

Redes Neurais Artificiais

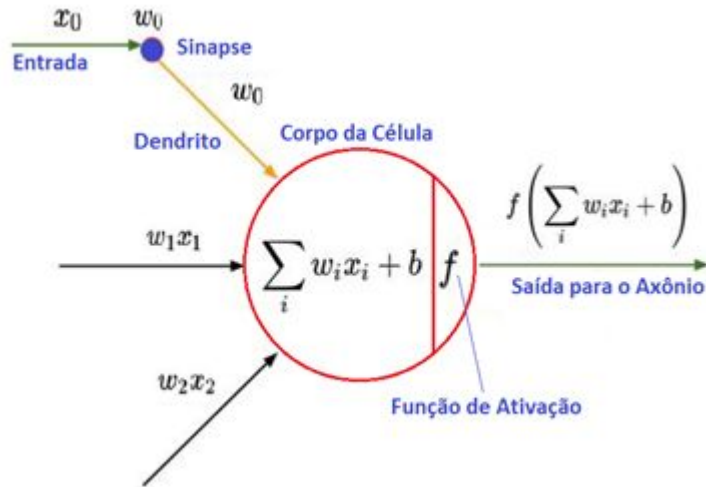
Professora: Anita Maria da Rocha Fernandes
Mestrando: Luiz Henrique A. Salazar

Agenda

- **Modelo Linear**
 - Perceptron como Classificador Linear
 - Função de Ativação
 - Problemas Multiclasse
- **Construção das Redes Neurais**
 - Problemas não linearmente separáveis
 - Arquitetura

Perceptron como Classificador Linear

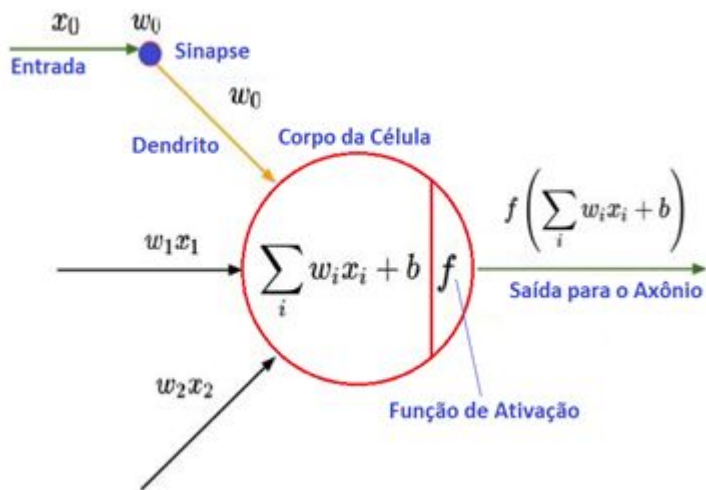
O perceptron é a unidade fundamental de redes neurais, ele recebe **d** conexões de entrada e produz um **único valor de saída**.



Cada **neurônio** de uma RNA é um **perceptron** como esse.

Perceptron como Classificador Linear

O perceptron é a unidade fundamental de redes neurais, ele recebe **d** conexões de entrada e produz um **único valor de saída**.

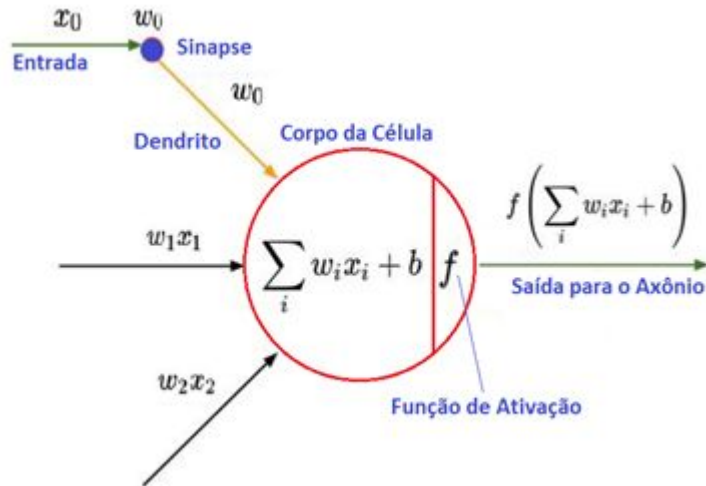


Seu processamento é composto:

- Função de mapeamento (transformação **linear**).
- Função de **ativação**.

Perceptron como Classificador Linear

O **treinamento** dos neurônios da rede busca encontrar os parâmetros $\mathbf{W} = \{w_1, w_2, \dots, w_n\}$ e \mathbf{b} , que minimizam os erros da rede.



Aula 2 P1 - Redes Neurais - Classificação Linear

PyTorch - Módulo *nn*

O PyTorch oferece o módulo **nn** (**n**eural **n**etworks) para a construção de redes neurais.

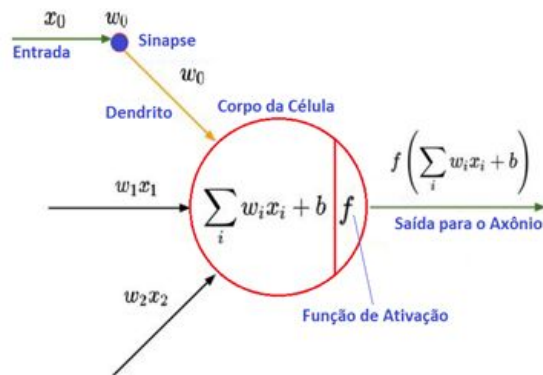
Uma ampla diversidade de **camadas neurais** podem ser instanciadas através deste módulo.

Para instanciar uma camada referente ao **perceptron clássico**, utilizamos a camada **LINEAR**.

PyTorch - Módulo *nn*

Esta camada recebe dois parâmetros:

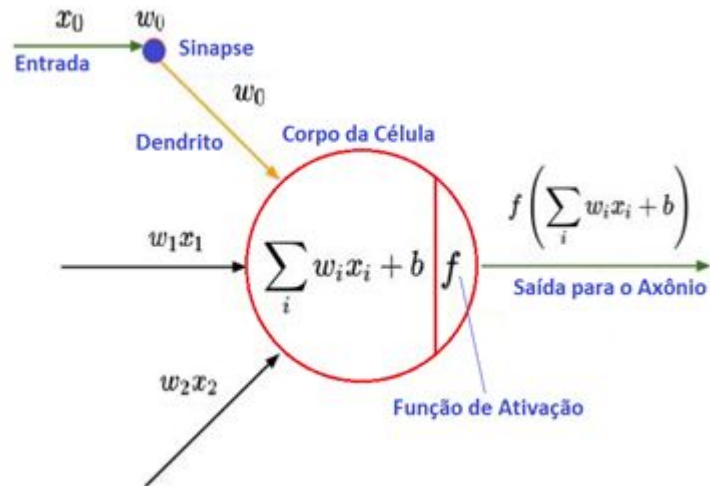
- **in_features:** dimensionalidade da entrada
- **out_features:** dimensionalidade da saída



PyTorch - Módulo *nn*

Neste exemplo:

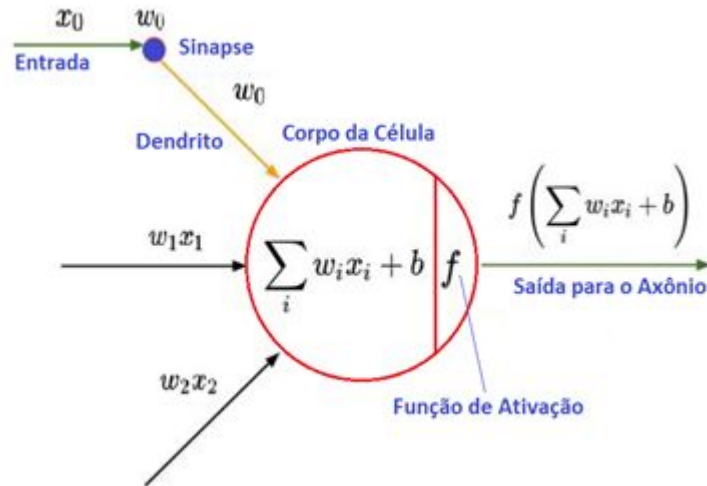
- **in_features:** 3
 - $X = \{x_1, x_2, x_3\}$
- **out_features:** 1
 - $W = \{w_1, w_2, w_3\}$
 - $y = \mathbf{WX} + \mathbf{b}$



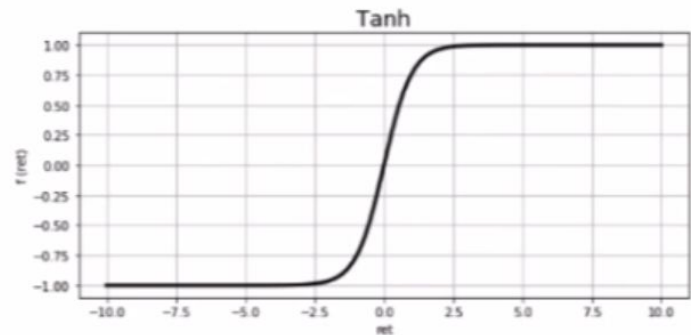
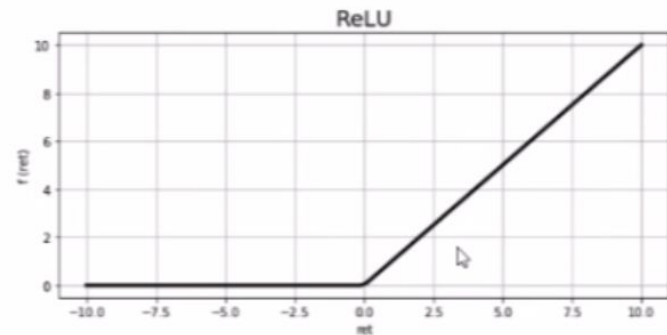
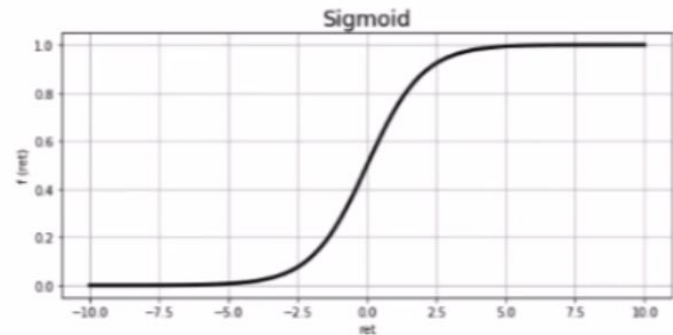
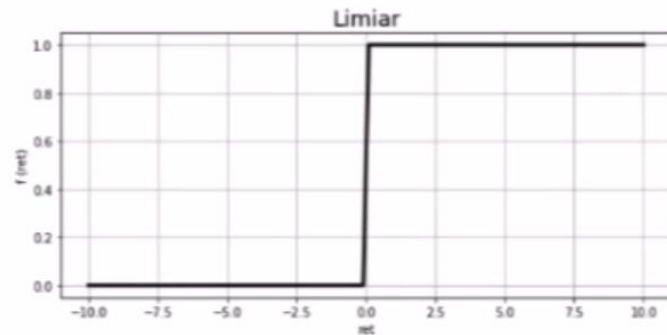
Aula 2 P2 - Redes Neurais - Módulo Linear Perceptron

Função de Ativação

O papel da **função de ativação** é definir se o neurônio vai **ativar** e qual a sua **força** de ativação.

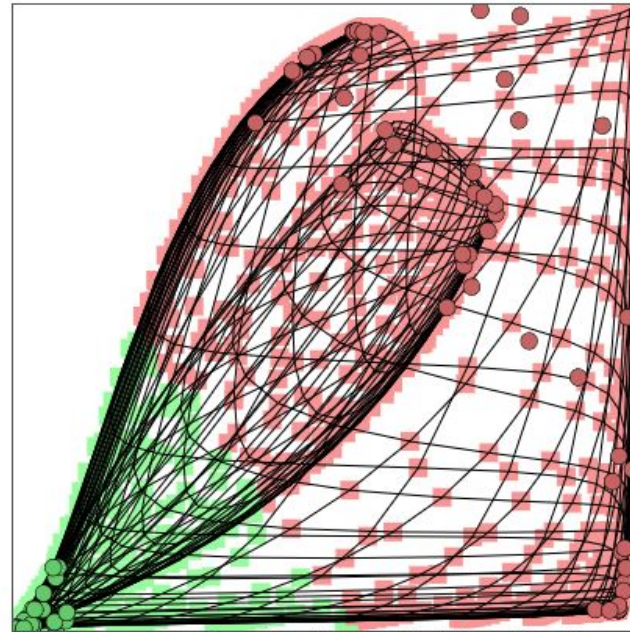
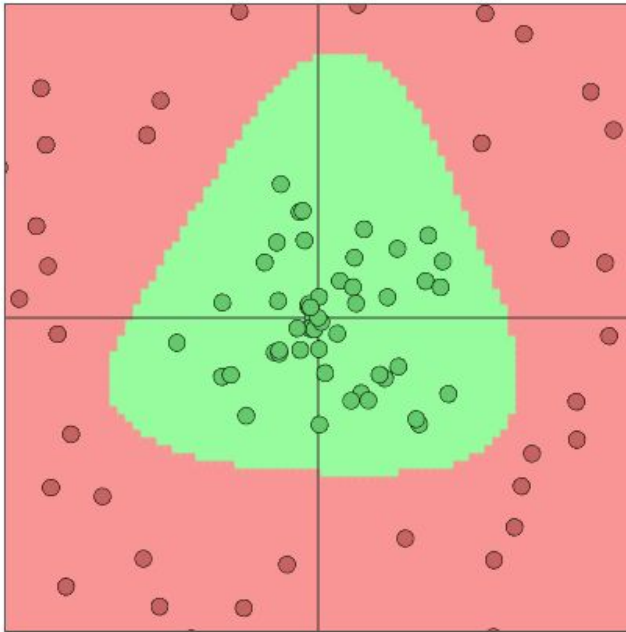


Função de Ativação



Função de Ativação

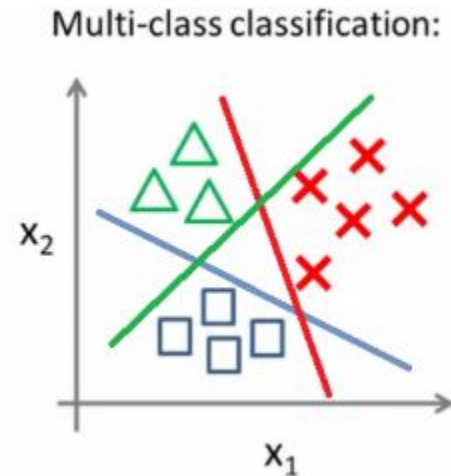
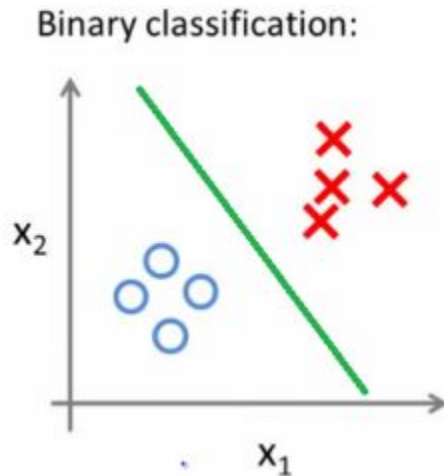
ConvnetJS demo



Aula 2 P2 - Redes Neurais - Função de Ativação

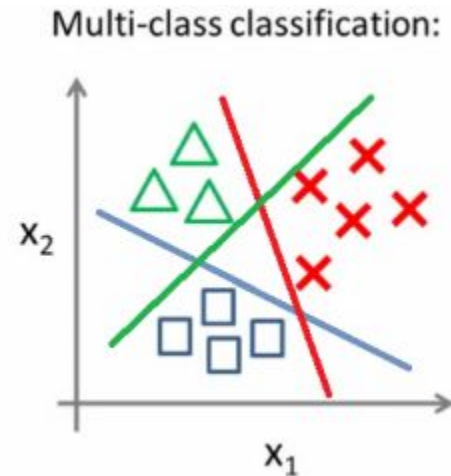
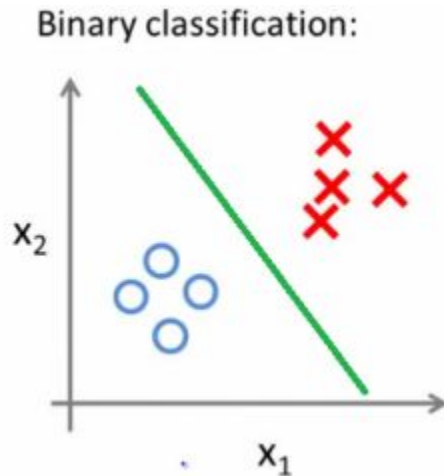
Problemas multi-classe

Limitação de um **único perceptron** de modelar uma única **reta** ou único **hiperplano**.



Problemas multi-classe

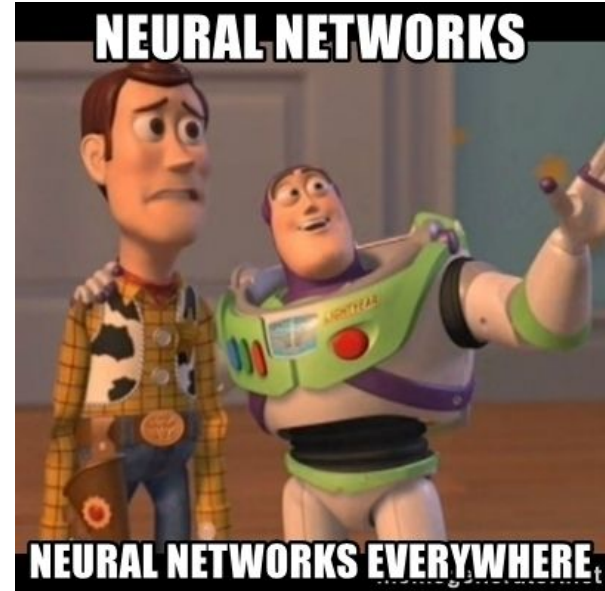
Mesmo que as classes sejam **linearmente separáveis**, um único perceptron não consegue separar todas as classes.



Redes Neurais

Treinar **múltiplos perceptrons** ao mesmo tempo, **especializando** cada um em um objetivo **diferente**.

- Classificador de triângulo
- Classificador de xis
- Classificador de quadrado

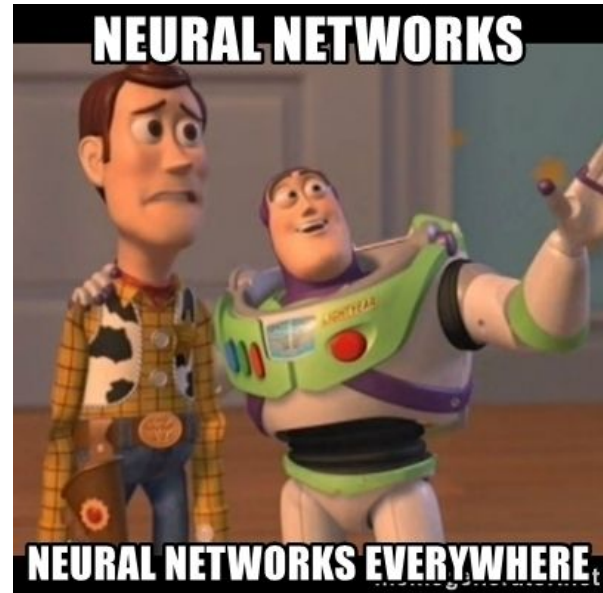


Redes Neurais

Camadas da rede neural:

- Input: 2 dimensões
- Output: **3 neurônios**

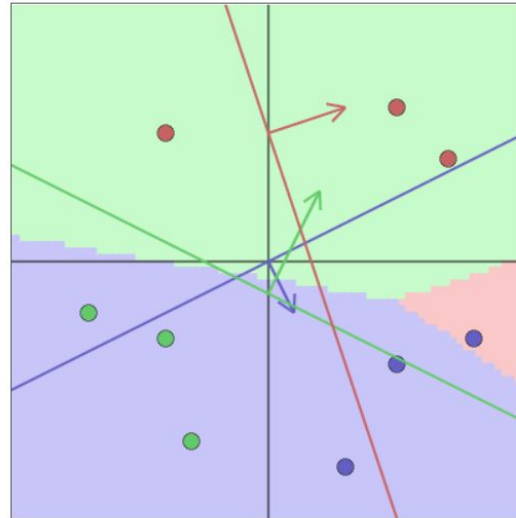
Comparar a saída dos 3 perceptrons e visualizar qual deles teve a ativação mais forte.



Redes Neurais

Multi Class Demo

the blue line shows the set of points (x_0, x_1) that give score of zero. The blue arrow draws the vector $(W_{0,0}, W_{0,1})$, which shows the direction of score increase and its length is proportional to how steep the increase is.
Note: you can drag the datapoints.



$w[0,0]$	$w[0,1]$	$b[0]$
▲	▲	▲
1.00 -0.38	2.00 0.07	0.00 0.11
▼	▼	▼
$w[1,0]$	$w[1,1]$	$b[1]$
▲	▲	▲
2.00 0.51	-4.00 -0.58	0.50 -0.11
▼	▼	▼
$w[2,0]$	$w[2,1]$	$b[2]$
▲	▲	▲
3.00 0.17	-1.00 0.36	-0.50 0.00
▼	▼	▼

Step size: 0.10000

Single parameter update

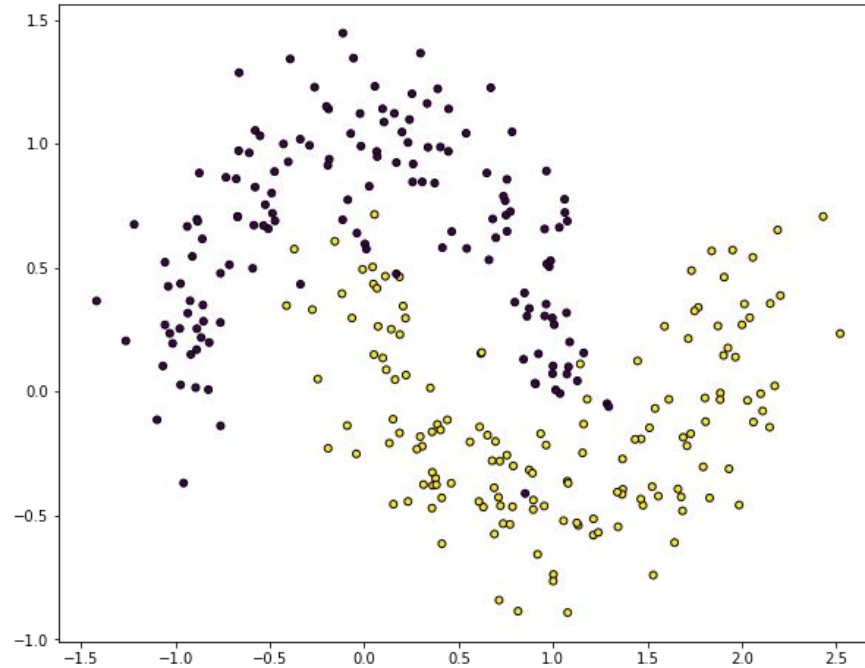
Start repeated update

Redes Neurais

No PyTorch, uma rede neural com múltiplos neurônios na mesma camada, é representada da seguinte maneira:

```
[ ] perceptron = nn.Linear(in_features=2, out_features=3)
```

Problemas não linearmente separáveis



Aula 2 P3 - Redes Neurais - Arquitetura