



Aprendizado Automático



João Paulo Pordeus Gomes

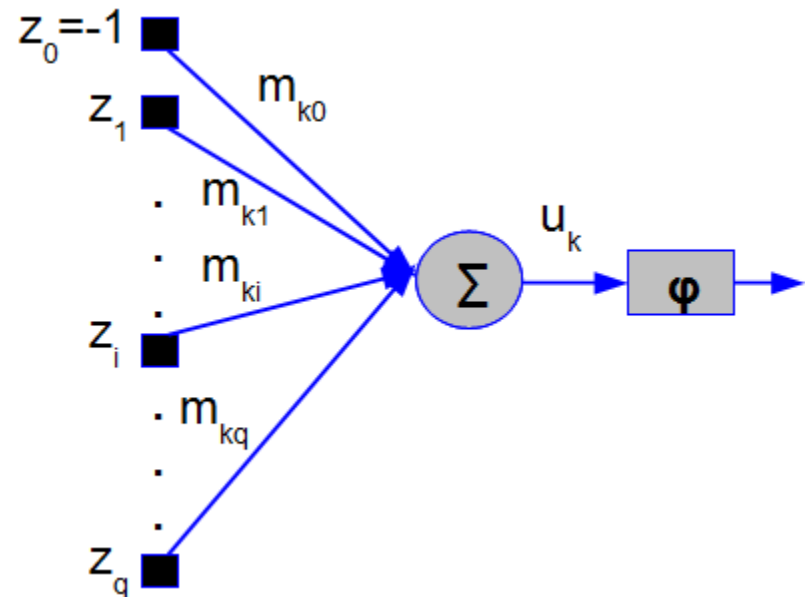
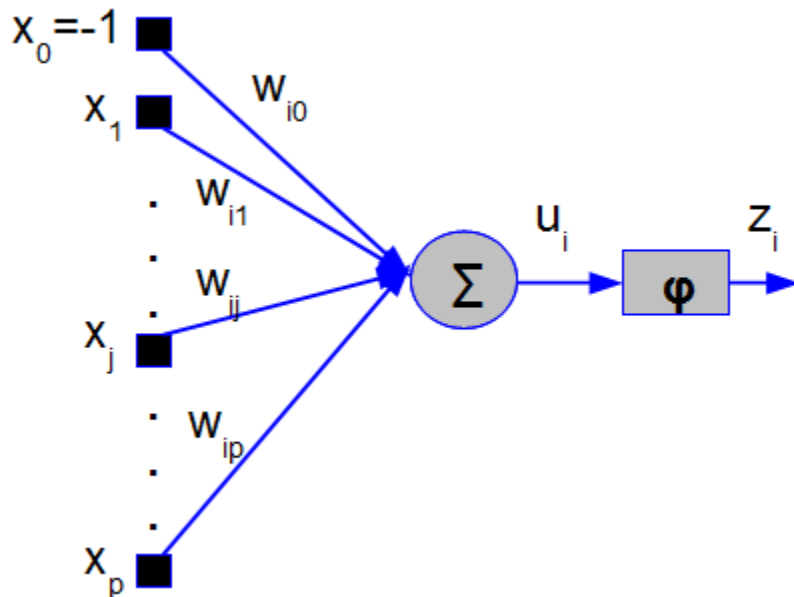


Aspectos Práticos

Redes Neurais

Redes MLP e Backpropagation

- ▶ $w_{ij} = w_{ij} + \alpha \delta_i x_j$
- ▶ $m_{ki} = m_{ki} + \alpha \delta_k z_i$
- ▶ Gradiente local
 - ▶ $\delta_k = e_k \varphi'(u_k)$
 - ▶ $\delta_i = \varphi'(u_i) \sum_{k=1}^r \delta_k m_{ki}$



Definição da Arquitetura

Redes MLP

Número de camadas ocultas



Número de camadas ocultas

- ▶ Sem camadas ocultas

Número de camadas ocultas

- ▶ Sem camadas ocultas
 - ▶ Problemas linearmente separáveis



Número de camadas ocultas

- ▶ Sem camadas ocultas
 - ▶ Problemas linearmente separáveis
- ▶ Uma camada oculta



Número de camadas ocultas

- ▶ Sem camadas ocultas
 - ▶ Problemas linearmente separáveis
- ▶ Uma camada oculta
 - ▶ Problemas não linearmente separáveis (função contínua)



Número de camadas ocultas

- ▶ Sem camadas ocultas
 - ▶ Problemas linearmente separáveis
- ▶ Uma camada oculta
 - ▶ Problemas não linearmente separáveis (função contínua)
- ▶ Duas camadas ocultas



Número de camadas ocultas

- ▶ Sem camadas ocultas
 - ▶ Problemas linearmente separáveis
- ▶ Uma camada oculta
 - ▶ Problemas não linearmente separáveis (função contínua)
- ▶ Duas camadas ocultas
 - ▶ Problemas não linearmente separáveis (qualquer função)



Número de Neurônios

- ▶ Camada de saída
 - ▶ Número de classes (problemas de classificação)



Número de Neurônios

- ▶ Camada de saída
 - ▶ Número de classes (problemas de classificação)
- ▶ Camadas ocultas
 - ▶ Conjunto de validação
 - ▶ Heurísticas



Função de Ativação

▶ Logística (0 e 1)

- ▶ $f(x) = \frac{1}{1+e^{-x}}$

- ▶ $f'(x) = f(x)[1 - f(x)]$

▶ Tangente Hiperbólica (-1 e 1)

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

- ▶ $f'(x) = 1 - f(x)^2$



Convergência

Redes MLP

Critério de Parada

- ▶ *Overfitting*



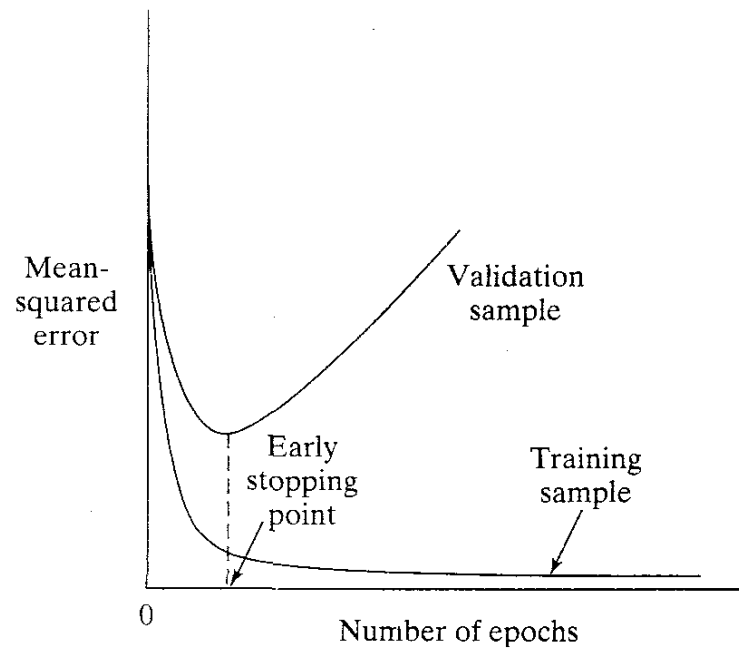
Critério de Parada

- ▶ *Overfitting*
- ▶ Solução
 - ▶ Conjunto de Validação



Critério de Parada

- ▶ *Overfitting*
- ▶ Solução
 - ▶ Conjunto de Validação



Técnicas para Aceleração da Convergência

- ▶ Normalização dos dados de entrada
- ▶ Taxa de aprendizagem variável
- ▶ Termo de momento



Normalização dos Dados de Entrada

- ▶ Dados normalizados entre 0 e 1 (-1 e 1)
- ▶ Dados normalizados para ter variância 1



Taxa de aprendizado

- ▶ Tamanho do passo de aprendizado



Taxa de aprendizado

- ▶ Tamanho do passo de aprendizado
- ▶ Estratégia
 - ▶ Diminuir a taxa a medida que o número de épocas aumenta
 - ▶ Decaimento linear
 - ▶ Decaimento exponencial



Fator de Momento

- ▶ Acelera o aprendizado em superfícies planas
- ▶ Ajuda a fugir de mínimos locais
- ▶ Penaliza mudanças bruscas de direção da busca



Fator de Momento

- ▶ Acelera o aprendizado em superfícies planas
- ▶ Ajuda a fugir de mínimos locais
- ▶ Penaliza mudanças bruscas de direção da busca
- ▶ Regra de Aprendizado
 - ▶ $w_{ij} = w_{ij} + \alpha \delta_i x_j + \beta (w_{ij} - w_{ij}^{anterior})$



Regressão com Redes MLP

Redes MLP

- ▶ Redes MLP também podem ser utilizadas para tarefa de regressão



Redes MLP

- ▶ Redes MLP também podem ser utilizadas para tarefa de regressão
- ▶ Utilizar a camada de saída sem função de ativação
 - ▶ Função de ativação linear



Redes MLP

- ▶ Redes MLP também podem ser utilizadas para tarefa de regressão
- ▶ Utilizar a camada de saída sem função de ativação
 - ▶ Função de ativação linear
 - ▶ Saída
 - $\bar{y}_k = \varphi(u_k)$
 - ▶ Regra de Aprendizado
 - $w_{ij} = w_{ij} + \alpha \delta_i x_j$
 - $m_{ki} = m_{ki} + \alpha \delta_k z_i$
 - $\delta_k = e_k \varphi'(u_k)$
 - $\delta_i = \varphi'(u_i) \sum_{k=1}^r \delta_k m_{ki}$



Redes MLP

- ▶ Redes MLP também podem ser utilizadas para tarefa de regressão
- ▶ Utilizar a camada de saída sem função de ativação
 - ▶ Função de ativação linear
 - ▶ Saída
 - $\overline{y}_k = u_k$
 - ▶ Regra de Aprendizado
 - $w_{ij} = w_{ij} + \alpha \delta_i x_j$
 - $m_{ki} = m_{ki} + \alpha \delta_k z_i$
 - $\delta_k = e_k$
 - $\delta_i = \varphi'(u_i) \sum_{k=1}^r \delta_k m_{ki}$





Dúvidas ?