



Inteligência Artificial Aplicada



| 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. |

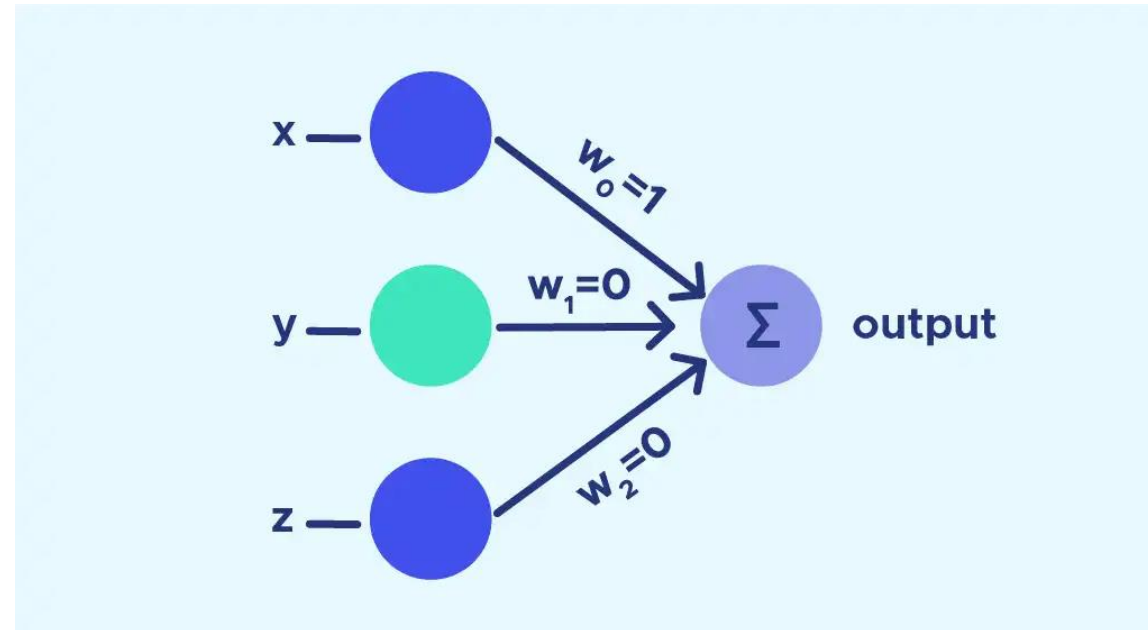


Redes Neurais (Perceptron)





Um dos primeiros modelos de rede neural (1958);
Modelo matemático inspirado em neurônios biológicos.





Composto por:

1. Entradas: vetor $x[x_1, x_2, \dots, x_n]$
2. Pesos: vetor $w[w_1, w_2, \dots, w_n]$
3. Soma ponderada: $y = f(x_1w_1 + x_2w_2 + \dots + x_nw_n + b)$
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 | η | Controla o tamanho do ajuste nos pesos e no bias. |
| Época | — | Rodadas de treinamento. |

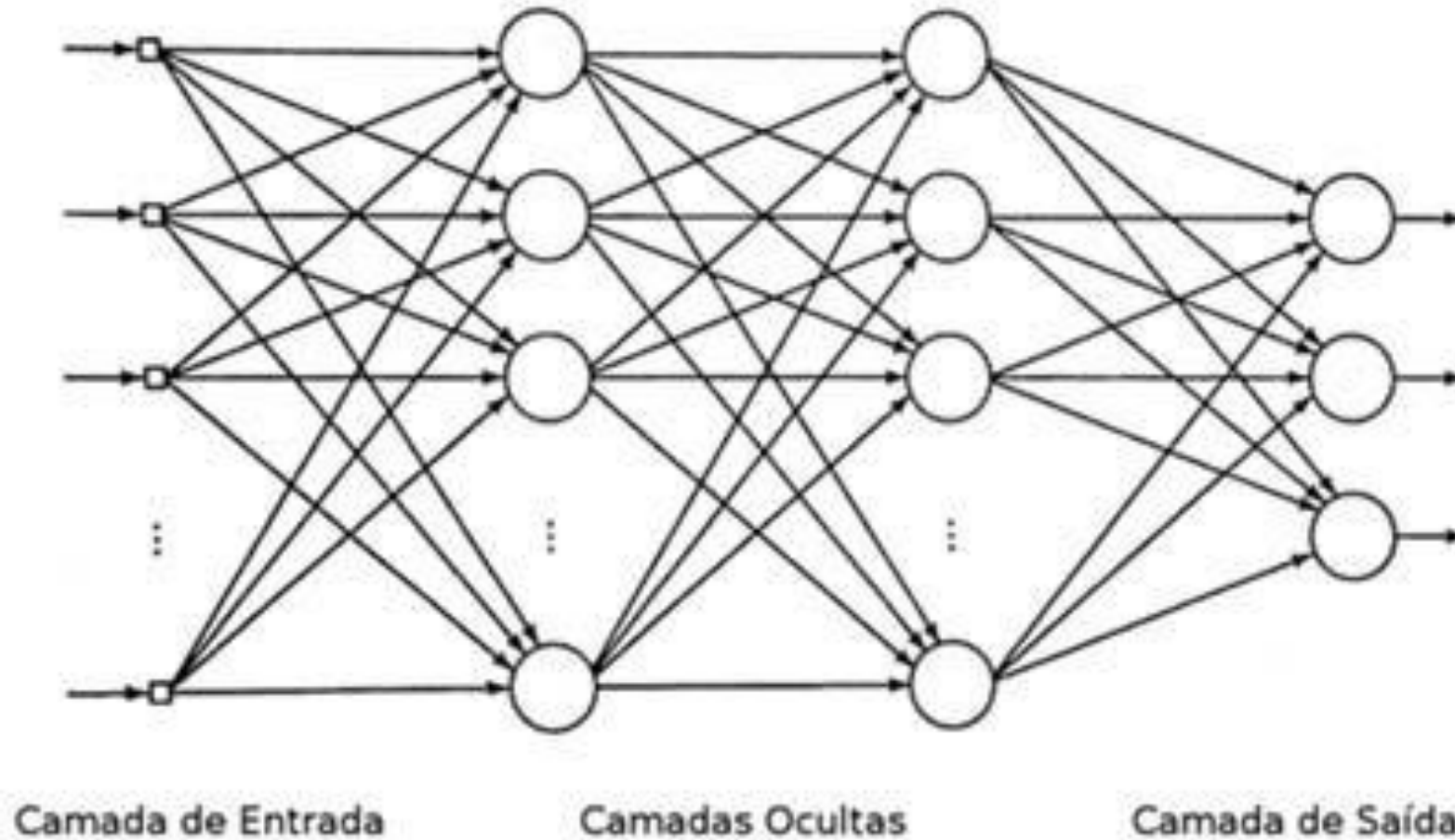


Redes Neurais (*Multi Layer Perceptron-MLP*)





Rede neural artificial composta por múltiplas camadas.





Composto por:

1. Camada de entradas: recebe os dados;
2. Camada oculta: processa os dados; permite resolver problemas mais complexas;
3. Camada de saída: produz a resposta final.



Quando aplicar?





Imagine que a rede neural é como uma **dobradura de papel**.

- Função linear ($f(x) = x$) é como **tentar dobrar um papel com régua — só permite linhas retas. Isso funciona se a figura que você quer formar é plana;**
- Para formar **um avião de papel, é necessário dobrar em ângulos, curvas, torções — ou seja, não linearidades.**

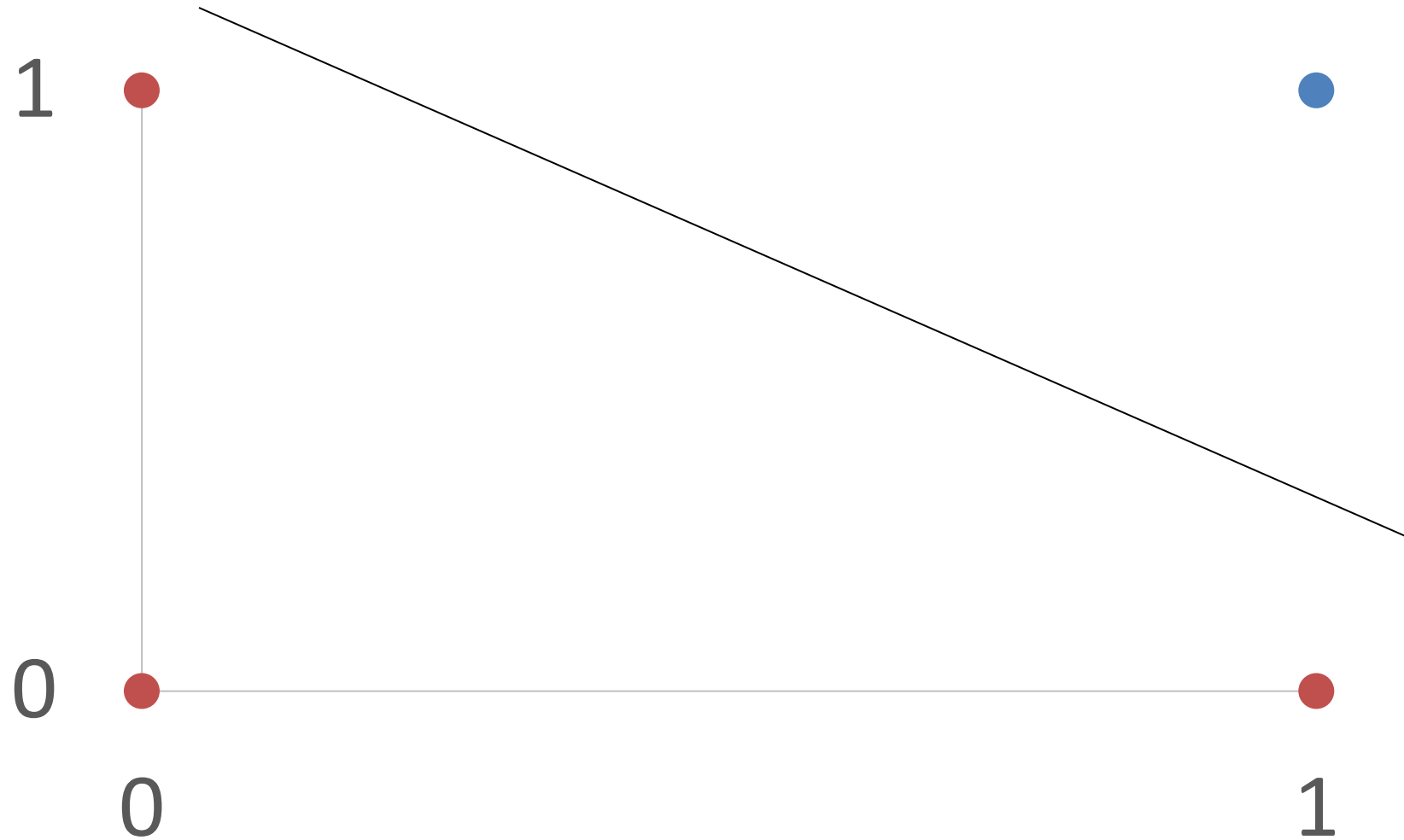


Um MLP com ao menos uma camada oculta com função de ativação não linear consegue resolver esse problema, porque:

- **A camada oculta transforma os dados em um espaço onde a separação linear é possível.**
- **A não linearidade da função de ativação permite que o modelo aprenda limites de decisão mais complexos.**



Exemplo solucionado com Perceptron (AND)



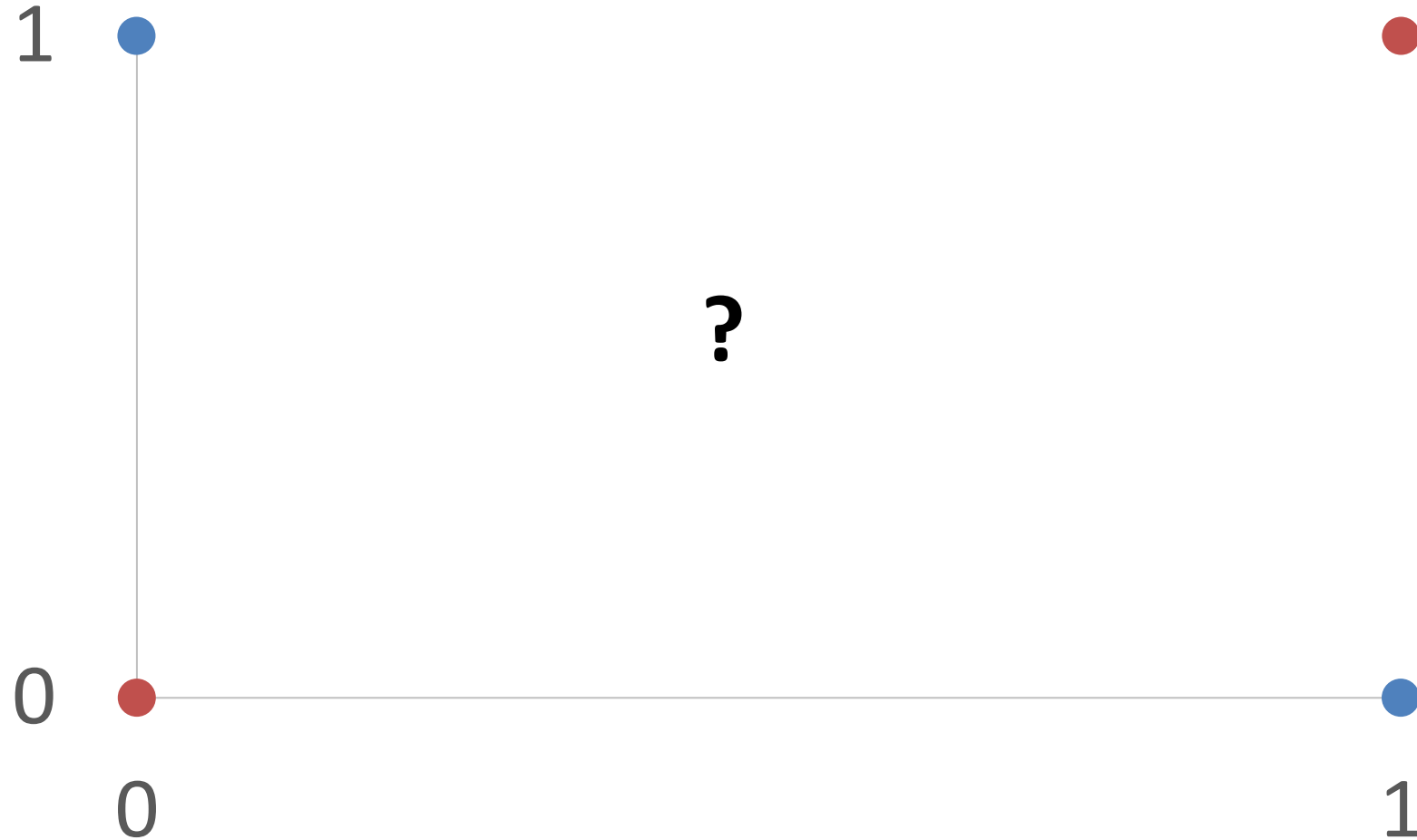


Exemplo solucionado com Perceptron (OR)





Exemplo não solucionado com Perceptron (XOR)



Funcionamento



Passo a passo:

1. Recebe os dados de entrada;
2. Cada neurônio da camada oculta faz:
 - Multiplica cada entrada pelo seu peso;
 - Soma os resultados + bias;
 - Passa essa soma por uma função de ativação (ReLU, sigmoide ou degrau).
3. A saída da camada oculta é usada como entrada na camada de saída;
4. O resultado final é a previsão da rede.
5. A rede compara com a resposta real;
6. Se estiver errada, ajusta os pesos com base no erro.

Elementos fundamentais



| Elemento | O que é | Para que serve | Exemplo/Nota |
|--------------------|---|---|-------------------------------------|
| Entrada | Valores de entrada da rede (x_1, x_2, \dots, x_n) | Alimenta a rede com dados reais | Ex: pixel, nota, sensor, etc. |
| Pesos | Valores multiplicadores entre neurônios | Controlam a importância de cada entrada | Iniciados aleatoriamente |
| Bias | Valor extra somado antes da ativação | Garante flexibilidade na decisão | Pode ser 0 no início |
| Camada Oculta | Neurônios intermediários | Extraem padrões complexos dos dados | Pode ter 1 ou mais camadas |
| Função de Ativação | Função matemática (sigmoide, ReLU, etc.) | Decide se o neurônio “dispara” (ativa) | Ex: $f(x) = \frac{1}{(1 + e^{-x})}$ |
| Saída | Resposta final da rede | Resultado da predição | Ex: 0 ou 1, ou valor contínuo |



| Elemento | O que é | Para que serve | Exemplo/Nota |
|-----------------------------------|--|--|---|
| <i>Forward Propagation</i> | Passagem dos dados pela rede | Gera a saída com os pesos atuais | Multiplica + soma + ativa |
| Erro | Diferença entre saída real e prevista | Mede o quanto a rede errou | $\text{erro} = y - y_{pred}$ |
| <i>Backpropagation</i> | Ajuste dos pesos e bias com base no erro | Treinamento: reduz o erro nas próximas rodadas | Derivadas da função de ativação |
| Derivada da Ativação | Inclinação da função de ativação | Mede sensibilidade do erro em relação ao peso | Ex: $\text{sig}(x) = 1 - \text{sig}(x)$ |
| Taxa de Aprendizado | Fator de velocidade de aprendizado | Evita mudanças muito grandes ou pequenas nos pesos | Típico: 0.01 a 0.1 |
| Época | Uma passada por todo o conjunto de dados de treino | Define o tempo de treinamento | Varia: 10, 100, 1000... |



| Elemento MLP | Analogia com alunos/sala de aula | Explicação Didática |
|--------------------|---|---|
| Entrada | As perguntas da prova | O que os alunos recebem como desafio ou estímulo |
| Pesos | O quanto o aluno estudou para cada tema | Mais estudo em um assunto = maior peso ao resolver aquela questão |
| Bias | Ajuda extra, como revisão do professor | Um empurrãozinho a mais para o aluno acertar, mesmo sem entender tudo |
| Camada Oculta | O “raciocínio interno” do aluno | Onde o aluno processa a questão, conecta ideias, faz contas |
| Função de Ativação | A decisão de acertar ou errar | Se o raciocínio foi suficiente para chegar na resposta correta |
| Saída | A resposta final na prova (certa ou errada) | O que o aluno entrega depois de pensar |



| Elemento MLP | Analogia com alunos/sala de aula | Explicação Didática |
|----------------------|---|---|
| Forward Propagation | Fazer a prova com o conhecimento atual | Aplicar o que sabe para resolver |
| Erro | Diferença entre a nota esperada e a nota real | Quanto o aluno errou na resposta |
| Backpropagation | Correção da prova + feedback do professor | Aprender com o erro e ajustar o modo de pensar para ir melhor da próxima vez |
| Derivada da Ativação | Quanto o erro depende do que foi pensado | Se o aluno quase acertou, aprende mais fácil; se errou feio, tem mais dificuldade |
| Taxa de Aprendizado | A vontade ou velocidade do aluno para aprender com o erro | Se for muito alta, ele muda tudo de uma vez; se for baixa, aprende bem devagar |
| Época | Quantas vezes o aluno faz simulados antes da prova final | Cada repetição melhora o desempenho até ele aprender de verdade |



| Função | Fórmula | Curva / Comportamento | Analogia Didática |
|----------|---------------------------------------|---|--|
| Sigmoide | $f(x) = \frac{1}{(1 + e^{-x})}$ | Saída entre 0 e 1 | Como um aluno que decide “quase acertei”, "errei um pouco", "acertei bem” |
| Tanh | $f(x) = \tanh(x)$ | Saída entre -1 e 1 | Como um aluno que pode pensar positivamente ou negativamente sobre o resultado |
| ReLU | $f(x) = \max(0, x)$ | Zera valores negativos, passa positivos | Se o aluno não entende nada, dá 0. Se entende, segue com a resposta |
| Softmax | $f(x_i) = \frac{e^{x_i}}{\sum_e x_j}$ | Distribui probabilidades (usada na saída) | Como um aluno que escolhe a resposta mais provável numa múltipla escolha |

MLP - implementação