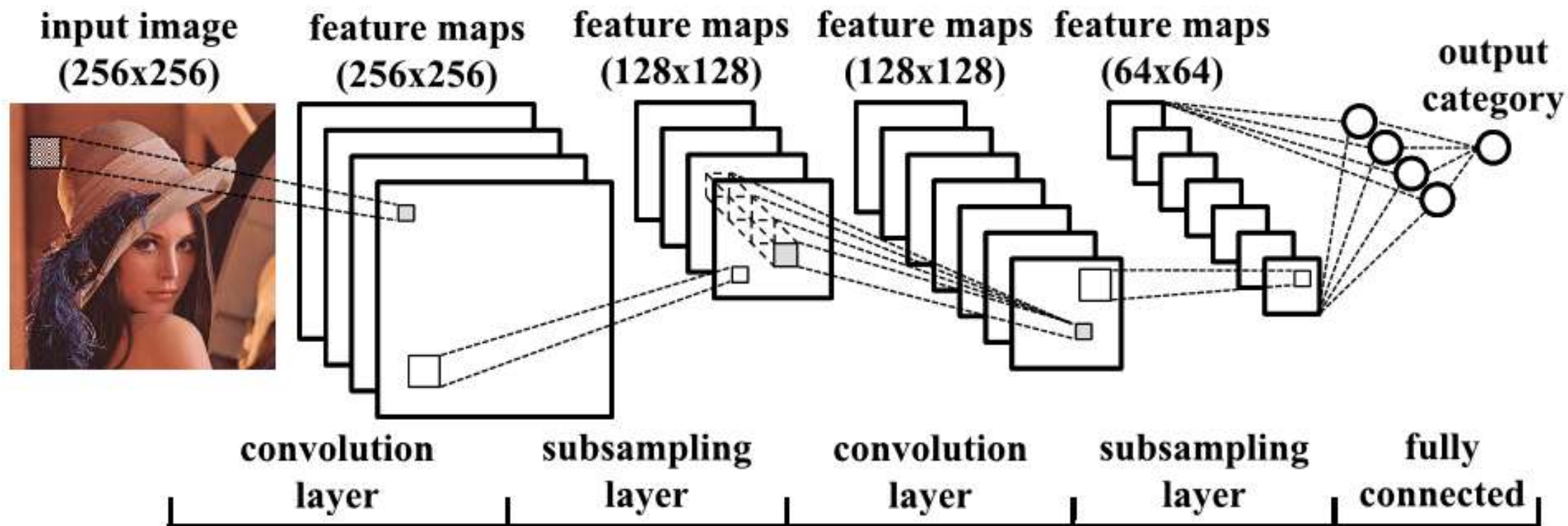
# Convolutional Neural Networks





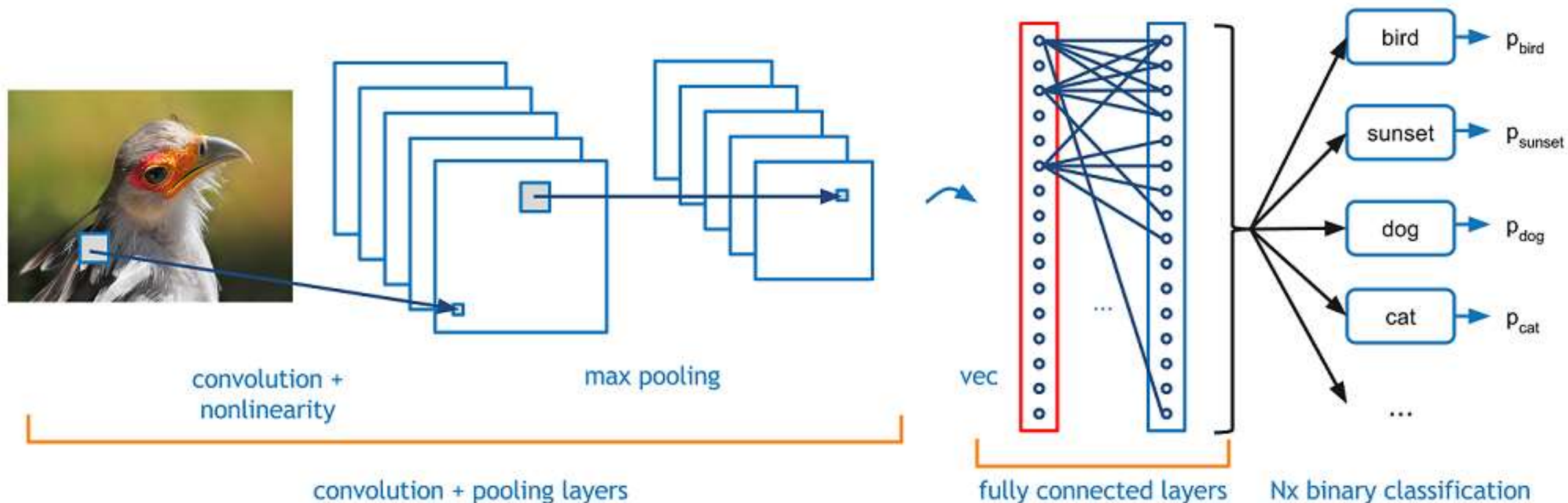


# Convolutional Neural Networks





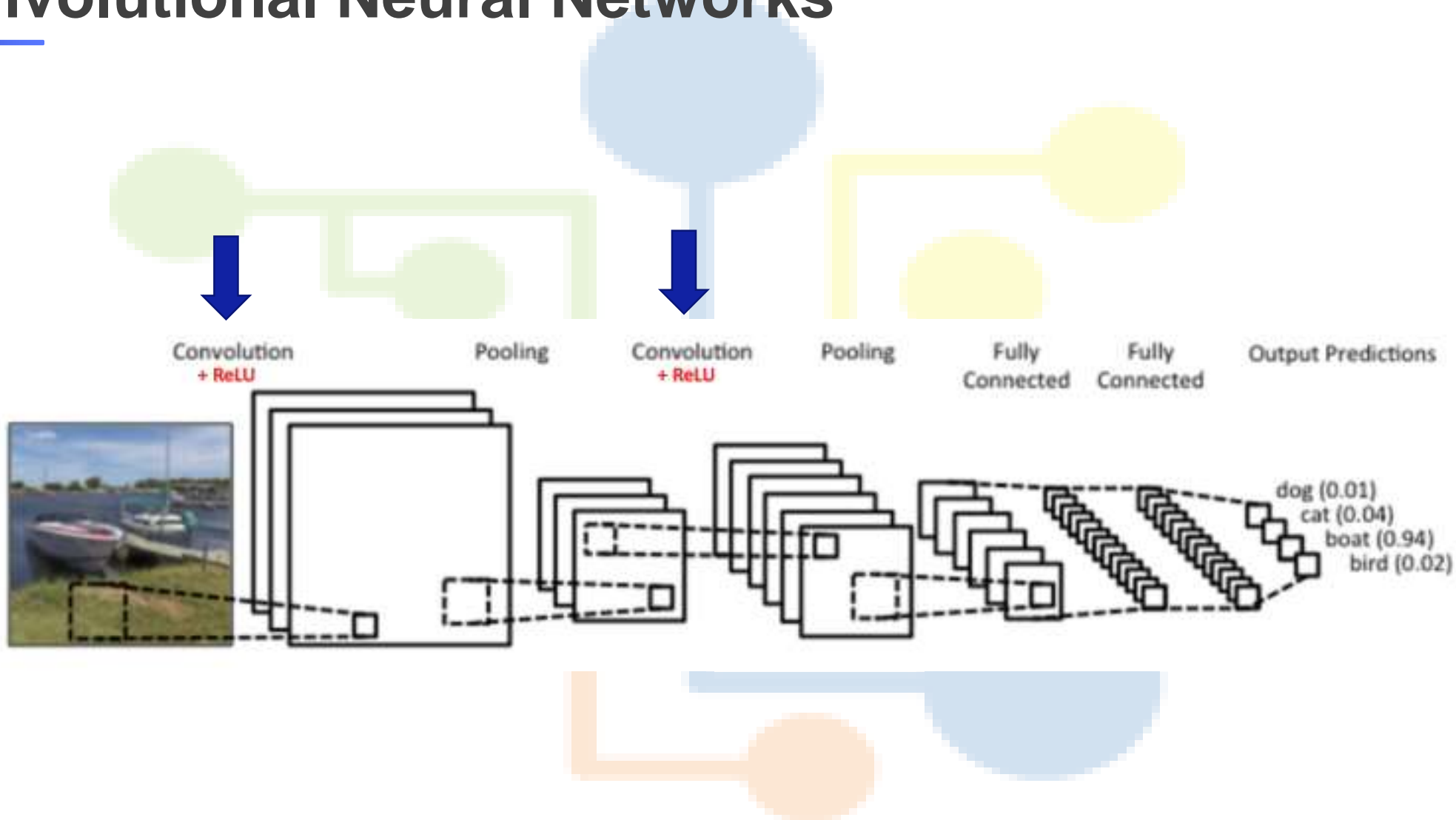
# Convolutional Neural Networks





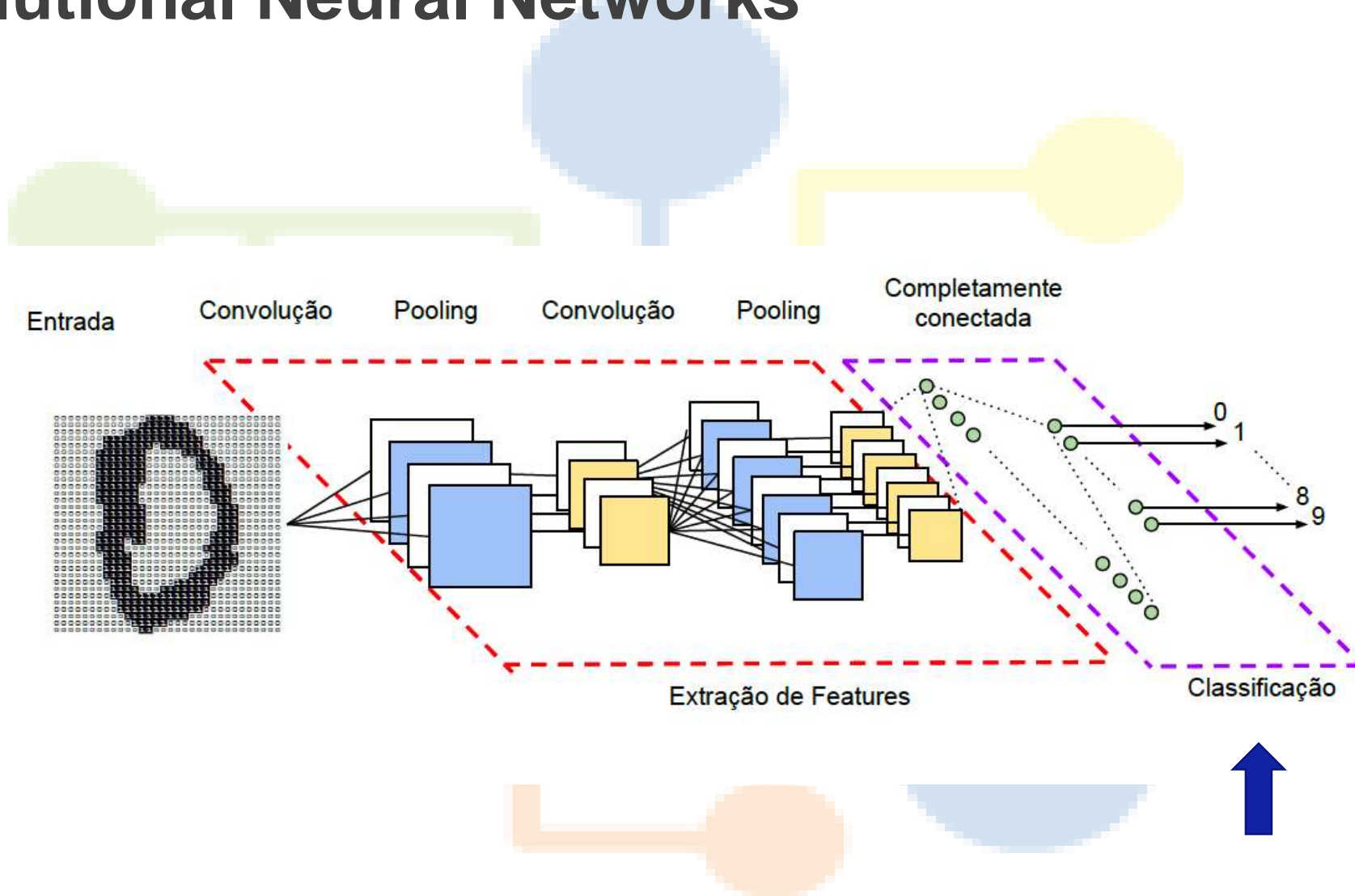


# Convolutional Neural Networks





# Convolutional Neural Networks



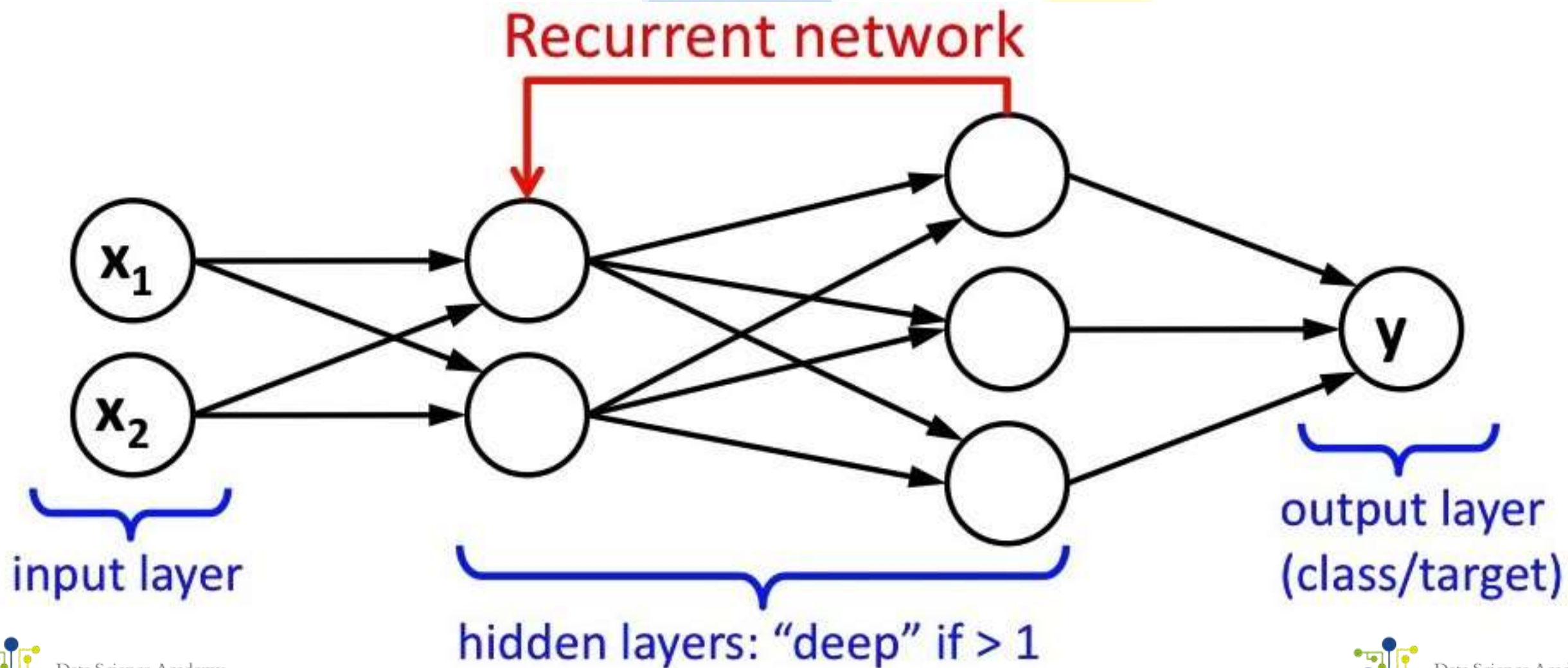


# Recurrent Neural Networks





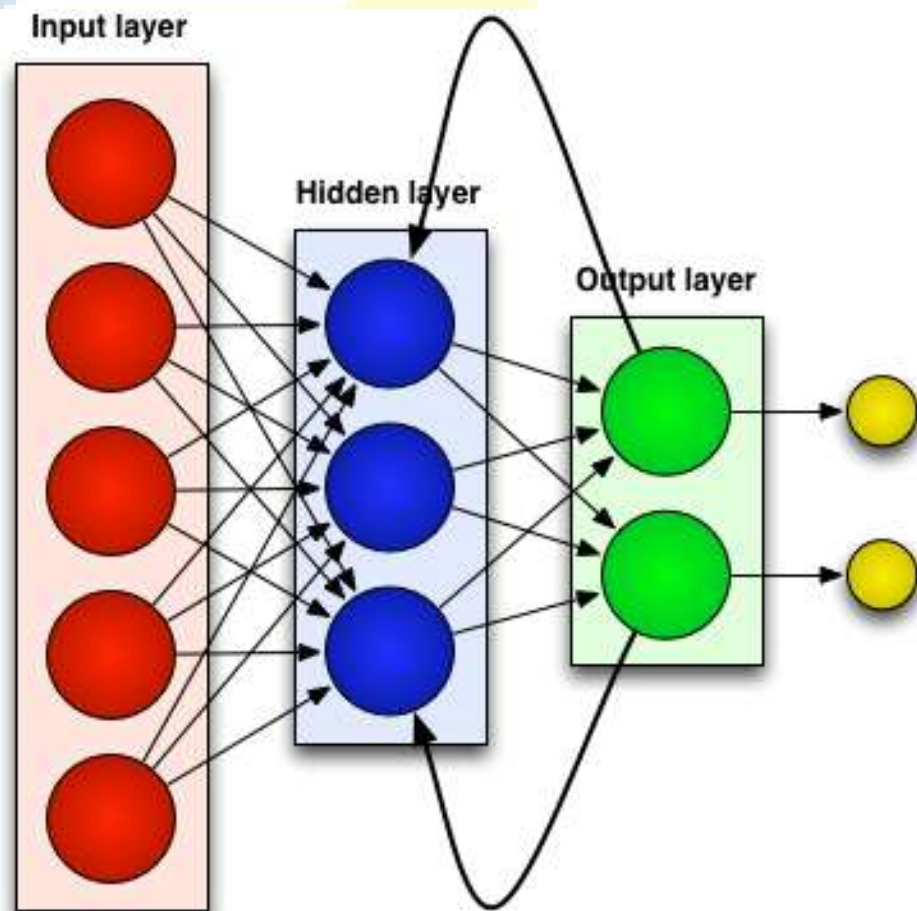
# Recurrent Neural Networks





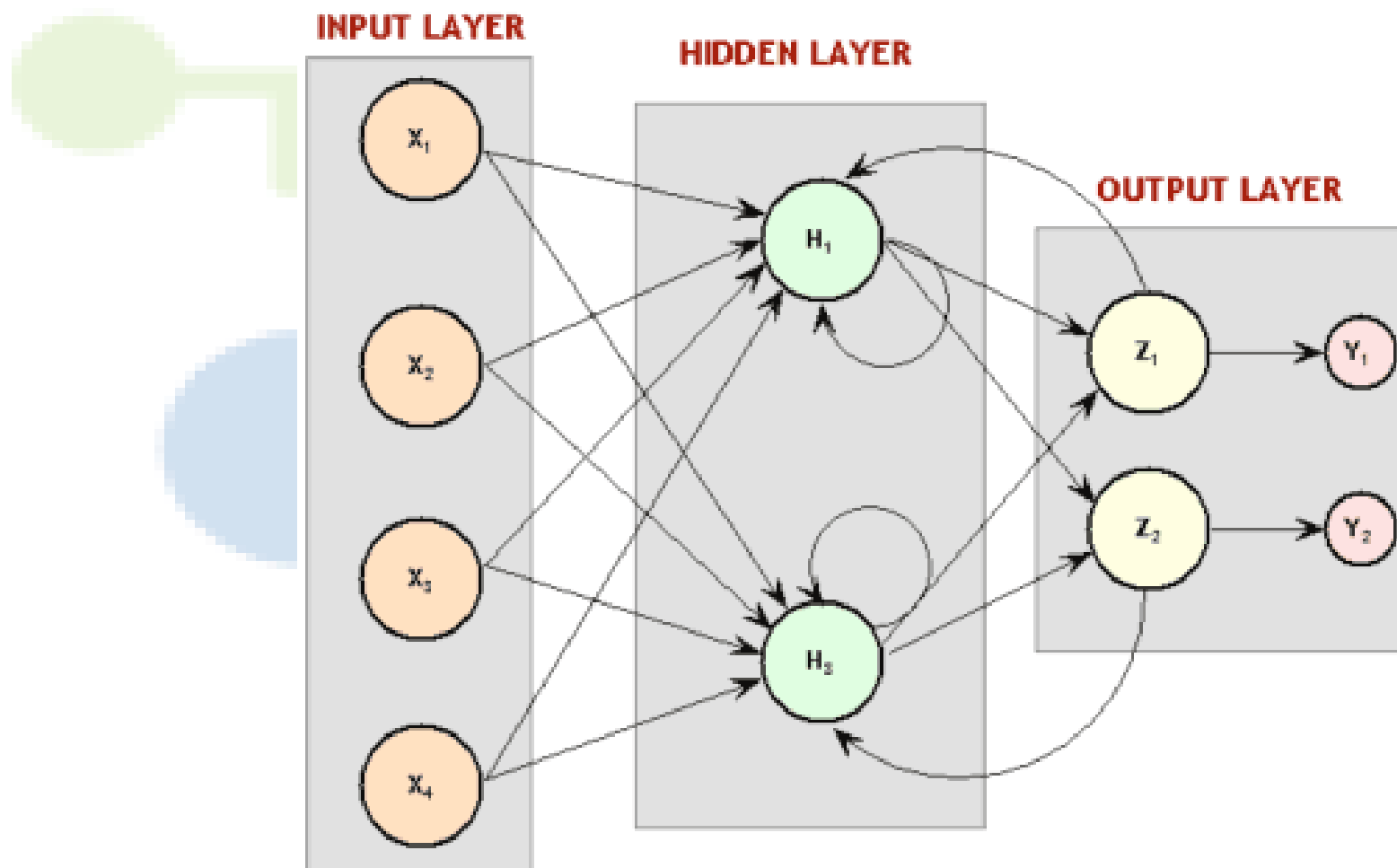
# Recurrent Neural Networks

A ideia por trás das RNNs é  
fazer uso de informações  
sequenciais





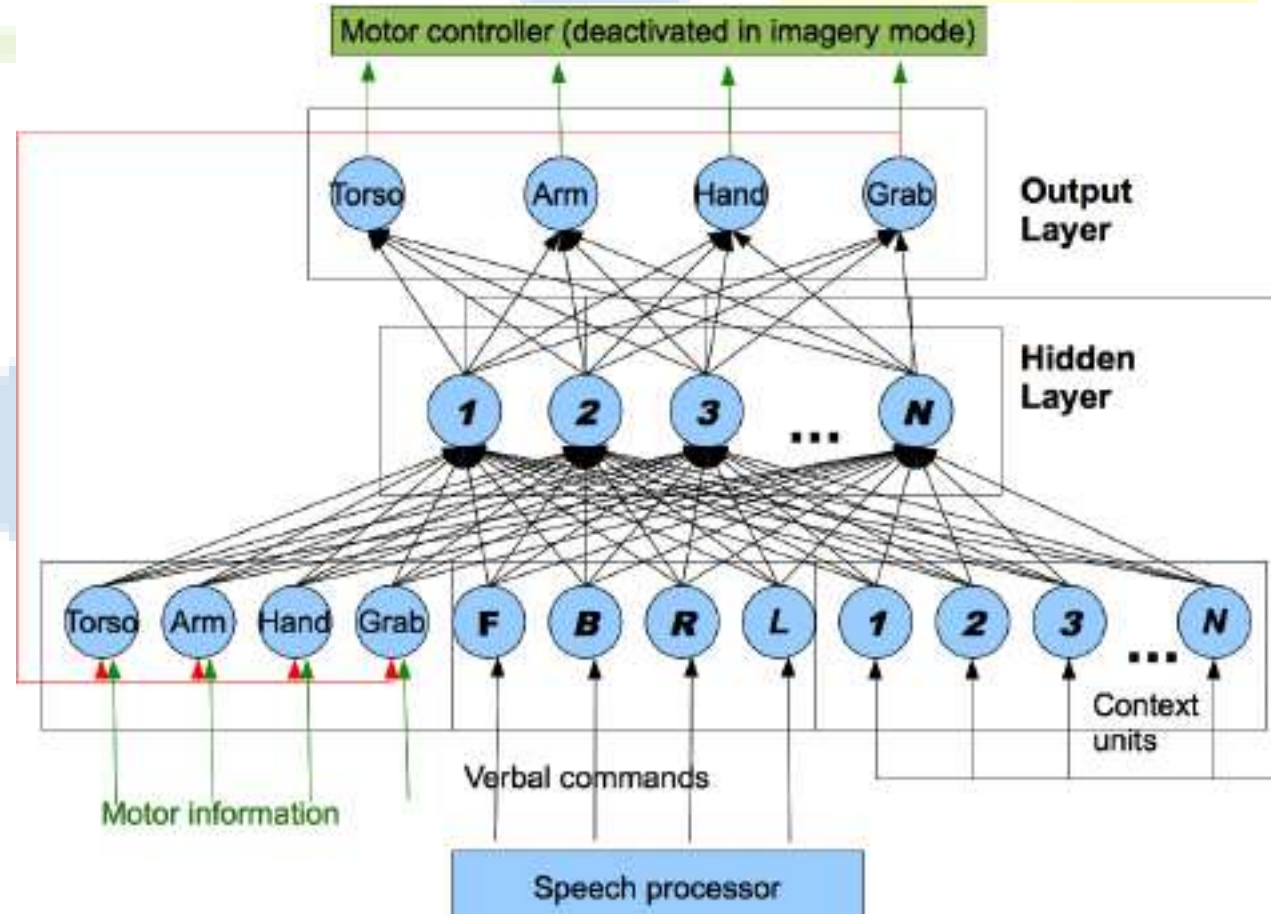
# Recurrent Neural Networks







# Recurrent Neural Networks



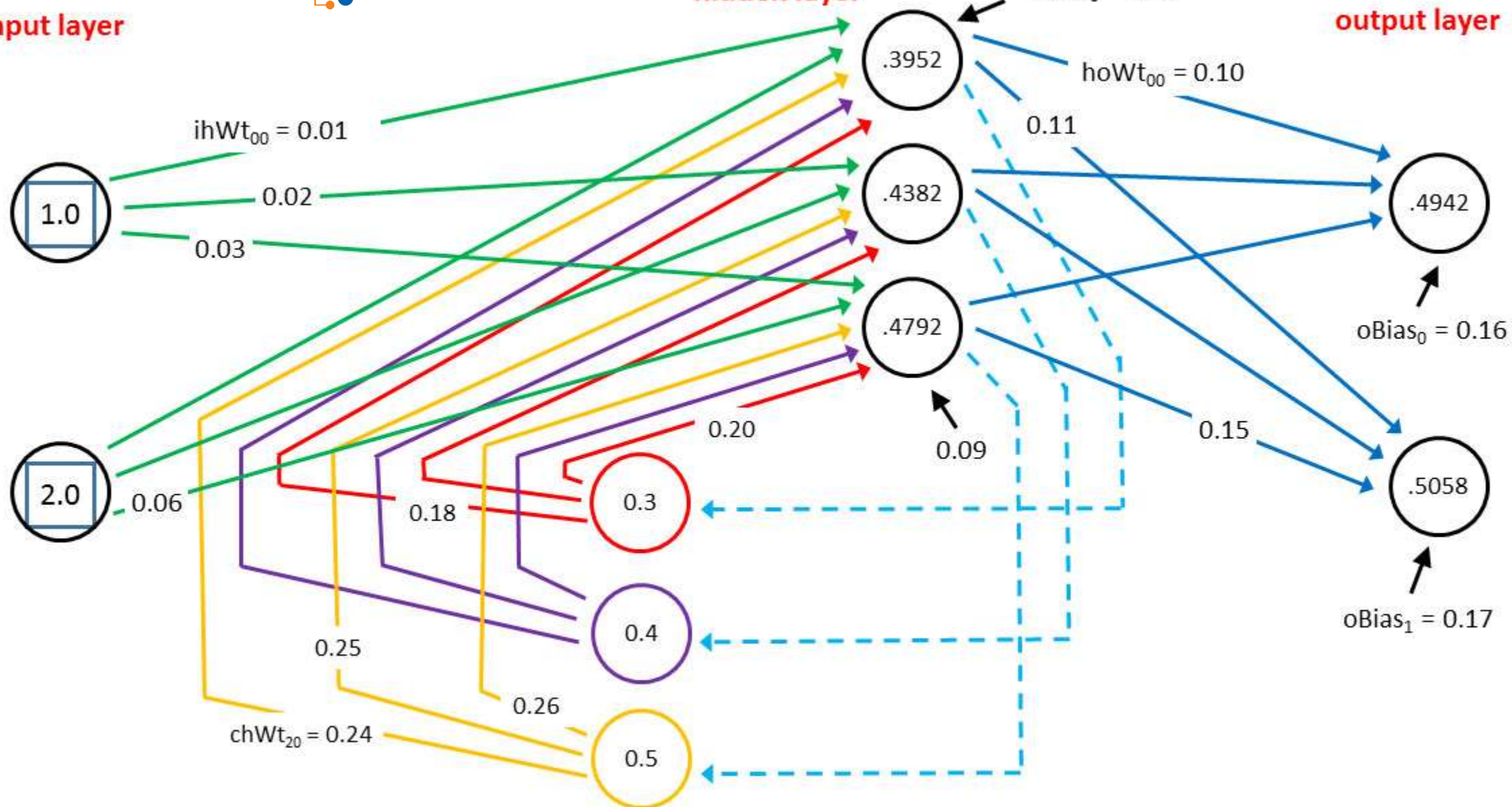




input layer

hidden layer

output layer

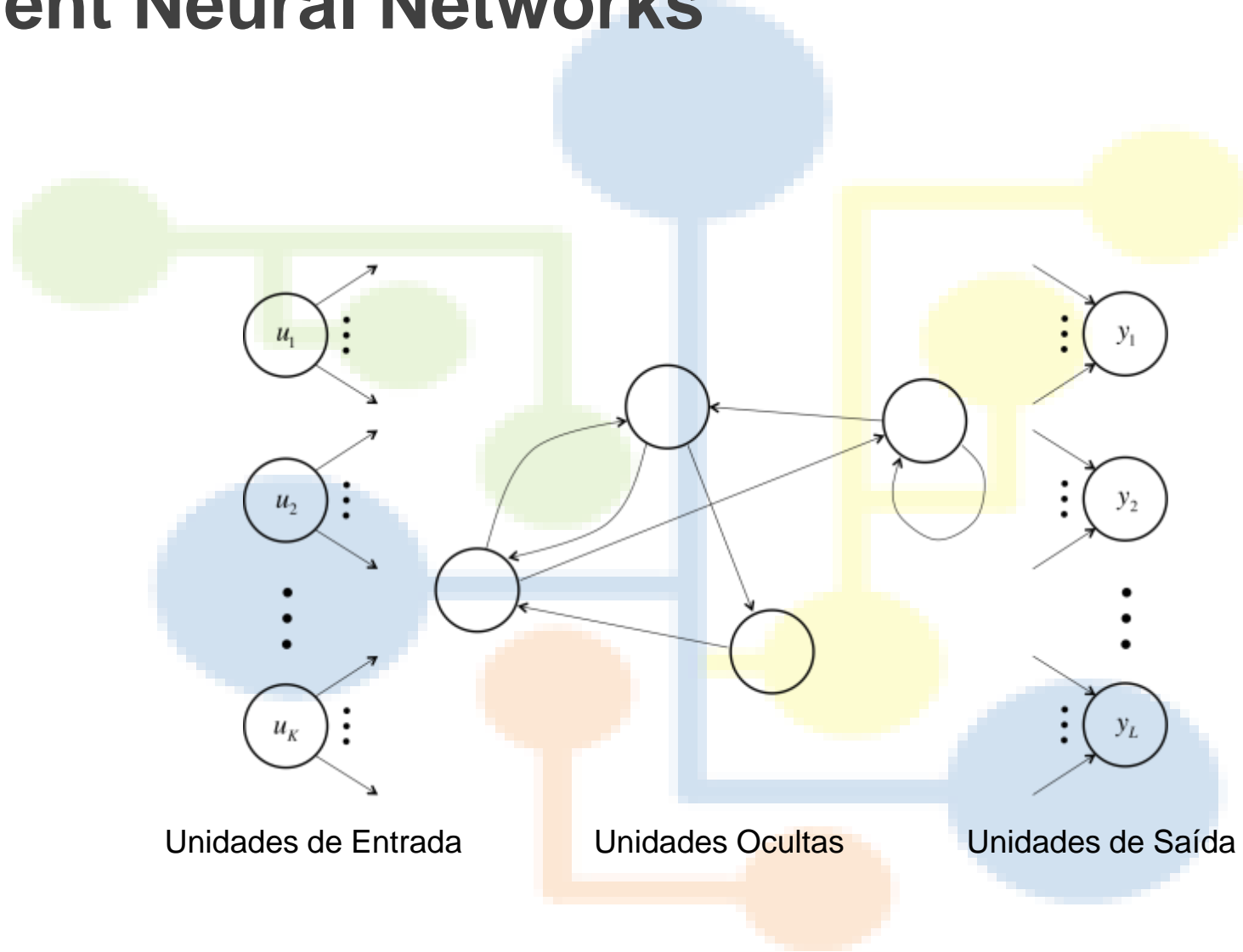


context nodes





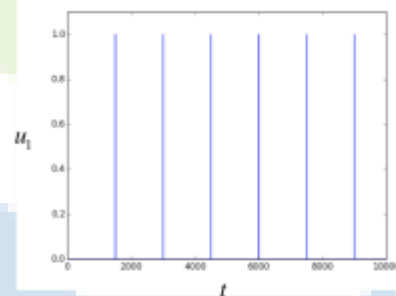
# Recurrent Neural Networks



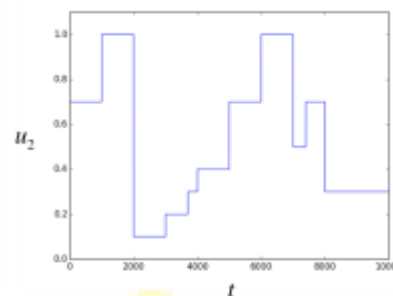


# Recurrent Neural Networks

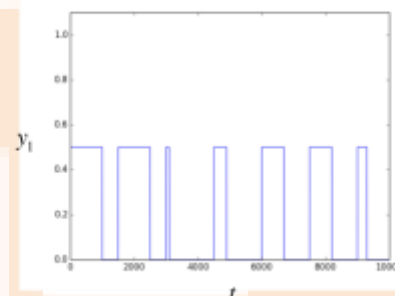
$u_1$  corresponde a um switch binário que dispara para o valor 1 quando a RNN inicializa o temporizador



$u_2$  é uma variável discreta que varia entre 0,1 e 1,0 inclusive, o que corresponde a quanto tempo a saída deve ser ligada se o temporizador for iniciado naquele instante

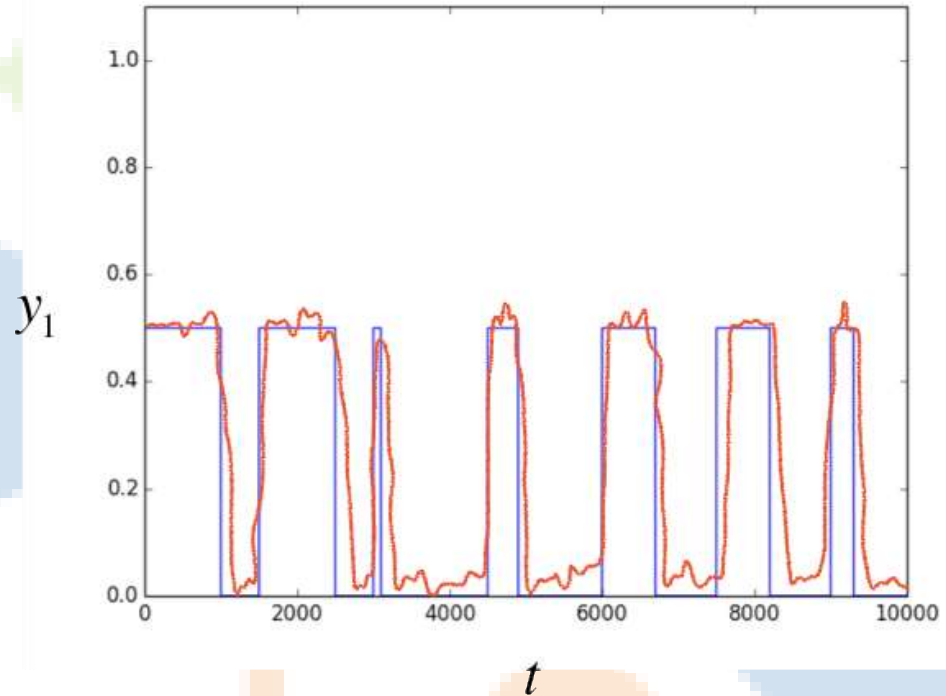


Saída do treinamento



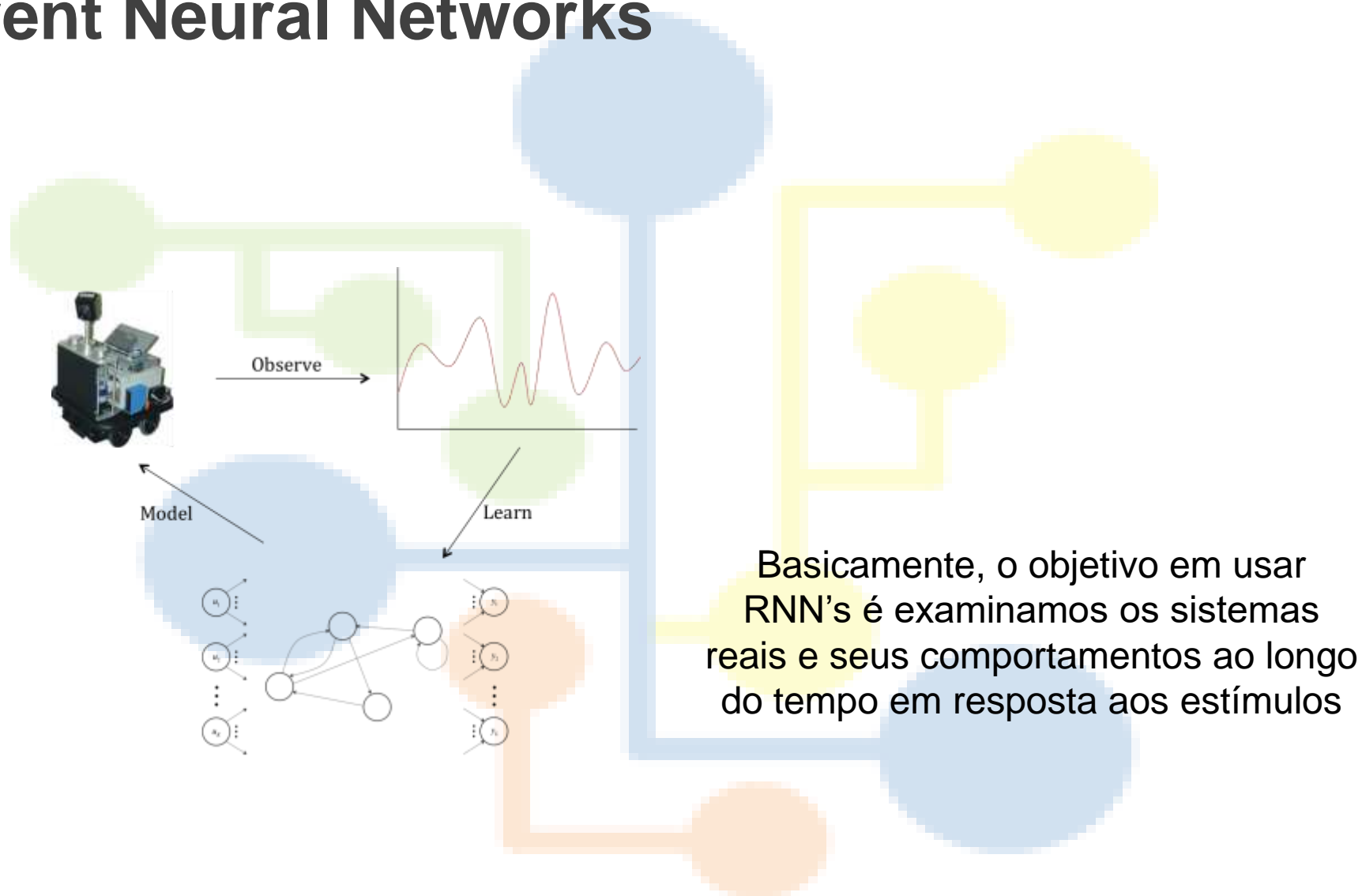


# Recurrent Neural Networks





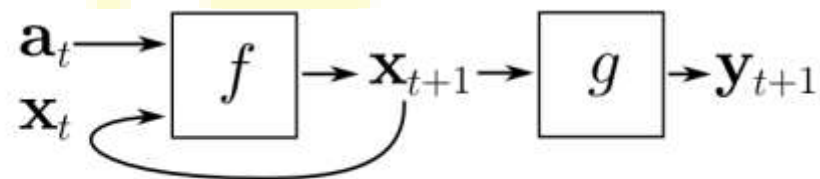
# Recurrent Neural Networks



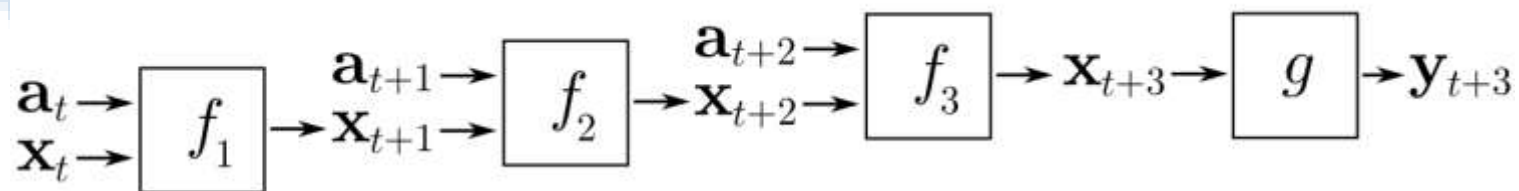


# Recurrent Neural Networks

Retropropagação através do tempo  
(Backpropagation Through Time, BPTT)



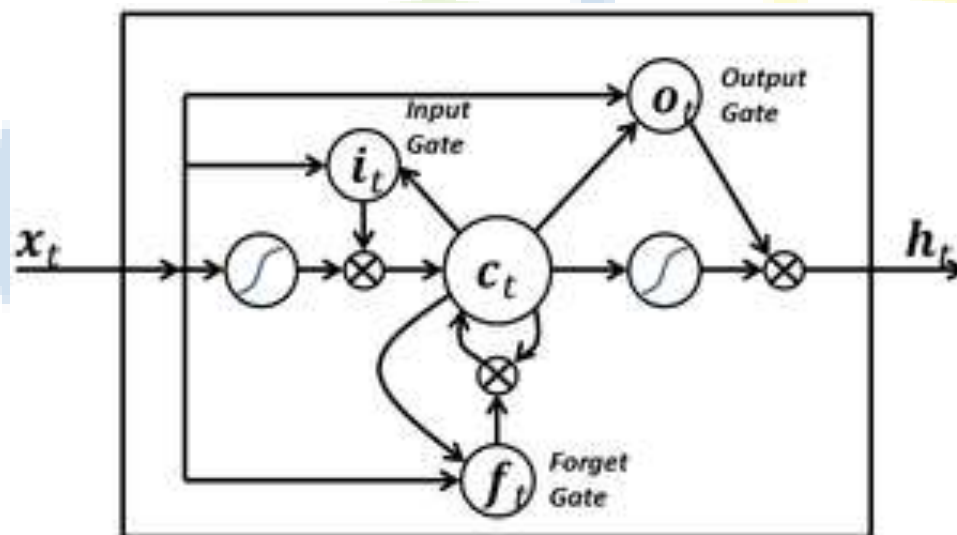
↓ unfold through time ↓





# Recurrent Neural Networks

LSTM (Long Short-Term Memory)



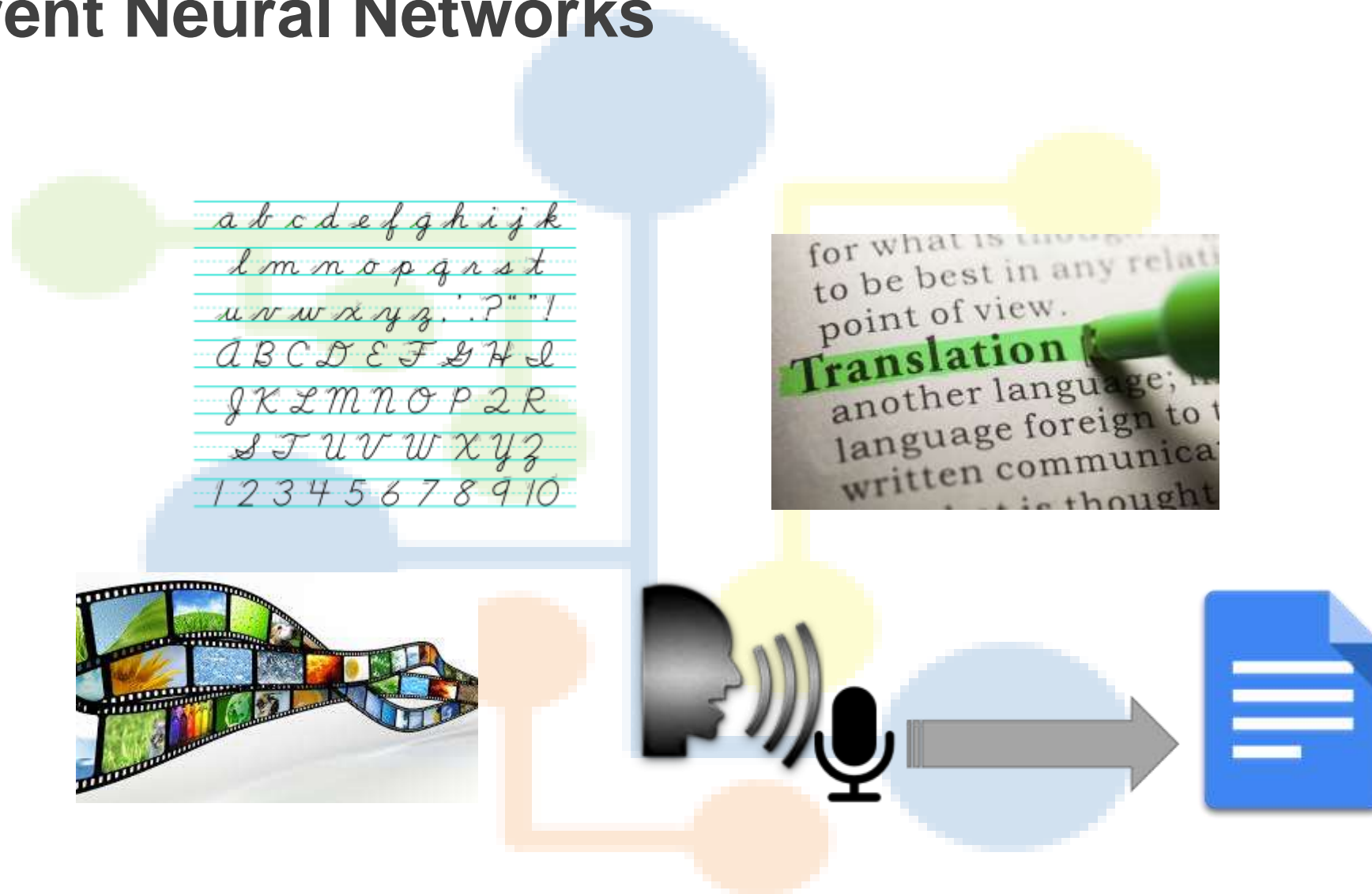
GRU (Gated Recurrent Unit)







# Recurrent Neural Networks





Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# LSTM



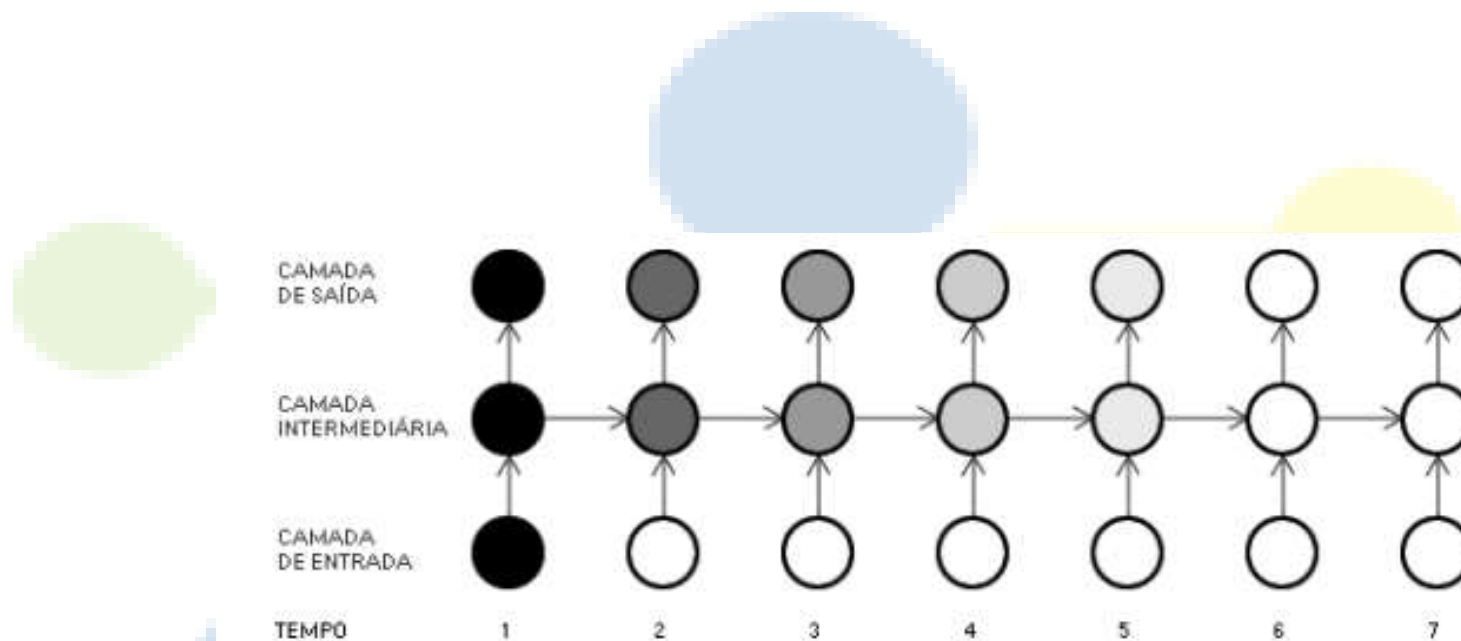
Data Science Academy



Data Science Academy



# LSTM



RNN's e o problema da dissipação do gradiente (vanishing gradient problem)

Problema da dissipação do gradiente. Os nós representam neurônios e seu conteúdo representa a sensibilidade para a entrada, quanto mais escuro o nó, mais sensível à entrada preta.

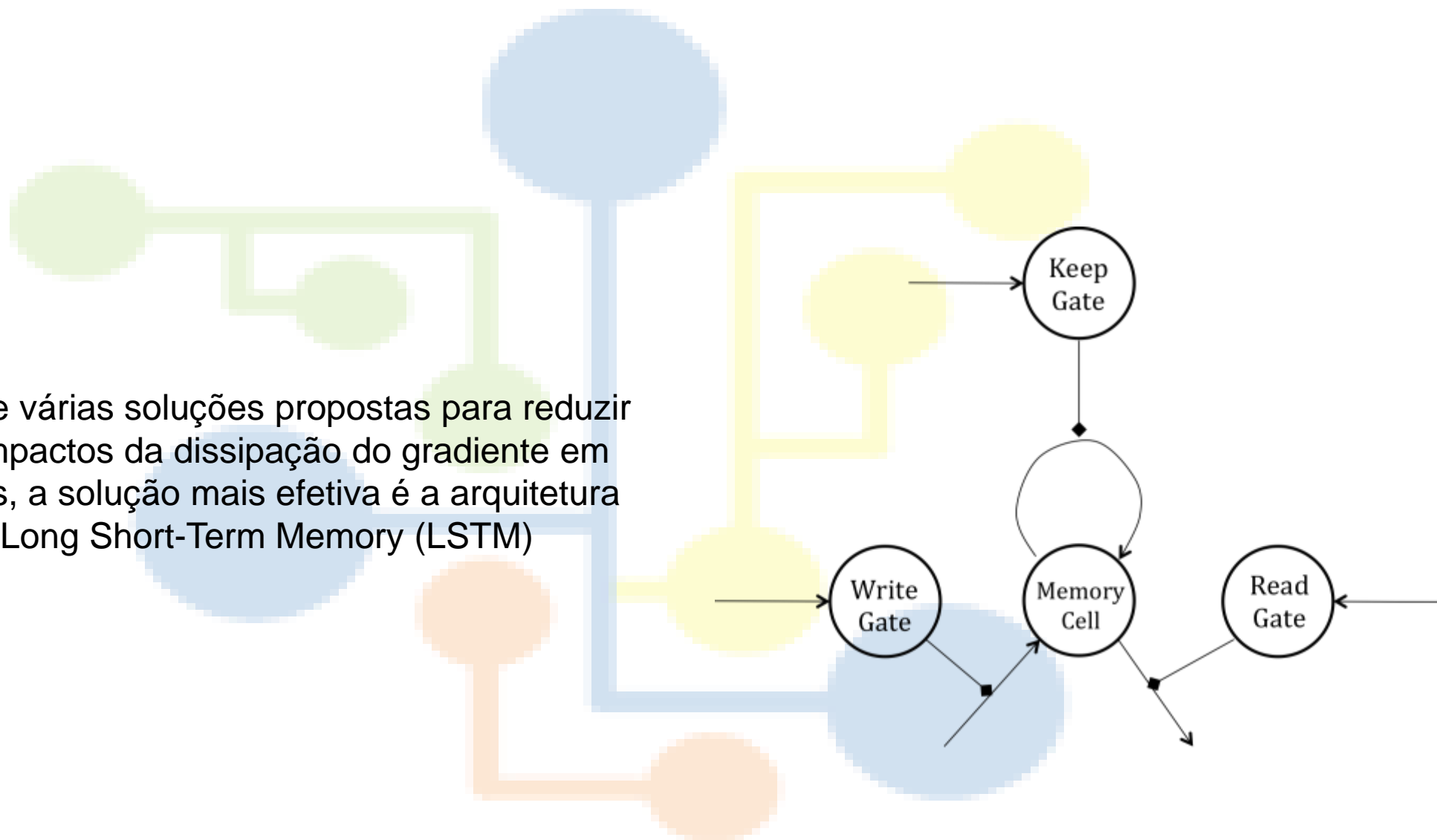
À medida que novas entradas são apresentadas à rede, a ativação da camada intermediária é sobrescrita e a rede "esquece" as primeiras entradas.





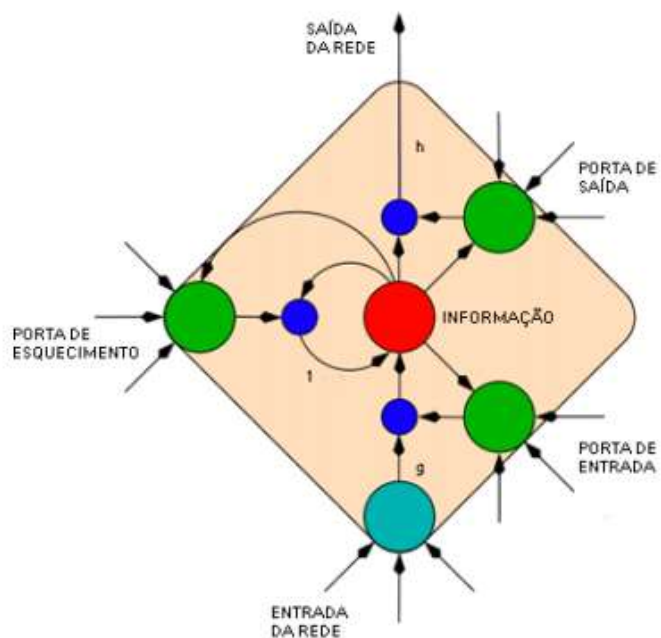
# LSTM

Dentre várias soluções propostas para reduzir os impactos da dissipação do gradiente em RNNs, a solução mais efetiva é a arquitetura Long Short-Term Memory (LSTM)





# LSTM



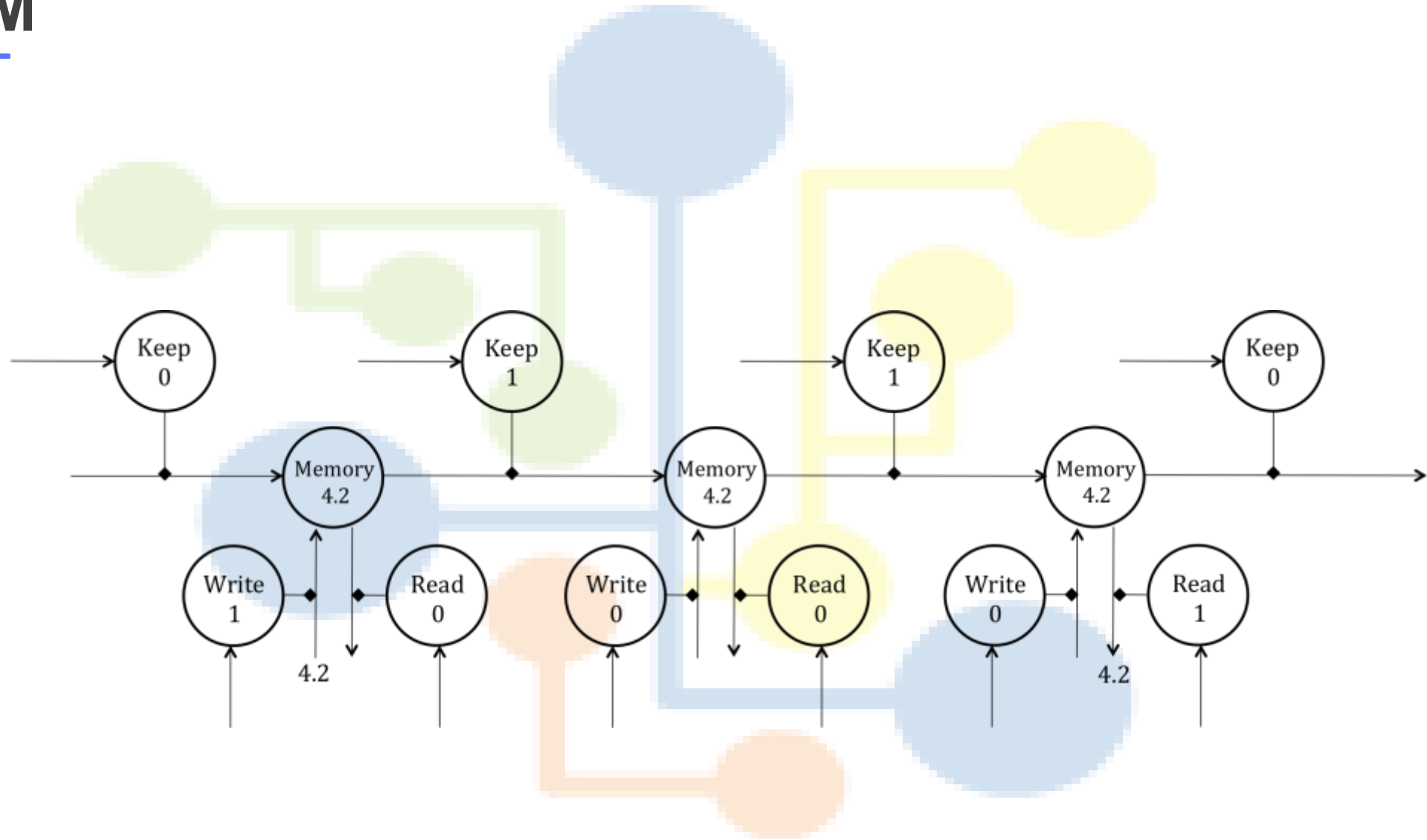
Cada bloco possui uma ou mais células de memória autoconectadas e três unidades de multiplicação que definem a operação que deve ser realizada, as portas de entrada, saída e de esquecimento.





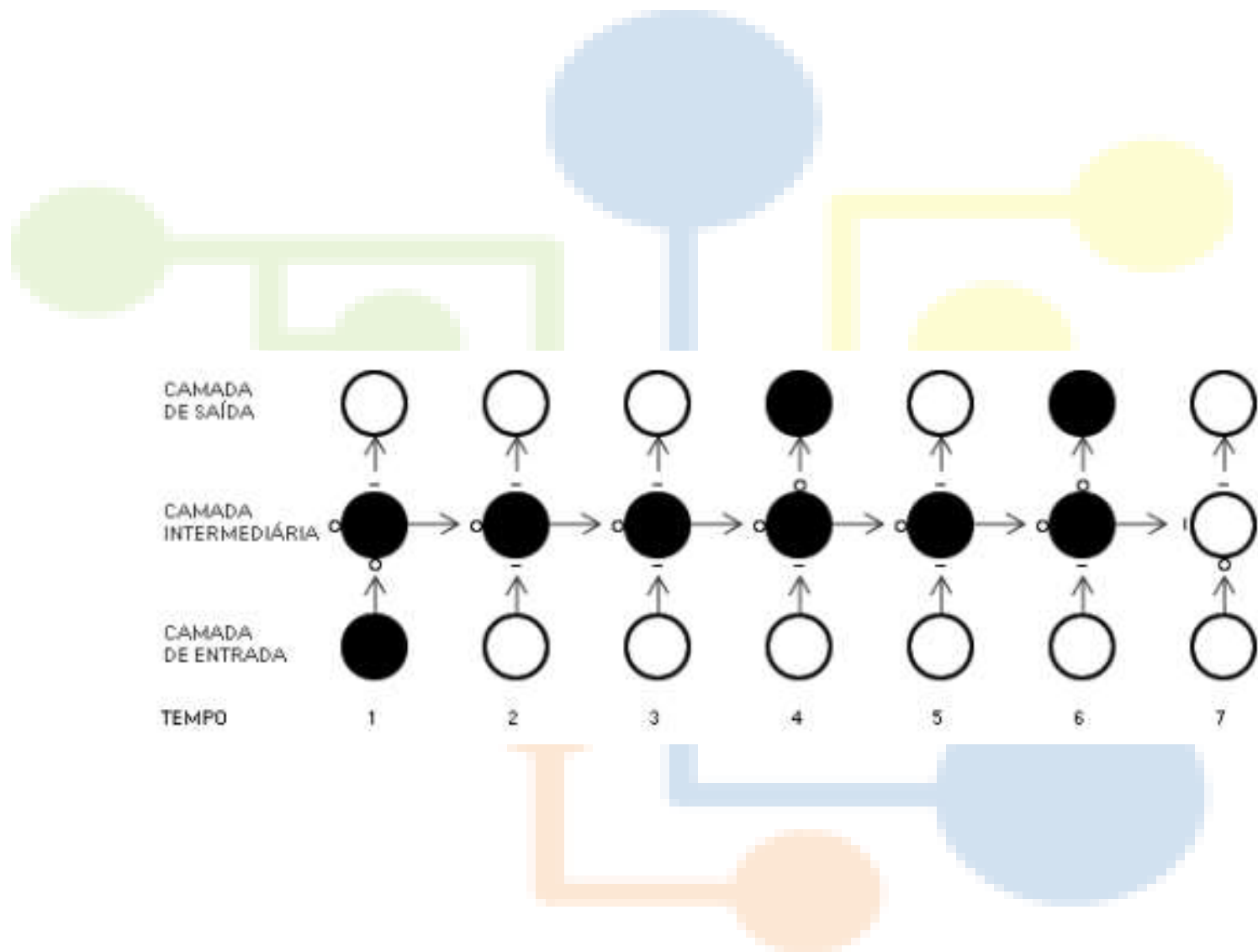


# LSTM





# LSTM





Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# Função Softmax



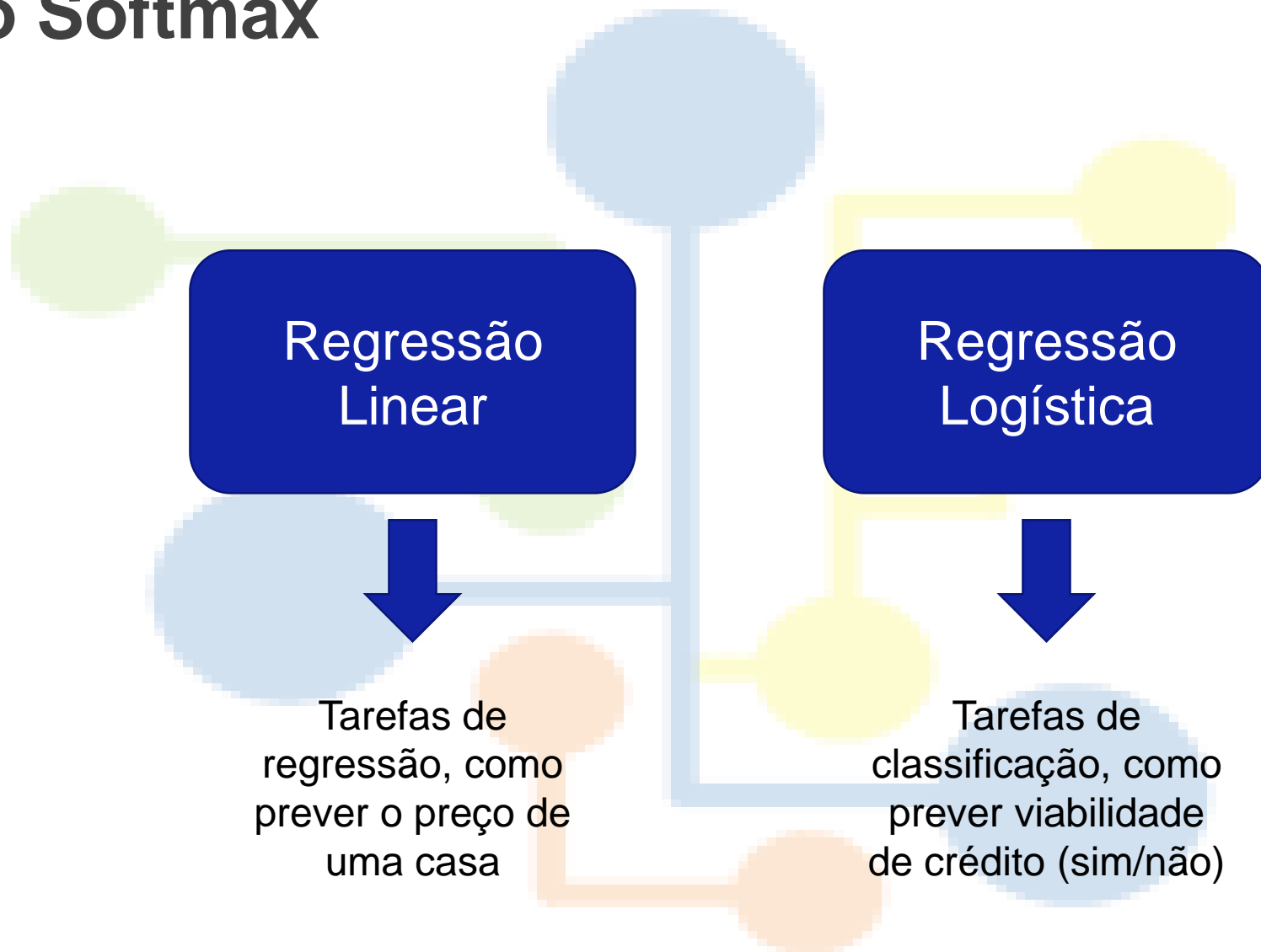
Data Science Academy



Data Science Academy



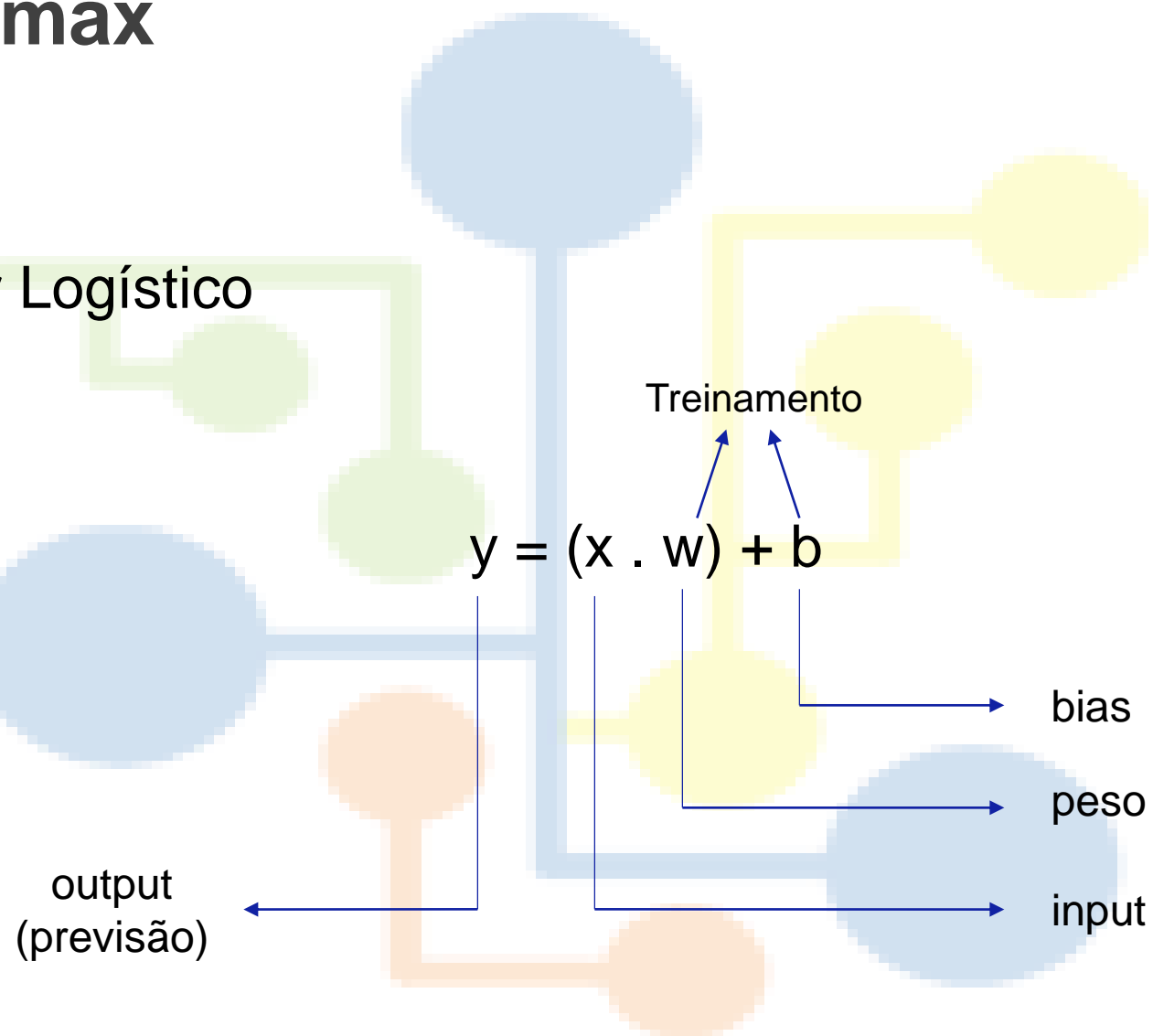
# Função Softmax





# Função Softmax

Classificador Logístico

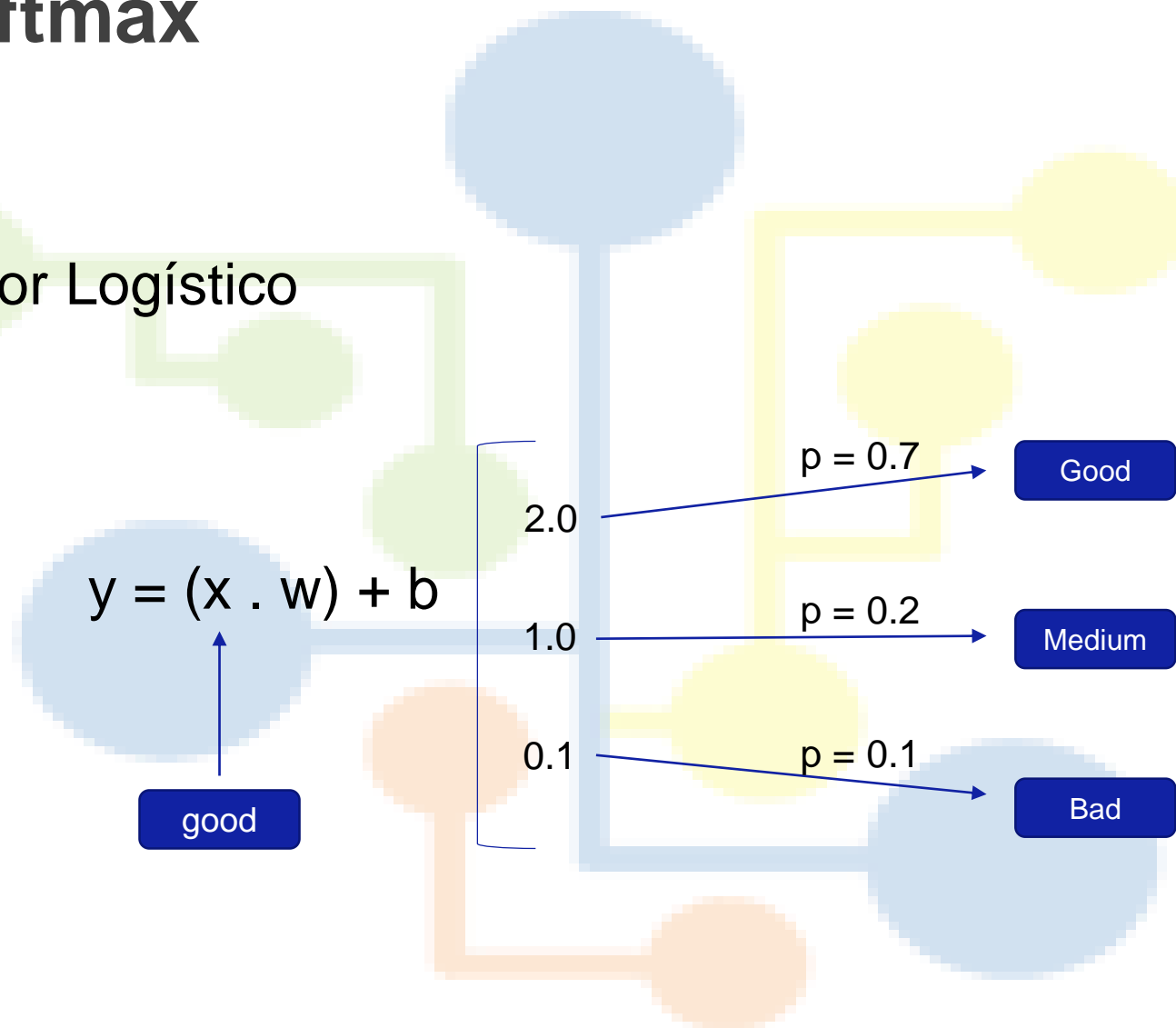






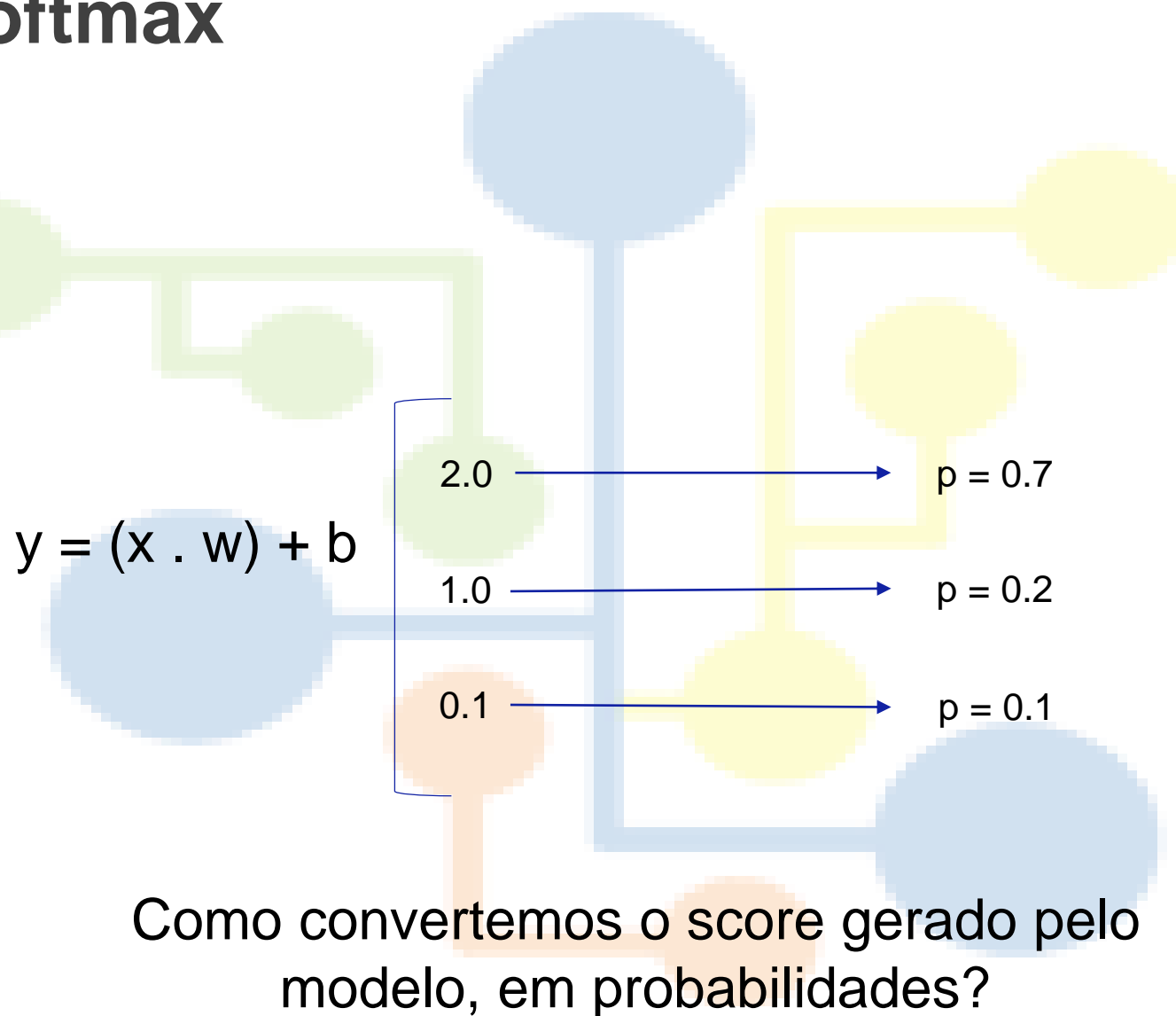
# Função Softmax

Classificador Logístico



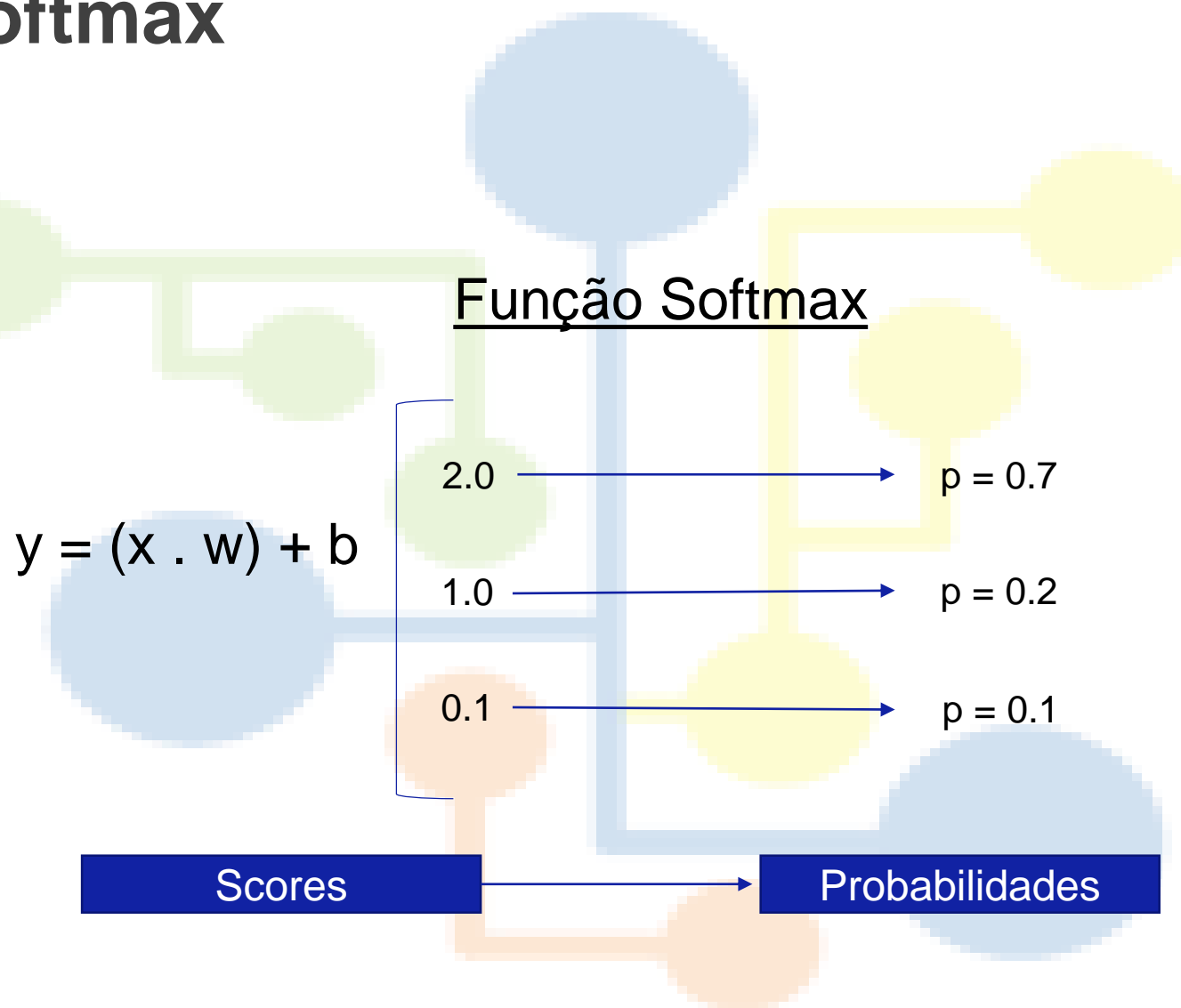


# Função Softmax





# Função Softmax





# Função Softmax

## Função Softmax

```
import numpy as np

scores = [3.0, 1.0, 0.2]

def softmax(x):
    return np.exp(x) / np.sum(np.exp(x), axis = 0)

print(softmax(scores))
```

```
[ 0.8360188  0.11314284  0.05083836]
```





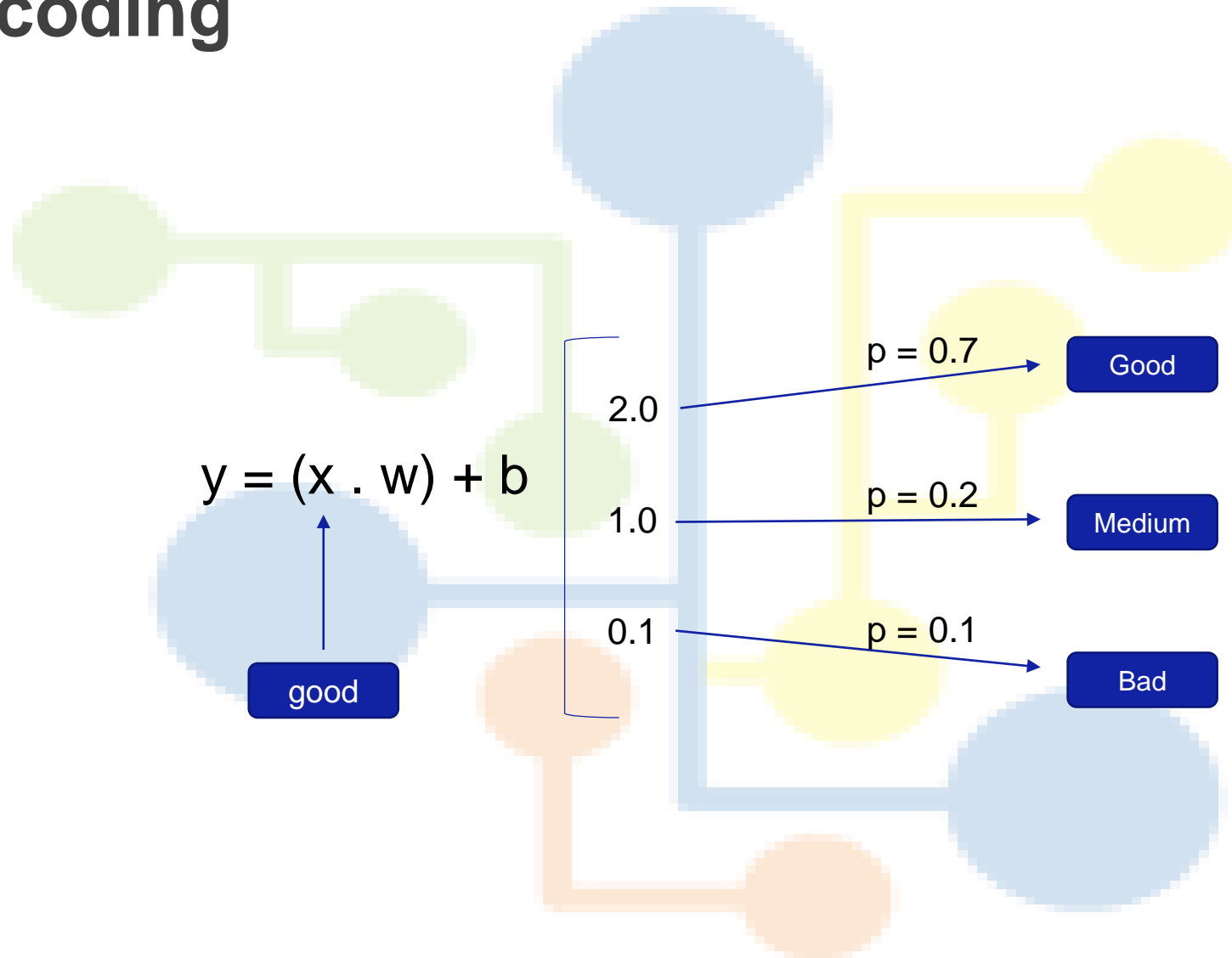
# Hot Encoding





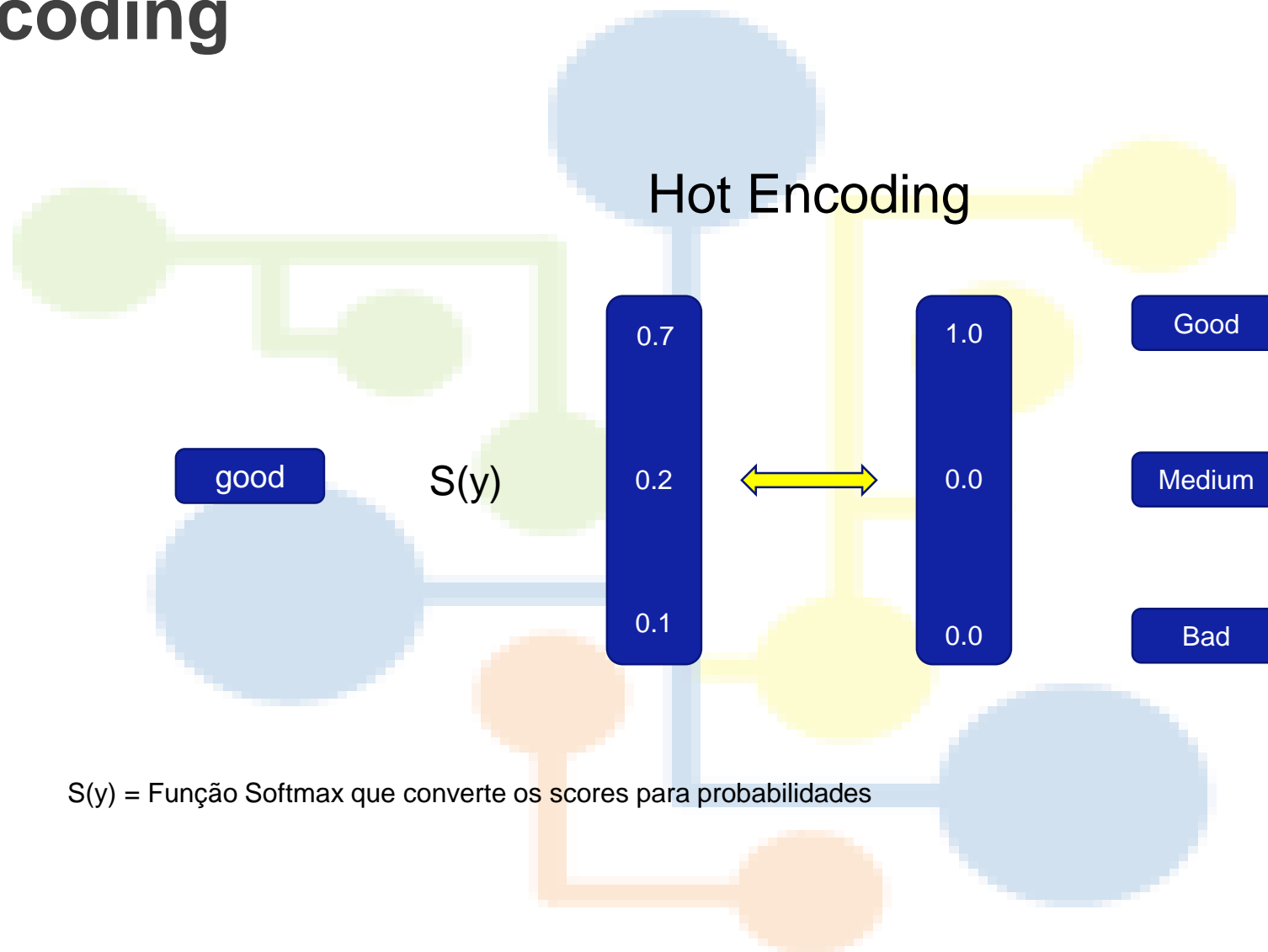


# Hot Encoding





# Hot Encoding





Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# Stochastic Gradient Descent



Data Science Academy

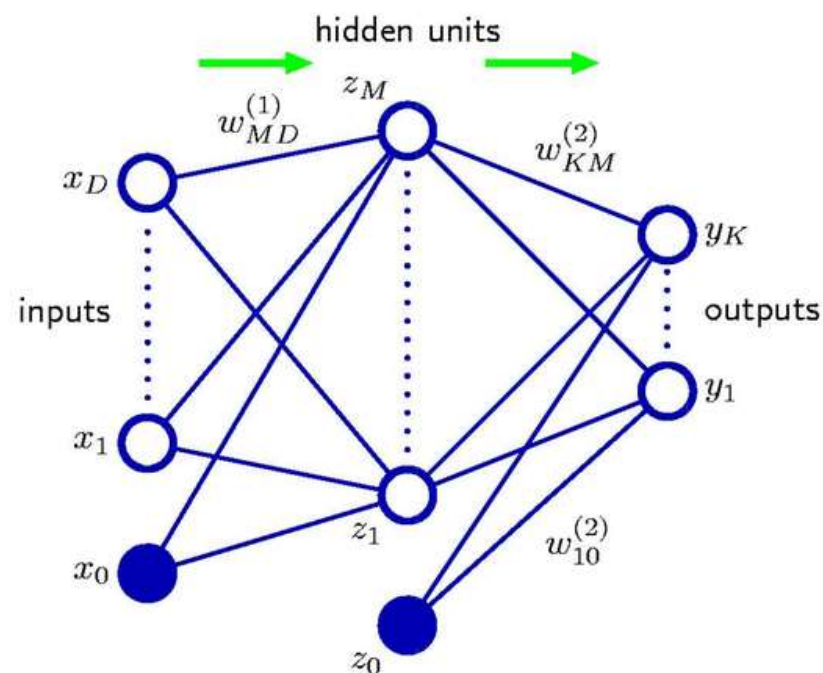


Data Science Academy



# Stochastic Gradient Descent

O treinamento de uma rede neural é convertido em um problema de otimização, cujo objetivo é minimizar o erro cometido pela rede, quando considerados todos os exemplos de treinamento.

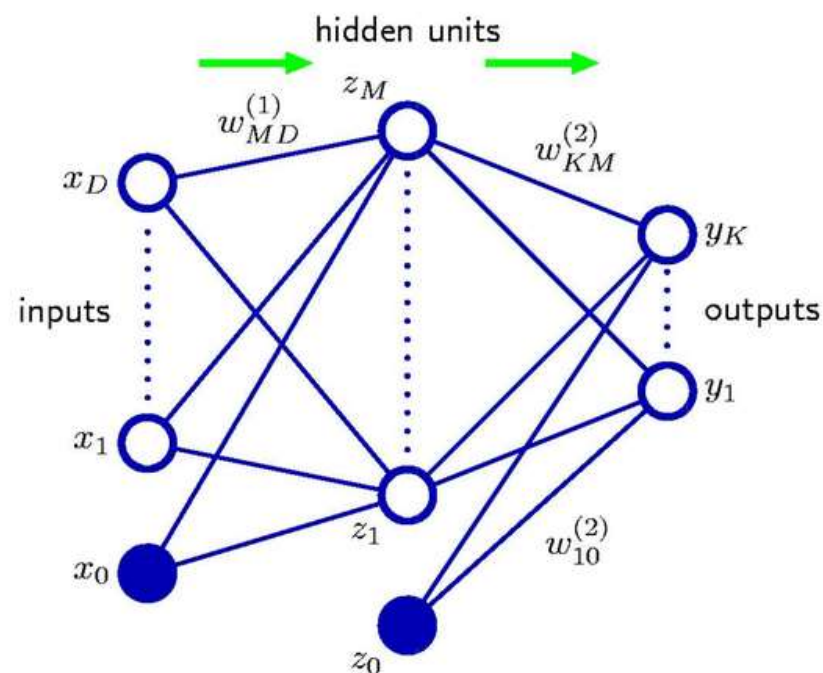




# Stochastic Gradient Descent

$$J( f(x(t); \theta), y(t) )$$

$$J(\theta) = \underbrace{\frac{1}{T} \sum_{t=1}^T J(f(\mathbf{x}^{(t)}; \theta), \mathbf{y}^{(t)})}_{\text{custo dos dados}} + \underbrace{\frac{\lambda}{2T} \sum_k \sum_l W_{k,l}^2}_{\text{custo de regularização}}$$



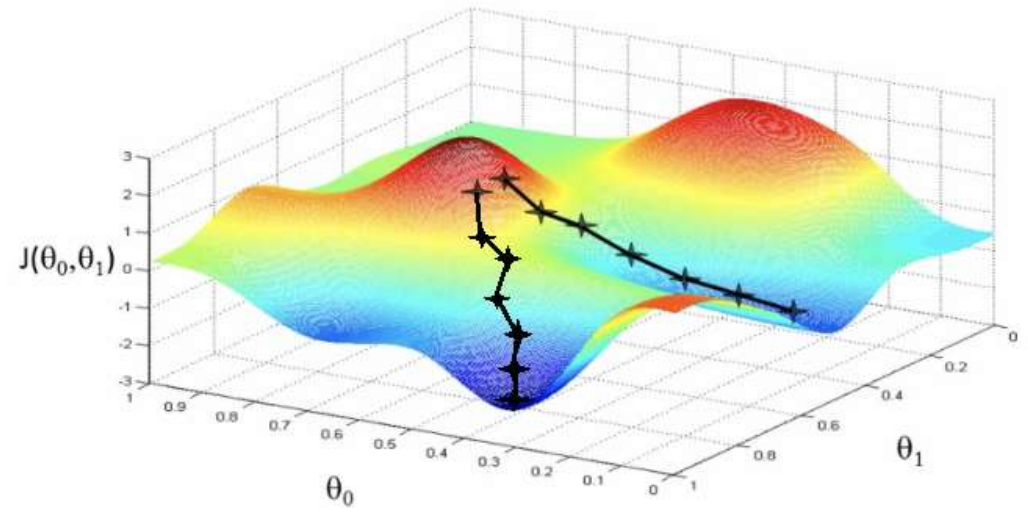




# Stochastic Gradient Descent

$$J(f(x(t); \theta), y(t))$$

$$J(\theta) = \underbrace{\frac{1}{T} \sum_{t=1}^T J(f(\mathbf{x}^{(t)}; \theta), y^{(t)})}_{\text{custo dos dados}} + \underbrace{\frac{\lambda}{2T} \sum_k \sum_l W_{k,l}^2}_{\text{custo de regularização}}$$





# Stochastic Gradient Descent

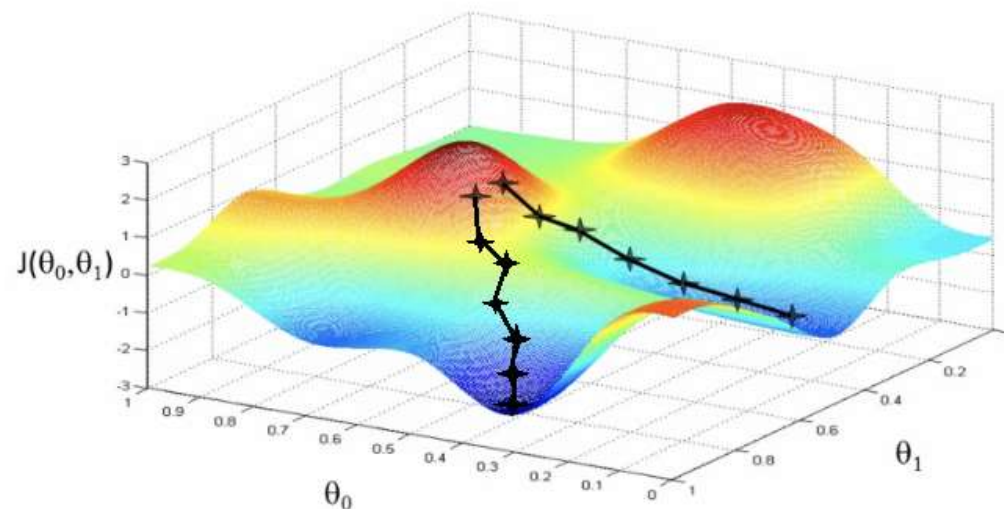
O gradiente de uma função  $f$  mede o quanto  $f$  varia uma vez que seus argumentos são alterados. Se  $f$  for uma função multivariada de  $n$  variáveis, então  $\nabla f$  é um vetor  $n$ -dimensional cujas componentes são as derivadas parciais de  $f$





# Stochastic Gradient Descent

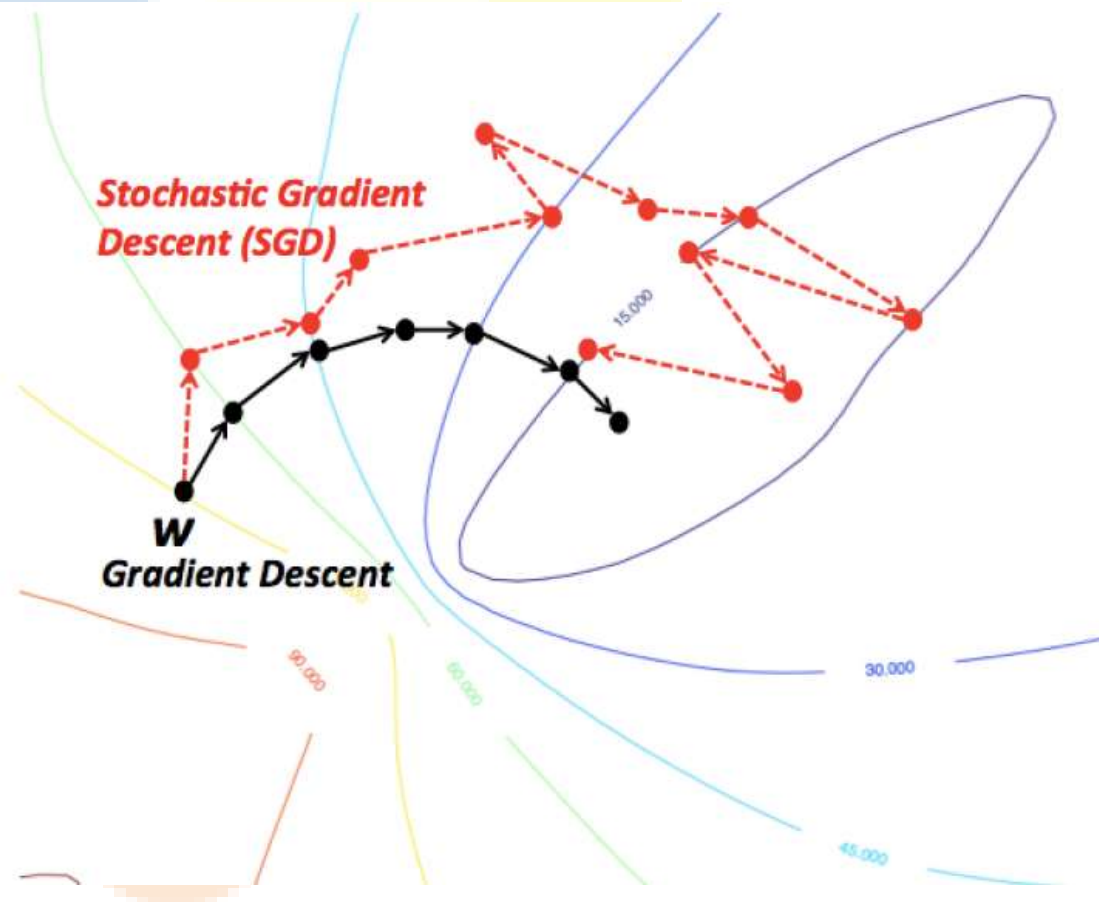
Além de ser computacionalmente intensivo, com Gradient Descent você precisa calcular o gradiente de cada elemento do seu conjunto de treinamento, o que pode levar muito tempo em grandes conjuntos de dados





# Stochastic Gradient Descent

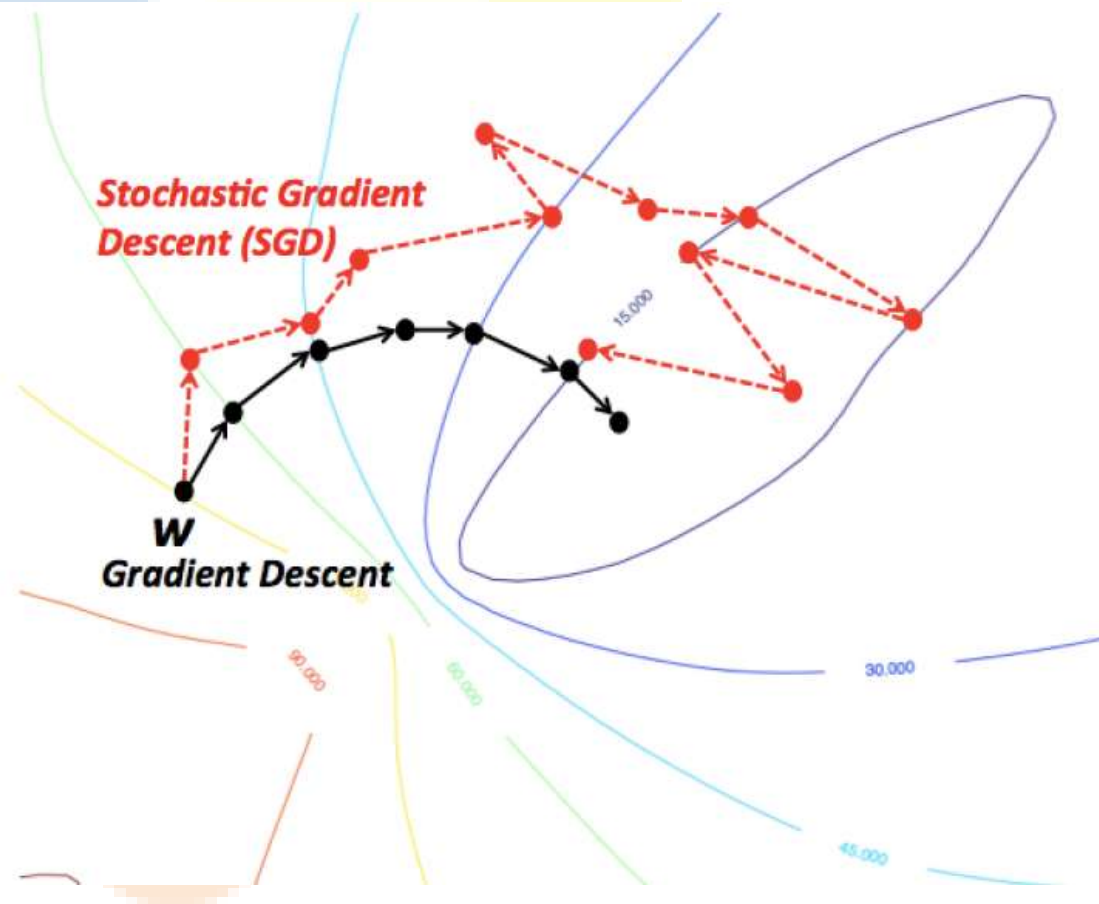
A solução encontrada para esse problema, foi o Stochastic Gradient Descent (SGD) que é uma versão do Gradient Descent, em que trabalhamos com amostras aleatórias





# Stochastic Gradient Descent

SGD é uma aproximação de Gradient Descent e quanto mais lotes processados pela rede neural (ou seja, mais amostra aleatórias), melhor a aproximação

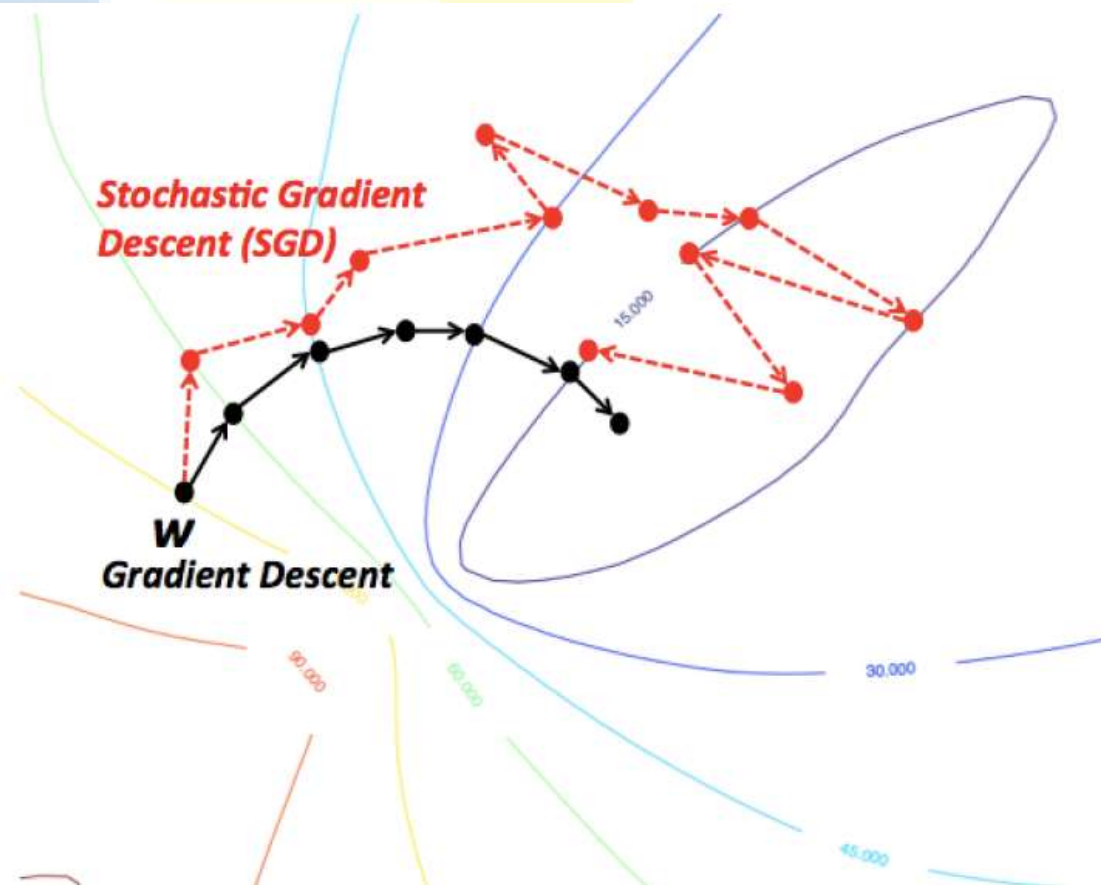




# Stochastic Gradient Descent

A implementação do SGD compreende:

1. Amostragem aleatória de um lote de dados do conjunto de dados total.
2. Executar a rede para frente e para trás para calcular o gradiente (com dados gerados no item 1).
3. Aplicar a atualização de descida do gradiente.
4. Repetir os passos 1 a 3 até a convergência ou o ciclo seja interrompido por outro mecanismo, como no caso de ser alcançado o número de épocas (epochs).



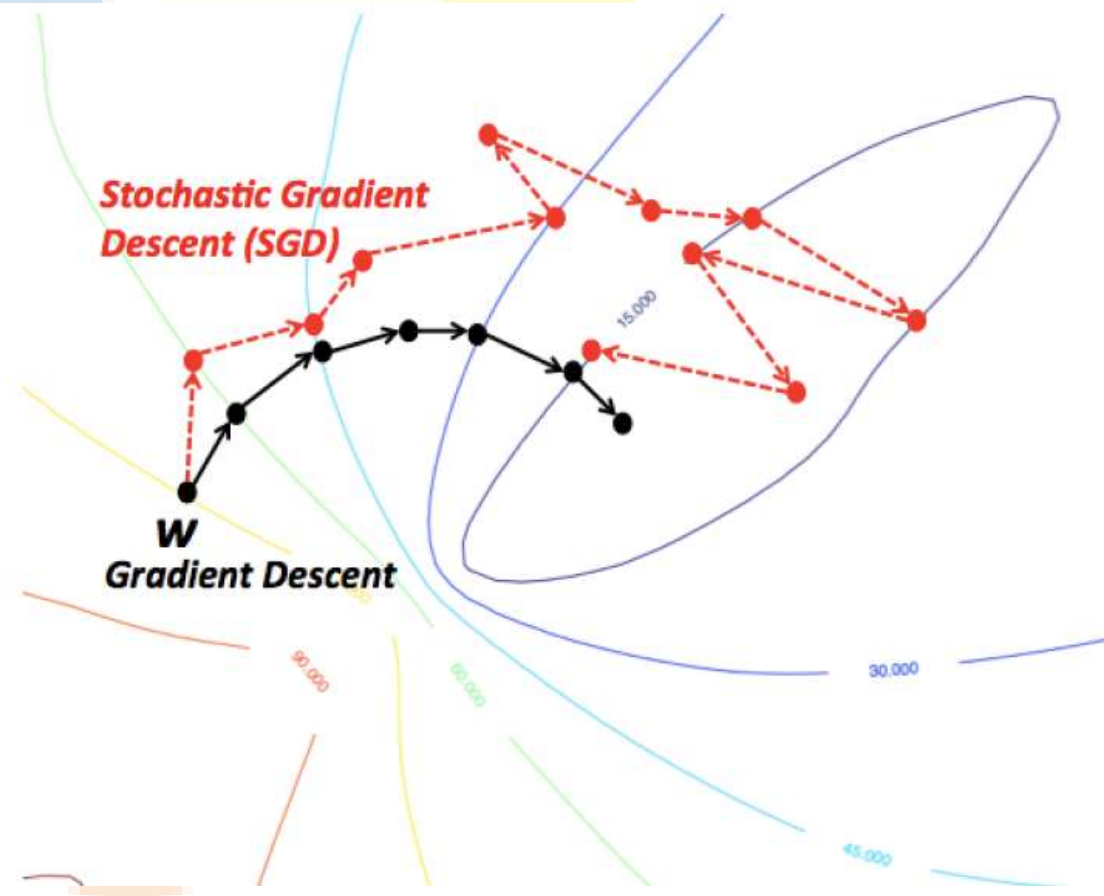




# Stochastic Gradient Descent

A técnica de Stochastic Gradient Descent está no cerne do Deep Learning.

Isso ocorre porque o SGD se equilibra bem com os dados e o tamanho do modelo, e queremos trabalhar com Big Data e modelos com muitas camadas ocultas.

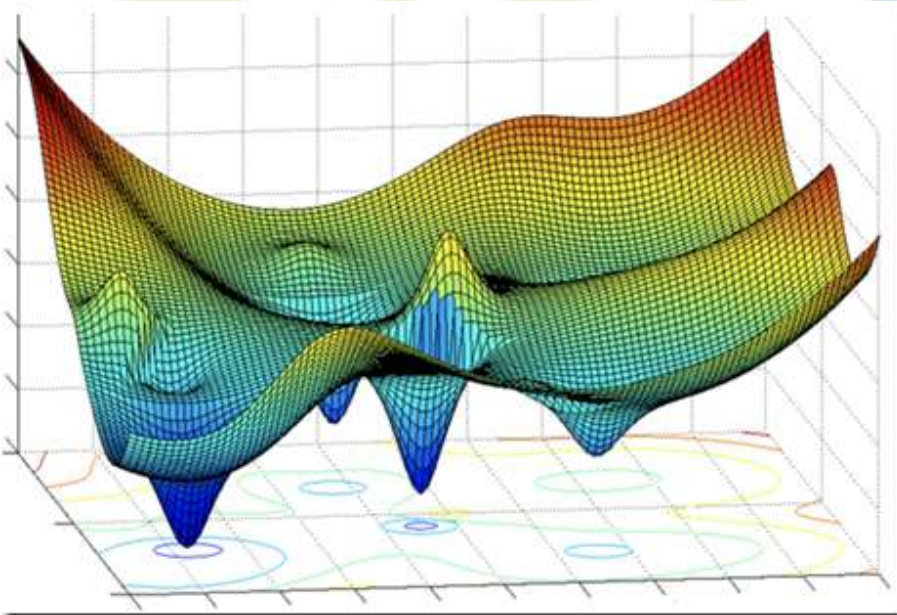




# Momentum e Learning Rate



# Momentum e Learning Rate



A forma mais comum para se treinar uma rede neural é conhecida como Stochastic Gradient Descent





# Momentum e Learning Rate

Tanto no GD quanto no SGD, você atualiza um conjunto de parâmetros de forma iterativa para minimizar a cost function.

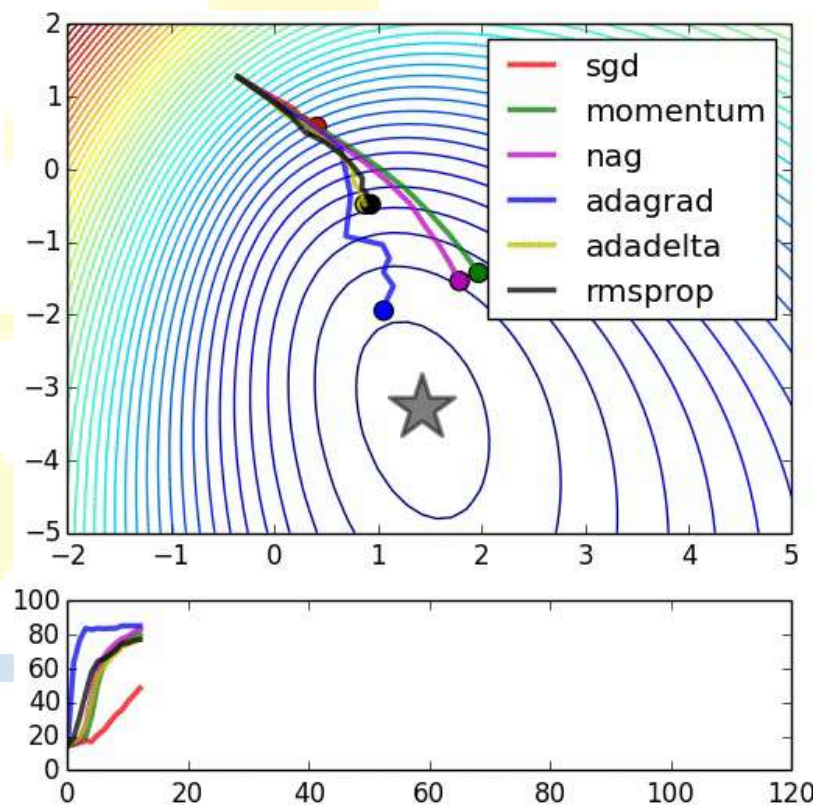
Enquanto com GD, você tem que percorrer TODAS as amostras em seu conjunto de treinamento para fazer uma única atualização para um parâmetro em uma iteração particular, com SGD você usa SOMENTE uma amostra de treinamento, do seu conjunto inteiro de treinamento, para fazer a atualização para um parâmetro em uma iteração específica.





# Momentum e Learning Rate

Os pesos da rede neural podem ser atualizados conforme os dados são processados e os erros são calculados (abordagem conhecida como online learning) ou ao final do processo (abordagem conhecida como batch learning).





# Momentum e Learning Rate

## Learning Rate

A função de perda (cost function) de redes neurais é como uma superfície, onde cada direção em que você pode se mover representa o valor de um peso. A descida do gradiente é como dar saltos na direção atual da inclinação, e a taxa de aprendizado é como o comprimento do salto que você dá.







# Momentum e Learning Rate

## Momentum

Incorpora as propriedades da atualização de pesos anterior e faz com que os pesos continuem sendo atualizados na mesma direção mesmo quando o erro diminui.

## Learning Rate Decay

Learning rate decay, é usado para diminuir o valor do learning rate conforme os erros diminuem.



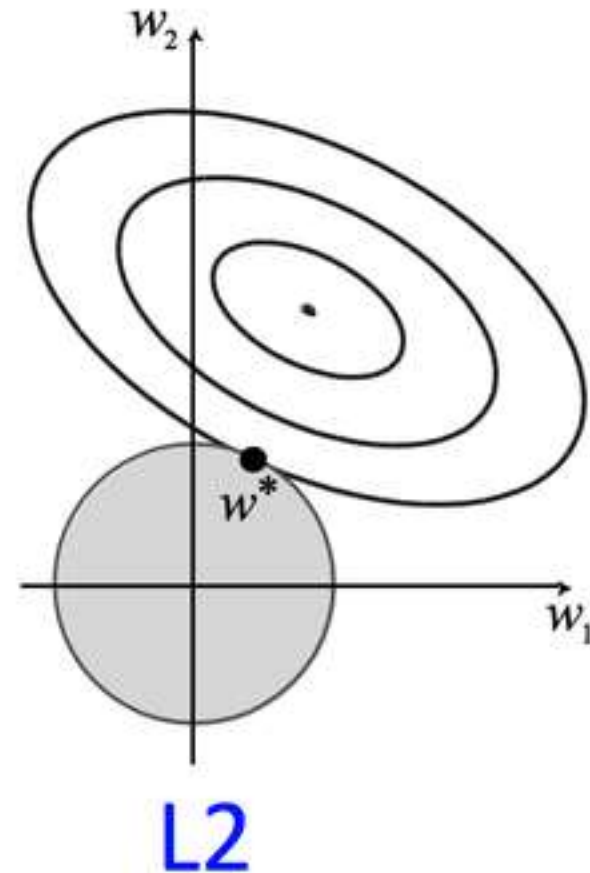
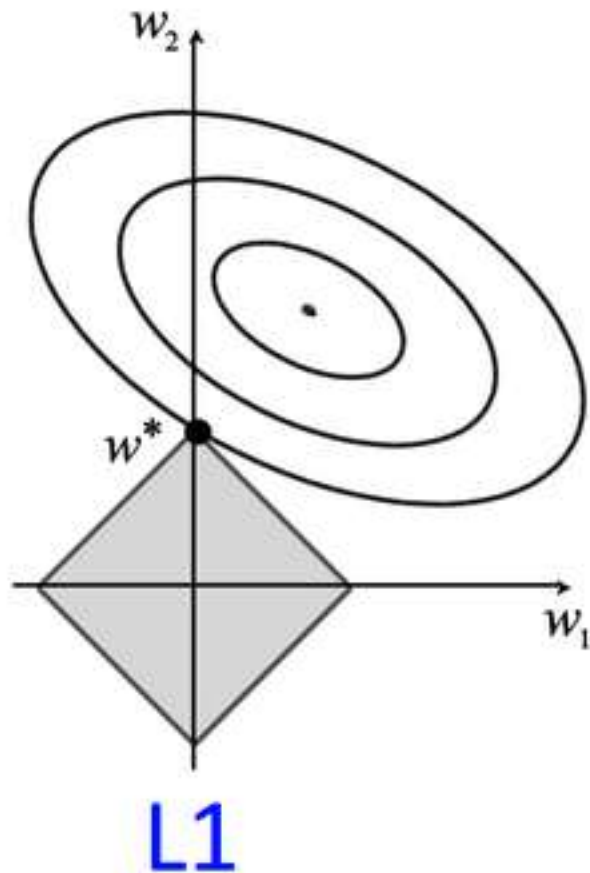


# Regularização e Dropout





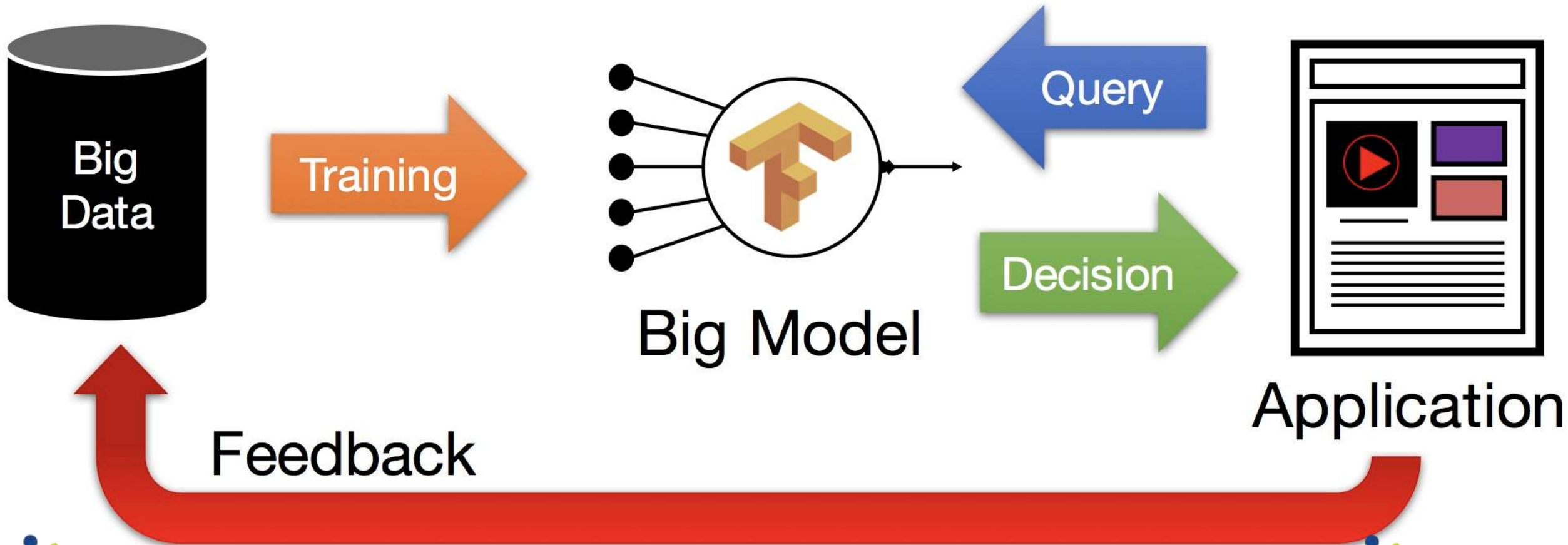
# Regularização e Dropout





# Learning

# Inference





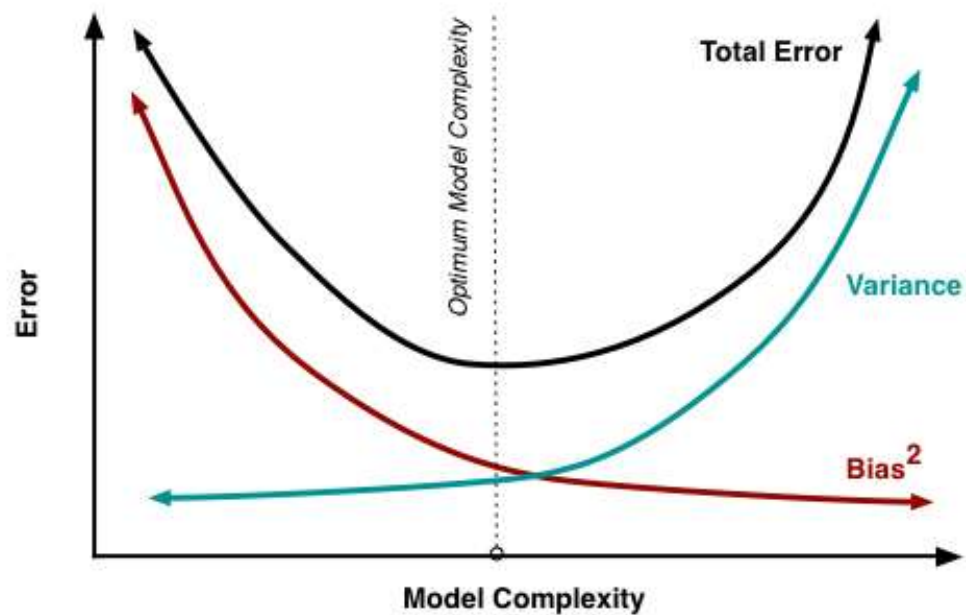
# Regularização e Dropout

Regularização é qualquer modificação que nós fazemos no algoritmo de aprendizagem, com o objetivo de reduzir o erro de generalização e não o erro de treinamento, a fim de evitar o overfitting.





# Regularização e Dropout







# Regularização e Dropout

Portanto, seu objetivo em Deep Learning é encontrar um modelo que seja grande e profundo o suficiente para representar a complexidade nos dados e que possa ser aplicado a novos conjuntos de dados, com um bom desempenho. A regularização é uma das formas usadas para se alcançar esse objetivo.





# Regularização e Dropout



Regularização  
L2

Regularização  
L1

Dropout





Data Science  
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

# Obrigado



Data Science Academy



Data Science Academy