

Inteligência Artificial Aplicada à Física

Fundamentos e Aplicações com Redes Neurais

AULA 2

Alexandre Suaide

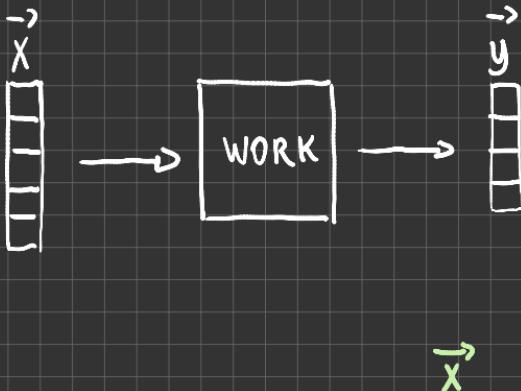
Escola Jayme Tiomno
IFUSP 2025

Objetivos das aulas de hoje

- Entender a estrutura de uma rede neural
 - camadas, função de ativação, etc
- Treinamento de uma rede
- Métricas de desempenho
- Exemplo prático com física
 - ↳ classificação de estrelas

O problema básico de classificação

Dado un vetor de entradas \vec{x} , determinar uns saídas \vec{y}



0	2	15	0	0	11	10	0	0	0	0	0	9	9	0	0
0	0	0	4	0	0	11	236	255	255	277	95	81	32	0	0
0	10	16	0	238	255	244	45	243	250	249	255	222	0	10	0
0	14	18	255	255	244	47	255	253	245	255	249	253	251	124	0
2	255	228	255	251	254	251	111	111	111	111	215	251	258	255	49
12	217	243	255	255	255	255	255	255	255	255	255	255	255	255	36
12	229	255	254	254	254	254	254	254	254	254	254	254	254	254	62
145	245	255	252	252	252	252	252	252	252	252	252	252	252	252	37
0	133	255	255	245	255	255	255	255	255	255	255	255	255	255	137
0	252	250	248	251	251	251	251	251	251	251	251	251	251	251	6
0	133	255	255	245	255	255	255	255	255	255	255	255	255	255	139
1	0	1	1	1	251	255	241	255	247	255	241	242	17	0	7
0	0	0	4	0	251	255	246	254	253	253	253	253	253	253	1
0	0	0	4	1	255	255	255	248	252	252	255	244	255	239	10
0	22	208	252	246	251	241	241	241	241	241	245	255	234	9	0
0	133	255	242	253	253	253	253	253	253	253	253	253	253	253	56
0	253	251	250	251	251	251	251	251	251	251	251	251	251	251	3
0	177	255	251	251	251	251	251	251	251	251	251	251	251	251	125
0	177	255	251	251	251	251	251	251	251	251	251	251	251	251	61
0	177	251	251	251	251	251	251	251	251	251	251	251	251	251	52
0	18	146	250	255	247	255	255	255	249	255	249	240	255	255	129
0	0	23	113	215	255	255	255	255	255	255	255	255	248	248	14
0	0	6	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	5	9	0	0	0	0	0	0	0	0	0	0	0	0

0	2	15	0	0	11	10	0	0	0	0	0	9	9	0	0
0	0	0	4	0	0	11	236	255	255	277	95	81	32	0	0
0	10	16	0	238	255	244	45	243	250	249	255	222	0	10	0
0	14	18	255	255	244	47	255	253	245	255	249	253	251	124	0
2	255	228	255	251	251	251	251	251	251	251	251	251	251	251	49
12	217	243	255	255	255	255	255	255	255	255	255	255	255	255	36
12	229	255	254	254	254	254	254	254	254	254	254	254	254	254	62
145	245	255	252	252	252	252	252	252	252	252	252	252	252	252	37
0	133	255	255	245	255	255	255	255	255	255	255	255	255	255	137
0	252	250	248	251	251	251	251	251	251	251	251	251	251	251	6
0	133	255	255	245	255	255	255	255	255	255	255	255	255	255	139
1	0	1	1	1	251	255	241	255	247	255	241	242	17	0	7
0	0	0	4	0	251	255	246	254	253	253	253	253	253	253	1
0	0	0	4	1	255	255	255	248	252	252	255	244	255	239	10
0	22	208	252	246	251	241	241	241	241	241	245	255	234	9	0
0	133	255	242	253	253	253	253	253	253	253	253	253	253	253	56
0	253	251	250	251	251	251	251	251	251	251	251	251	251	251	3
0	177	255	251	251	251	251	251	251	251	251	251	251	251	251	125
0	177	255	251	251	251	251	251	251	251	251	251	251	251	251	61
0	177	251	251	251	251	251	251	251	251	251	251	251	251	251	52
0	18	146	250	255	247	255	255	255	249	255	249	240	255	255	129
0	0	23	113	215	255	255	255	255	255	255	255	255	248	248	14
0	0	6	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	5	9	0	0	0	0	0	0	0	0	0	0	0	0



→ posição que identifica
8 no vetor de saídas

→ posição que identifica
8 no vetor de saídas

→ \vec{x} pode ser uma imagem,
um texto, um conjunto
de dados

→ \vec{y} pode ser um conjunto
de rótulos, outra imagem,
etc

A estrutura básica de uma rede neural artificial

- Neurônios artificiais

- Um pedaço de memória (número)
- Recebe sinais a partir de diferentes conexões, com intensidades diferentes
- Produz um resultado que pode ser enviado a outros neurônios

- A rede de neurônios

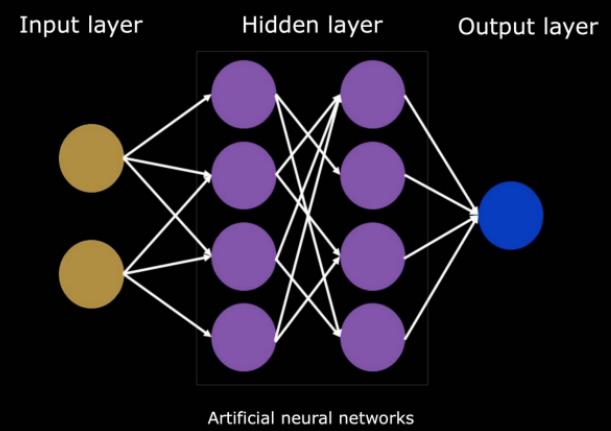
- Organizadas em camadas



- Input (\vec{x}) $\rightarrow \vec{x}^0$

- hidden (internas) $\rightarrow \vec{x}^i$

- output (\vec{y}) $\rightarrow \vec{x}^n$

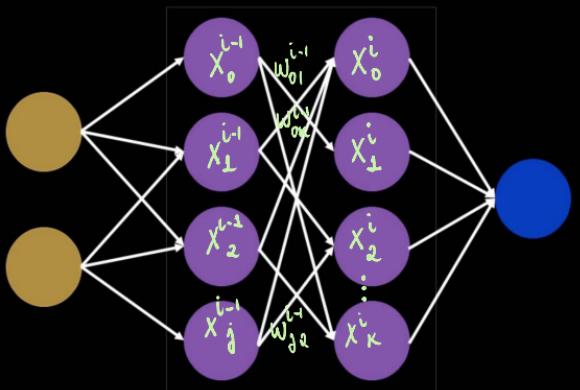


Artificial neural networks

Input layer

Hidden layer

Output layer



Artificial neural networks

$$X_0^i = W_{00}^{i-1} X_0^{i-1} + W_{10}^{i-1} X_1^{i-1} + \dots + W_{j0}^{i-1} X_j^{i-1} + b_0^{i-1}$$

$$\vdots$$

$$X_K^i = W_{0K}^{i-1} X_0^{i-1} + W_{1K}^{i-1} X_1^{i-1} + \dots + W_{jK}^{i-1} X_j^{i-1} + b_K^{i-1}$$

$$X^i = \begin{bmatrix} X_0^i \\ X_1^i \\ \vdots \\ X_K^i \end{bmatrix} \quad i-1 = \begin{bmatrix} X_0^{i-1} \\ X_1^{i-1} \\ \vdots \\ X_j^{i-1} \end{bmatrix} \quad b = \begin{bmatrix} b_0^{i-1} \\ b_1^{i-1} \\ \vdots \\ b_K^{i-1} \end{bmatrix}$$

K pode ser diferente de i

$$W^{i-1} = \begin{bmatrix} W_{00}^{i-1} & W_{10}^{i-1} & \dots & W_{j0}^{i-1} \\ W_{01}^{i-1} & W_{11}^{i-1} & \dots & W_{j1}^{i-1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{0K}^{i-1} & W_{1K}^{i-1} & \dots & W_{jK}^{i-1} \end{bmatrix}$$

$$X^i = W^{i-1} X^{i-1} + b^{i-1}$$

Funções de Ativação

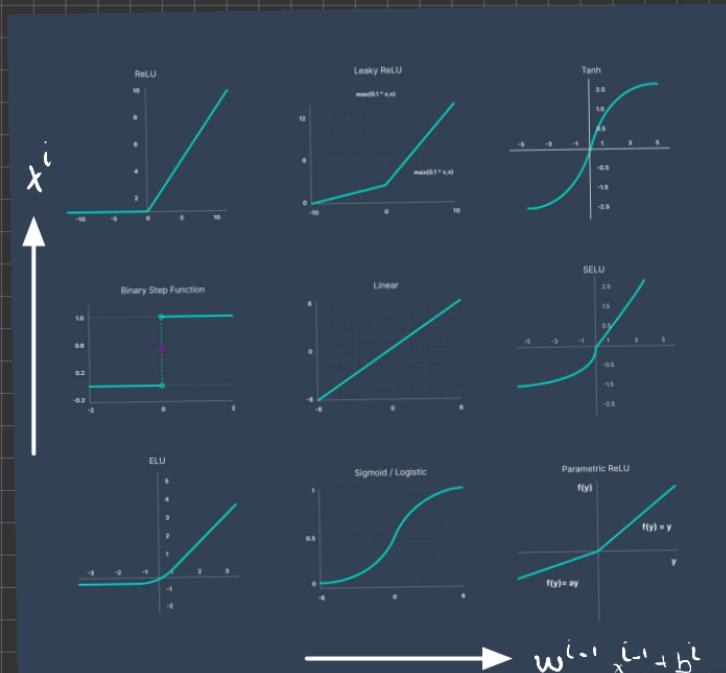
A expressão $x^i = w^{i-1}x^{i-1} + b^{i-1}$ é linear!

Para introduzir não linearidades na rede e também modular os sinais dos neurônios. A resposta de cada neurônio passa por uma função de ativação. logo:

$$x^i = h[w^{i-1}x^{i-1} + b^{i-1}]$$

$h \rightarrow$ função de ativação

Vários tipos à escolha do usuário



Fazendo uma rede Aprender (treinamento)

- Como determinamos os valores de pesos (w_{jk}^i) e biases (b_k^i)?
- Precisamos encontrar a menor discrepância possível entre os vetores \hat{y} previstos pela rede e os queles conhecidos nos dados de treino y_L

↳ Função de custo

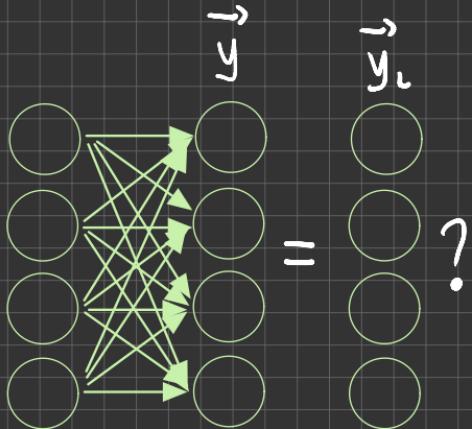


Função de custo

VÁRIOS tipos de custos

- Erro absoluto médio

$$MAE = \frac{1}{N} \sum_{i=1}^N |\vec{y} - \vec{y}_L|$$



- Erro quadrático MÉDIO

$$MSE = \frac{1}{N} \sum_{i=1}^N (\vec{y} - \vec{y}_L)^2$$

- Entropia cruzada (classificação)

Para um conjunto de dados de treino \vec{x} , calcula-se a saída da rede \vec{y} e compara com o rótulo \vec{y}_L

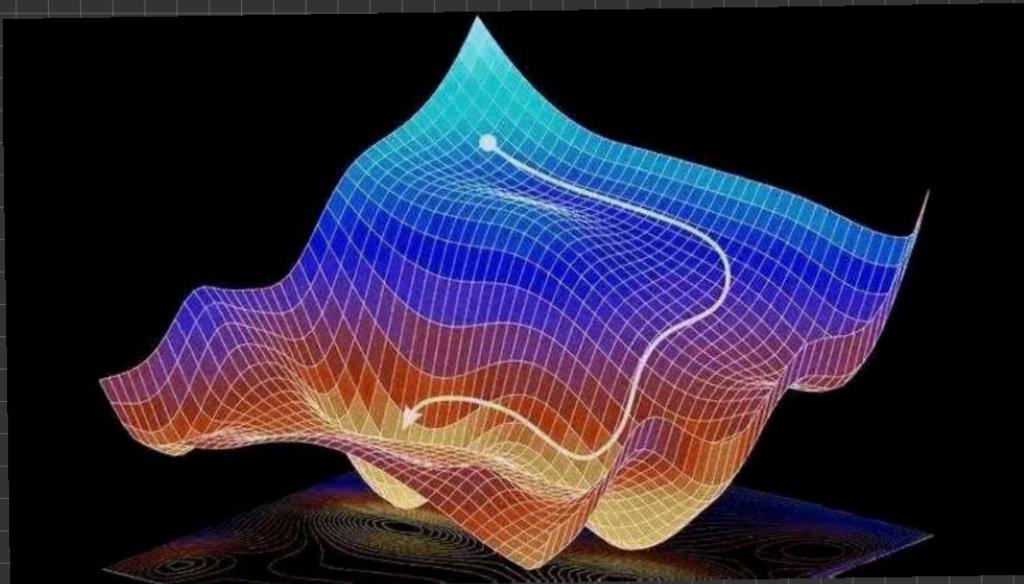
$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log(y_{ic})$$

- A escolha depende do contexto

Como encontrar o valor mínimo?

A estratégia mais comum é começar de um local aleatório e avaliar o gradiente da função de perda, até atingir o valor mínimo

$$\rightarrow -\nabla \text{Perda}(w, b)$$



Taxa de Aprendizado

Correça com um conjunto w, b
de modo que

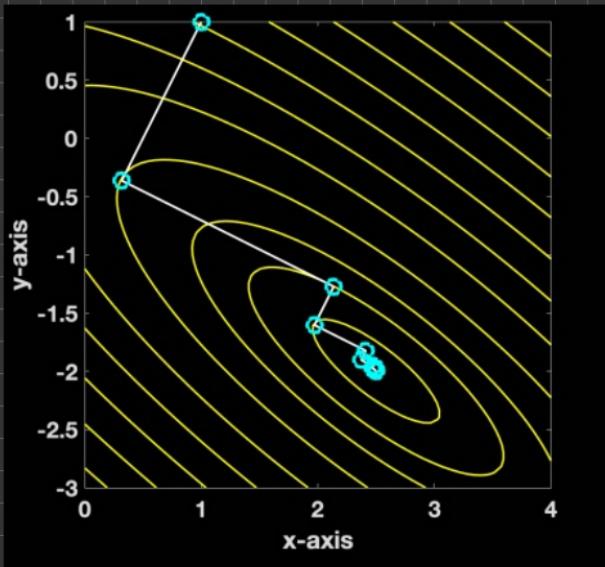
$$w(t+1) = w(t) - \eta \nabla C(w(t), b(t))$$

$$b(t+1) = b(t) - \eta \nabla C(w(t), b(t))$$

η = taxa de aprendizagem

η muito grande \rightarrow passar do ponto

η muito pequeno \rightarrow demora e risco
de mínimos locais



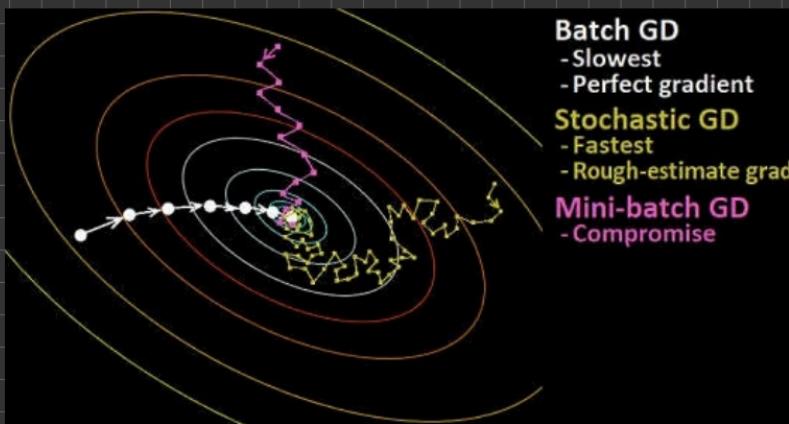
Back propagation

- Para um conjunto de dados de treinamento
 - - calcule o output da rede
 - calcule o custo
 - volte para trás a camada calculando o ∇C
 - ajuste os valores de w e b
 - repita até convergir

↳ CADA REPETIÇÃO => EPOCH

• Algunas estrategias comuns

1. Atualizar w, b para cada dado de treino
 - RÁPIDO com MUITAS FLUTUAÇÕES
2. Usar todo conjunto de treino p/ atualizar
 - Lento e suave
3. mini batches → INTERMEDIÁRIO



Métricas de desempenho/qualidade

- Depois de treinado → modelo deve ser validado
- Definições:

$$\begin{array}{l} \left. \begin{array}{l} TP = \text{True Positive} \\ TN = \text{True Negative} \\ FP = \text{False Positive} \\ FN = \text{False Negative} \end{array} \right\} \text{Resultados corretos} \\ \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{Resultados errados} \end{array}$$

- Acurácia → Percentual de respostas certas

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precisão → Proporção de TP entre todos positivos

$$P = \frac{TP}{TP + FP}$$

- Recall → Proporção de TP identificados corretamente

$$R = \frac{TP}{TP + FN}$$

- F1-score → Média harmônica Precisão/recall

$$F1 = 2 \frac{P \times R}{P + R}$$

Matriz de confusão

FORMA VISUAL de VISUALIZAR ESSAS GRANDEZAS

Ex: CLASSIFICAÇÃO entre 3 categorias de animais

- CACHORROS, GATOS, ESQUILOS

	CACHORRO PREVISÃO	GATO PREVISÃO	ESQUILO PREVISÃO
CACHORRO REAL	50	5	3
GATO REAL	2	45	8
ESQUILO REAL	1	4	40

$$\text{Precisão} = \frac{TP}{TP+FP} = \frac{50}{50+2+1} = 94,3\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{50}{50+5+3} = 86,2\%$$

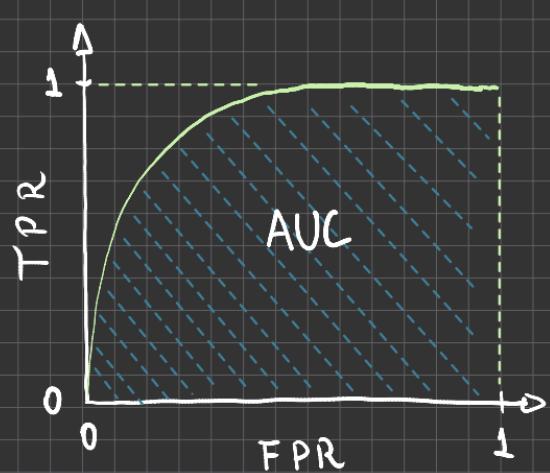
$$F1 = 2 \times \frac{\text{Precisão} \times \text{recall}}{\text{Precisão} + \text{recall}} = 90,1\%$$

CURVA ROC e AUC

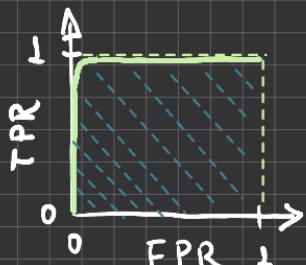
Grafico que ilustra o desempenho de um modelo de classificação binária
(para múltiplas classes normalmente faz-se 1 contra o resto)

Eixo-x \Rightarrow taxa de FALSOS POSITIVOS (QUANTO MENOR, MELHOR) FPR

Eixo-y \Rightarrow taxa de verdadeiros positivos (QUANTO MAIOR, MELHOR) TPR

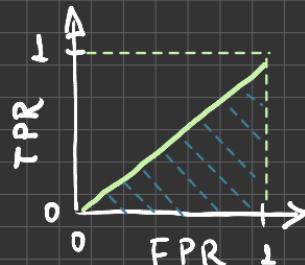


AUC \rightarrow Área under curve



Modelo ideal

$$AUC = 1$$



Modelo aleatório

$$AUC = 0.5$$

Atividade de hoje

Construir uma rede neural para classificar estrelas
com base em parâmetros como tamanho, massa, luminosidade, etc

Realizar treinamento, calcular métricas, e fazer previsões

Biblioteca KERAS do Python

Algumas bibliotecas de ML

SCIKIT-LEARN → GERAL DE ML, VÁRIOS ALGORÍTMOS
REDES NEURAIS SIMPLES, CPU

KERAS

→ INTERFACE COM TENSORFLOW (GOOGLE)

PyTorch

→ INTERFACE COM TORCH (META)