Sistema Fiep

FIEP SESI SENAI IEL

Inteligência Artificial Aplicada

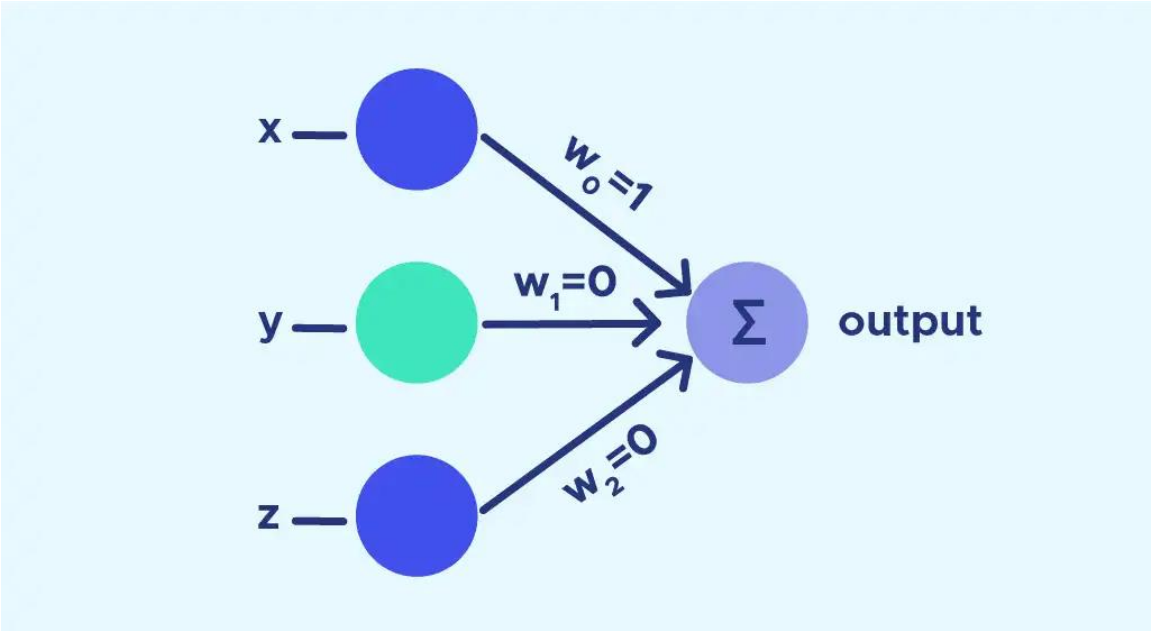
UniSenai PR– São José dos Pinhais

Categoria	Descrição	Algoritmos/técnicas	Exemplos de aplicação
Métodos de busca	**Encontram soluções em um espaço de estados.**	**BFS, A***	**GPS traçando a melhor rota.**
Raciocínio temporal	**Modelam eventos ao longo do tempo.**	**Cadeias de Markov, Redes Bayesianas**	**Previsão do tempo, reconhecimento de fala.**
Lógica fuzzy	**Lida com incertezas e valores intermediários.**	**Conjuntos fuzzy, Inferência fuzzy**	**Controle de temperatura em ar-condicionado.**
Modelos de aprendizado	Ajustam pesos para identificar padrões em dados.	Redes Neurais (Perceptron, MLP)	Reconhecimento facial, chatbots.

Redes Neurais (Perceptron)

Um dos primeiros modelos de rede neural (1958);

Modelo matemático inspirado em neurônios biológicos.



Elementos fundamentais

Composto por:

1. Entradas: vetor $x[x_1, x_2, \ldots, x_n]$
2. Pesos: vetor $w[w_1, w_2, \ldots, w_n]$
3. Soma ponderada: $y = f(x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b)$
4. *Bias* b (termo de ajuste)
5. Função de ativação (degrau): $f(u) = \begin{cases} 1, se\ u \geq 0 \\ 0, se\ u < 0 \end{cases}$

Elemento	Símbolo	Função
Entrada	$x_1, x_2, \dots, x_n$	Características do dado de entrada.
Pesos	$w_1, w_2, \dots, w_n$	Definem a importância de cada entrada.
Bias (viés)	b	Constante adicionada à soma ponderada; permite deslocar a decisão.
Soma ponderada	$u = \vec{x} * \vec{w} + b$	Combina entradas com seus pesos e o bias
Função de ativação	$f(u)$	Define a saída com base na soma: degrau

Elemento	Símbolo	Função
Saída prevista	y_{pred}	Resultado da função de ativação; a predição do modelo.
Saída real	y	Valor esperado nos dados de treinamento.
Erro	$\varepsilon = y - y_{pred}$	Diferença entre saída esperada e prevista
Taxa de aprendizado	π	Controla o tamanho do ajuste nos pesos e no bias.
Época	–	Rodadas de treinamento.

Funcionamento

1. Iniciar pesos e bias com 0 ou valores aleatórios;
2. Para cada registro:
   - Calcular a saída do perceptron (y_{pred});
   - Comparar com a saída desejada ($y_{real} - y_{pred}$);
   - Ajustar os pesos e o bias:
     - $w_i = w_i + n * erro * x_i$
     - $b = b + n * erro$
3. Repetir por várias épocas, até o erro ser zero ou muito baixo.

Perceptron - implementação