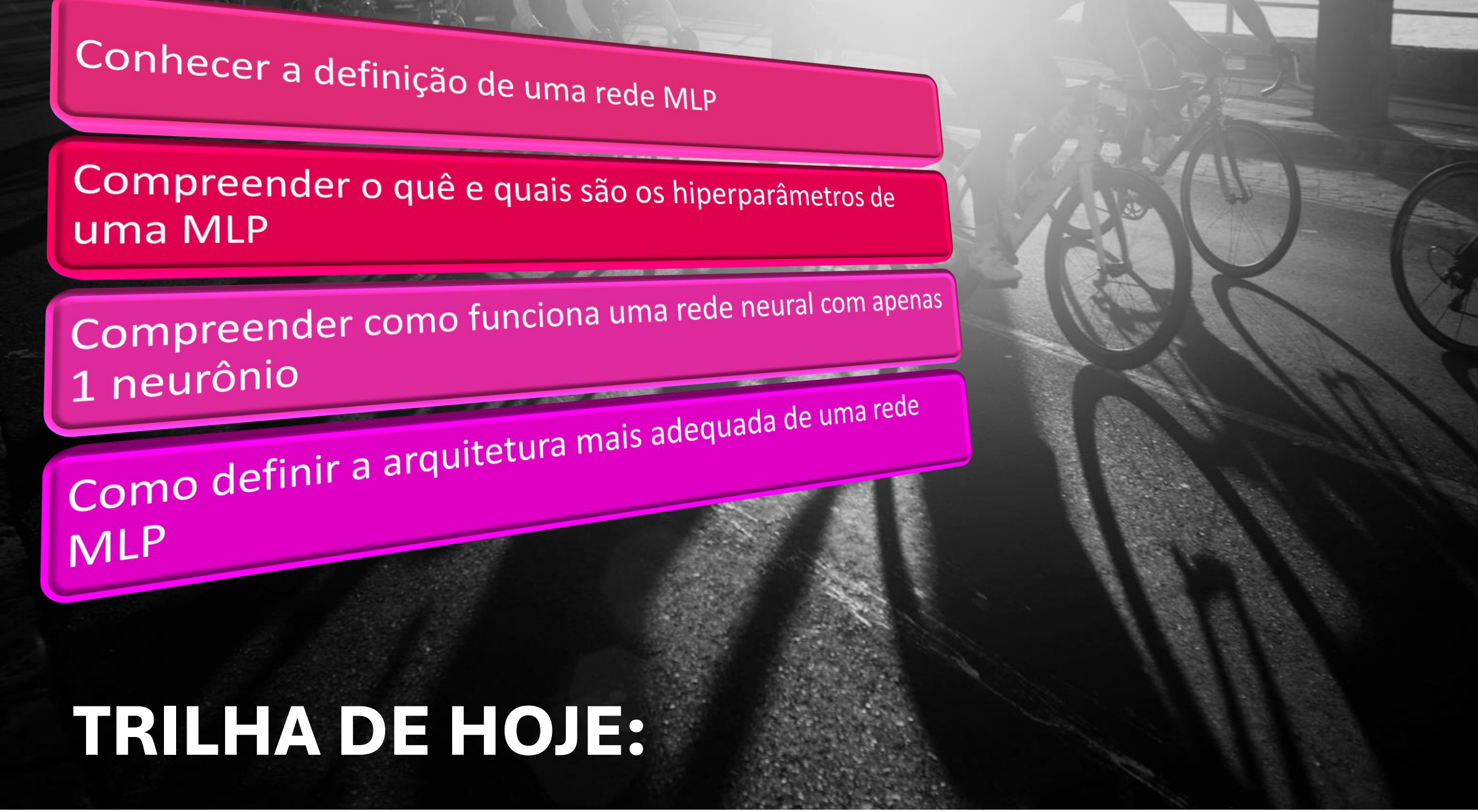
REDES MULTICAMADAS-MLP

Tópicos
Avançados II

Aula 4

Augusto Uchôa

Petran- Programa de Pós-graduação em Engenharia de Transportes



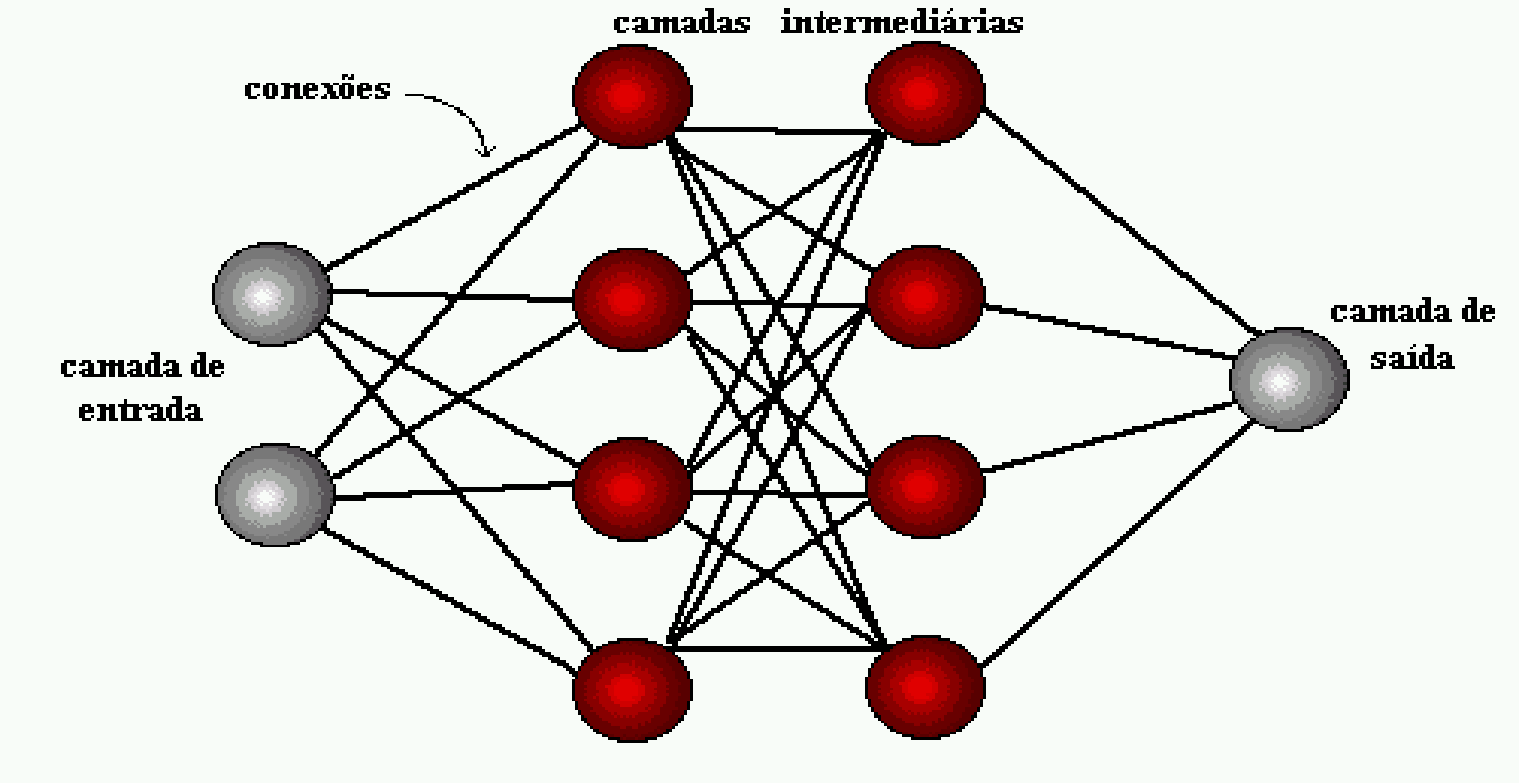
TRILHA DE HOJE:

PRAZER, SOU UMA MLP!

(do inglês "Multilayer Perceptrons")

São um tipo de rede neural artificial, e também são conhecidas como redes neurais de alimentação direta ou redes neurais multicamadas.

São uma extensão das redes neurais de camada única (**também chamadas de perceptrons**) e são usadas em muitas aplicações de aprendizado de máquina.



O QUE SÃO OS HIPERPARÂMETROS?

- São parâmetros que não são aprendidos durante o treinamento da rede, mas **precisam ser definidos, pelo modelador, antes do início da etapa de treinamento** da rede
- **Afetam diretamente o comportamento e o desempenho** da rede
- **Muitas vezes envolve experimentação e ajuste fino.**

HIPERPARÂMETROS FUNDAMENTAIS

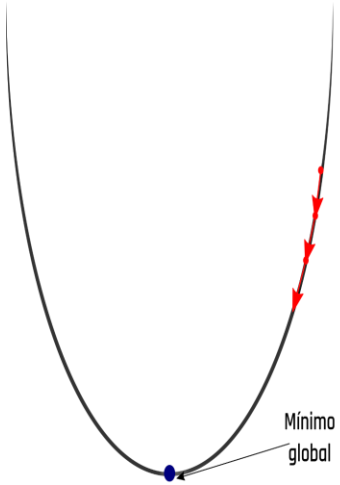
1. Taxa de aprendizado
2. Número de camadas e neurônios por camada
3. Funções de ativação
4. Épocas de treinamento
5. Inicialização de pesos

HIPERPARÂMETROS OPCIONAIS

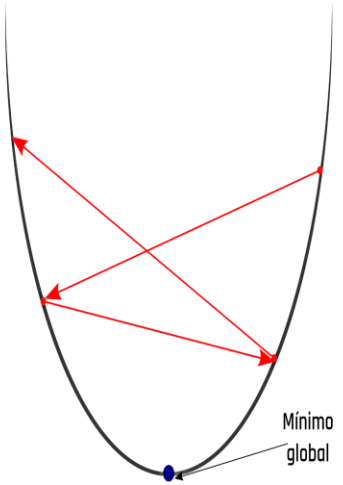
6. Regularização
7. Tamanho do lote (batch size)
8. Dropout
9. Momento (momentum)

TAXA DE APRENDIZAGEM (η)

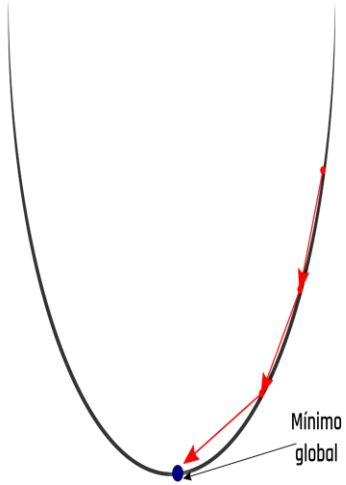
- Controla o tamanho dos passos que a rede dá durante o treinamento para ajustar os pesos das conexões entre neurônios



Taxa de aprendizado pequena



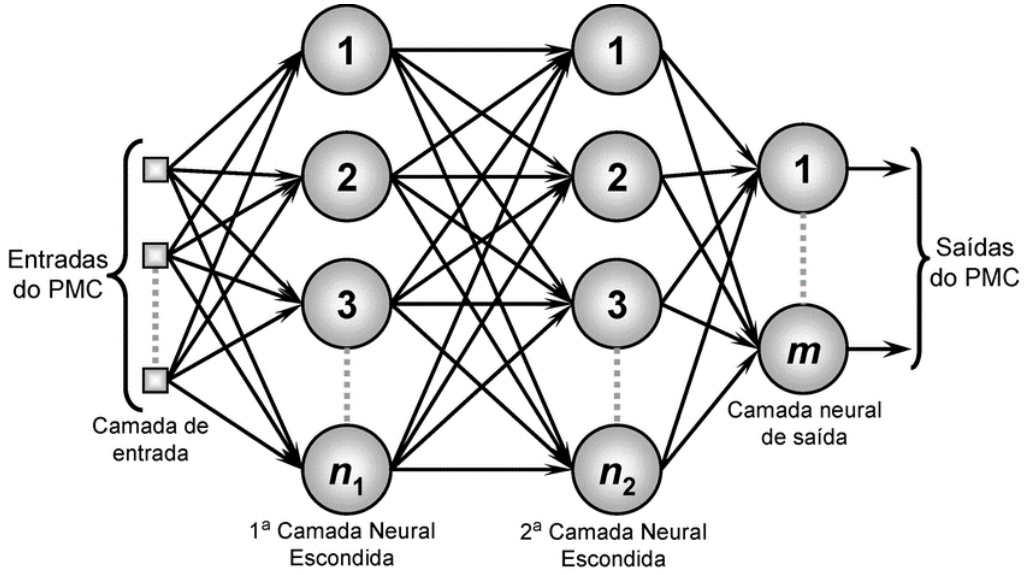
Taxa de aprendizado grande

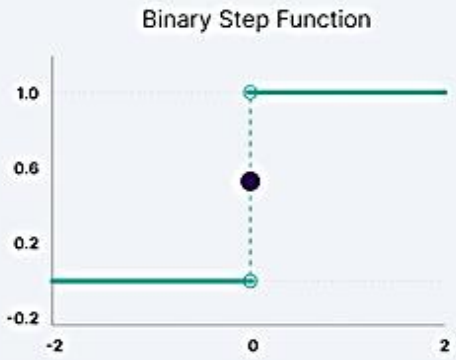


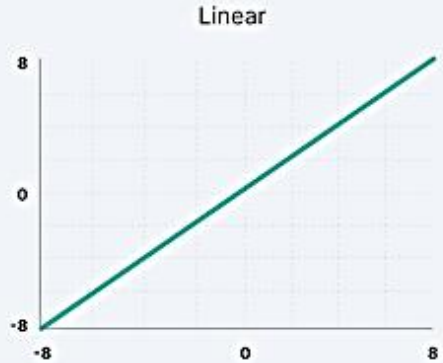
Taxa de aprendizado ideal

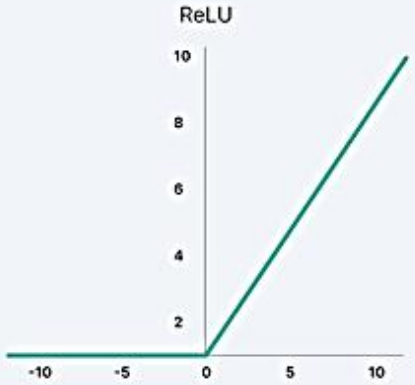
NÚMERO DE CAMADAS E NEURÔNIOS POR CAMADA

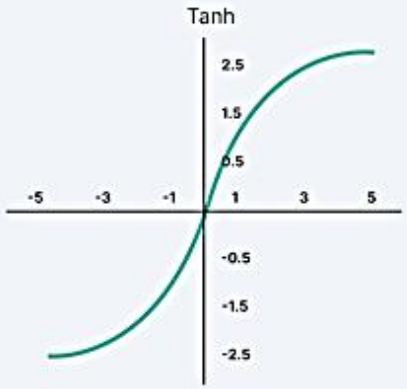
- Define a arquitetura da rede, especificando quantas camadas e quantos neurônios existem em cada camada.

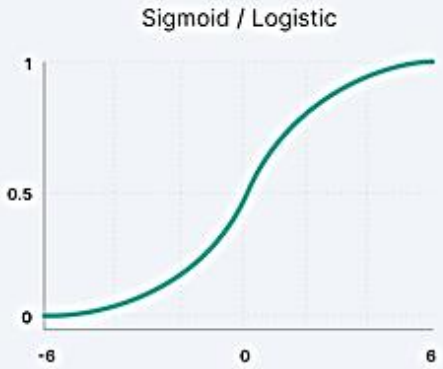


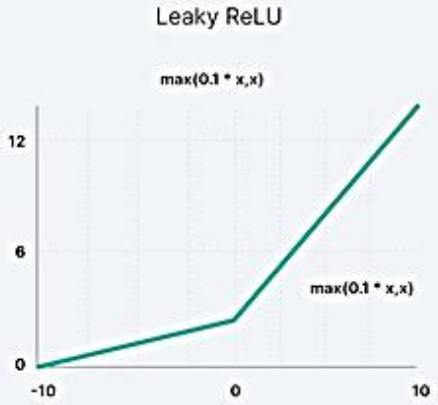


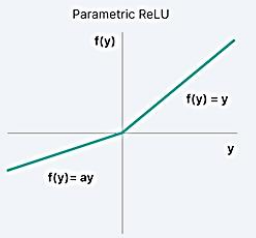


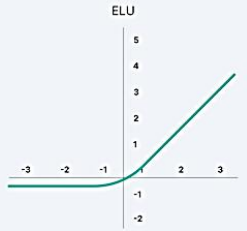


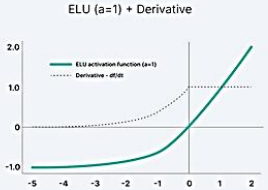


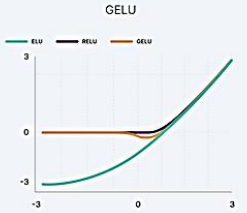


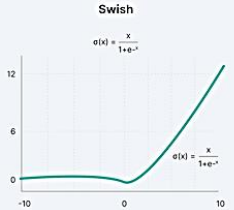












FUNÇÕES DE ATIVAÇÃO

Determina a função matemática que define a saída de cada neurônio, como ReLU (Rectified Linear Unit) ou sigmoide

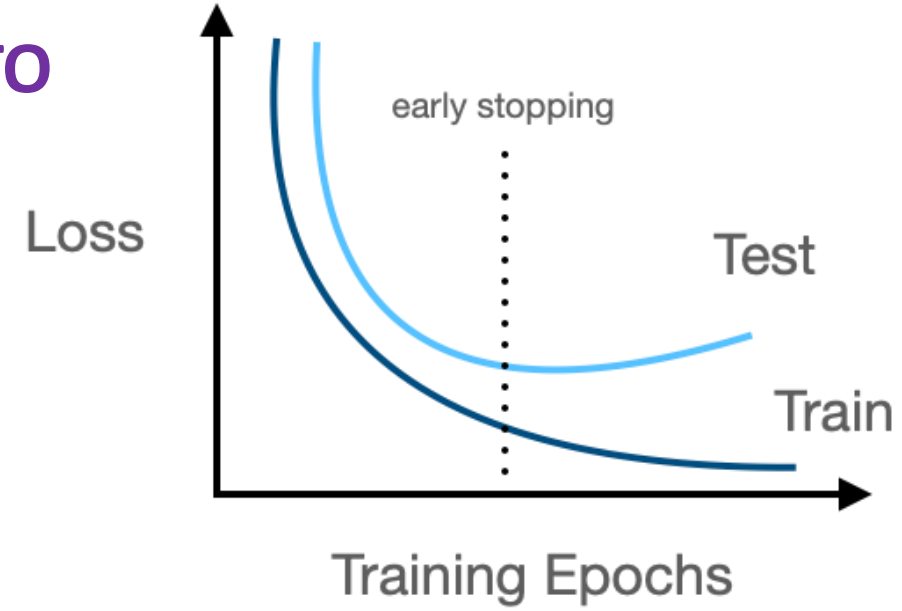
INICIALIZAÇÃO DE PESOS:

- Define como os pesos das conexões são inicializados antes do treinamento, como pesos aleatórios ou pré-treinados.

- Se todos os pesos fossem iniciados com zero, então cada neurônio da rede iria:
  - calcular a mesma saída,
  - calcular os mesmos gradientes durante o backprop,
  - passar pelas mesmas atualizações de parâmetros.
- Portanto, não haveria fonte de assimetria entre os neurônios se seus pesos forem inicializados com o mesmo valor.

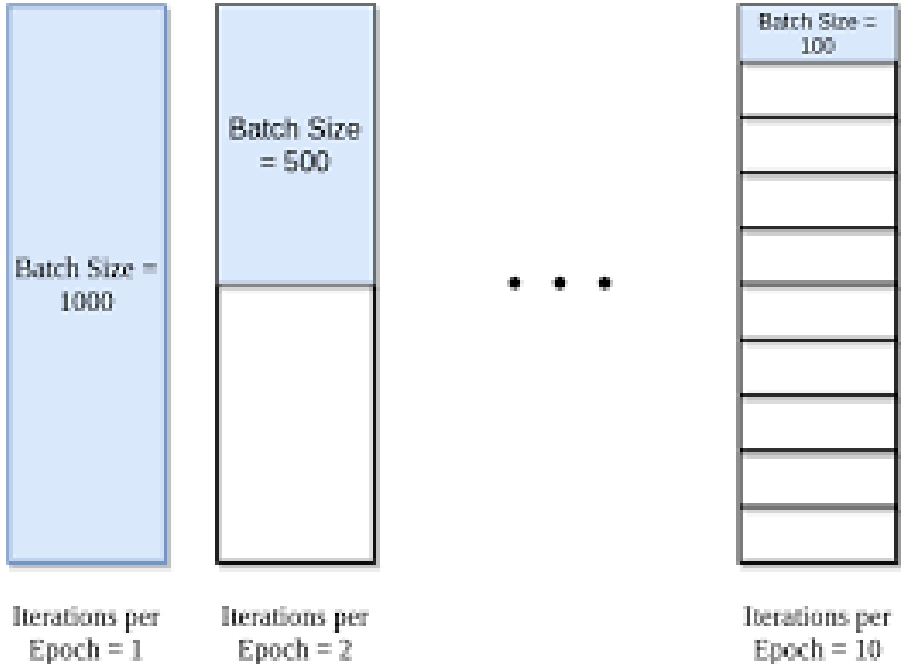
ÉPOCAS DE TREINAMENTO

- Indica quantas vezes todo o conjunto de treinamento é percorrido durante o treinamento.



TAMANHO DO LOTE (BATCH SIZE)

- Define quantos exemplos de treinamento são usados em cada iteração durante o treinamento.

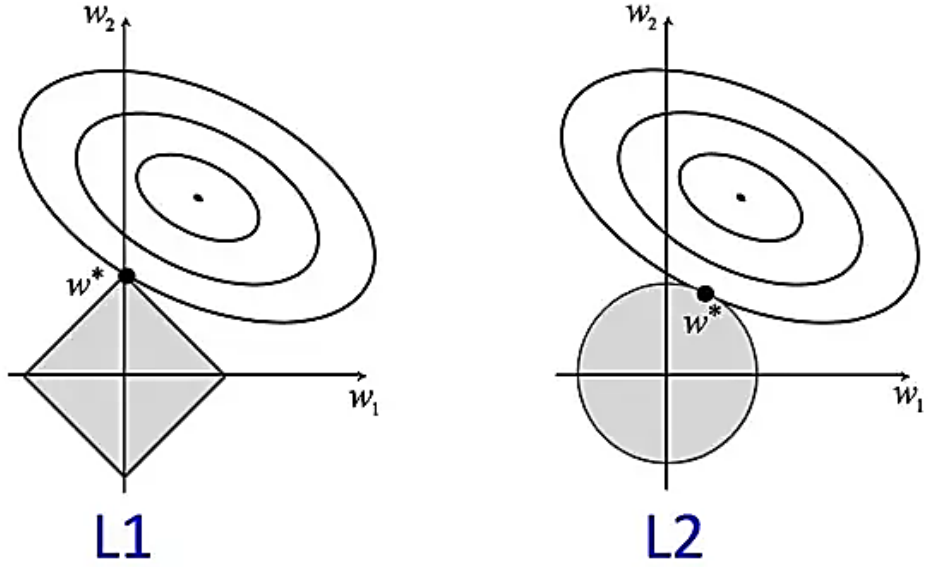


MOMENTO (MOMENTUM) α

- Ajuda a acelerar o treinamento, ajustando os pesos com base em gradientes anteriores. Ajuda a reduzir a influênica da escolha da taxa de aprendizagem. Adiciona uma fração α do valor anterior de atualização dos pesos ao atual

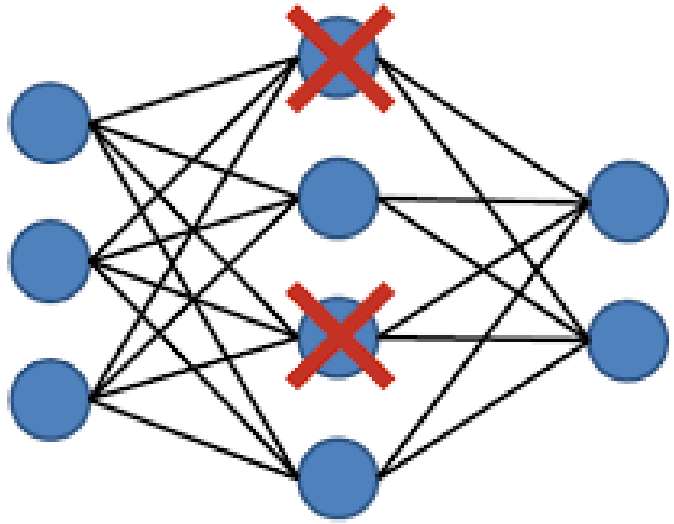
REGULARIZAÇÃO:

- Controla a penalização de pesos grandes para evitar o overfitting, incluindo L1 ou L2 regularization.



DROPOUT:

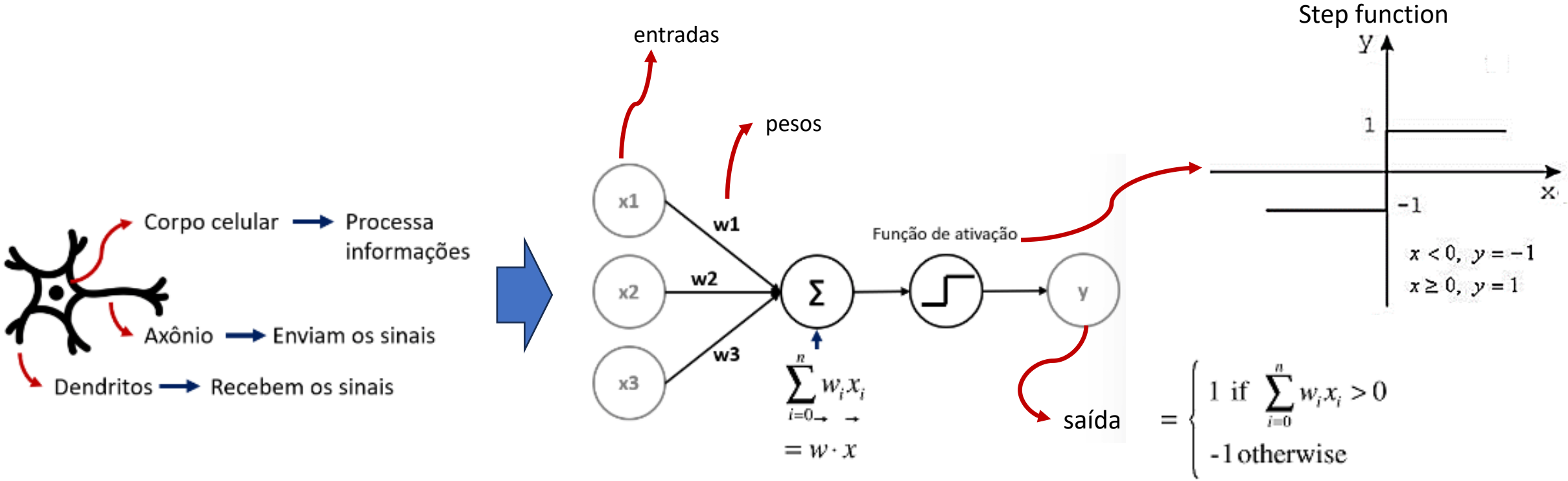
- Uma técnica de regularização que desativa aleatoriamente neurônios durante o treinamento para reduzir o overfitting.



Eu sou demais!

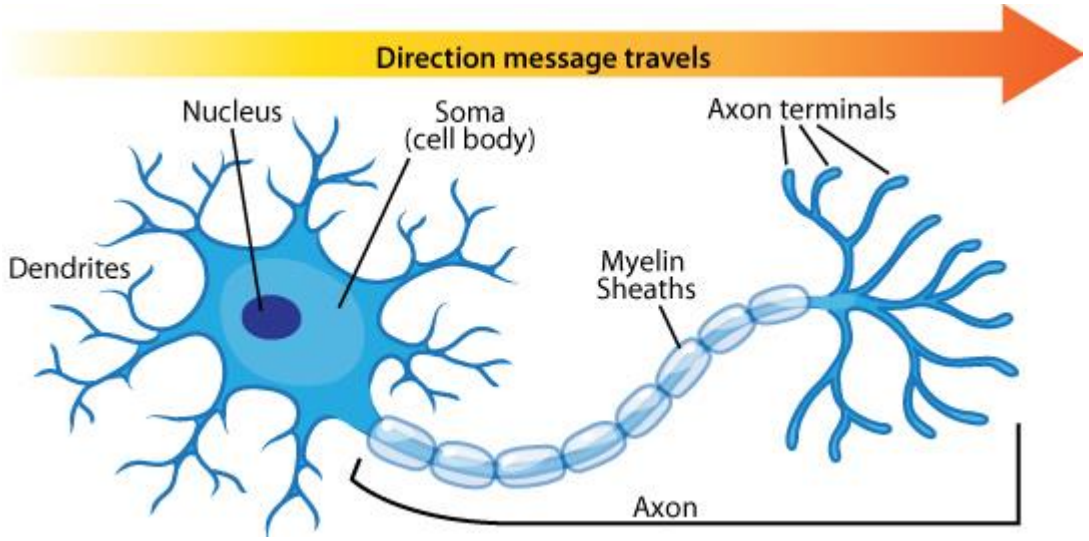
É preciso antes entender o **PERCEPTRON** para poder entender uma MLP!

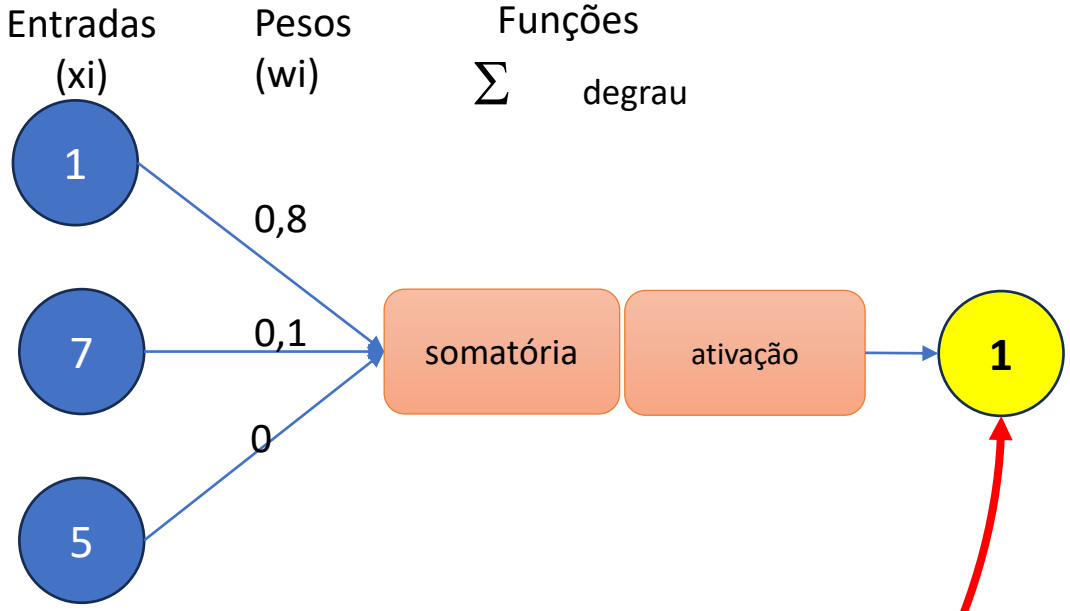
ENTENDENDO O PERCEPTRON!

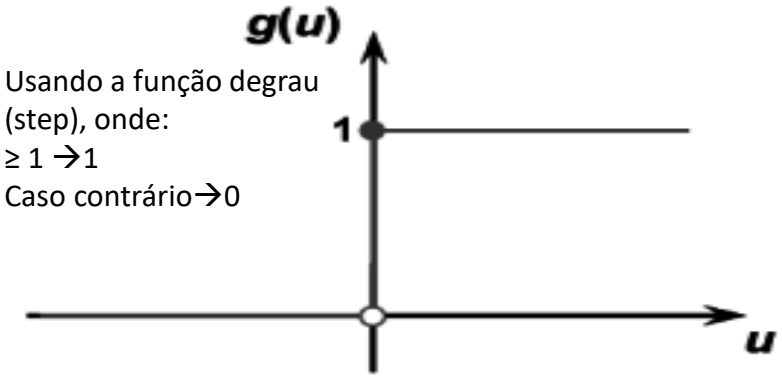


"É uma "rede neural" composta por um **único neurônio** artificial, cujos os pesos podem ser treinados a partir de um vetor de entradas para produzir um vetor saída correspondente"

O QUE PODE FAZER 1 NEURÔNIO?





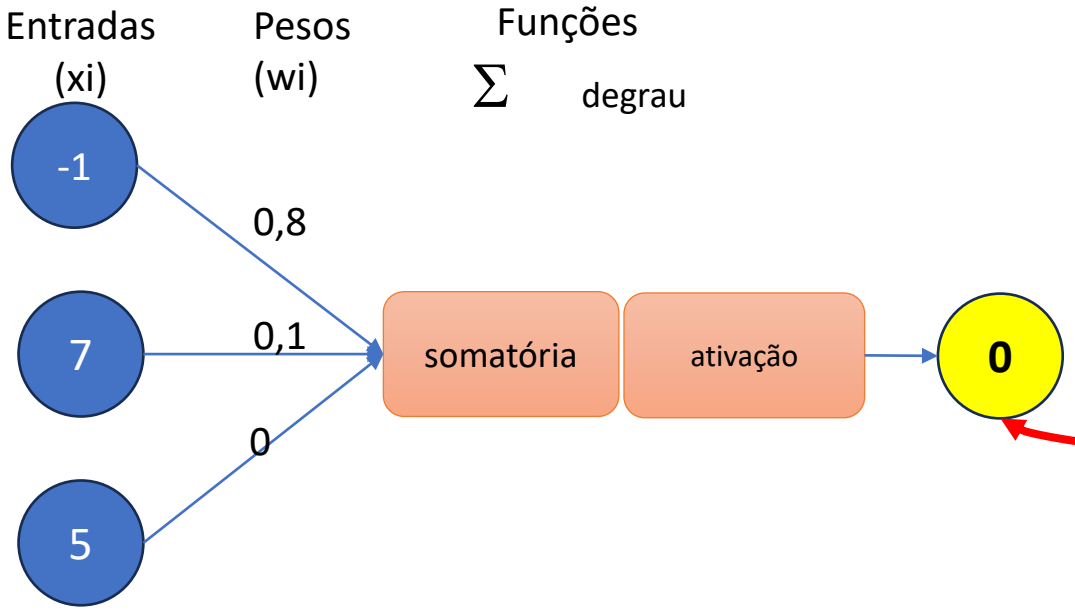


Usando a função degrau (step), onde:
≥ 1 →1
Caso contrário→0

$$soma = \sum_{i=1}^{n} xi * wi$$

Soma =
(1*0,8)+(7*0,1)+(5*0) →1,5
1,5> 1, logo a saída será 1

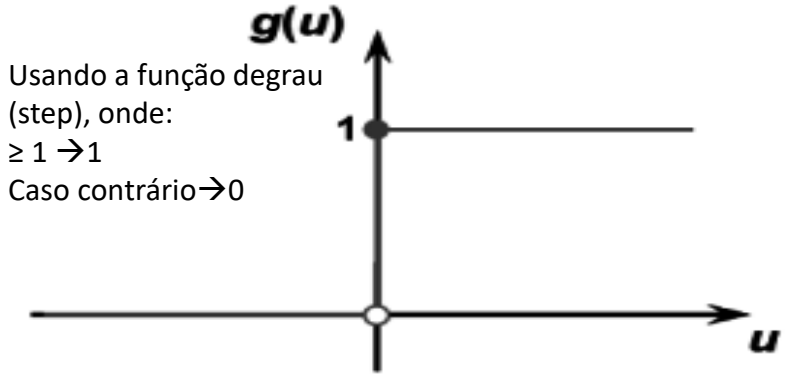
MUDANDO APENAS 1 DAS ENTRADAS



$$soma = \sum_{i=1}^{n} xi * wi$$

Soma = (-1*0,8)+(7*0,1)+(5*0)→-0,1

-0,1< 1, logo a saída será 0



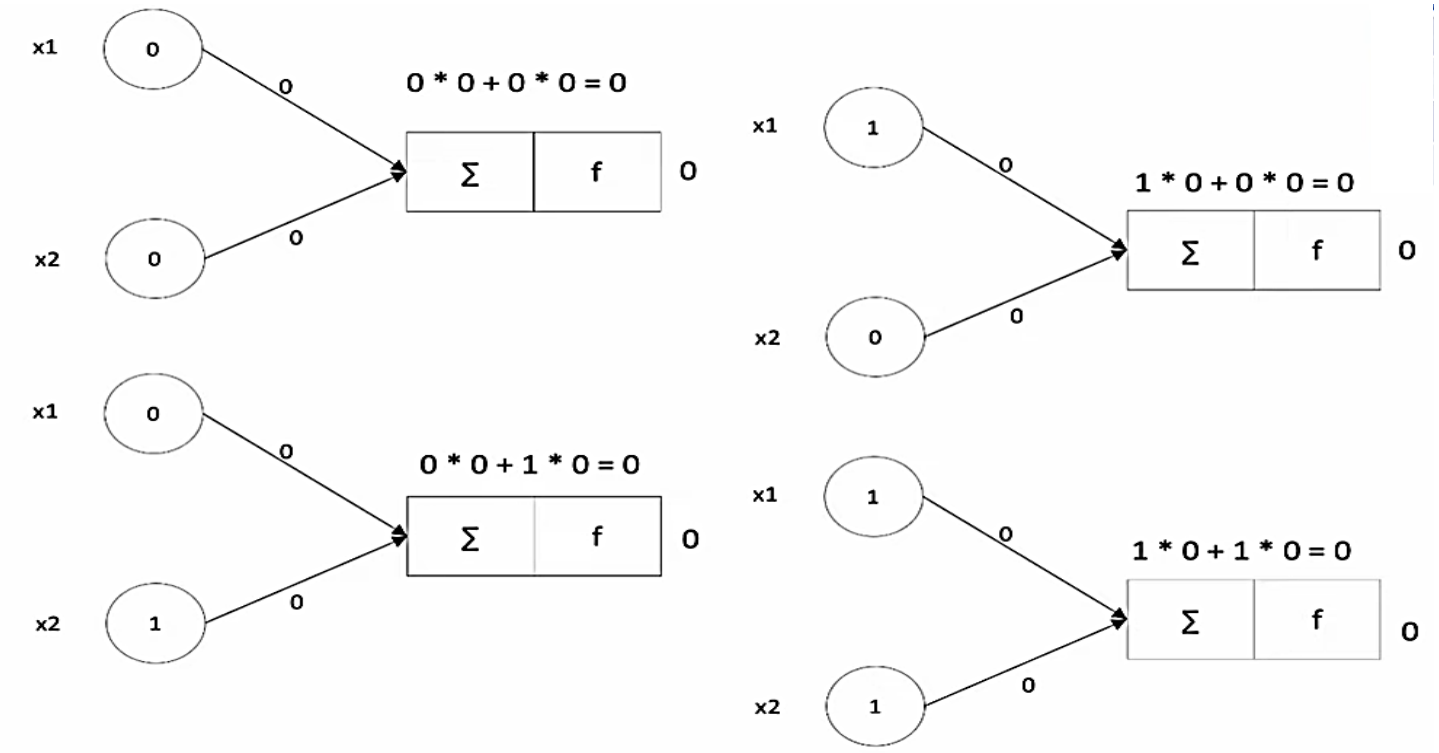
Usando a função degrau (step), onde:
≥ 1 →1
Caso contrário→0

PERCEPTRON VERSUS OPERADORES LÓGICOS?

X1	X2	saida
falso	falso	falso
falso	verdadeiro	falso
verdadeiro	falso	falso
verdadeiro	verdadeiro	verdadeiro

AND

X1	X2	saida
0	0	0
0	1	0
1	0	0
1	1	1



Cálculo dos erros:

0-0=0

0-0=0

0-0=0

1-0=1 **25%**

Atualização dos pesos

Peso(n+1)= peso(n)+(taxa de aprendizagem*entrada*erro

Considerando a taxa de aprendizagem = 0,1 (10%)

Peso(n+1)=0+(0,1*1*1)=0,1

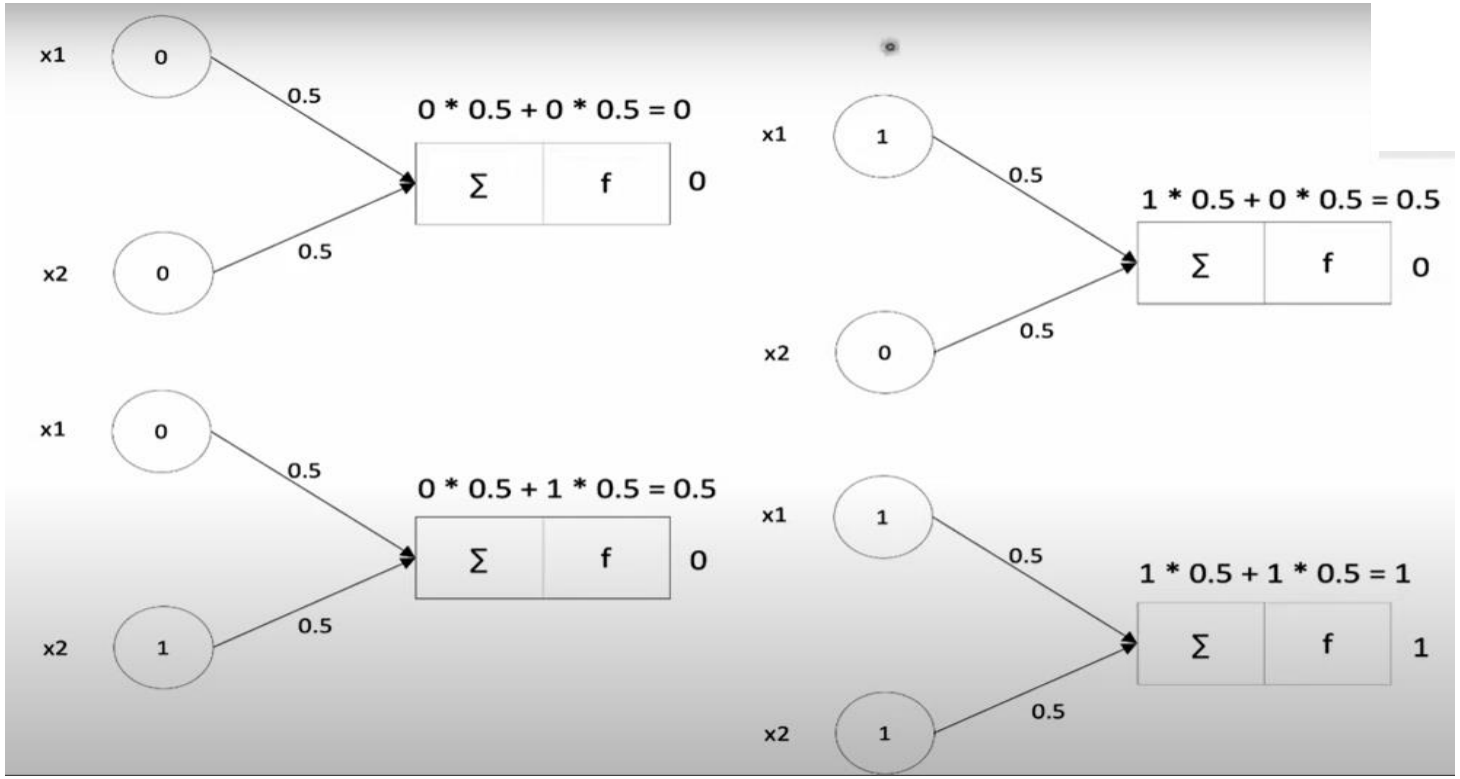
O objetivo então é: encontrar o conjunto de pesos para que a rede possua o menor erro possível

ATUALIZANDO OS PESOS PARA 0,5

X1	X2	saida
falso	falso	falso
falso	verdadeiro	falso
verdadeiro	falso	falso
verdadeiro	verdadeiro	verdadeiro

AND

X1	X2	saida
0	0	0
0	1	0
1	0	0
1	1	1



Cálculo dos erros:

0-0=0

0-0=0

0-0=0

1-1=0

0%

Atualização dos pesos

Peso(n+1)= peso(n)+(taxa de aprendizagem*entrada*erro

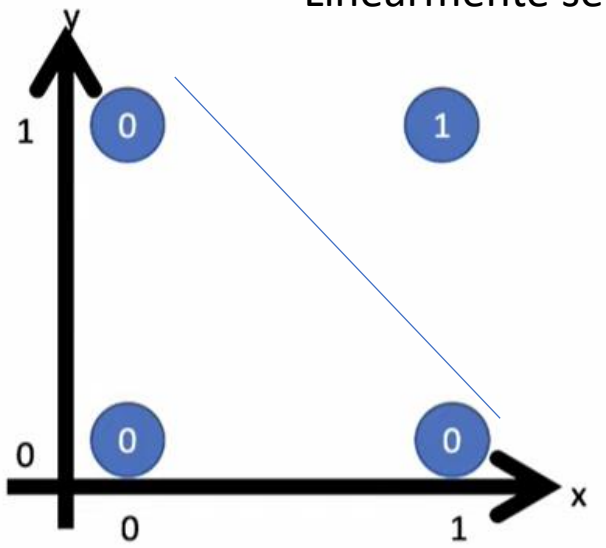
Considerando a taxa de aprendizagem = 0,1

Peso(n+1)=0+(0,5*1*1)=0,1

O objetivo então é: encontrar o conjunto de pesos para que a rede possua o menor erro possível

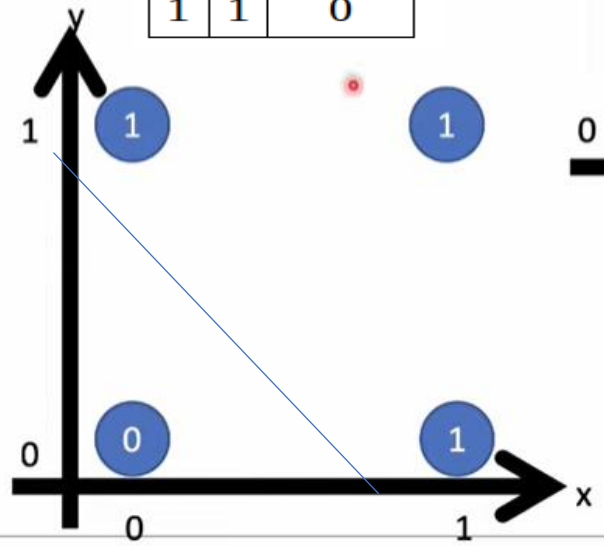
UM NEURÔNIO SÓ NÃO DÁ CONTA DE PROBLEMAS COMPLEXOS

Linearmente separáveis

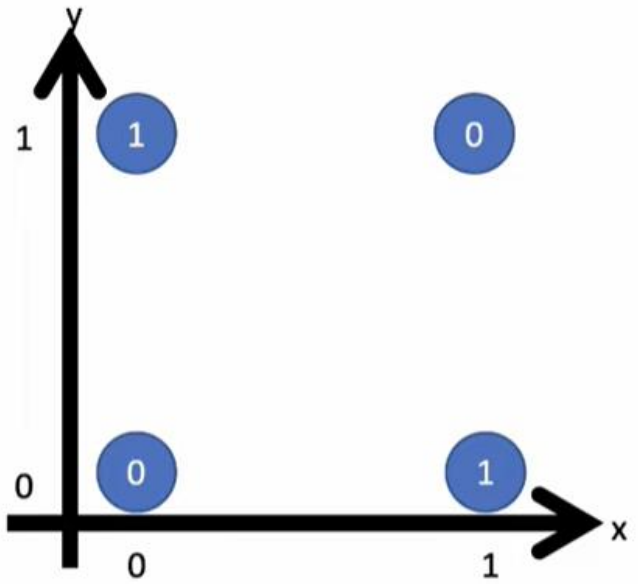


A	B	AND
0	0	1
0	1	0
1	0	0
1	1	0

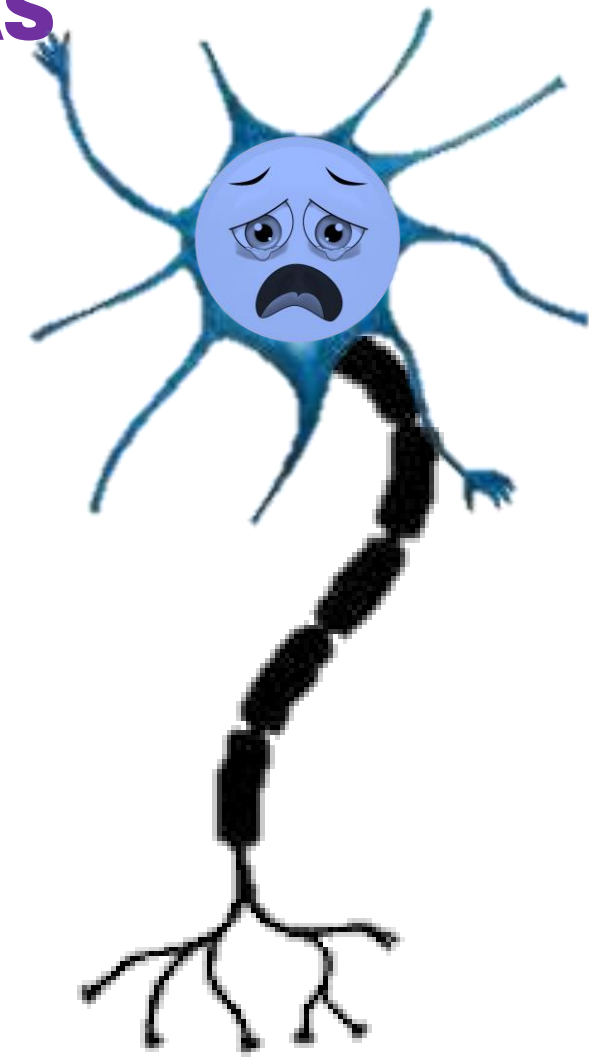
A	B	OR
0	0	1
0	1	0
1	0	0
1	1	0



NÃO Linearmente separáveis



A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0




• 1 ou 2 camadas?
• Quantos neurônios em cada camada?
Eis a Questão!
COMO DEFINIR A ARQUITETURA DE SUA REDE NEURAL?

ÁRVORE DE MODELAGEM

A Definição da arquitetura, ou topologia mais adequada de uma MLP para a geração de um modelo que melhor atenda aos objetivos da modelagem, pode ser bastante exaustiva e, vai de regra consome bastante tempo do modelador, é um trabalho de experimentação, mas conhecer bem o fenômeno modelado ajuda e aprender com os erros dos outros também!

ME AJUDA, TÔ PERDIDO (A, E)!

1. Quantas **entradas** terá seu modelo → 2,3,4,5,...n
2. Quantas **Saídas** terá seu modelo → (estimar, prever (1) ou classificar (2, 3, 4...n)
3. Inicie sua árvore com **uma camada** intermediária→ depois de varrer as possibilidades e não conseguir bons resultados insira mais uma camada intermediária e repita a árvore de modelagem só que com 2 camadas
4. **Varie o número de neurônios** nessa camada intermediária→ tem algumas **dicas** para iniciar
5. **Varie os hiperparâmetros**→ taxa de aprendizagem, função de ativação, momentum, etc..
6. **Varie o algoritmo de treinamento**→ Backpropagation padrão ou **Levemberg Marquardt**

DICAS VALIOSAS!!!

1. LIPPMANN (1987)→ No caso de apenas 1 camada intermediária, ela deverá ter s.(i+1) neurônios, onde: s é o número de neurônios na camada de saída e i é o número de neurônios na camada de entrada. Se houver uma 2ª camada, ela deve ter o dobro de neurônios da camada de saída;
2. CYBENKO (1988)→ apenas 1 camada intermediária é suficiente para aproximar qualquer função contínua e 2 camadas aproximam qualquer função matemática
3. HECHT; NIELSEN (1989)→ Afirmam que apenas 1 camada intermediária já é possível calcular uma função arbitrária qualquer a partir dos dados fornecidos. A camada intermediária deve ter aproximadamente (2i+1), onde i=número de entradas;
4. Alguns autores defendem a ideia que em pequenas redes MLP de apenas 1 camada intermediária, o número de neurônios nesta camada deve ser igual à média geométrica dada pela multiplicação entre o número de neurônios da camada de entrada pelo número de neurônios da camada de saída.
5. Busquem outras dicas na literatura!

- Exemplo:

i=4 (entradas)
S=1 (saída)
LIPPMANN:
1 camada intermediária terá → 1.(4+1)= 5 neurônios, caso o fenômeno seja bastante complexo, faça variações desse número iniciando por ele

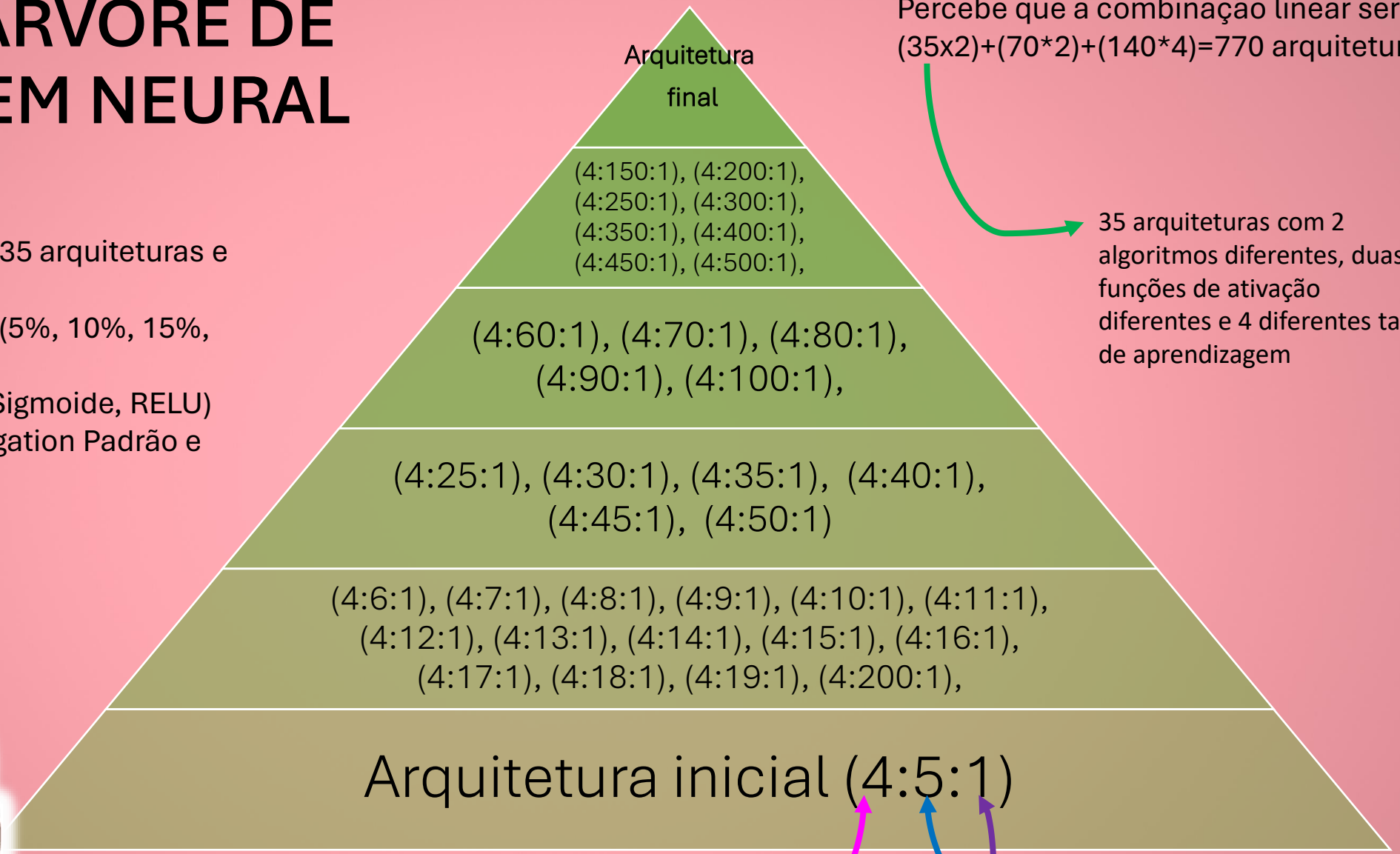
HECHT; NIELSEN:
1 camada intermediária terá→ (2.4+1)= 9 neurônios

Outros autores:
1 camada intermediária terá→ (4.1)= 4 neurônios

MINHA 1ª ÁRVORE DE MODELAGEM NEURAL

Dicas:
Experimente as primeiras 35 arquiteturas e
Varie os hiperparâmetros:

- Taxa de aprendizagem (5%, 10%, 15%, 20%)
- Funções de ativação (Sigmoide, RELU)
- Algoritmos backpropagation Padrão e Levemberg Marquardt



Percebe que a combinação linear seriam:
(35x2)+(70*2)+(140*4)=770 arquiteturas

35 arquiteturas com 2 algoritmos diferentes, duas funções de ativação diferentes e 4 diferentes taxas de aprendizagem

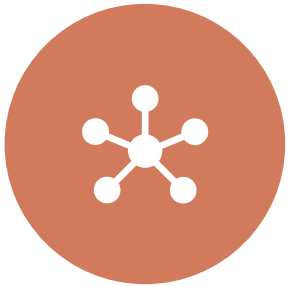
Exemplo:
i=4 (entradas)
S=1 (saída)
LIPPMANN:
1 camada intermediária → 1.(4+1)= 5

PULGA ATRÁS DA ORELHA?

- Perguntas?
- Dúvidas?

PONTOS CHAVE

O que é uma rede MLP?



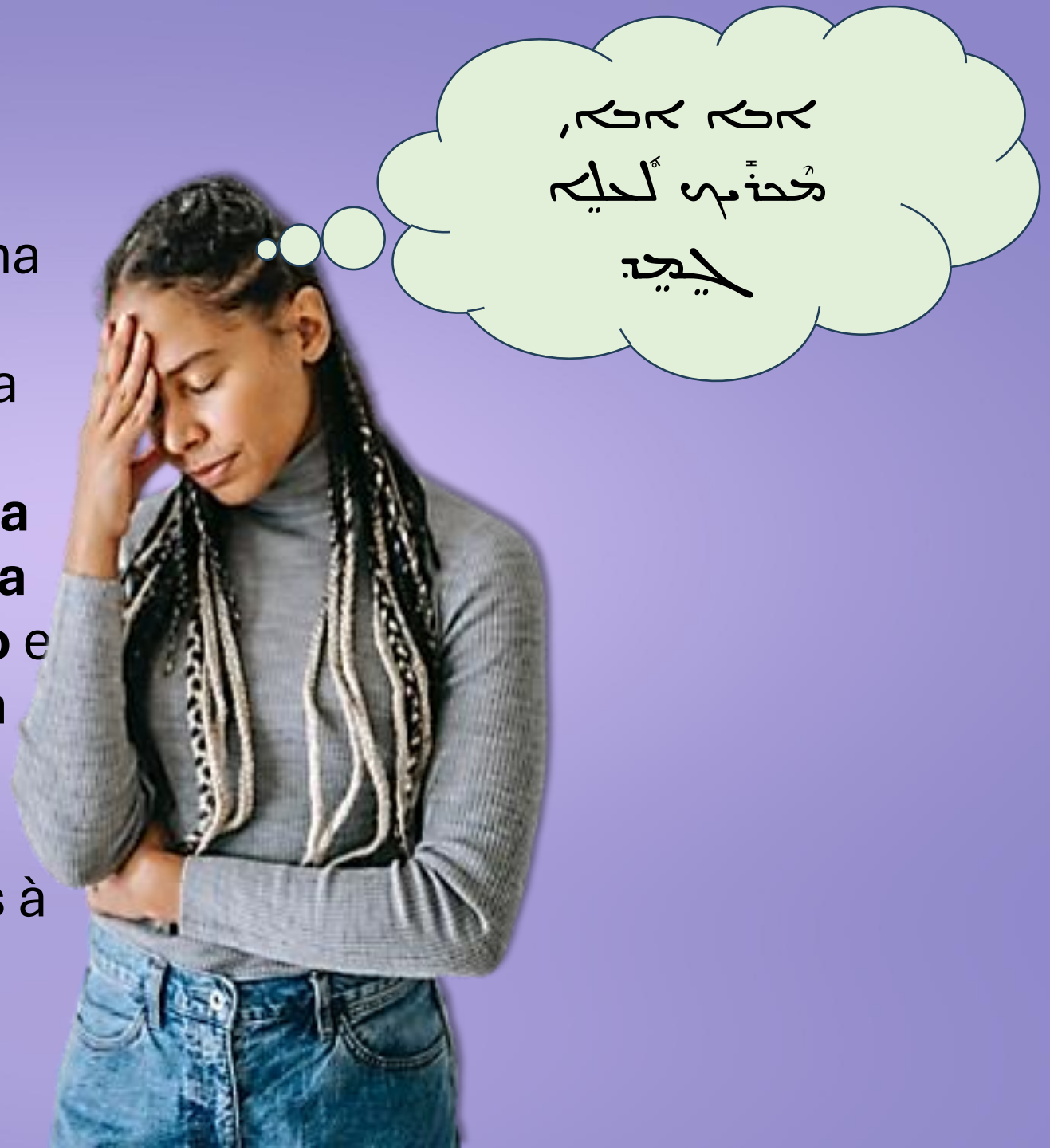
O que são os hiperparâmetros?

Como funciona um perceptron?

Como implementar uma árvore de modelagem?

ATIVIDADE 3:

Imaginando que você já tem alguma ideia de quais são as variáveis potencialmente envolvidas em sua modelagem: **entradas e saídas**. Solicita-se que você **construa uma árvore de modelagem neural para o fenômeno por você investigado** e descubra quantas redes você teria de testar para descobrir a combinação de arquitetura + hiperparâmetros, mais adequados à modelagem de seu fenômeno.




A SEGUIR, CENAS DO
PRÓXIMO
CAPÍTULO
O Algoritmo de treinamento backpropagation