Machine Learning e MLP na Classificação

Ana Luiza Ramos, Ana Luiza Moraes, Beatriz Vêncio,
Maria Fernanda Prado, Arthur de Sousa

CEUB

Fundamentos da MLP

Trata-se de uma rede neural com uma ou mais camadas ocultas com um número indeterminado de neurônios. A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias.

Versatilidade da MLP

- **Adaptação a Complexidade**: MLP é capaz de lidar com problemas de classificação simples e complexos, ajustando-se dinamicamente à complexidade dos dados.

- **Multiclasses**: Enfatiza a habilidade da MLP de lidar com problemas de classificação multiclasses, tornando-a adequada para diversas aplicações.

MLP Architecture:

Camada de entrada, camada oculta e camada de saída.

- Input layer
  - Um neurônio por variável problema: mesma quantidade de variáveis que compõem uma entrada do problema.

Exemplo: RN para classificar nível de colesterol.

— Variáveis por entrada: sexo, idade, IMC, classe social, alimentação de risco

— Neurônios sensoriais: 5

2. Output Layer

- Um neurônio por classe: o valor de saída de cada neurônio será a probabilidade de a entrada pertencer aquela classe.

Exemplo: classe de colesterol > normal, alto, médio.

— Neurônios na camada de saída: 3

3.Hidden Layer

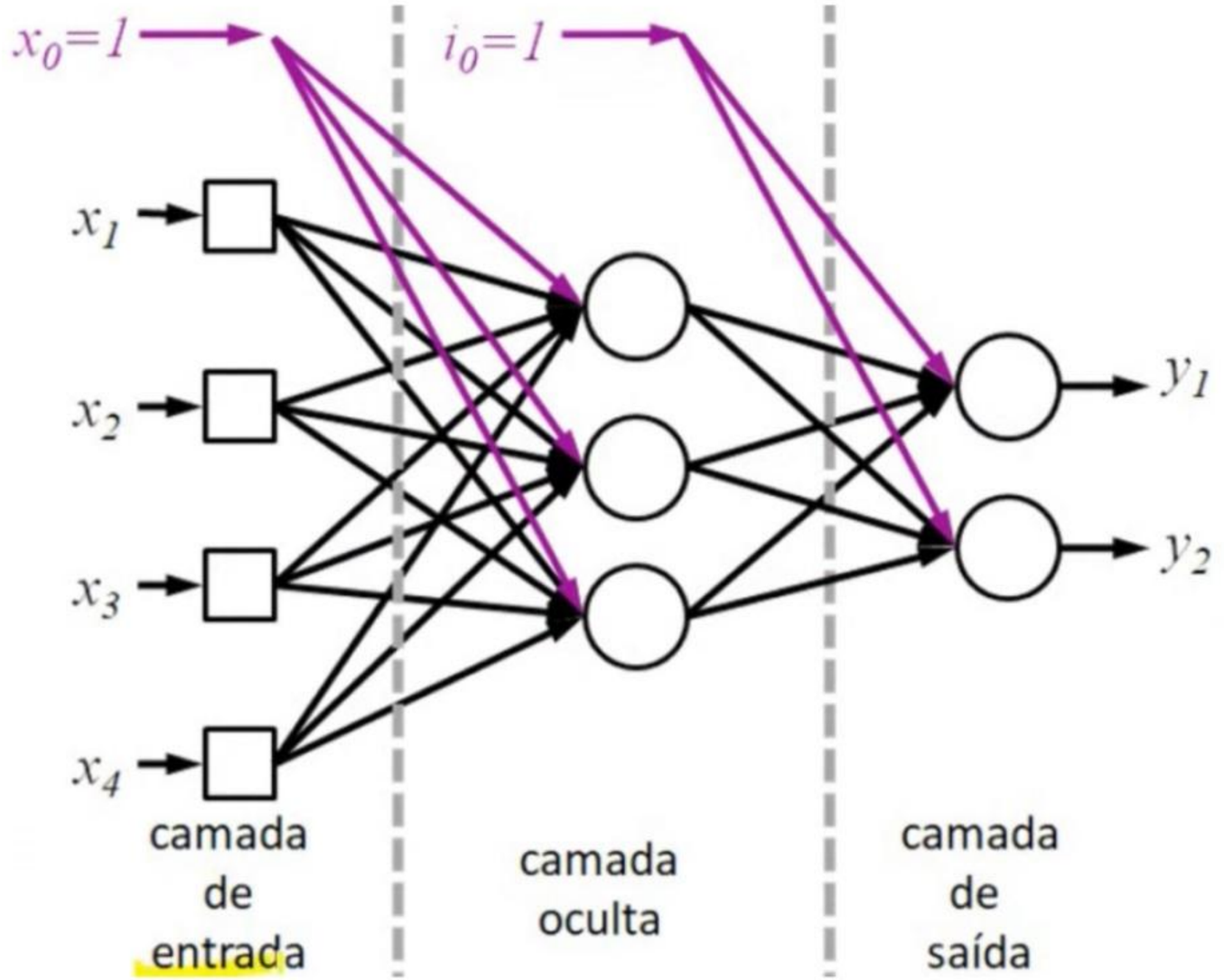
- Seguir o princípio da Navalha de Ockha: Se houver opções equivalentes, preferir a mais simples

—Treinar RNs com diferentes quantidades de camadas ocultas e neurônios nestas camadas.

—Verificar a capacidade de generalização (taxa de acertos) de cada RN.

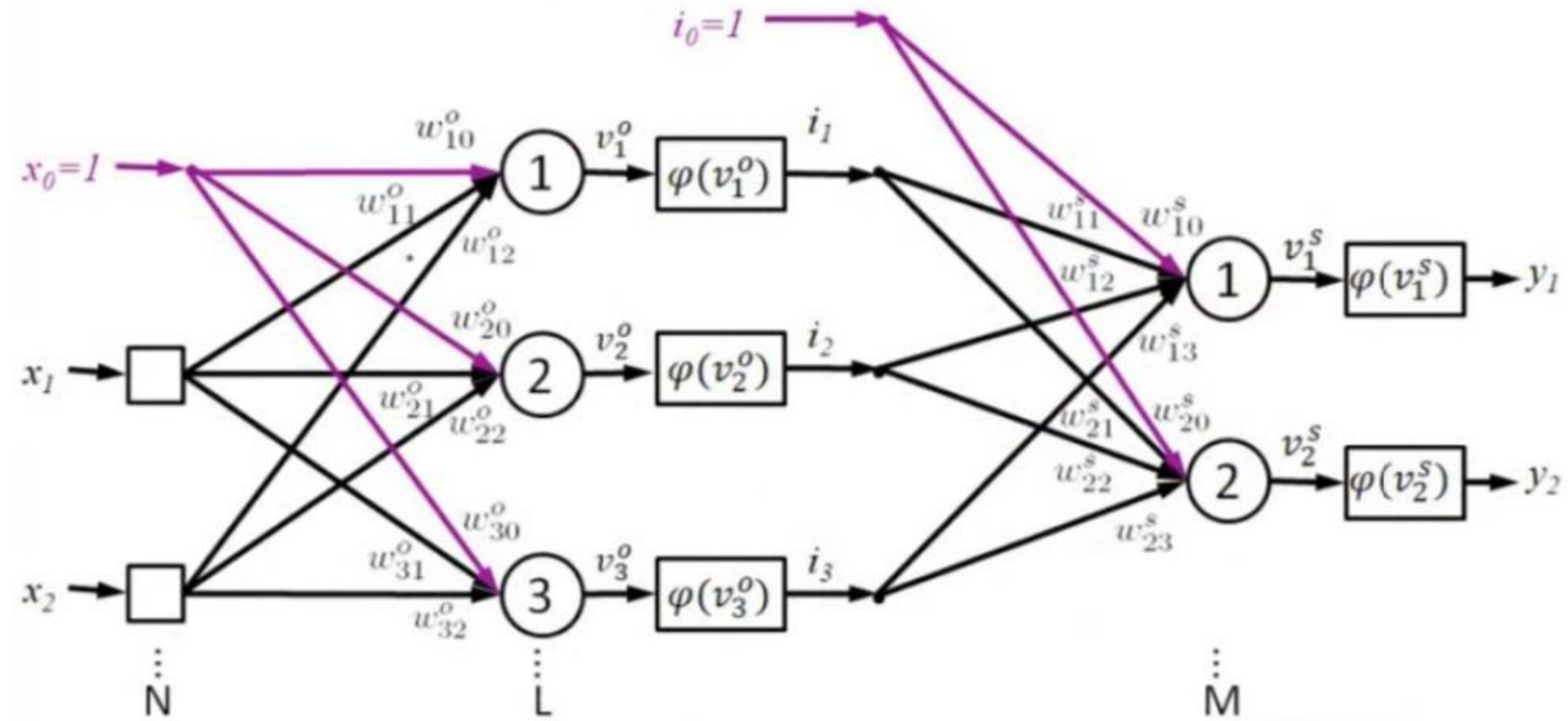
— Adotar a mais simples

Exemplos de MLP:



Propagação das entradas no MLP

Os sinais de entradas são propagados para frente, por toda a rede, camada por camada.



Função de ativação Sigmoid

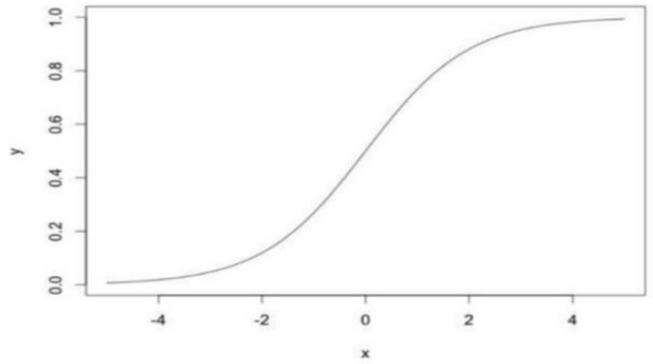
A função de ativação sigmoid é comumente utilizada por redes neurais com propagação positiva (Feedforward) que precisam ter como saída apenas números positivos, em redes neurais multicamadas e em outras redes com sinais contínuos.

Apesar de seu grande uso, a função de ativação tangente hiperbólica é geralmente uma escolha mais adequada.

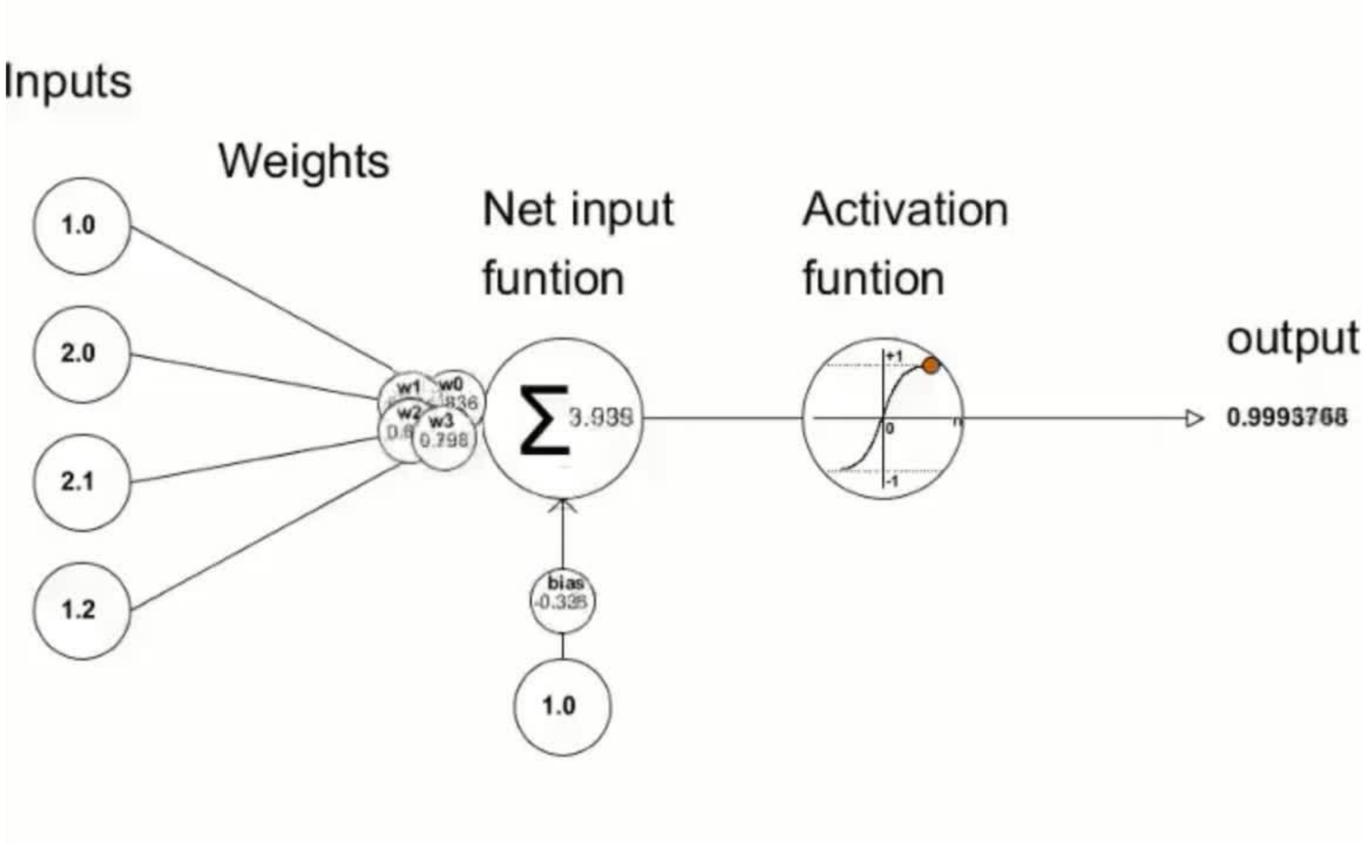
Equação:

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

Gráfico:



Backpropagation



Na etapa final, um ***error*** é calculado. Nosso objetivo é reduzir o erro, e a única opção que temos é ajustar os pesos. Para isso, utilizamos um algoritmo chamado "***backpropagation***."

Diferentemente do Perceptron e Adaline, onde existe apenas um único neurônio de saída {y}, a MLP pode relacionar o conhecimento a vários neurônios de saída.

O algoritmo de aprendizado (backpropagation), é composto de 4 passos:

1º Passo: Inicialização

- Atribuir valores aleatórios para os pesos e limites
- Escolha dos valores iniciais influencia o comportamento da rede
- Na ausência de conhecimento prévio os pesos e limites devem ter valores iniciais aleatórios e pequenos uniformemente distribuídos

2º Passo: Ativação

- Calcular os valores dos neurônios da camada oculta
- Calcular os valores dos neurônios da camada de saída

3º Passo: Treinar os Pesos

- Calcular os erros dos neurônios das camadas de saída e oculta
- Calcular a correção dos pesos
- Atualizar os pesos dos neurônios das camadas de saída e oculta

4º Passo: Iteração

- Repetir o processo a partir do passo 2 até que satisfaça o critério de erro

Fontes Bibliográficas

https://www.youtube.com/playlist?list=PLPrYObOisEDGCe-bt-x2tbUjh-RgM2FL0

https://www.youtube.com/watch?v=Ilg3gGewQ5U

https://www.linkedin.com/feed/update/urn:li:activity:7134239751540232193?utm_source=share&utm_medium=member_ios

https://scikit-learn.org/stable/modules/neural_networks_supervised.html#algorithms

https://www.kaggle.com/code/kfurudate/multi-layer-perceptron-mlp-network