



Universidade Federal do Piauí
Laboratório de Inteligência Artificial - LINA

Introdução à Deep Learning

Bruno Vicente Alves de Lima

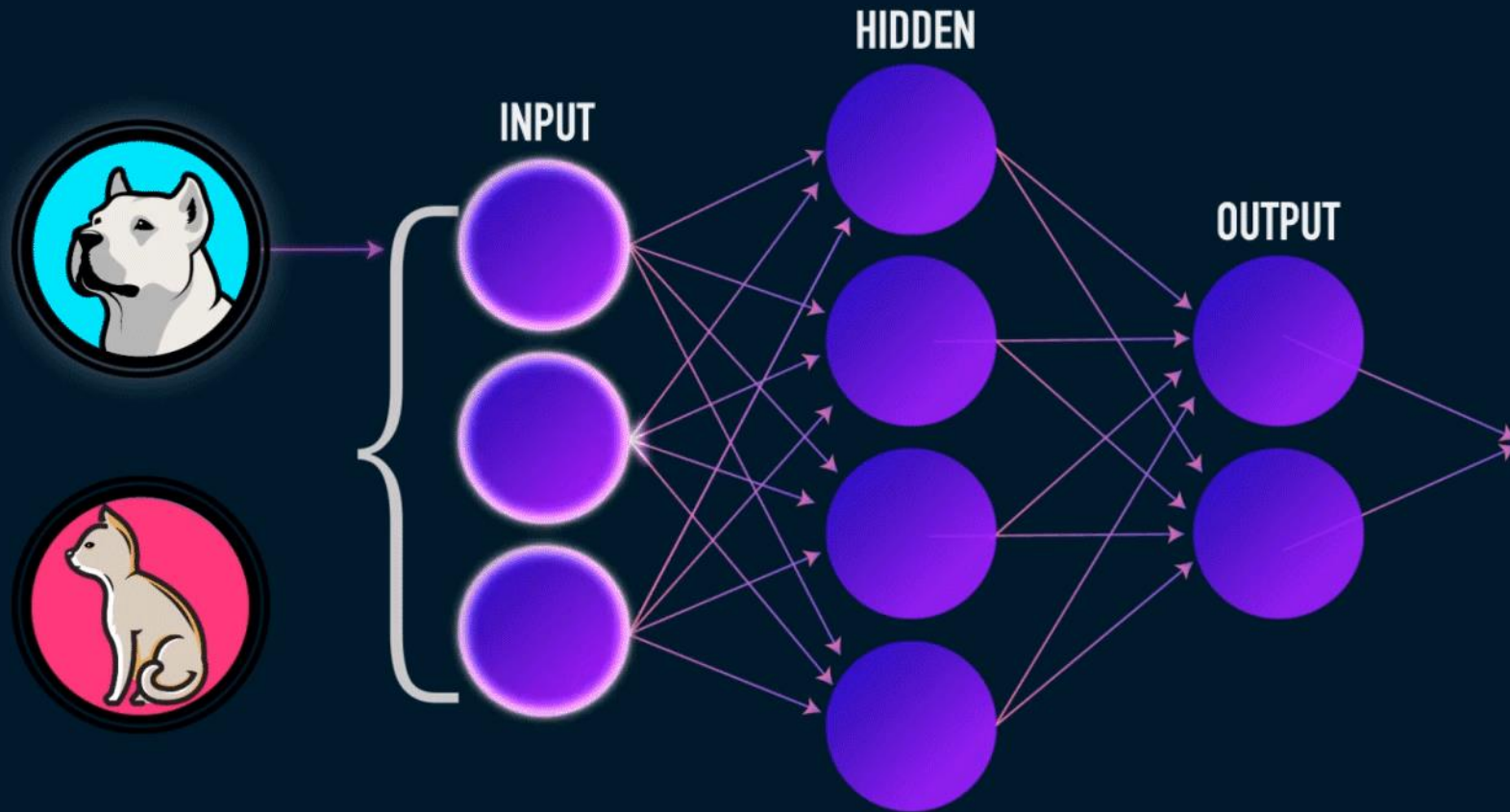
🌐 GitHub:

https://github.com/brunnovicente/curso_deeplearning

🌐 Bases de Dados

- Lua
- Círculos
- Iris
- Sementes
- Vinhos
- MNIST -> https://drive.google.com/file/d/1-50dESSsuW9jYGQes6YiU4PtPRGr_V2a/view?usp=sharing

Rede Perceptron Multicamadas



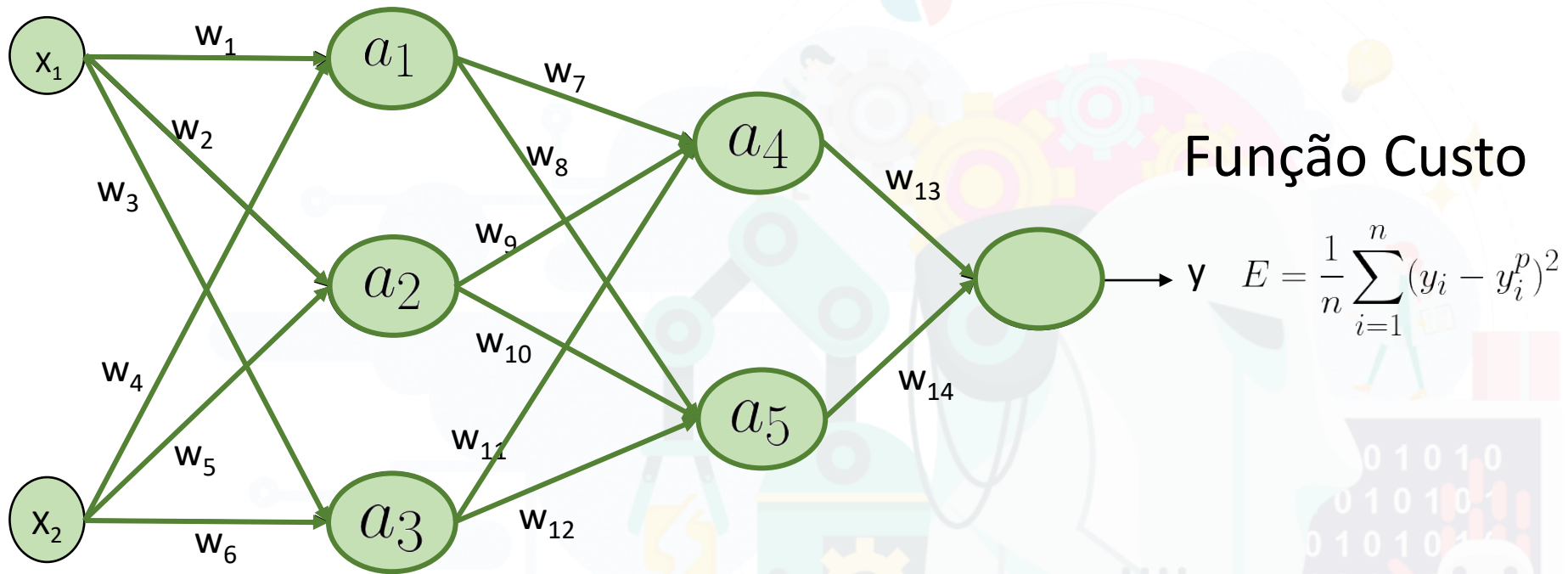
Aprendizado pelo Gradiente

- Processo de otimização utilizado no Backpropagation para encontrar o melhor conjunto de pesos;
- É especificada uma função custo que deve ser otimizada. Esta função avalia a qualidade da rede durante o treinamento

● Ex.:

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2$$

Aprendizado pelo Gradiente



Aprendizado pelo Gradiente

● Gradiente: $\nabla E[\vec{w}] = \left\{ \frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right\}$

● Regra do treinamento para descida do gradiente:

$$w_i = w_i + \Delta w_i$$

Onde:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Aprendizado pelo Gradiente

Aplicando a regra da cadeia na função custo.

Derivada da Função de Ativação

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial E}{\partial (w.x)} \cdot \frac{\partial (w.x)}{\partial w_i}$$

Derivada da função custo

Entrada

Substituindo

$$\frac{\partial E}{\partial w_i} = -(y_d - y_p) \cdot f'(w.x) \cdot x_i$$

Aprendizado pelo Gradiente

$$\frac{\partial E}{\partial w_i} = -(y_d - y_p) \cdot f'(w \cdot x) \cdot x_i$$

Considerando:

$$\delta = -(y_d - y_p) \cdot f'(w \cdot x)$$

Substituindo:

$$\frac{\partial E}{\partial w_i} = \delta \cdot x_i$$

Aprendizado pelo Gradiente

Considerando:

$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Substituindo:

$$w_i = w_i + \eta \cdot \delta \cdot x_i$$

Descida do Gradiente

- A descida do gradiente determina um vetor de pesos minimizando a função Custo diferenciável, começando com um vetor inicial de pesos arbitrário e modificando-o repetidamente em pequenos passos.
- A cada passo, o vetor de pesos é alterado na direção que produz a maior queda ao longo da superfície de erro.
- Este processo continua até atingir um erro mínimo global.

Descida do Gradiente



Descida do Gradiente

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

Pesos Iniciais

$$\begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \\ \vdots \\ w_{n1} \end{bmatrix}$$

$$\begin{bmatrix} w_{12} \\ w_{22} \\ w_{32} \\ \vdots \\ w_{n2} \end{bmatrix}$$

$$\begin{bmatrix} w_{13} \\ w_{23} \\ w_{33} \\ \vdots \\ w_{n3} \end{bmatrix}$$

Pesos Ótimos

Gradiente

Descida do Gradiente

Funções Custo

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^p)^2$$

Erro Médio Quadrático

$$f(x) = - \sum_{i=1}^n p(x_i) \log q(x_i)$$

Entropia Cruzada

$$f(x) = \frac{1}{n} \sum_{i=1}^n |y_i - y_i'|$$

Erro Médio Absoluto

Descida do Gradiente

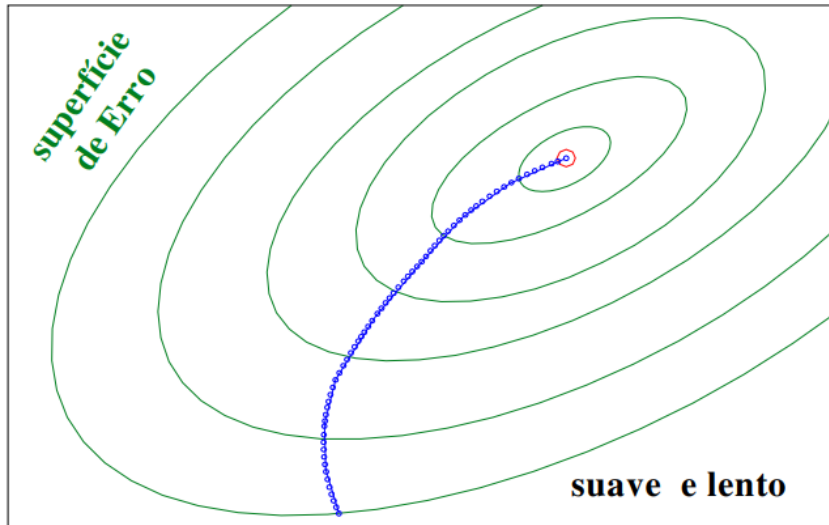
🌐 Momentum

- Permite alterar a taxa de aprendizado durante a execução do backpropagation;
- Leva em consideração mudanças anteriores dos pesos para definir a próxima alteração

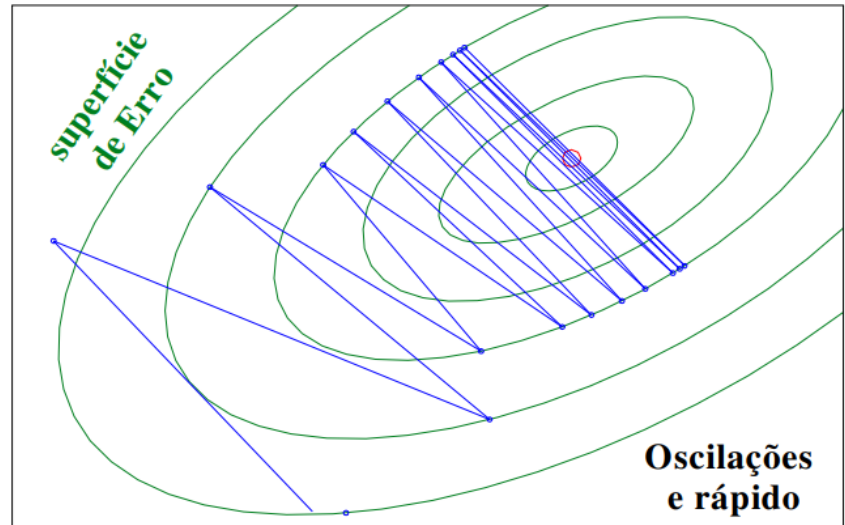
$$w_i = w_i \alpha + \Delta w_i$$

Descida do Gradiente

● Momentum



(a) η – Pequeno

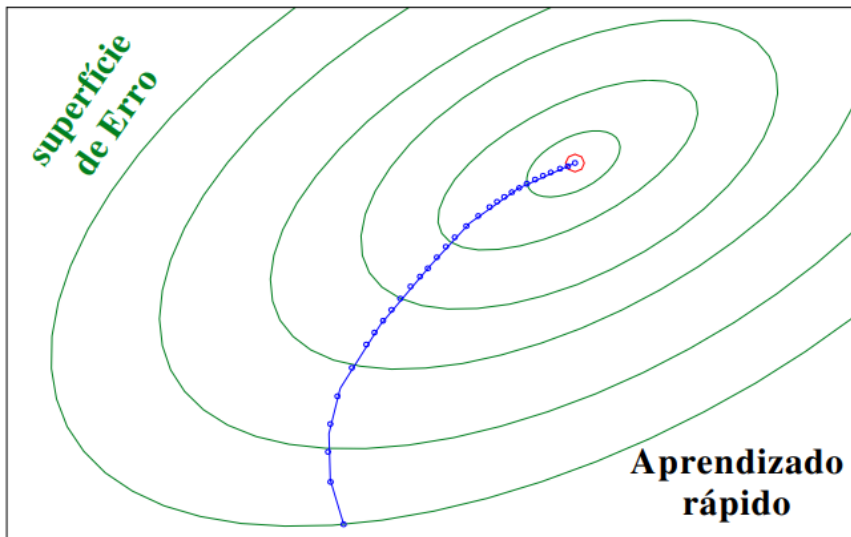


(b) η – Grande

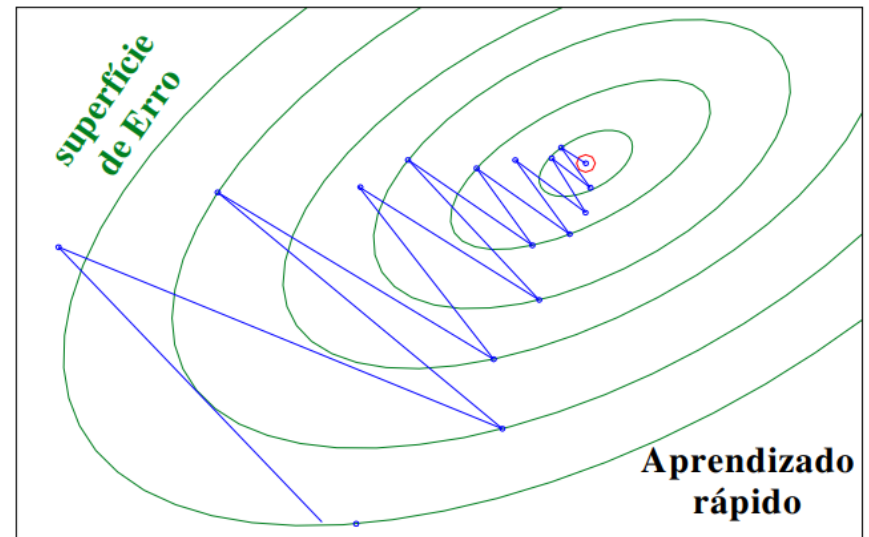
Aprendizado sem Termo Momento

Descida do Gradiente

Momentum



(a) η – Pequeno



(b) η – Grande

Aprendizado com Termo Momento

Gradiente Descendente Estocástico (SGD)

- Uma melhoria no Gradiente Descendente;
- Realiza o gradiente descendente em **lotes** na base de dados;
- Pode reduzir o tempo de treinamento;

Gradiente Descendente Estocástico (SGD)

Gradiente Descendente usa toda a base.

	A	B	C	D	E
1	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
2	5.1	3.5	1.4	0.2	Iris-setosa
3	4.9	3	1.4	0.2	Iris-setosa
4	4.7	3.2	1.3	0.2	Iris-setosa
5	4.6	3.1	1.5	0.2	Iris-setosa
6	5	3.6	1.4	0.2	Iris-setosa
7	5.4	3.9	1.7	0.4	Iris-setosa
8	4.6	3.4	1.4	0.3	Iris-setosa
9	5	3.4	1.5	0.2	Iris-setosa
10	4.4	2.9	1.4	0.2	Iris-setosa
11	4.9	3.1	1.5	0.1	Iris-setosa
12	5.4	3.7	1.5	0.2	Iris-setosa
13	4.8	3.4	1.6	0.2	Iris-setosa
14	4.8	3	1.4	0.1	Iris-setosa
15	4.3	3	1.1	0.1	Iris-setosa
16	5.8	4	1.2	0.2	Iris-setosa
17	5.7	4.4	1.5	0.4	Iris-setosa
18	5.4	3.9	1.3	0.4	Iris-setosa
19	5.1	3.5	1.4	0.3	Iris-setosa
20	5.7	3.8	1.7	0.3	Iris-setosa
21	5.1	3.8	1.5	0.3	Iris-setosa
22	5.4	3.4	1.7	0.2	Iris-setosa
23	5.1	3.7	1.5	0.4	Iris-setosa
24	4.6	3.6	1	0.2	Iris-setosa
25	5.1	3.3	1.7	0.5	Iris-setosa

Lote 1

Lote 2

Lote 3

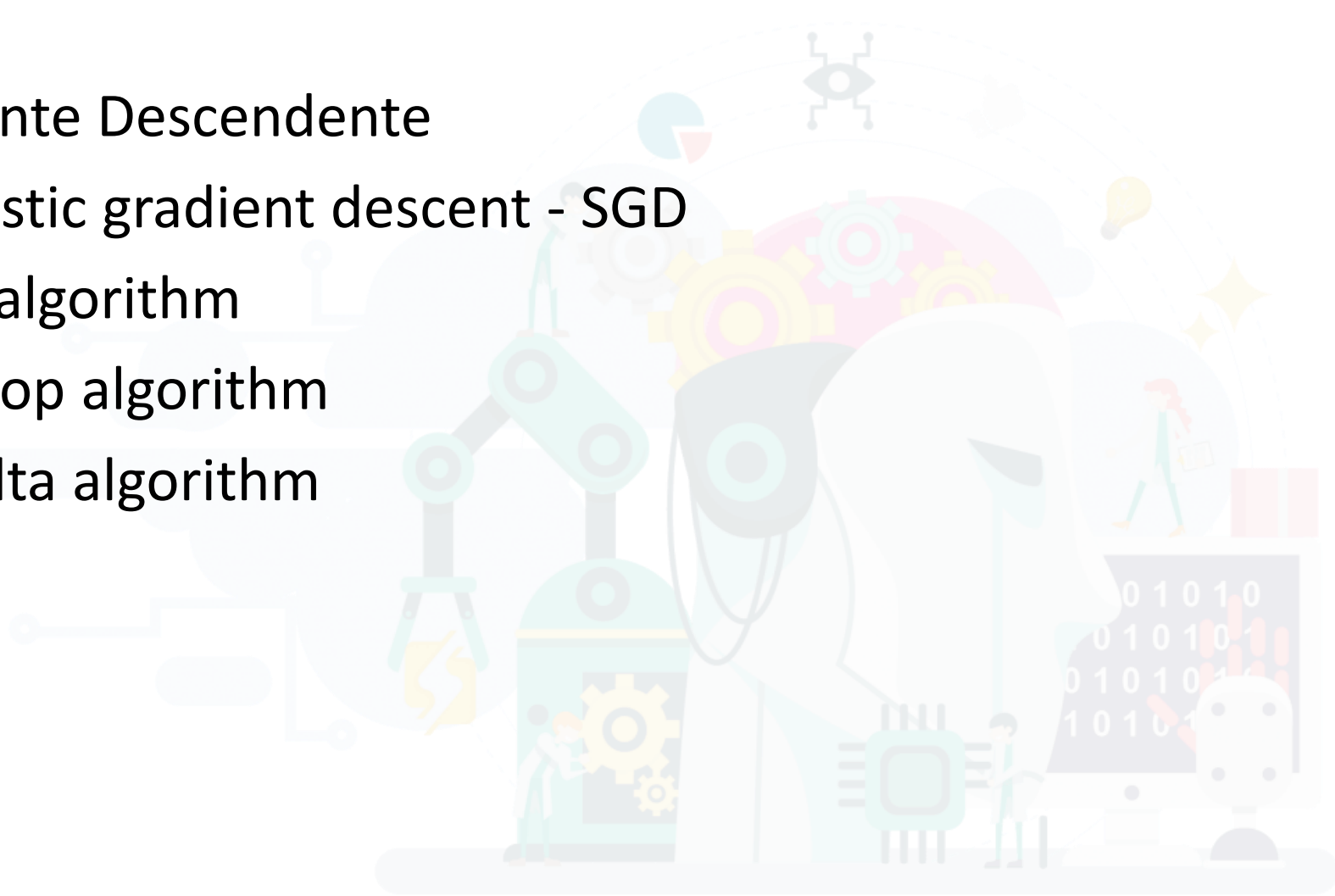
Lote 4

Lote 5

Gradiente Descendente Estocástico Divide a base em lotes

Métodos de Otimização

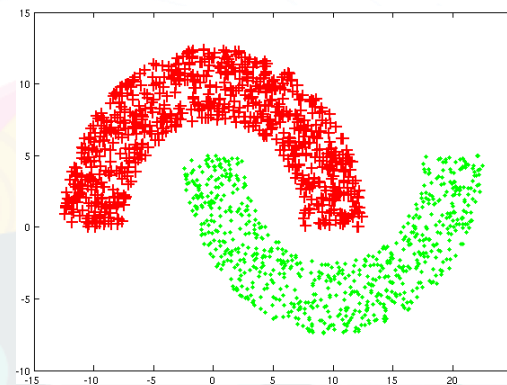
- Gradiente Descendente
- Stochastic gradient descent - SGD
- Adam algorithm
- RMSprop algorithm
- Adadelta algorithm



Exemplo Keras

● Classificação Binária

- Base Lua
- Base Câncer

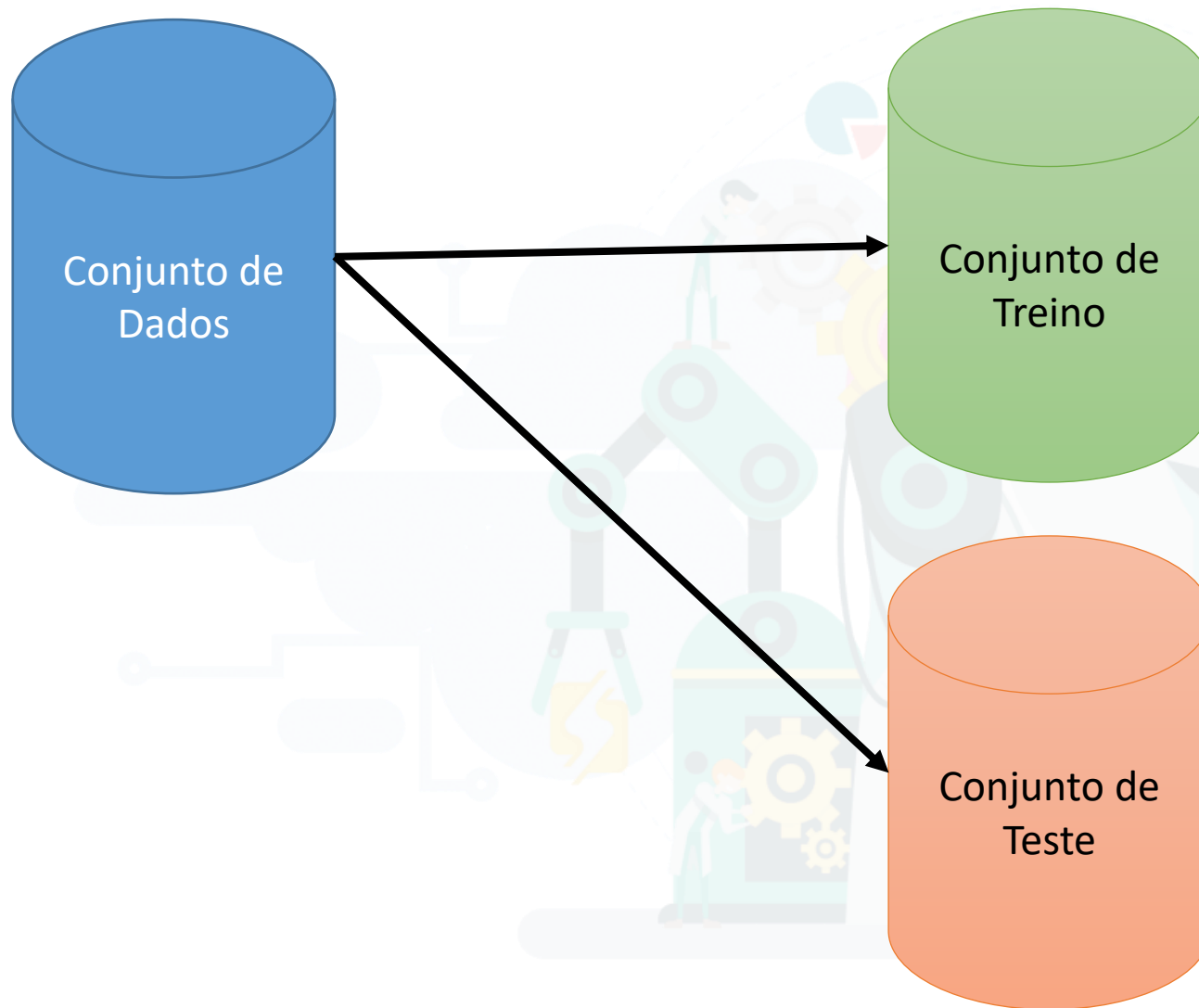


● Classificação Multiclasse

- Iris
- Vinhos
- MNIST64

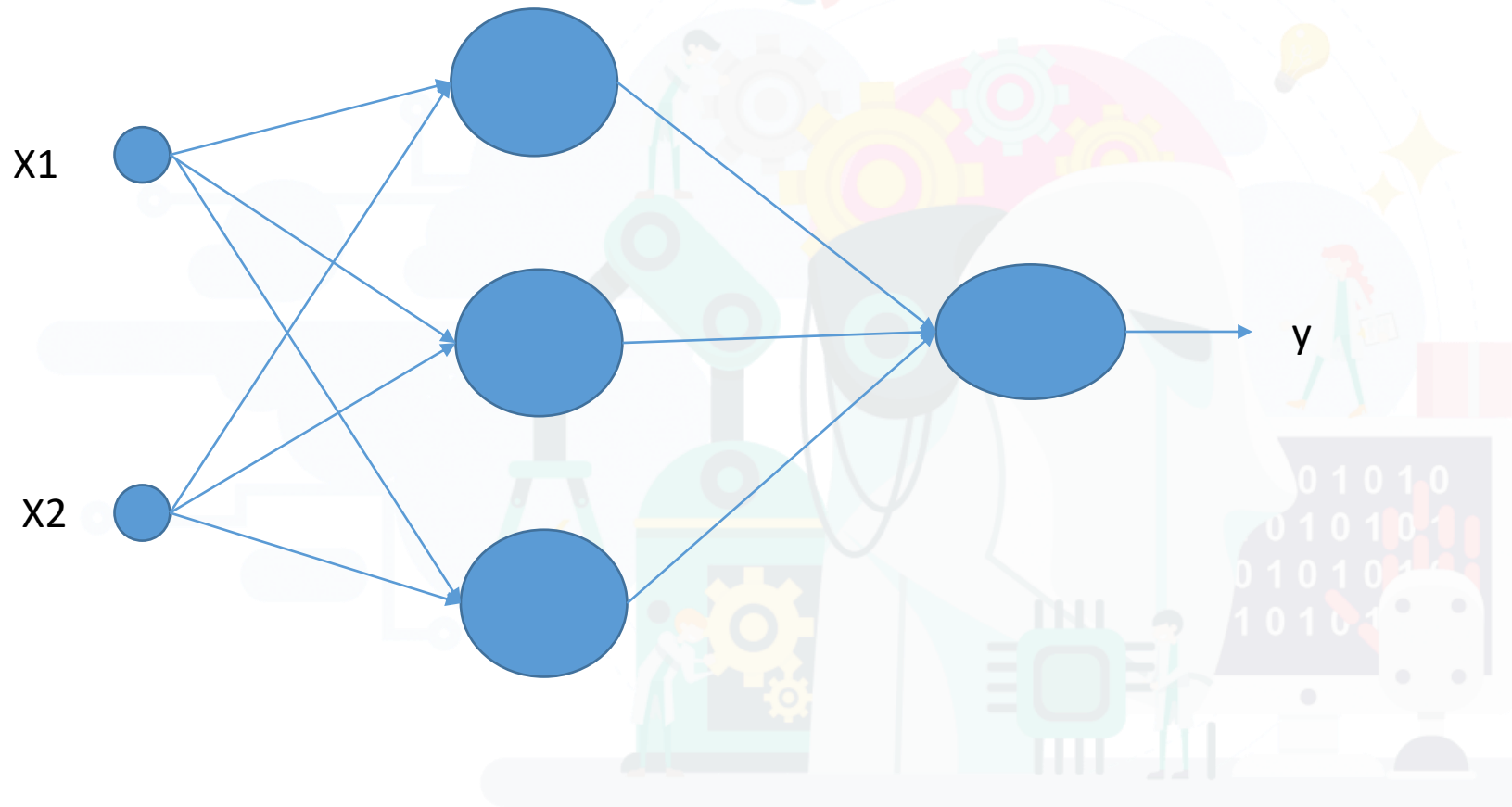


Exemplo Keras



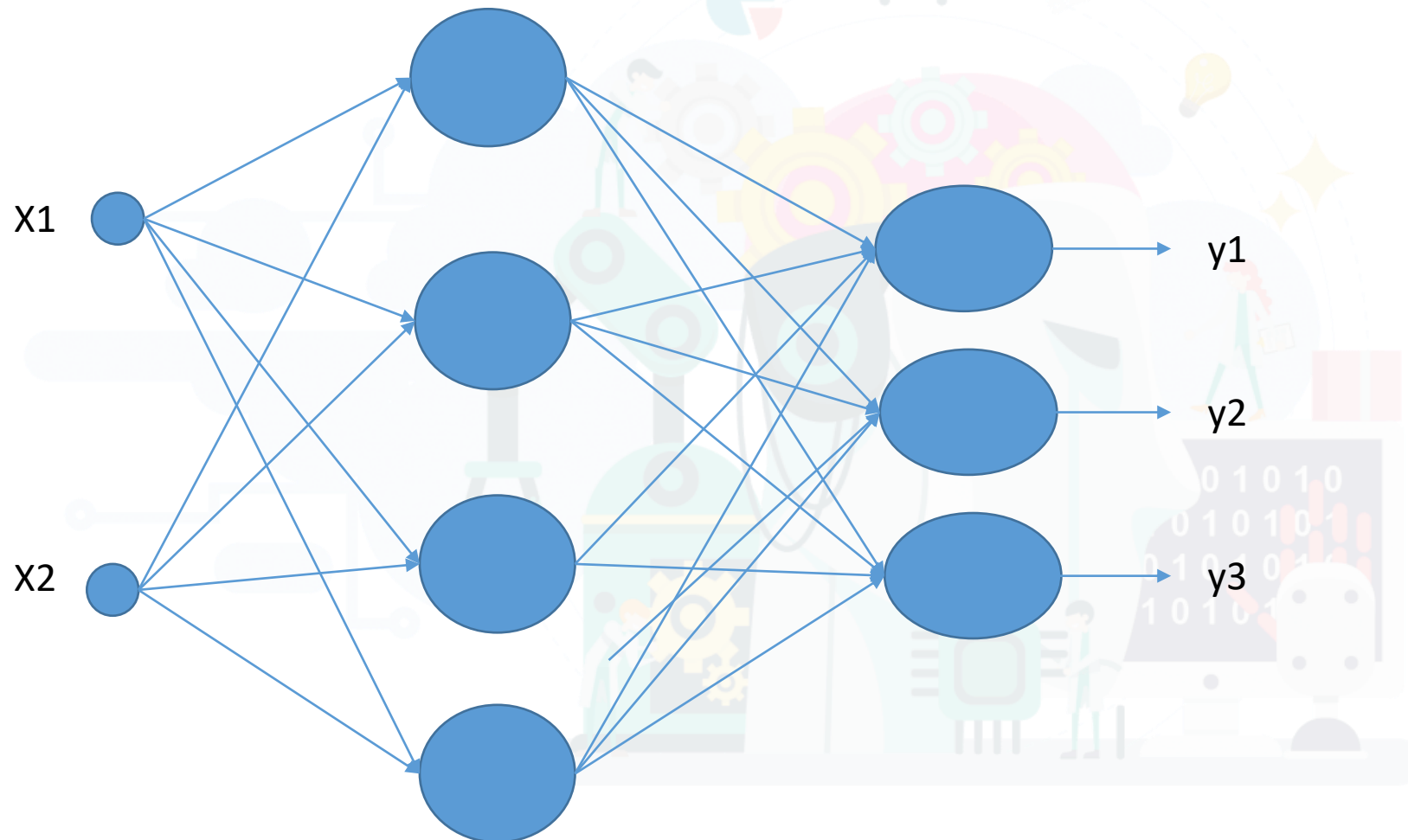
Exemplo Keras

Classificação Binária



Exemplo Keras

Classificação Multiclasse





Universidade Federal do Piauí
Laboratório de Inteligência Artificial - LINA

Introdução à Deep Learning

Bruno Vicente Alves de Lima