



# Algoritmo Backpropagation

Profa Carine G. Webber

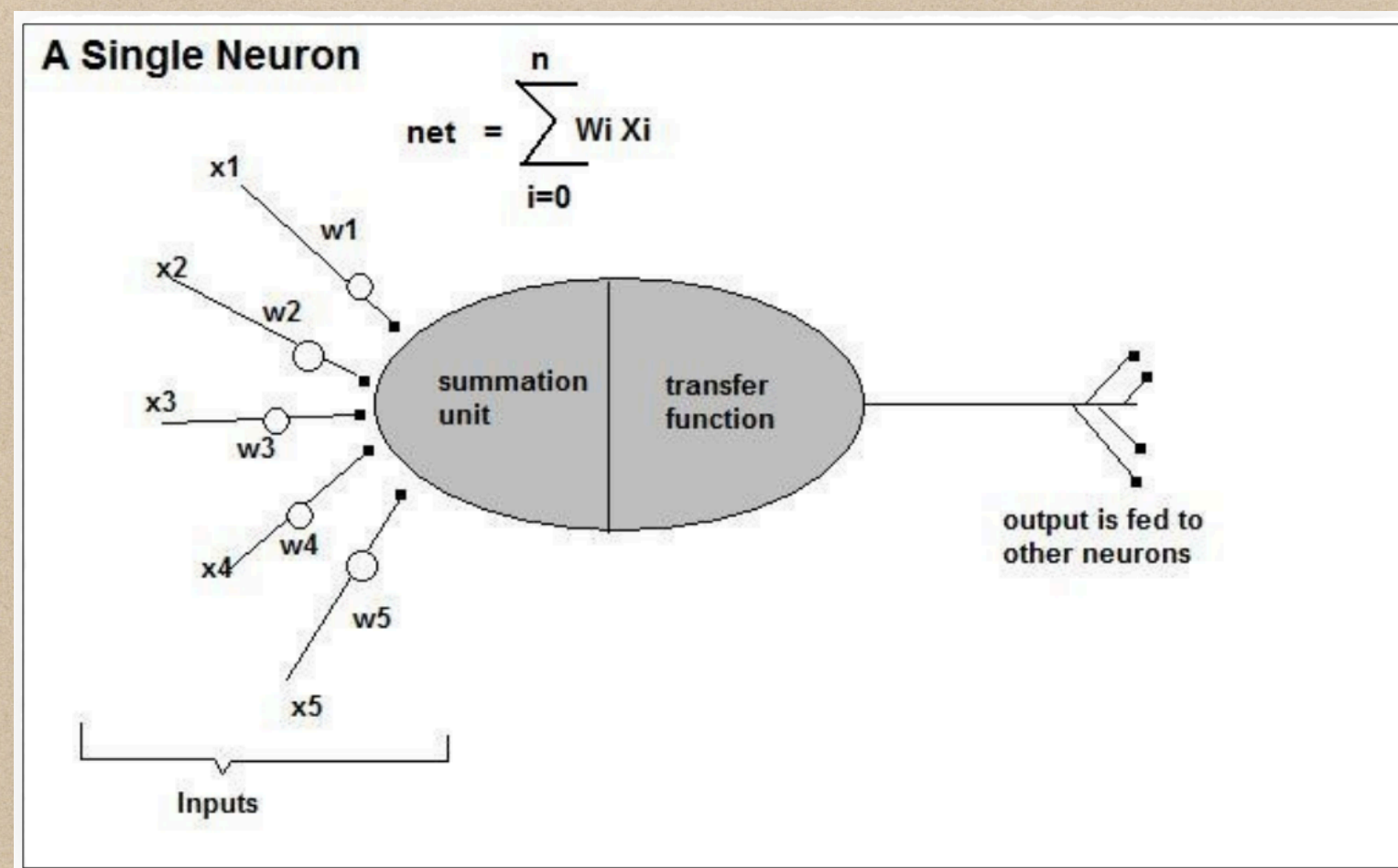


# Treínamento da Rede Neural

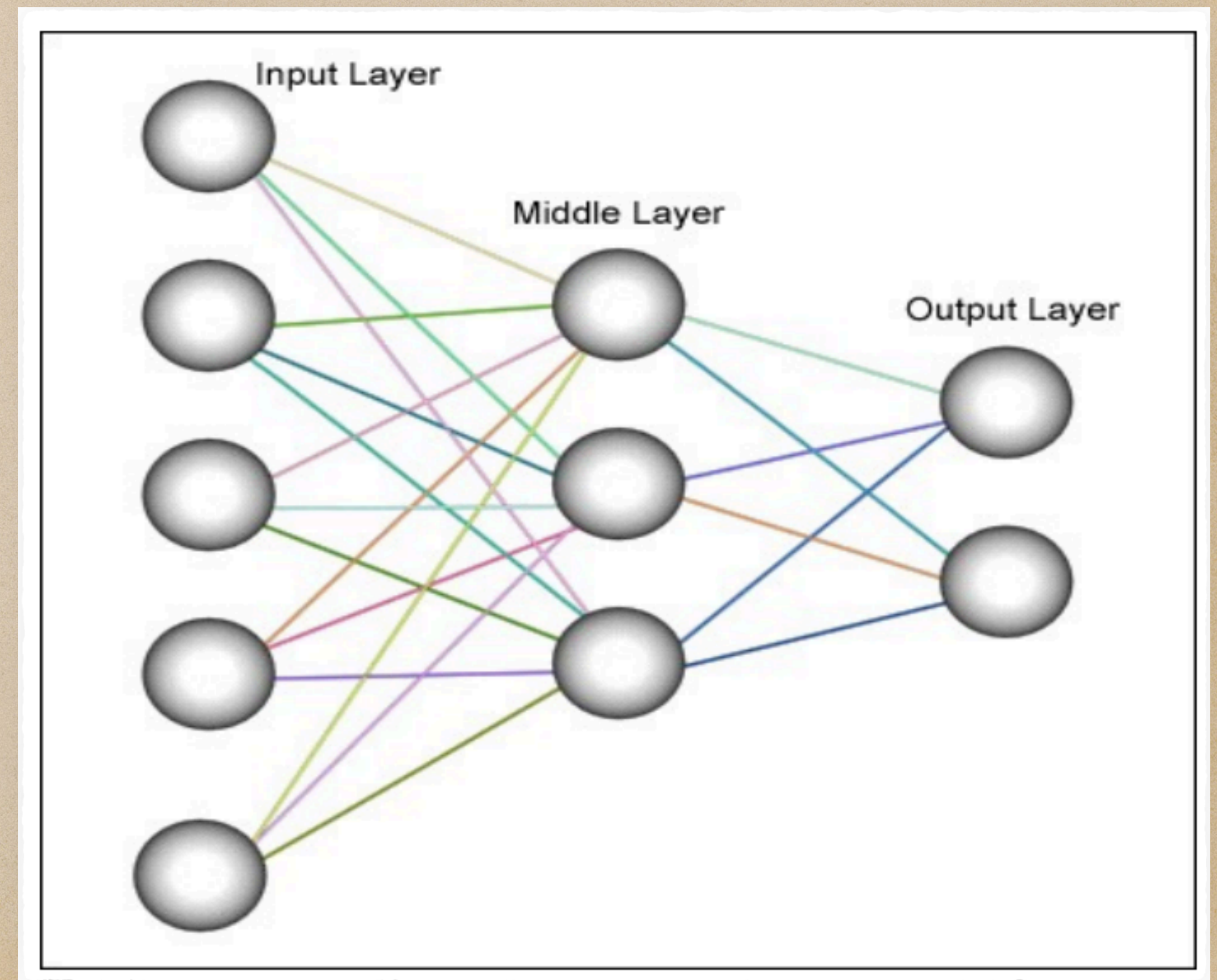
- ◆ Passo 1 - Propagar as entradas pelas camadas de neurônios da rede
- ◆ Passo 2 - Calcular os valores dos neurônios da camada de saída da rede
- ◆ Passo 3 - Calcular o erro de cada neurônio
- ◆ Passo 4 - Ajustar os pesos das conexões entre neurônios



# Passo 1 - Entradas

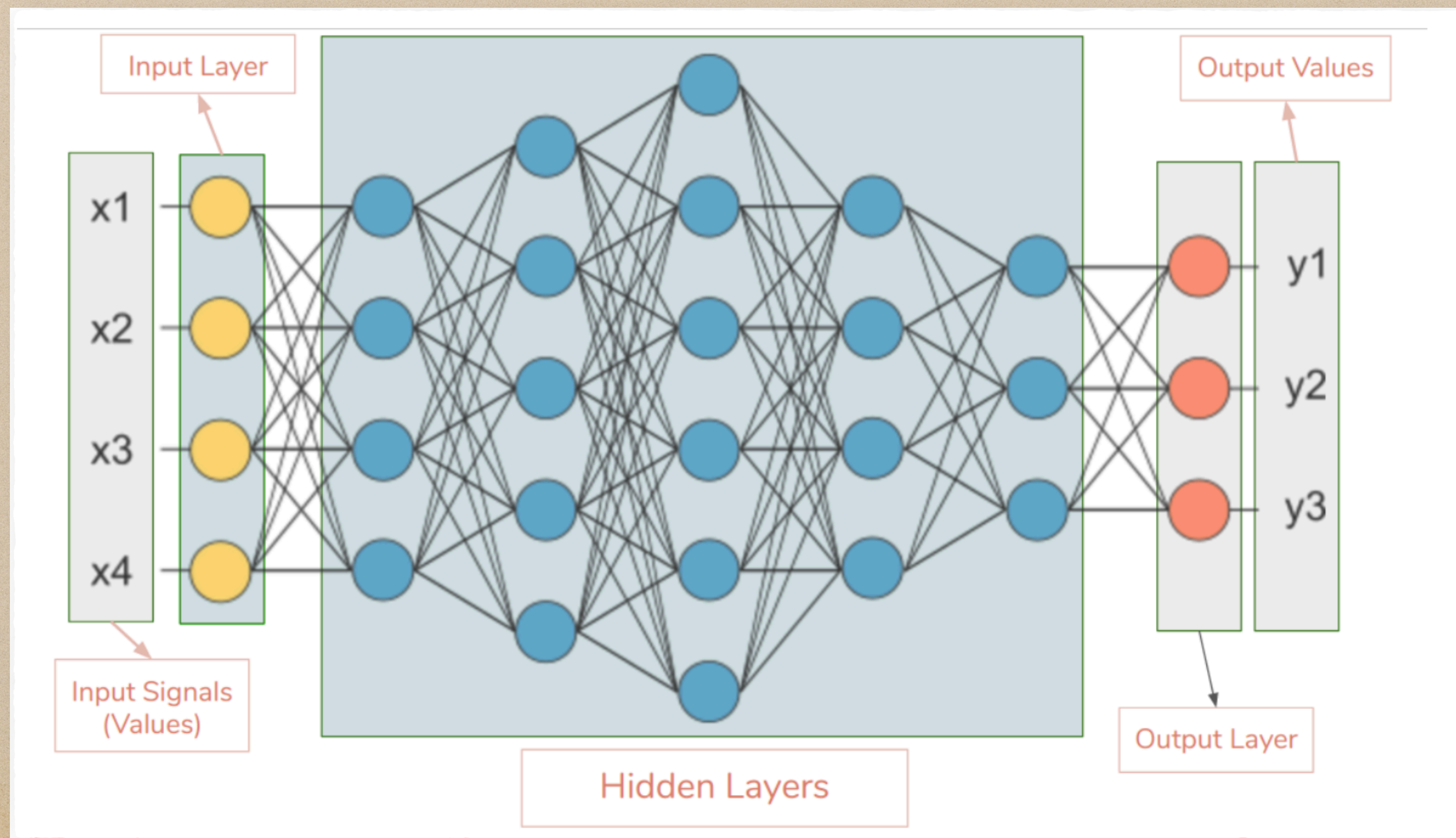


- ◆ Função Sigmoidal
- ◆ Gera um valor entre 0 e 1.
- ◆ saída  $\approx 1 / (1 + \text{Exp}(-\text{somatório}))$





# Passo 2 - Neurônios da Saída





# Passo 3 - Cálculo do Erro

- ◆ Camada de Saída
- ◆  $\text{FatorErro de neurónio na camada de saída} = \text{Saída Esperada} - \text{Saída Atual Neurónio}$
- ◆  $\text{Neurónio.Erro} = \text{Neurónio.Saída} * (1 - \text{Neurónio.Saída}) * \text{FatorErro}$



# Passo 3 - Cálculo do Erro

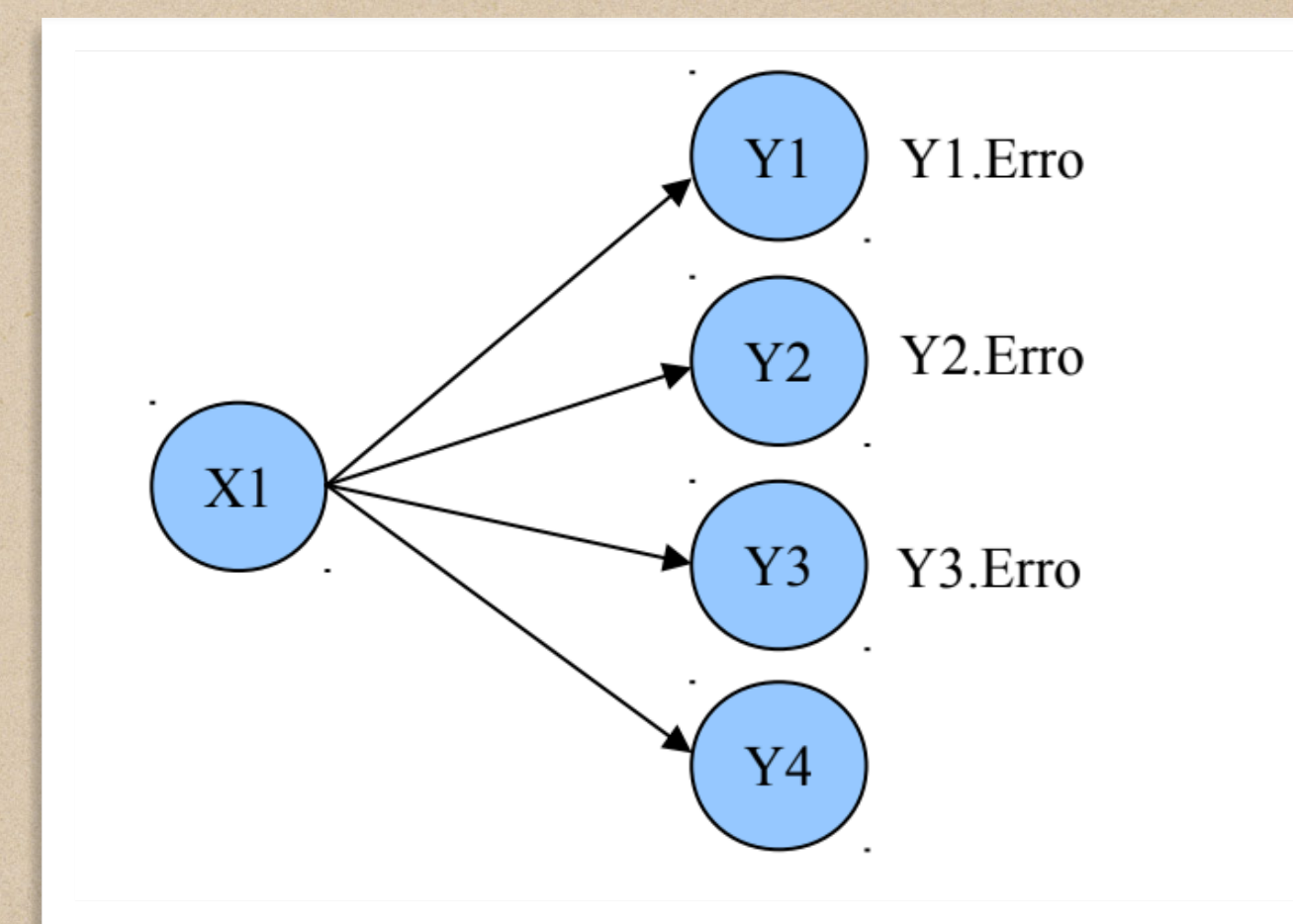
- ◆ Camadas Intermediárias

FatorErroX=0;

For each neurónio Y connected to X

$$\text{FatorErroX} = \text{FatorErroX} + (\text{Y.Erro} * \text{Peso da Conexão entre X e Y})$$

$$\text{NeurónioX.Erro} = \text{NeurónioX.Saída} * (1 - \text{NeurónioX.Saída}) * \text{FatorErroX}$$





# Passo 4 - Ajuste dos pesos

- ◆ Constante momentum = 0.9
- ◆ Constate taxa\_aprendizagem = 0.2
- ◆ Aplique a seguinte função para cada peso:
- ◆ 
$$\text{Novo\_peso} = \text{Peso\_anterior} * \text{momentum} + \text{Taxa\_aprendizagem} * \text{Saída\_neurônio\_anterior} * \text{Erro\_neurônio\_posterior}$$



# Repita o processo

- ◆ Treine cada instância do conjunto em sequência.
- ◆ Conte o número de épocas de treinamento.
- ◆ Verifique o erro gerado nos neurônios da camada de saída, e use esses valores como condições de parada do treinamento.
- ◆ Observe durante algumas épocas se o sistema converge (quando os pesos se estabilizam). Então, passe para a etapa de testes.