Redes Neurais

SCC0633 e SCC5908 Processamento de Linguagem Natural

Prof. Thiago A. S. Pardo

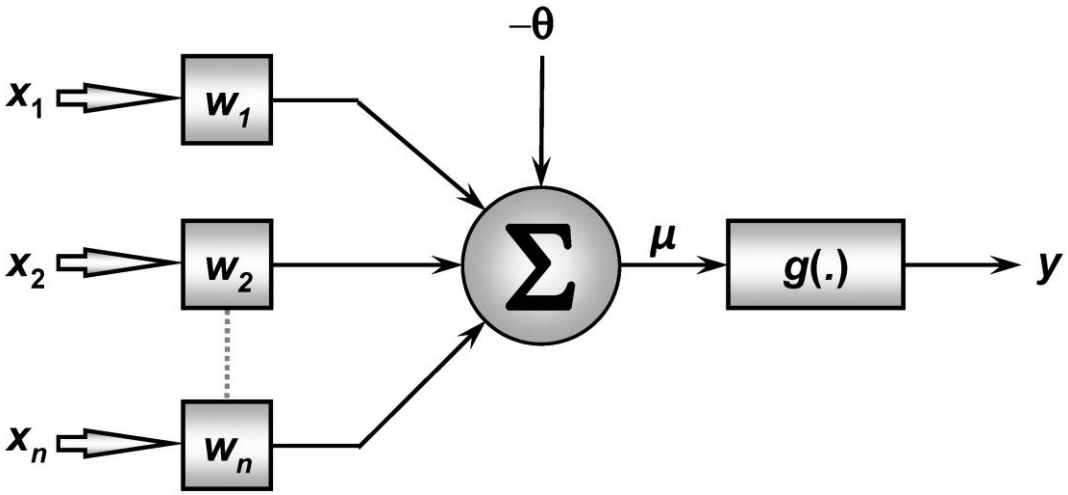
PAE: **Roney Lira de Sales Santos** roneysantos@usp.br

Redes Neurais Artificiais

- São modelos computacionais inspirados no sistema nervoso de seres vivos
  - Tentativa de modelar as capacidades de processamento de informação dos sistemas nervosos.
- Capacidade de aquisição e manutenção do **conhecimento**
- Conjunto de unidades de processamento
  - Neurônios artificiais
- Sinapses artificiais
  - Representadas por vetores ou matrizes de pesos sinápticos

Redes Neurais Artificiais

- Modelo de um neurônio artificial



- Sinais de entrada: $\{x_1, x_2, ..., x_n\}$
- Pesos sinápticos: $\{w_1, w_2, ..., w_n\}$
- Potencial de ativação μ: soma ponderada entre o combinador linear $\sum$ e o limiar de ativação θ
- Função de ativação: $g(.)$
- Saída: y

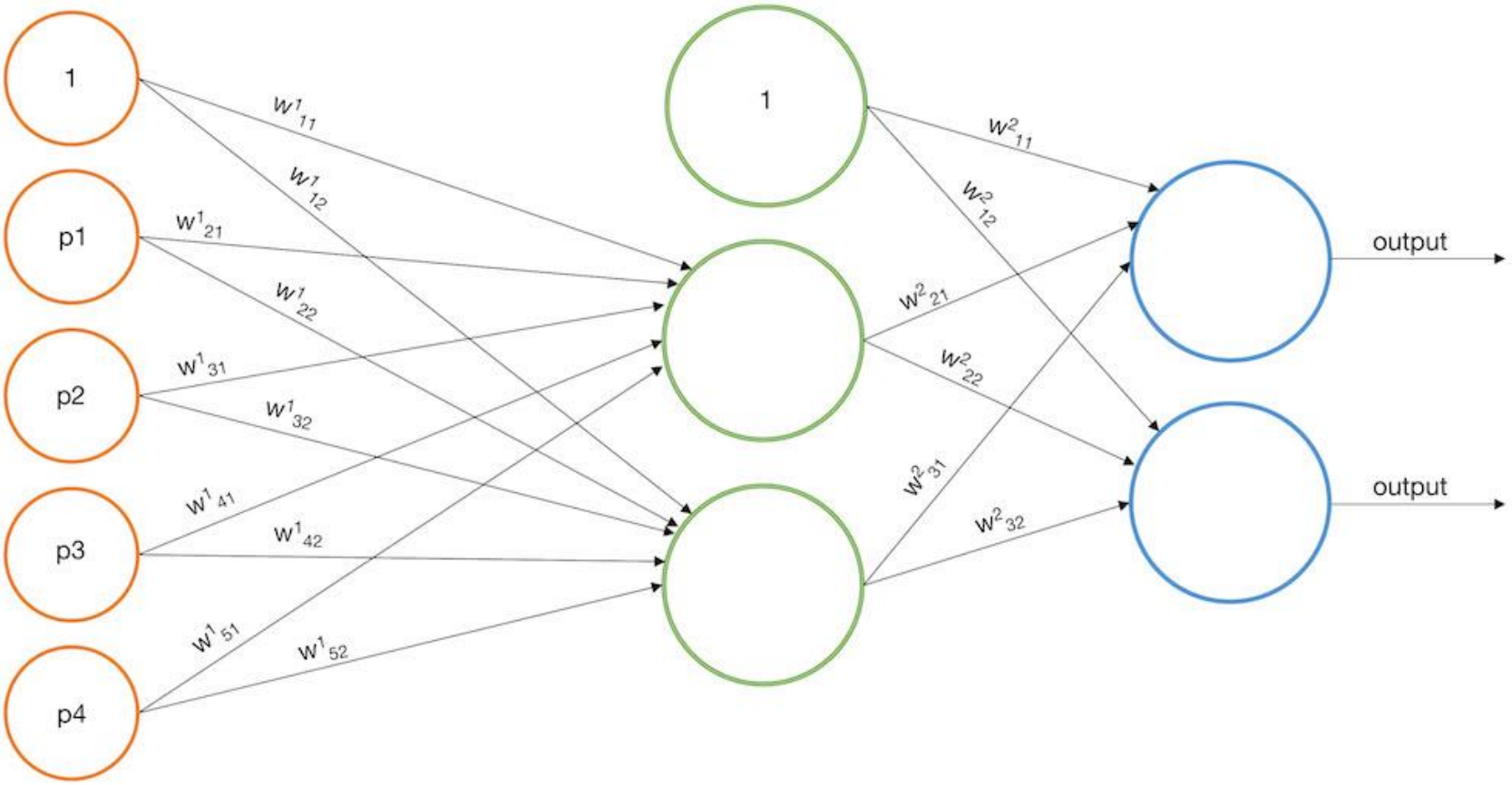
Redes Neurais Artificiais

- Uma **rede neural artificial** (RNA) é matematicamente representada por uma tripla

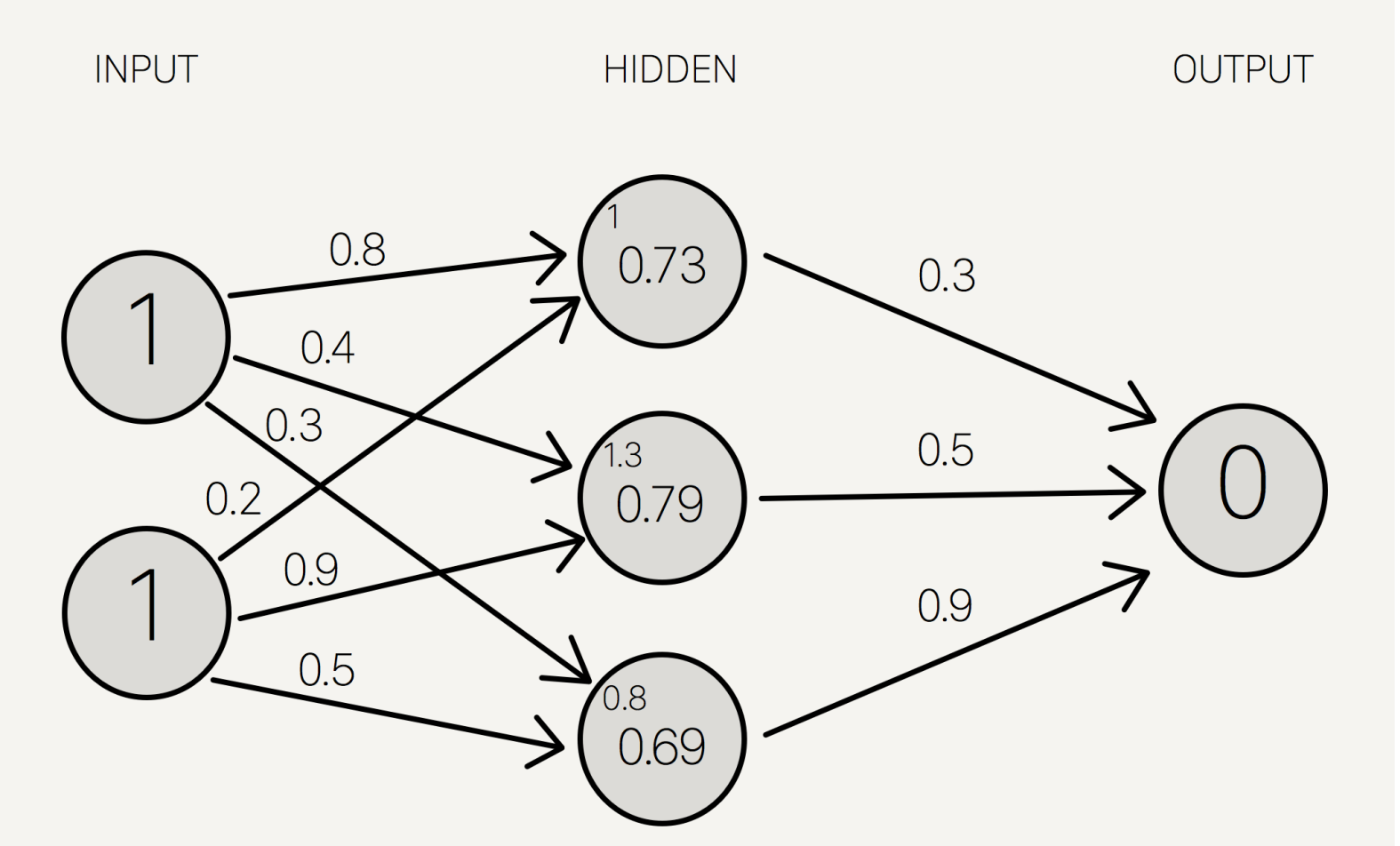
$$(\Omega, \gamma, w)$$

- onde:
  - Ω = conjunto de neurônios
  - γ = conjunto de conexões $\{(i, j) \mid i, j \in \mathbb{N}\}$
    - conexão entre o neurônio i e o neurônio j
  - w = conjunto de pesos sinápticos $\boldsymbol{w(i, j)}$
    - peso da conexão entre os neurônios i e j

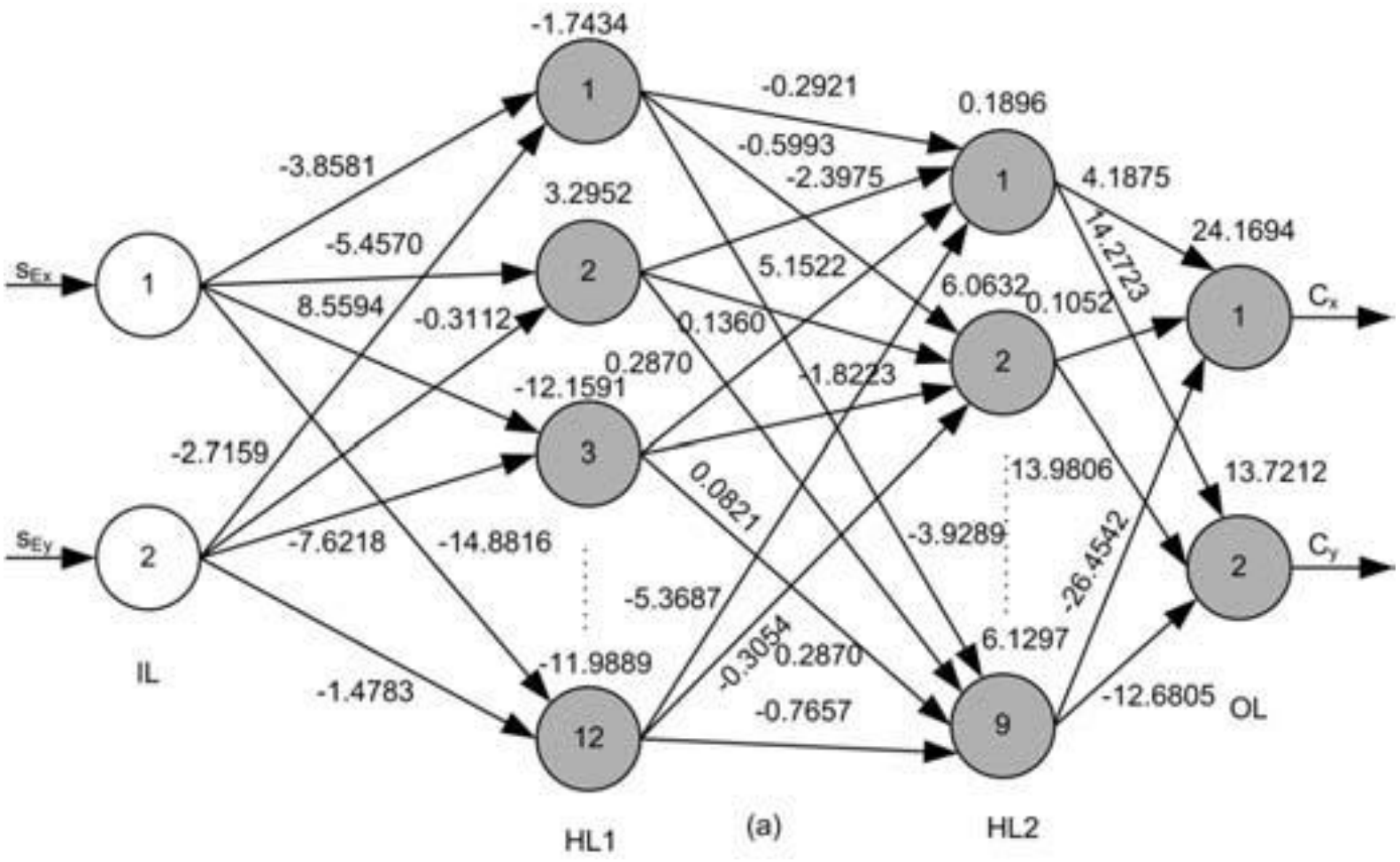
Redes Neurais Artificiais



Redes Neurais Artificiais

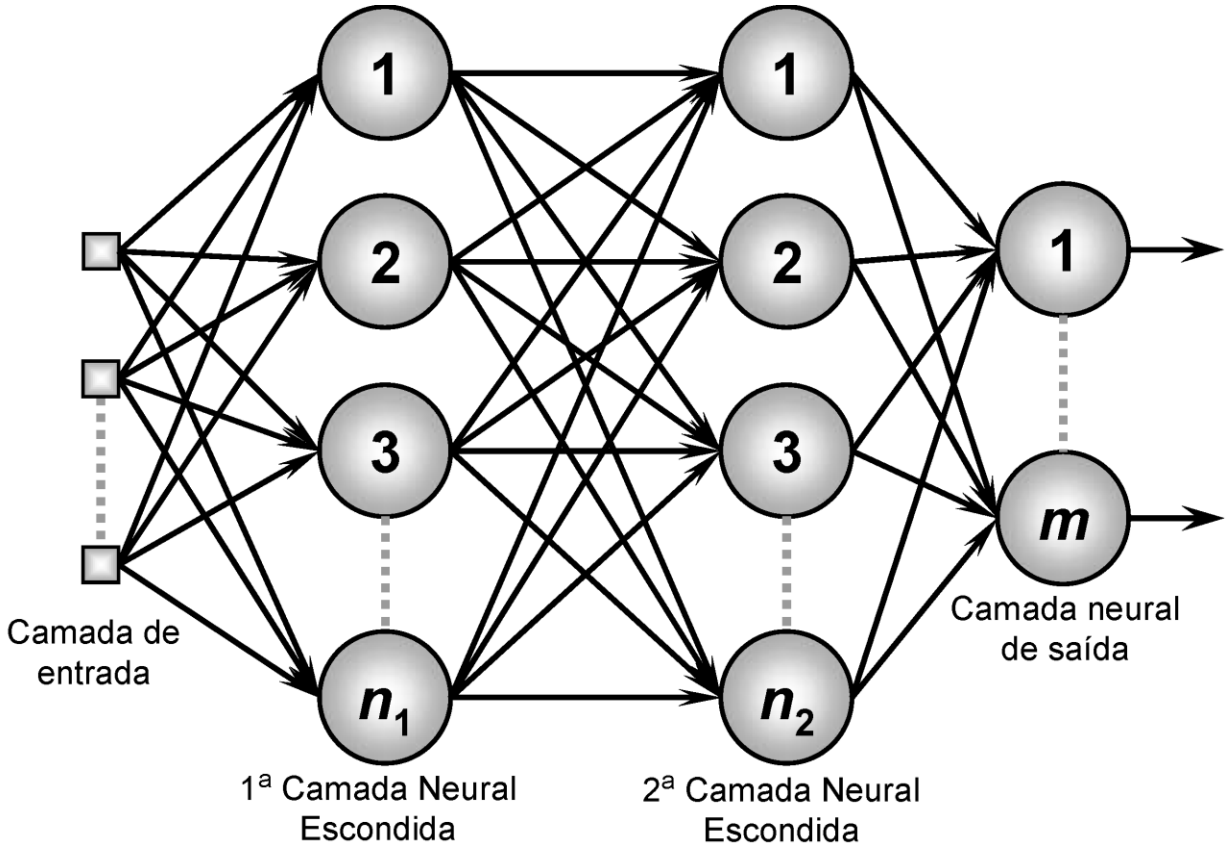


Redes Neurais Artificiais



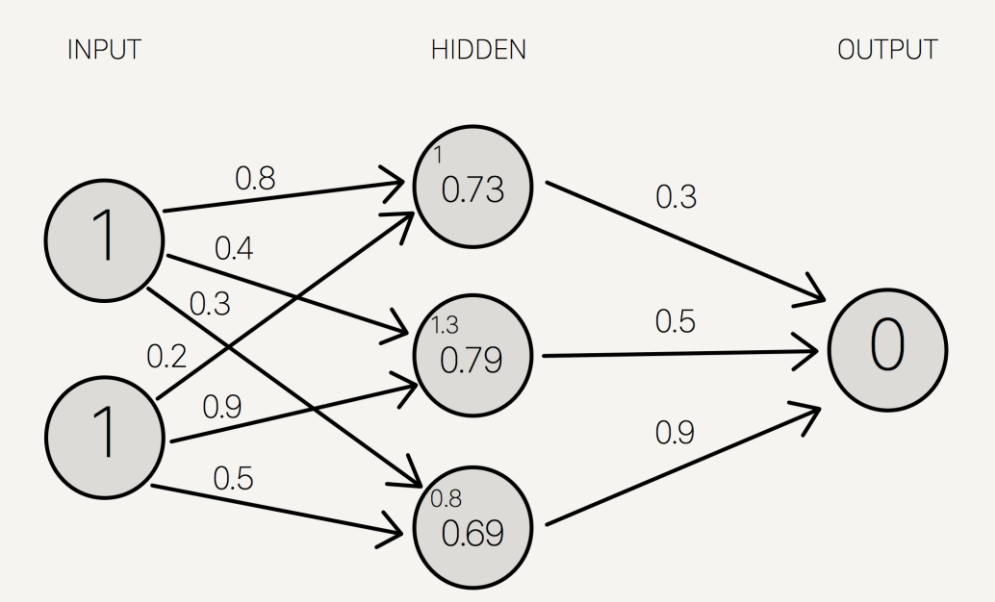
Multi-layer Perceptron (MLP)

- São redes neurais que apresentam uma ou mais camadas intermediárias de neurônios e uma camada de saída



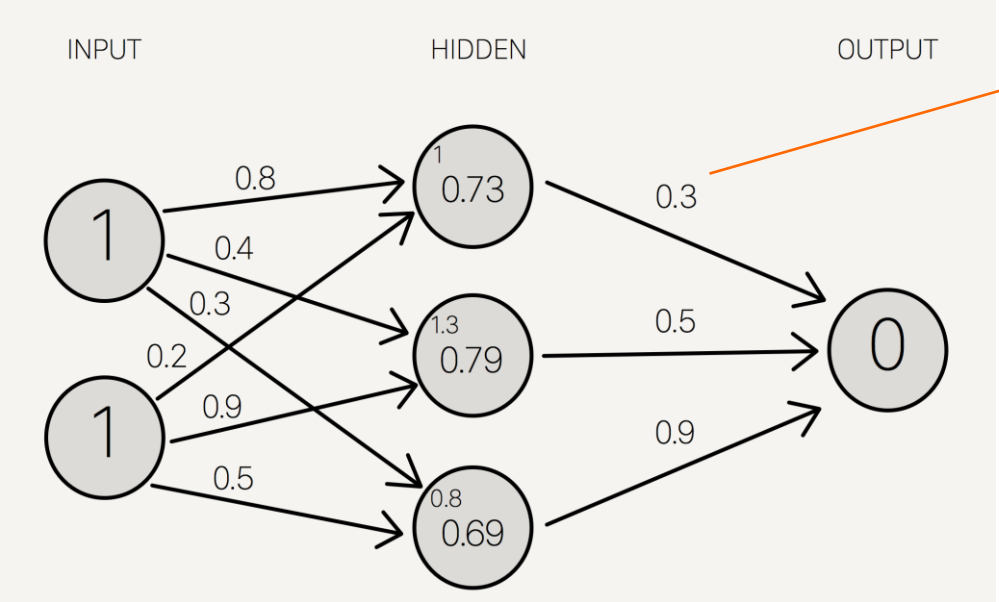
Multi-layer Perceptron (MLP)

- Cada neurônio realiza uma função específica e ela sempre depende dos neurônios da camada anterior que estão conectados



Multi-layer Perceptron (MLP)

- Cada neurônio realiza uma função específica e ela sempre depende dos neurônios da camada anterior que estão conectados



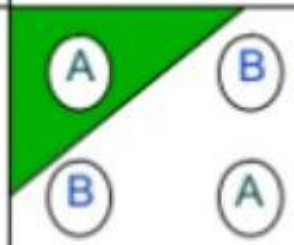
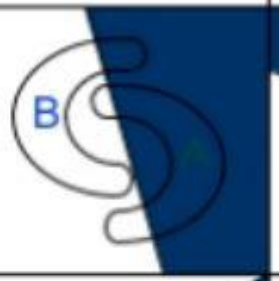
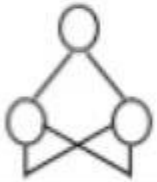
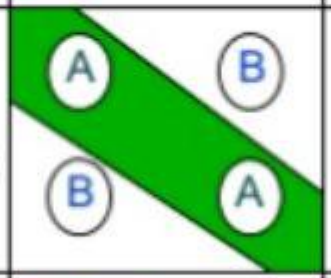
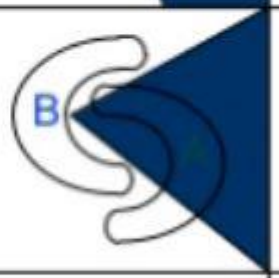
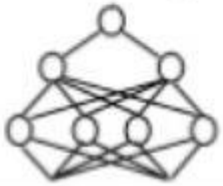
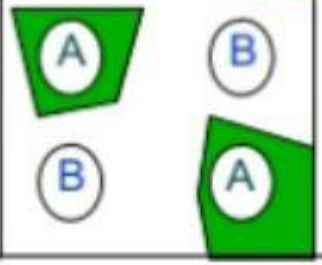
Esse valor 0.3 depende do processamento dos valores da camada anterior, por exemplo, aqueles 0.8 e 0.2!

Multi-layer Perceptron (MLP)

- Cada neurônio realiza uma função específica e ela sempre depende dos neurônios da camada anterior que estão conectados

- E cada camada escondida forma uma reta que consegue dividir o espaço das amostras para a possível classificação.

Multi-layer Perceptron (MLP)

Structure	Types of Decision Regions	Exclusive-OR Problem	Classes with Meshed regions	Most General Region Shapes
Single-Layer 	Half Plane Bounded By Hyperplane			
Two-Layer 	Convex Open Or Closed Regions			
Three-Layer 	Arbitrary (Complexity Limited by No. of Nodes)			

Fonte: Anderson Soares. UFG. Disponível em https://ww2.inf.ufg.br/~anderson/deeplearning/20181/Aula%202%20-%20Multilayer%20Perceptron.pdf

Multi-layer Perceptron (MLP)

- Cada neurônio realiza uma função específica e ela sempre depende dos neurônios da camada anterior que estão conectados

- E cada camada escondida forma uma reta que consegue dividir o espaço das amostras para a possível classificação.

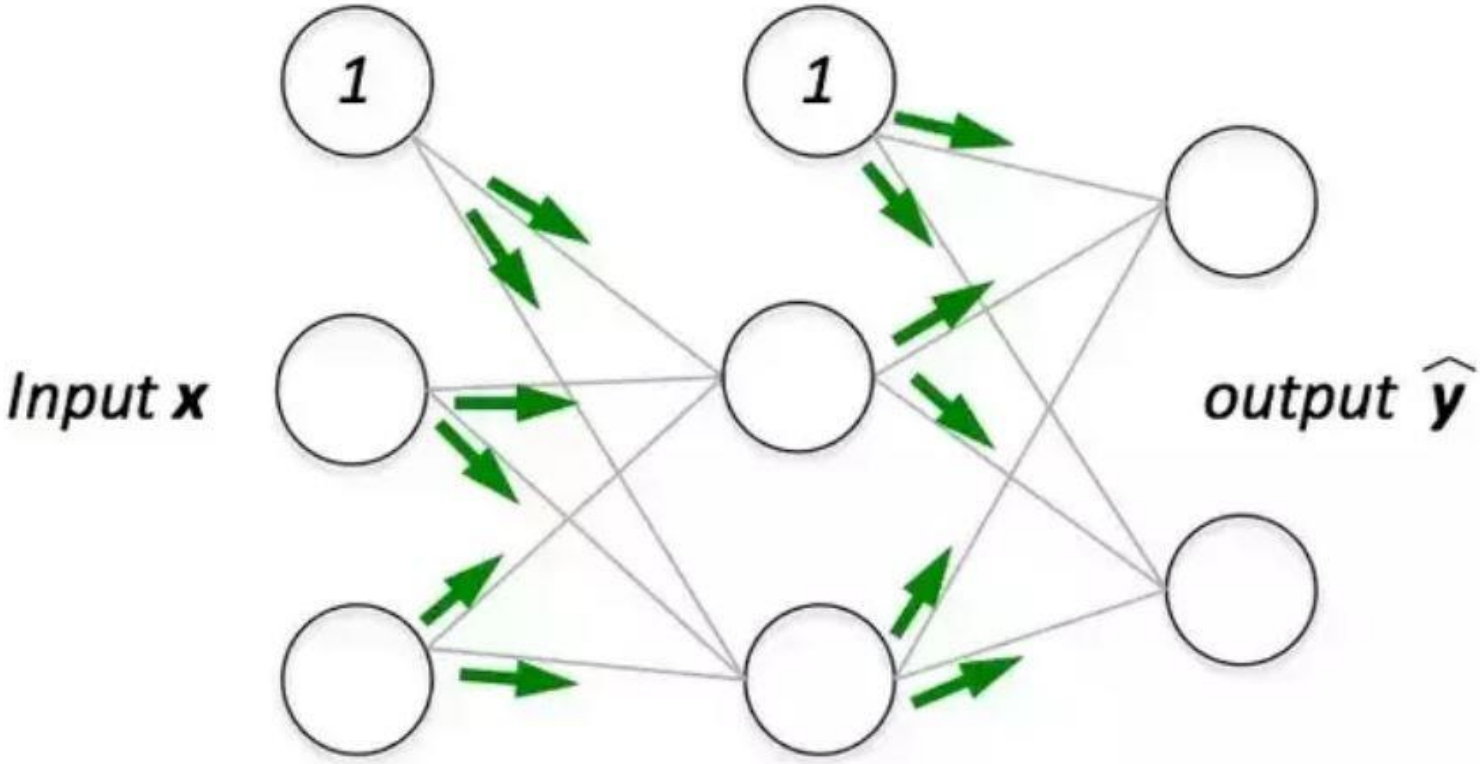
- À medida que o processamento avança de uma camada intermediária para a camada seguinte, o processamento realizado se torna mais complexo

Multi-layer Perceptron (MLP)

- Cada neurônio da camada de saída está associado a uma das classes presentes no conjunto de dados
- O erro cometido pela rede quando tenta classificar é peça importante para o treinamento da rede.
- Quando não há mais erro ou o erro está em um limite permitido, a rede estará treinada.
  - Então pode-se testar com dados nunca vistos!
  - Generalização, aprendizado.

O TREINAMENTO: *BACK-PROPAGATION*

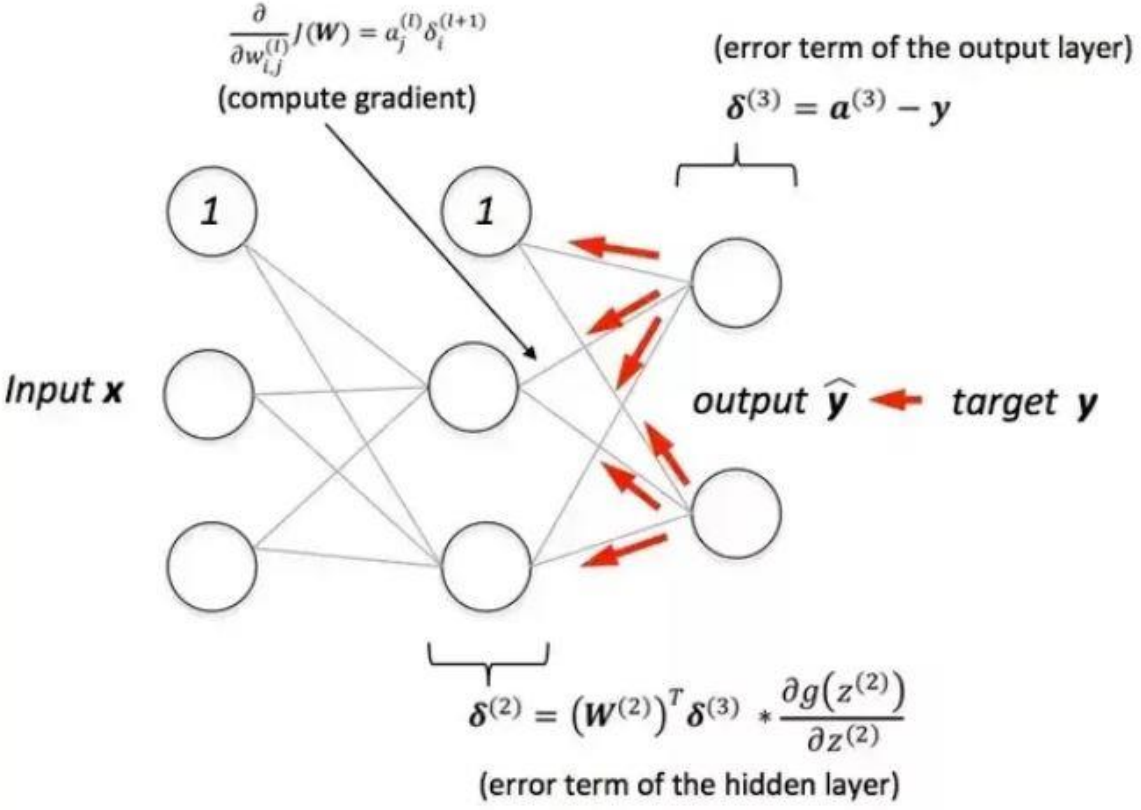
- Fase 1: **forward**
  - Os pesos são atualizados da entrada para a saída



Fonte: Anderson Soares. UFG. Disponível em https://ww2.inf.ufg.br/~anderson/deeplearning/20181/Aula%202%20-%20Multilayer%20Perceptron.pdf

O TREINAMENTO: *BACK-PROPAGATION*

- Fase 2: **backward**
  - Os pesos são atualizados de volta, da saída para a entrada, levando em consideração o erro e a taxa de aprendizado



Fonte: Anderson Soares. UFG. Disponível em https://ww2.inf.ufg.br/~anderson/deeplearning/20181/Aula%202%20-%20Multilayer%20Perceptron.pdf

O TREINAMENTO: *BACK-PROPAGATION*

- E quando para? Critérios de parada do treinamento
  - Número máximo de ciclos: épocas
  - Taxa máxima de erro

- Deve-se prestar atenção no chamado ***overfitting***!
  - Sobreajuste: quando o treinamento faz com que o modelo se adapte muito bem com os dados que estão sendo treinados, porém, não generaliza bem para dados novos.
    - ou seja, **DECOROU**!

O TREINAMENTO: *BACK-PROPAGATION*

- E quando para? Critérios de parada do treinamento
  - Número máximo de ciclos: épocas
  - Taxa máxima de erro

- Deve-se prestar atenção no chamado ***overfitting***!
  - Sobreajuste: quando o treinamento faz com que o modelo se adapte muito bem com os dados que estão sendo treinados, porém, não generaliza bem para dados novos.
    - ou seja, **DECOROU**!
  - Como resolve? Parte do conjunto de treinamento é separada, formando um conjunto de validação
  - Esse conjunto de validação é apresentado à rede a cada *n* épocas a fim de avaliar a rede.

O TREINAMENTO: *BACK-PROPAGATION*

- Deve-se prestar atenção no chamado ***overfitting***!
  - Então, toda vez que esse conjunto de validação for apresentado, haverá uma *taxa de erro relacionada*.
  - Caso esse erro comece a **subir**, é um indício que a rede parou de aprender e está se ajustando aos dados, característica do *overfitting*.
  - É nesse momento que se para o treinamento. Essa parada é chamada de *early stop*.
- Então a rede terá todos aqueles pesos em cada uma das ligações. Dizemos então que a rede está treinada!

Redes Neurais Artificiais: O MLP!

- Achou interessante? Quer aprender mais sobre? **Dicas de aprofundamento**!

- Videoaulas do Professor Rodrigo Mello
  - O que é MLP (13 min):
    - https://youtu.be/q3noPM9gcd8
  - Formulação da Camada de Saída da MLP (23 min):
    - https://youtu.be/cQ3x15UJsHA
  - Formulação da Camada Escondida da MLP (25 min):
    - https://youtu.be/nirNIh2Vmnk
- Slides-base desta nossa aula, com vários exemplos práticos e as formulações matemáticas:
  - https://ww2.inf.ufg.br/~anderson/deeplearning/20181/Aula%202%20-%20Multilayer%20Perceptron.pdf