

IMT – Instituto Mauá de Tecnologia

Redes Neurais Introdução

Prof. Jones Egydio

jones.egydio@maua.br

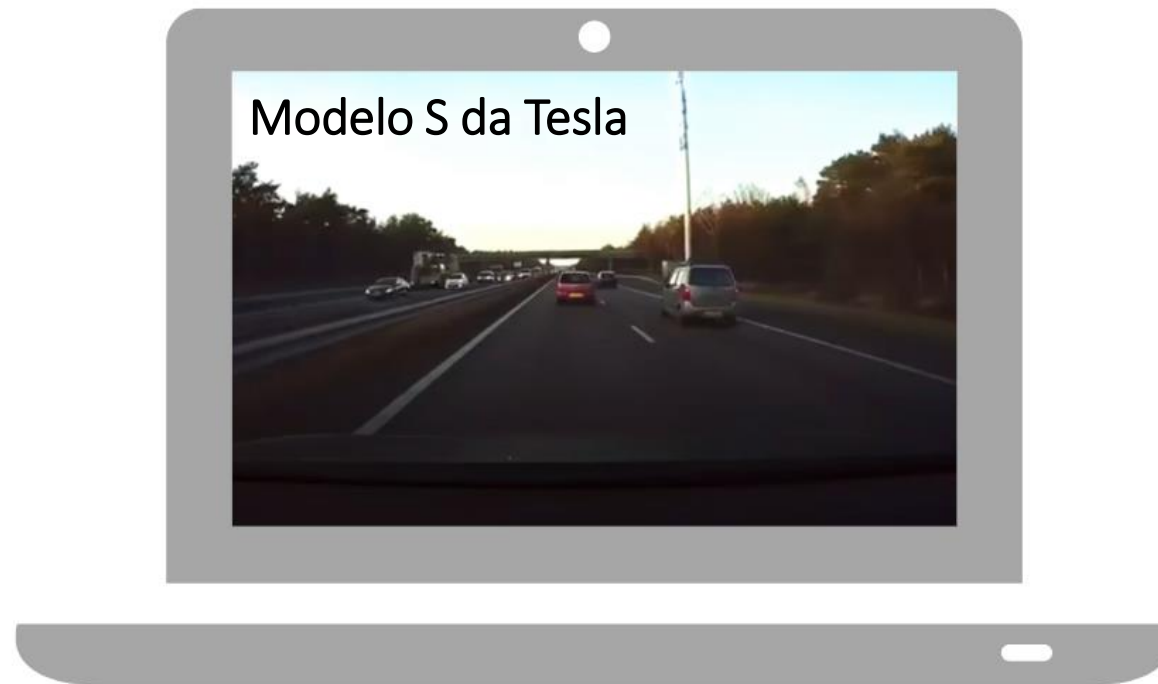
Objetivos

- Relevância das Redes Neurais Artificiais (RNAs) no atual contexto com alguns exemplos;
- Conceituação teórica e funcionamento de uma RNA;
 - Processo de desenvolvimento de uma RNA;
 - Processo de treinamento de uma RNA;
- Caso prático de solução de problema em Python;
 - Classificação multiclasse usando imagens de artigos de vestuário;
 - Conclusões;
- Perguntas.



RNNs – *Recurrent Neural Networks*

- Exemplo 1: É possível prever um acidente utilizando apenas uma imagem?



Fonte: Internet

RNNs – *Recurrent Neural Networks*

- Exemplo 2: Criação de música – Music Generation – sinfonia inacabada de Schubert.

Edição: ESPAÑA

ASSINE FAÇA LOGIN

EL PAÍS

CINEMA · LITERATURA · LIVROS · MÚSICA · ARTE · TELEVISÃO

Cultura

Um algoritmo completa a misteriosa ‘Sinfonia Inacabada’ de Schubert

Vários músicos consideram a iniciativa, que é parte de uma campanha da empresa de tecnologia chinesa Huawei, artificial e carente de alma

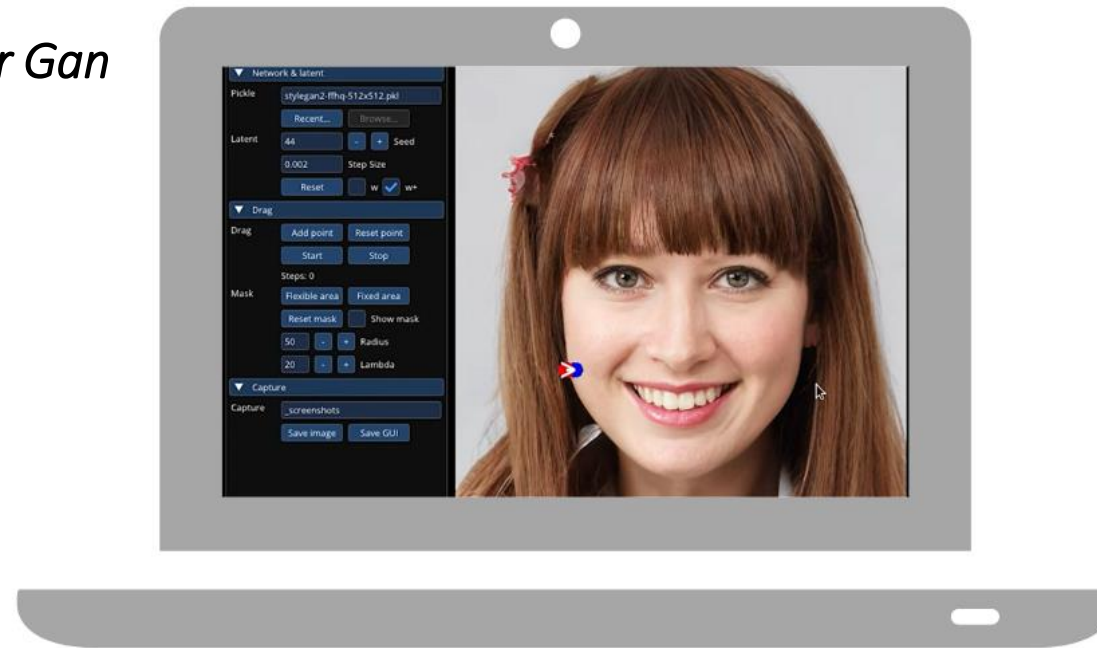
Fonte: El País



GANs – *Generative Adversarial Networks*

- Exemplo 3: É possível a geração de imagens realistas ou realizar edições de alta qualidade sem descaracterizar a naturalidade do objeto fotografado?

Drag Your Gan



Fonte: Internet

Artigo



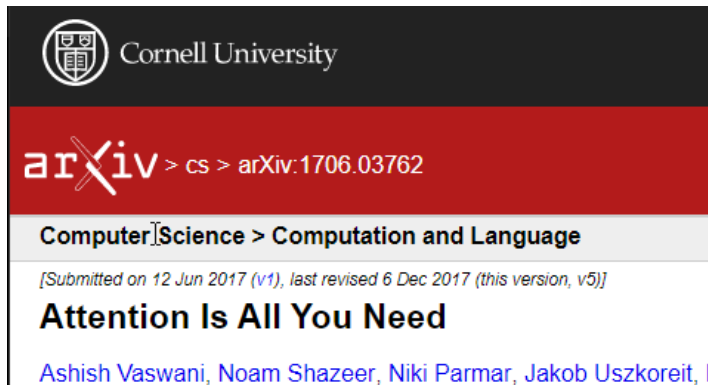


Transformers

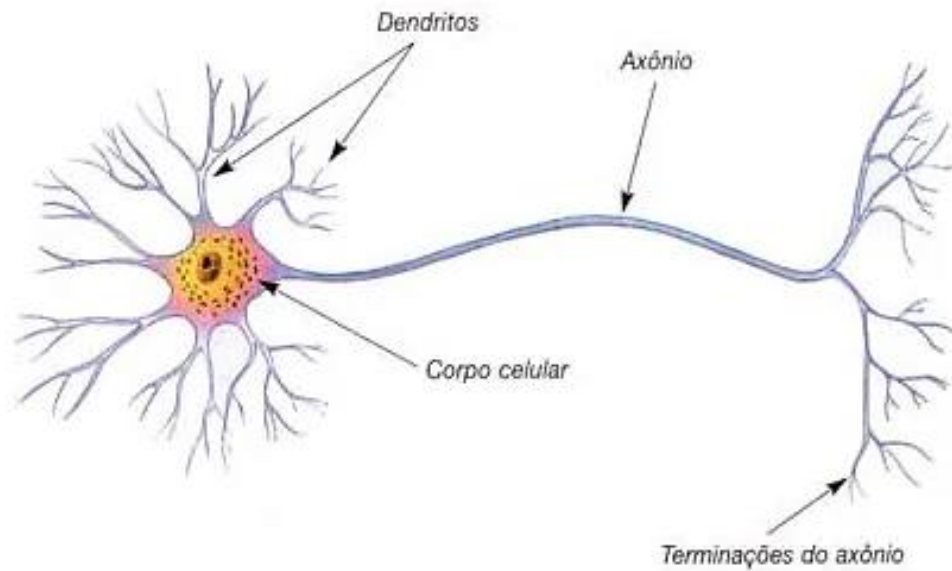
- Exemplo 4: Chat GPT (*Generative Pre-Trained Transformer*) permite ao usuário criar diálogos virtuais e gerar textos assertivos.



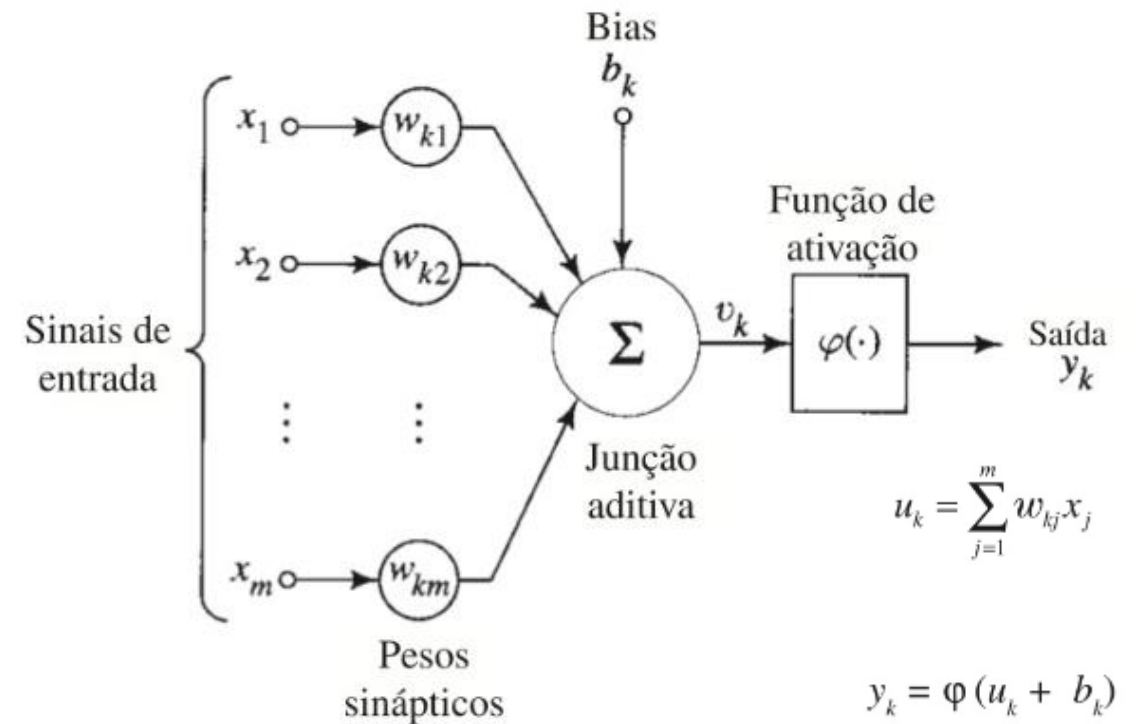
O ChatGPT, o popular chatbot da OpenAI, atingiu 100 milhões de usuários ativos mensais em janeiro, apenas dois meses após o lançamento, tornando-se o aplicativo de consumo de crescimento mais rápido da história, de acordo com comentário do banco suíço UBS hoje (1)



Neurônio biológico e *modelo* de um neurônio



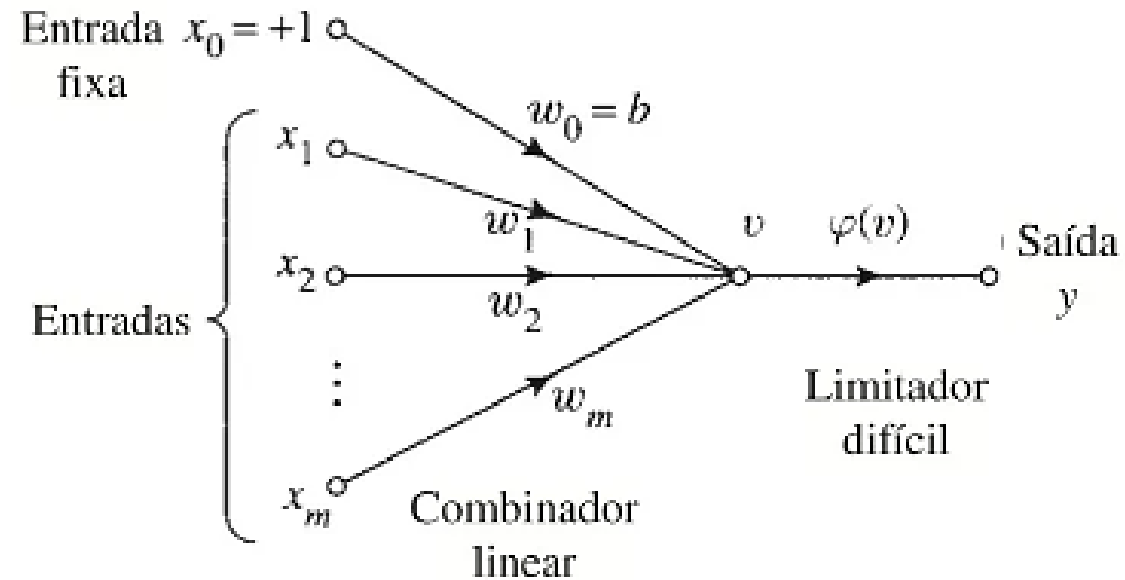
Fonte: Esquema de um neurônio biológico (Gomes, I., Visão Geral de RN)



Fonte: Redes Neurais – Simon Haykin

Perceptrons

O *perceptron* é a forma mais simples de uma rede neural usada para a classificação de padrões ditos *linearmente separáveis*, isto é, padrões que se encontram em lados opostos de um hiperplano.
(Haykin, S., Redes Neurais).



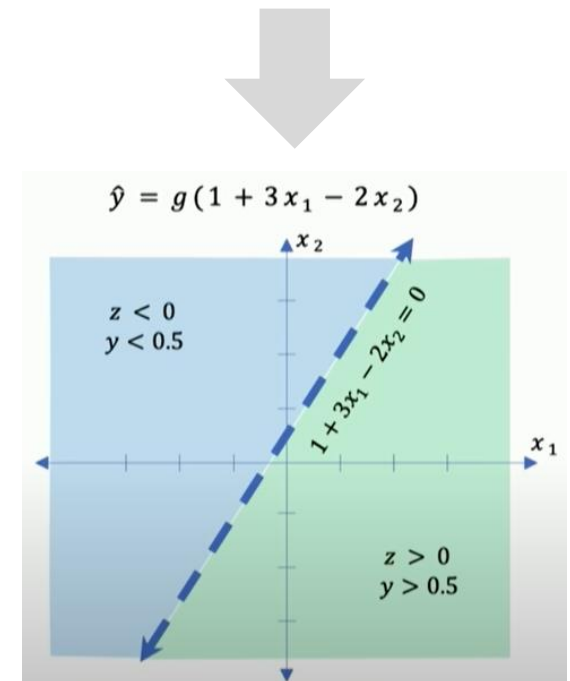
Correspondentemente, a saída do combinador linear pode ser escrita na forma compacta

$$v(n) = \sum_{i=0}^m w_i(n) x_i(n)$$

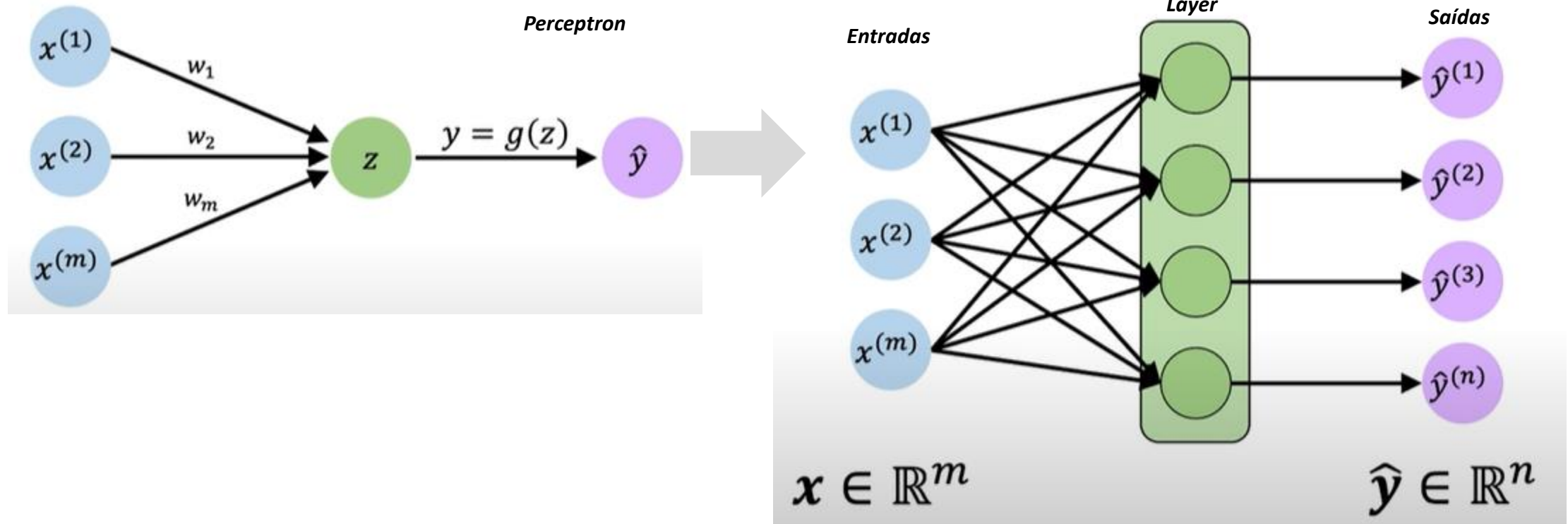
$$= \mathbf{w}^T(n) \mathbf{x}(n)_T$$

Fonte: Redes Neurais – Simon Haykin

Procedimento de aprendizagem desenvolvido por Rosenblatt (1958, 1962)



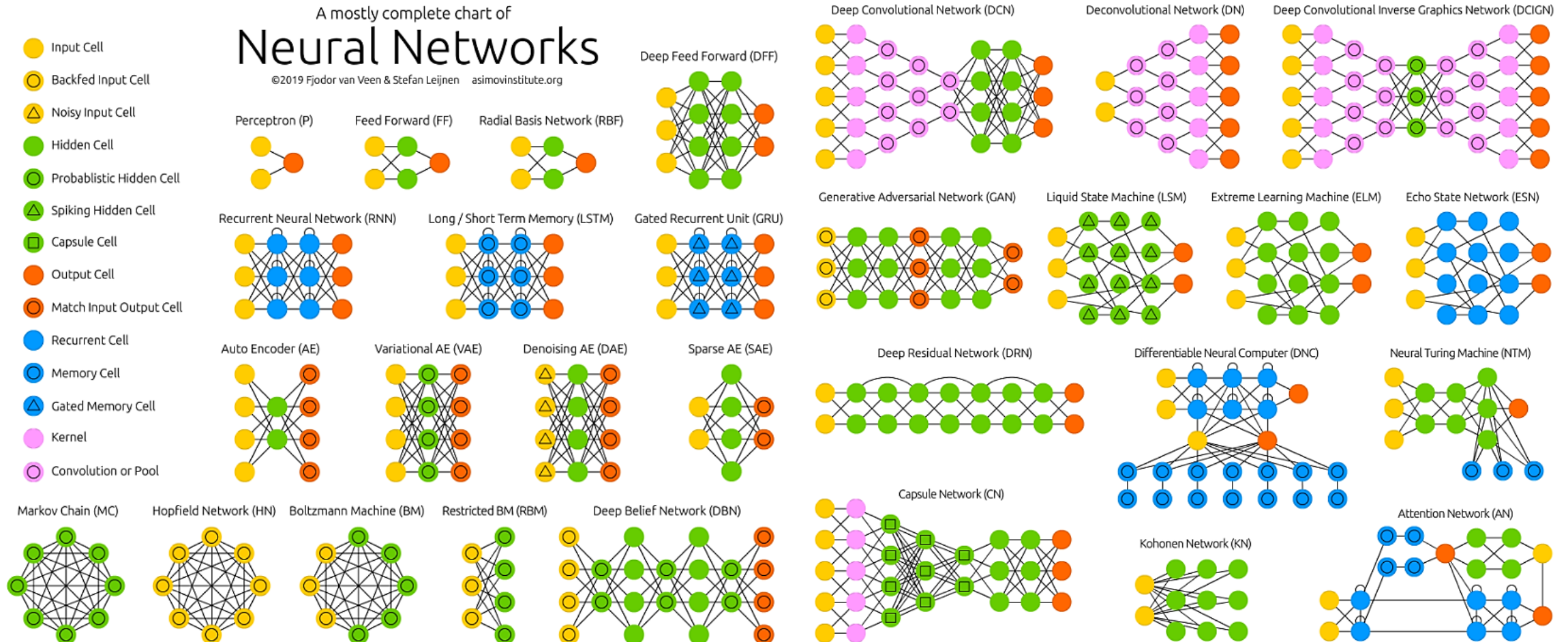
Feed-Forward Networks



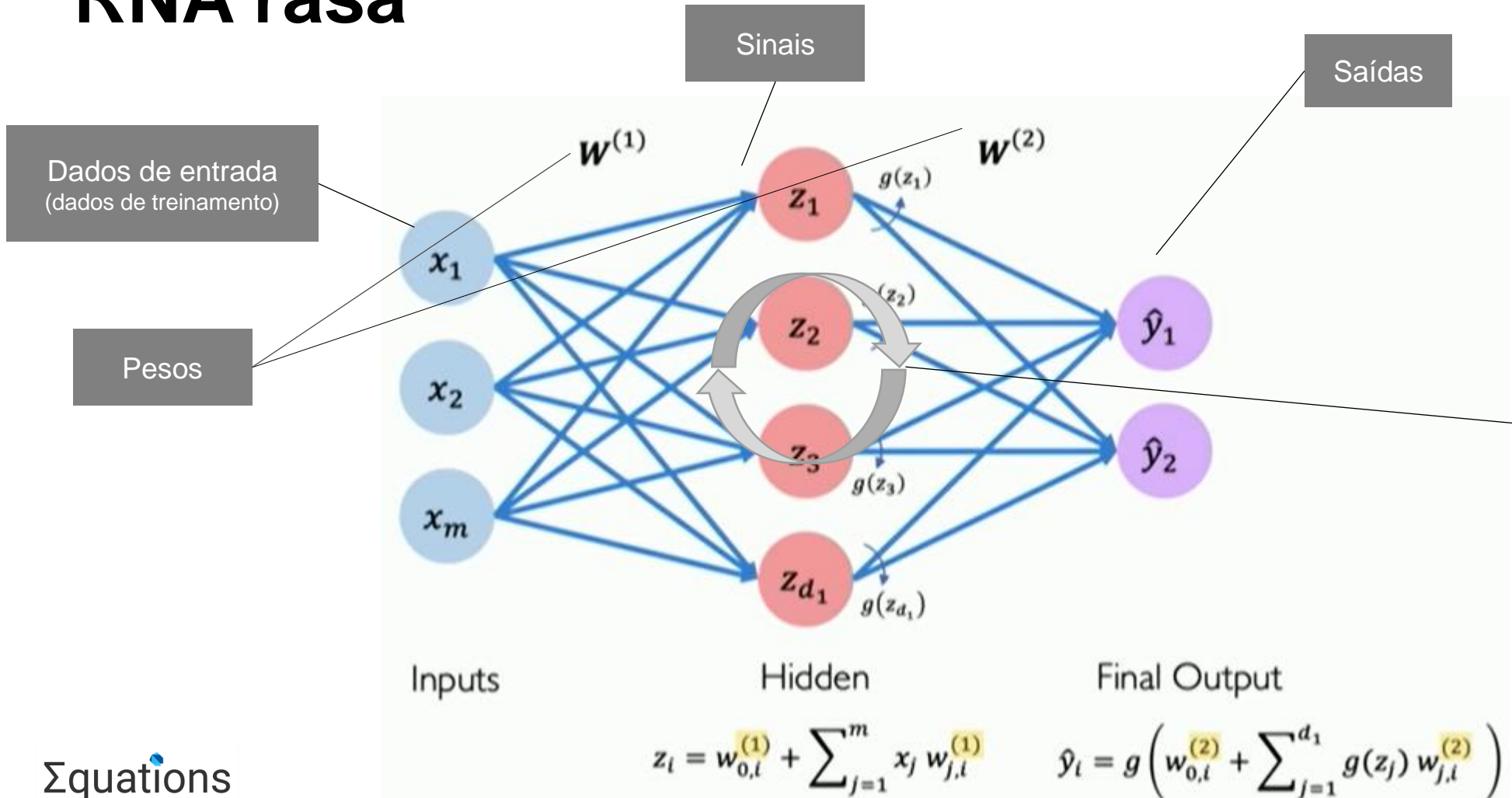
Fonte: MIT Introduction to Deep Learning



Arquiteturas de Redes Neurais



Conceituação teórica e funcionamento de uma RNA rasa

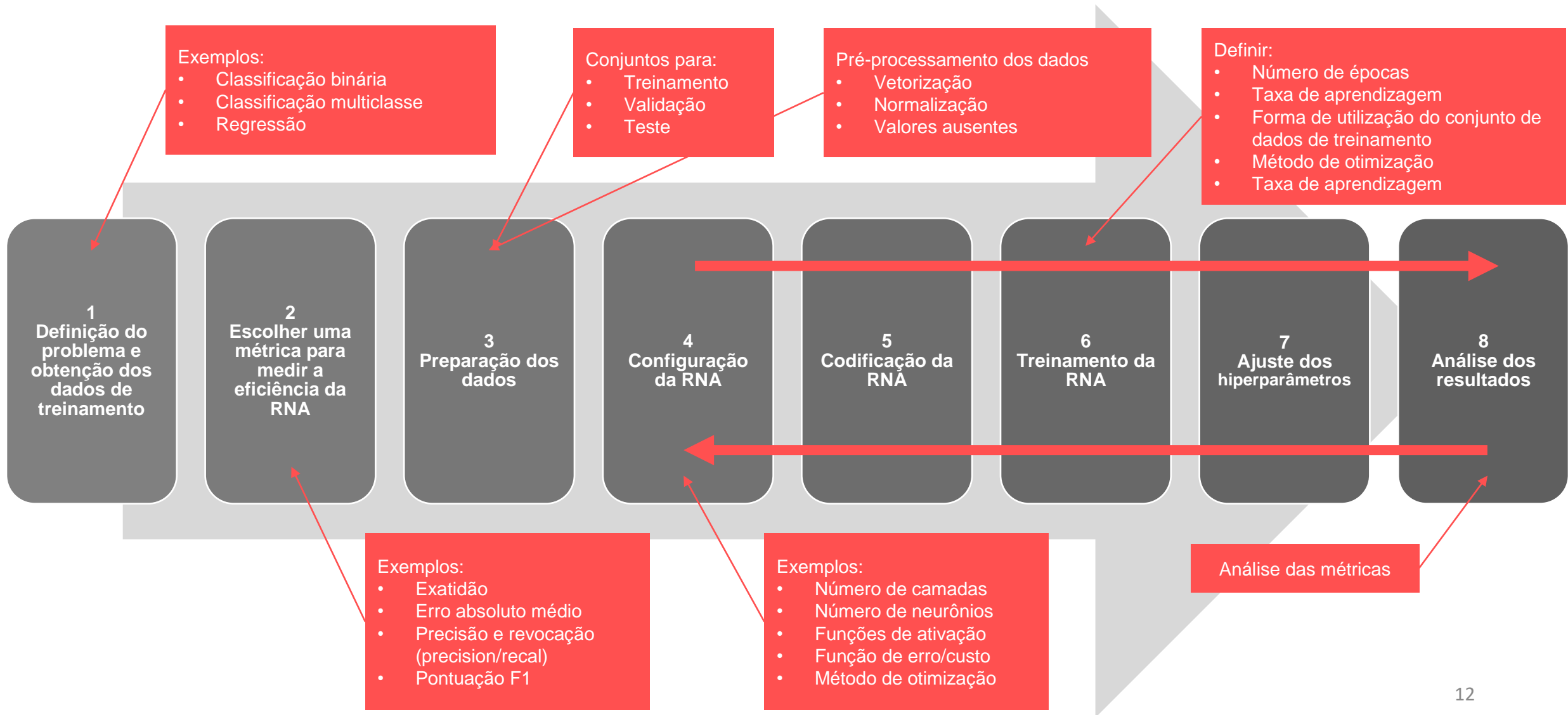


Aprendizado da RNA:

- Os pesos das ligações representam a informação usada para resolver o problema;
- Aprender a solução de um problema significa alterar os pesos das ligações;
- Aprendizado consiste em modificar os pesos das ligações, de forma iterativa, até a RNA ser capaz de reproduzir os dados usados no treinamento.



Processo de desenvolvimento de uma RNA



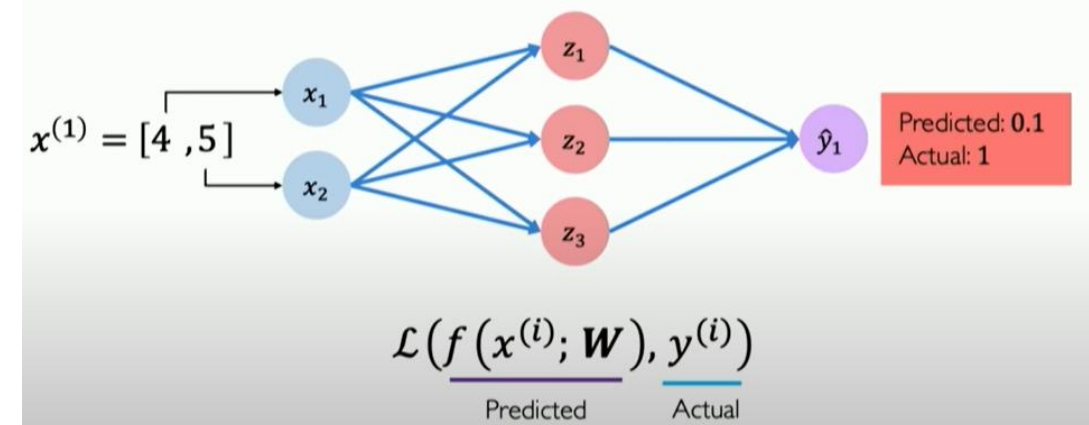
Função de custo (*Loss*)

- A escolha correta da função de custo é extremamente importante, pois o algoritmo de otimização, durante o treinamento, irá minimizar essa função de custo, portanto, se ela não estiver muito bem relacionada como problema o desempenho final da RNA não será bom.
- A função de custo deve ser adequada ao tipo de problema que estamos resolvendo.



Tipo de problema	Função de ativação da última camada	Função de custo
Classificação binária	Sigmoide	Função logística (entropia cruzada)
Classificação multi-classe	Softmax	Entropia cruzada para múltiplas classes
Regressão (ajuste de função) com valores arbitrários	Linear	Erro quadrático médio
Regressão com valores entre 0 e 1	Sigmoide ou ReLu	Erro quadrático médio ou entropia cruzada
Regressão com valores entre -1 e 1	Tangente hiperbólica	Erro quadrático médio

Fonte: CABRAL, E., Notas de Aula – IMT



Fonte: MIT Introduction to Deep Learning



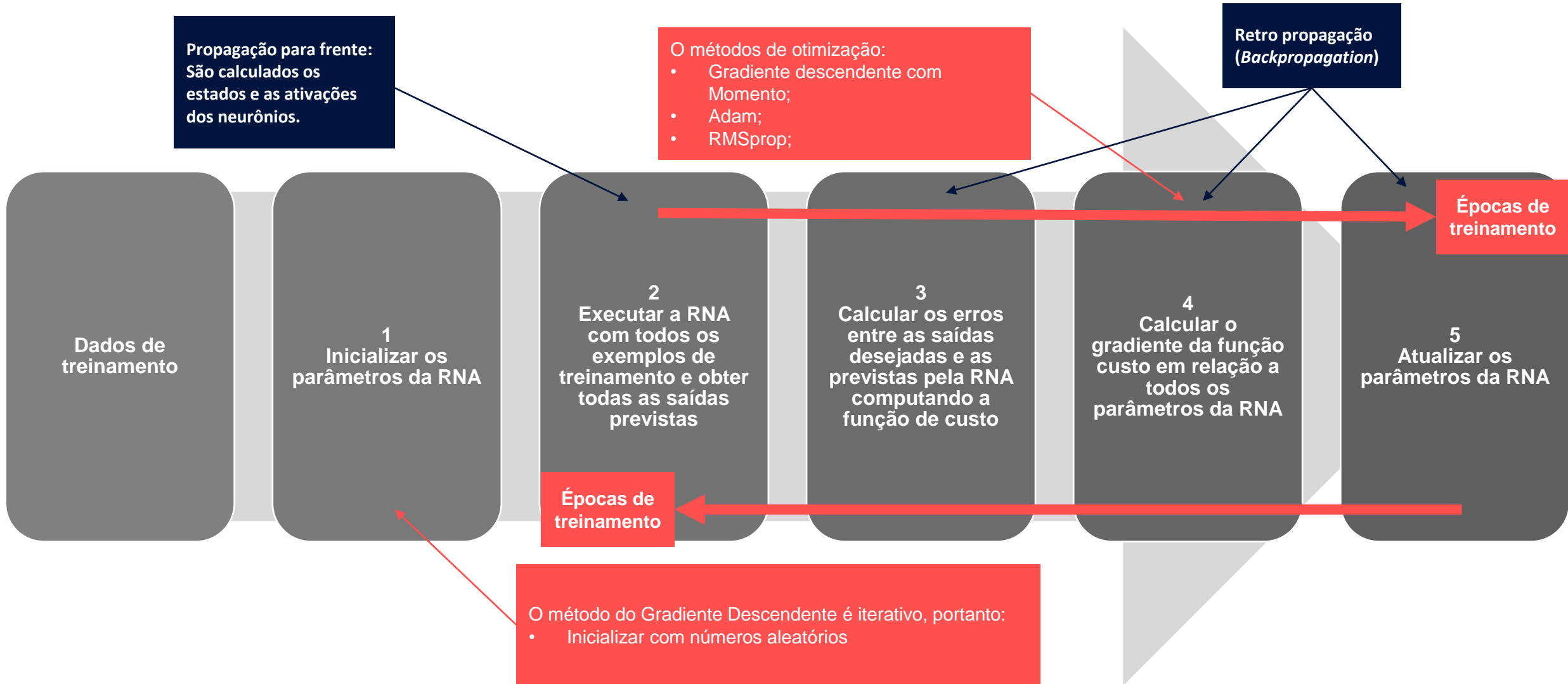
Treinamento de uma RNA

Propagação para frente:
São calculados os
estados e as ativações
dos neurônios.

Os métodos de otimização:

- Gradiente descendente com Momento;
- Adam;
- RMSprop;

Retro propagação
(Backpropagation)





Gradiente Descendente

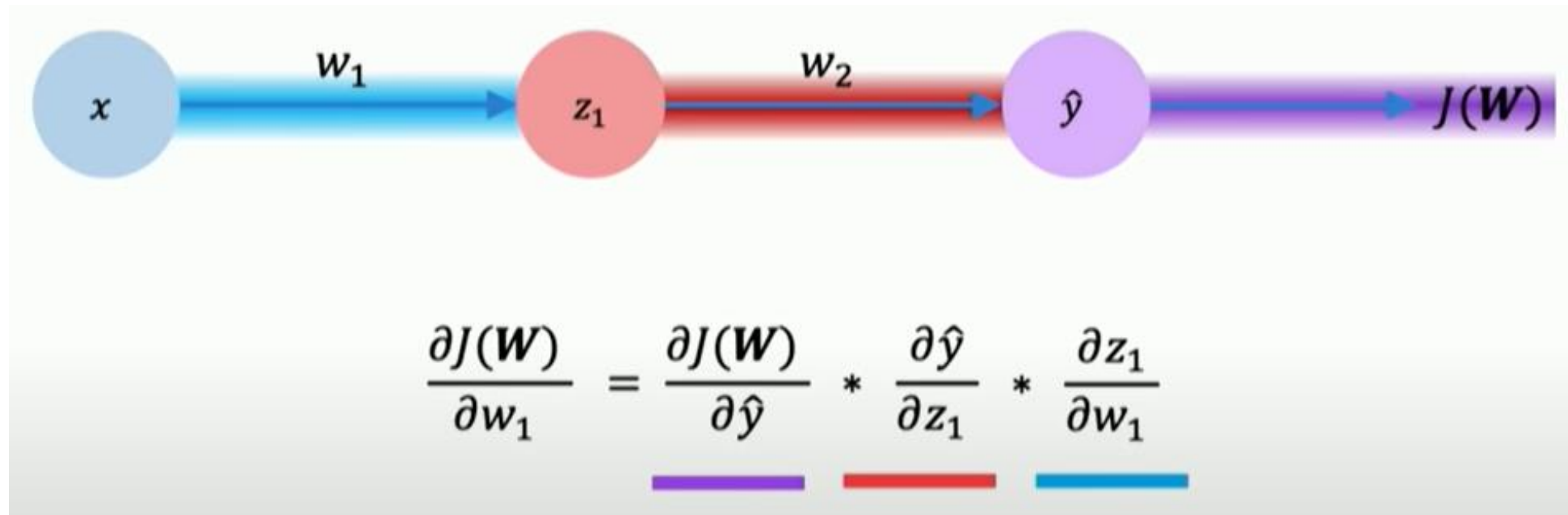
É um algoritmo de otimização usado para encontrar o valor mínimo de uma função. Frequentemente é usado em aprendizado de máquina e inteligência artificial para minimizar funções de erro ou perda, ajustando iterativamente os parâmetros do modelo.

Algoritmo:

1. Inicializar os pesos da rede randomicamente $\sim \mathcal{N}(0, \sigma^2)$;
2. Iterar até a convergência:
 1. Calcular o gradiente, $\frac{\partial J(W)}{\partial W}$
 2. Atualizar os pesos, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
3. Retornar

Gradiente Descendente e *backpropagation*

- Aplicação em um *perceptron* com uma entrada, um neurônio e uma saída:



Fonte: MIT Introduction to Deep Learning

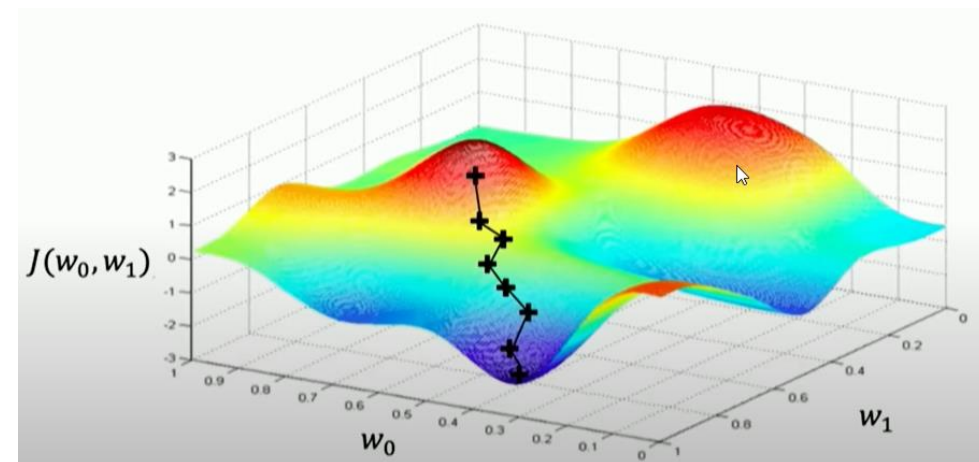
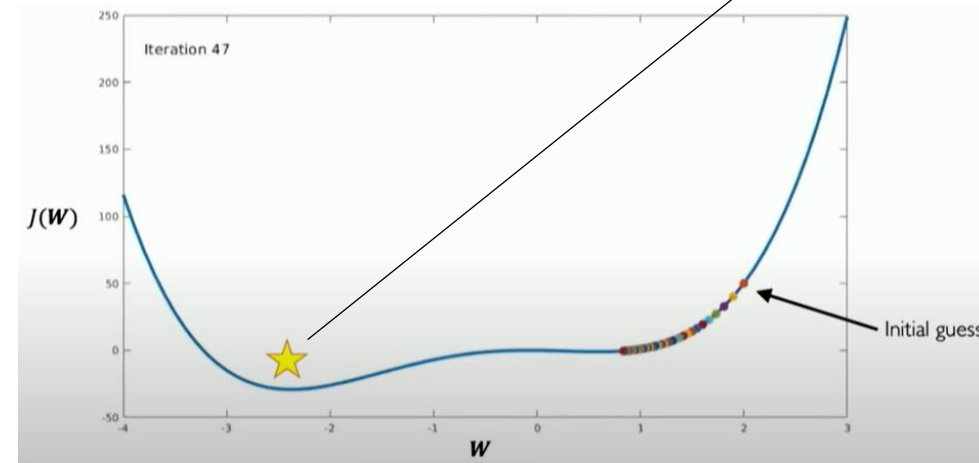
Gradiente Descendente e dificuldades no cálculo

- Otimização pelo Gradiente Descendente:

$$W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$$

Taxa de aprendizagem

Fonte: MIT Introduction to Deep Learning





Caso prático – Classificação multiclasse

- **Objetivo**

- Demonstrar como realizar uma tarefa de classificação multiclasse usando uma RNA.
- Nesse problema queremos determinar o tipo de vestuário que é mostrado em uma imagem.
- O problema consiste em dada uma imagem, a RNA avalia a probabilidade de existirem determinados tipos de vestuário na imagem e determina qual o tipo mais provável entre dez possíveis.
- O objetivo desse problema é treinar uma RNA que recebe como entrada uma imagem e determina qual tipo de vestuário é mostrado na imagem.



Referências bibliográficas

- Neural Network Zoo – The Asimov Institute, disponível em: <https://www.asimovinstitute.org/neural-network-zoo/>, acesso em 24/5/2023;
- MIT – Introduction to Deep Learning, disponível em: <http://introtodeeplearning.com/>, acesso em 24/5/2023;
- GOMES, I., Visão Geral de uma Rede Neural, disponível em: <https://medium.com/neuronio-br/vis%C3%A3o-geral-de-redes-neurais-ee4b882d32af>, acesso em 24/5/2023;
- HAYKIN, S., Redes Neurais – Princípios e Prática, 2ª Edição. Porto Alegre: Bookman, 2007;
- CABRAL, E. L. L., Notas de Aula – Pós Graduação em Ciência de Dados e Inteligência Artificial, Redes Neurais Artificiais, 2023.
- ASSUNÇÃO, R., Fonte: ASSUNÇÃO, R., Notas de Aula – Deep Learning – UFMG, disponível em: <https://homepages.dcc.ufmg.br/~assuncao/AAP/>, Acesso em 24/05/2023.

Material Extra



Equações da RNA (Slide 11)

Equações da RNA:

- Dimensões:
 - Número de entradas $\Rightarrow n_x = 3$
 - Número de neurônios das camadas:
 - Primeira camada $\Rightarrow n^{[1]} = 4$
 - Segunda camada $\Rightarrow n^{[2]} = 2$
 - Número de estados = número de níveis de ativação = número de vieses = número de neurônios da camada
 - Matriz de pesos:
 - Primeira camada \Rightarrow dimensão $(n^{[1]}, n_x) = (4, 3)$
 - Segunda camada \Rightarrow dimensão $(n^{[2]}, n^{[1]}) = (2, 4)$
 - Número de saídas $\Rightarrow n_y =$ número de neurônios da 2ª camada ($n^{[2]} = 2$)
- Estados dos neurônios da primeira camada ($\mathbf{z}^{[1]}$):

$$\begin{cases} z_1^{[1]} = w_{1,1}^{[1]}x_1 + w_{1,2}^{[1]}x_2 + w_{1,3}^{[1]}x_3 + b_1^{[1]} \\ z_2^{[1]} = w_{2,1}^{[1]}x_1 + w_{2,2}^{[1]}x_2 + w_{2,3}^{[1]}x_3 + b_2^{[1]} \\ z_3^{[1]} = w_{3,1}^{[1]}x_1 + w_{3,2}^{[1]}x_2 + w_{3,3}^{[1]}x_3 + b_3^{[1]} \\ z_4^{[1]} = w_{4,1}^{[1]}x_1 + w_{4,2}^{[1]}x_2 + w_{4,3}^{[1]}x_3 + b_4^{[1]} \end{cases} \Rightarrow \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & w_{1,3}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} & w_{2,3}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} & w_{3,3}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} & w_{4,3}^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}$$

ou simplesmente:

$$\mathbf{z}^{[1]} = \mathbf{W}^{[1]}\mathbf{x} + \mathbf{b}^{[1]}$$

onde:

$w_{k,j}^{[1]}$ = peso da ligação do neurônio k da primeira camada com a entrada j

$b_k^{[1]}$ = viés do neurônio k da primeira camada

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{(n_x,1)} = \text{vetor de entradas}$$

$$\mathbf{z}^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix}_{(n^{[1]},1)} = \text{vetor de estados dos neurônios da primeira camada}$$

$$\mathbf{W}^{[1]} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & w_{1,3}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} & w_{2,3}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} & w_{3,3}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} & w_{4,3}^{[1]} \end{bmatrix}_{(n^{[1]},n_x)} = \text{matriz de pesos da primeira camada}$$

$$\mathbf{b}^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}_{(n^{[1]},1)} = \text{vetor de vieses dos neurônios da primeira camada}$$



Equações da RNA (Slide 11)

- Nível de ativação dos neurônios da primeira camada ($\mathbf{a}^{[1]}$):

$$\begin{cases} a_1^{[1]} = g^{[1]}(z_1^{[1]}) \\ a_2^{[1]} = g^{[1]}(z_2^{[1]}) \\ a_3^{[1]} = g^{[1]}(z_3^{[1]}) \\ a_4^{[1]} = g^{[1]}(z_4^{[1]}) \end{cases} \Rightarrow \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = g^{[1]} \left(\begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} \right) \Rightarrow \boxed{\mathbf{a}^{[1]} = g^{[1]}(\mathbf{z}^{[1]})}$$

onde:

$g^{[1]}$ = função de ativação da primeira camada

$$\mathbf{a}^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix}_{(n^{[1]},1)} = \text{vetor de ativação dos neurônios da primeira camada}$$

- Estados dos neurônios da segunda camada ($\mathbf{z}^{[2]}$):

$$\begin{cases} z_1^{[2]} = w_{1,1}^{[2]}a_1^{[1]} + w_{1,2}^{[2]}a_2^{[1]} + w_{1,3}^{[2]}a_3^{[1]} + w_{1,4}^{[2]}a_4^{[1]} + b_1^{[2]} \\ z_2^{[2]} = w_{2,1}^{[2]}a_1^{[1]} + w_{2,2}^{[2]}a_2^{[1]} + w_{2,3}^{[2]}a_3^{[1]} + w_{2,4}^{[2]}a_4^{[1]} + b_2^{[2]} \end{cases}$$

Matricialmente,

$$\begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} \\ w_{2,1}^{[2]} & w_{2,2}^{[2]} & w_{2,3}^{[2]} & w_{2,4}^{[2]} \end{bmatrix} \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} + \begin{bmatrix} b_1^{[2]} \\ b_2^{[2]} \end{bmatrix} \Rightarrow \boxed{\mathbf{z}^{[2]} = \mathbf{W}^{[2]}\mathbf{a}^{[1]} + \mathbf{b}^{[2]}}$$

onde:

$w_{k,j}^{[2]}$ = peso da ligação do neurônio k da camada 2 com o neurônio j da primeira camada

$b_k^{[2]}$ = viés do neurônio k da segunda camada

$$\mathbf{z}^{[2]} = \begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \end{bmatrix}_{(n^{[2]},1)} = \text{vetor de estados dos neurônios da segunda camada}$$

$$\mathbf{W}^{[2]} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} \\ w_{2,1}^{[2]} & w_{2,2}^{[2]} & w_{2,3}^{[2]} & w_{2,4}^{[2]} \end{bmatrix}_{(n^{[2]},n^{[1]})} = \text{matriz de pesos da segunda camada}$$

$$\mathbf{b}^{[2]} = \begin{bmatrix} b_1^{[2]} \\ b_2^{[2]} \end{bmatrix}_{(n^{[2]},1)} = \text{vetor de vieses dos neurônios da segunda camada}$$

- Nível de ativação dos neurônios da segunda camada ($\mathbf{a}^{[2]}$):

$$\begin{cases} a_1^{[2]} = g^{[2]}(z_1^{[2]}) \\ a_2^{[2]} = g^{[2]}(z_2^{[2]}) \end{cases} \Rightarrow \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \end{bmatrix} = g^{[2]} \left(\begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \end{bmatrix} \right) \Rightarrow \boxed{\mathbf{a}^{[2]} = g^{[2]}(\mathbf{z}^{[2]})}$$

onde:

$g^{[2]}$ = função de ativação da segunda camada

$$\mathbf{a}^{[2]} = \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \end{bmatrix}_{(n^{[2]},1)} = \text{vetor de ativação dos neurônios da segunda camada}$$

⇒ Observe que as funções de ativações das diversas camadas em geral são diferentes.



Equações da RNA (Slide 11)

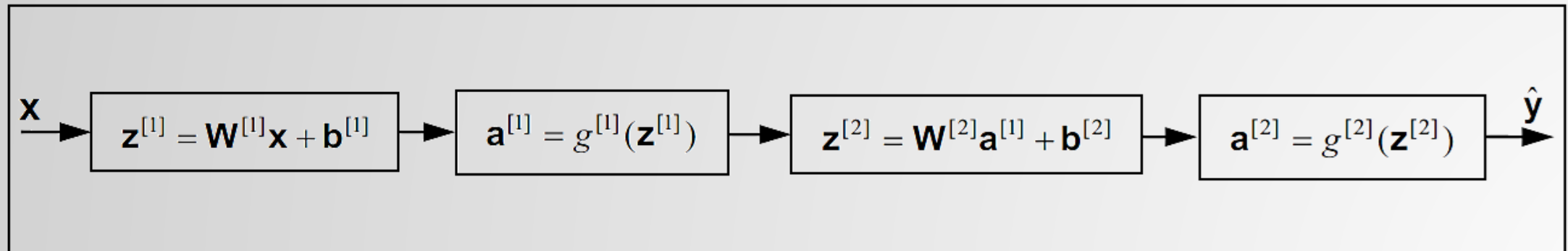
- Saídas da rede ($\hat{\mathbf{y}}$):

$$\begin{cases} \hat{y}_1 = a_1^{[2]} \\ \hat{y}_2 = a_2^{[2]} \end{cases} \Rightarrow \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \end{bmatrix} \Rightarrow \boxed{\hat{\mathbf{y}} = \mathbf{a}^{[2]}}$$

onde:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}_{(n^{[2]}, 1)} = \text{vetor de saída da RN}$$

Fluxo de cálculo da RNA
rasa de uma camada
intermediária

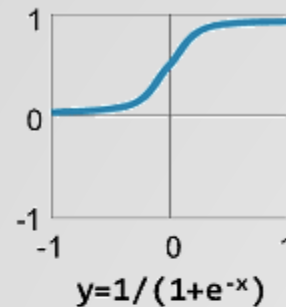




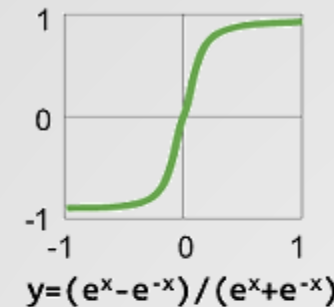
Funções de ativação e custo

Traditional
Non-Linear
Activation
Functions

Sigmoid

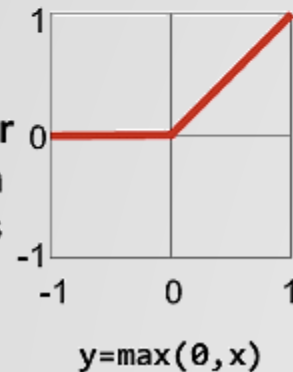


Hyperbolic Tangent

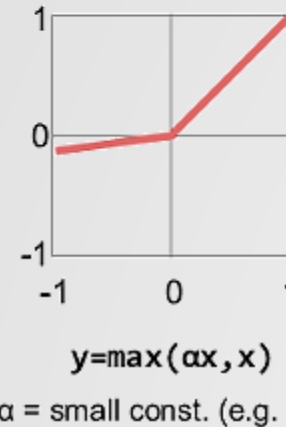


Modern
Non-Linear
Activation
Functions

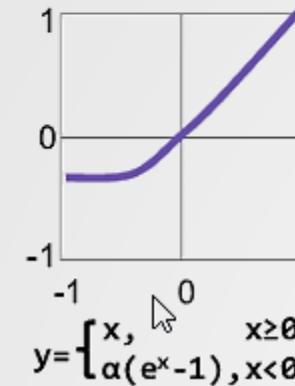
Rectified Linear Unit
(ReLU)



Leaky ReLU



Exponential LU



Fonte: ASSUNÇÃO, R., Notas de Aula – Deep Learning – UFMG

