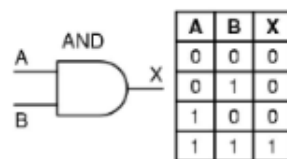


Teoria de Redes Neurais Artificiais

Neurônio Artificial

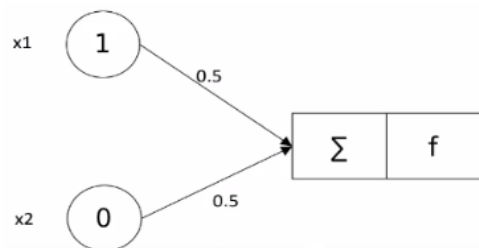
É a unidade fundamental de redes neurais artificiais, se inspirando em neurônios biológicos. Recebendo entradas, processá-las e gerar uma saída.

Usando como exemplo um neurônio com duas entradas (x_1 e x_2) e uma saída, podemos criar uma porta AND.



Entradas:

- O neurônio recebe duas entradas (A e B) que podem ser 0 ou 1.
- Cada entrada é multiplicada por um peso associado. No exemplo de uma porta AND é necessário ser 0.5.



Soma:

- As entradas serão multiplicadas pelos seus respectivos pesos e somadas.
$$\text{soma} = (1 \times 0.5) + (0 \times 0.5) = 0.5$$

Step Function:

- A soma é passada por uma step function. No exemplo é usada uma função de degrau, que retorna 1 se a soma for maior ou igual a 1, e 0 caso contrário.
- No caso de 0.5, a função irá retornar 0.

Saída:

- A saída do neurônio é o resultado da step function. Com base nas entradas x_1 e x_2 .

Saídas para cada caso:

$$\text{soma} = (0 \times 0.5) + (0 \times 0.5) = 0 \Rightarrow \text{saída} = 0$$

$$\text{soma} = (1 \times 0.5) + (0 \times 0.5) = 0.5 \Rightarrow \text{saída} = 0$$

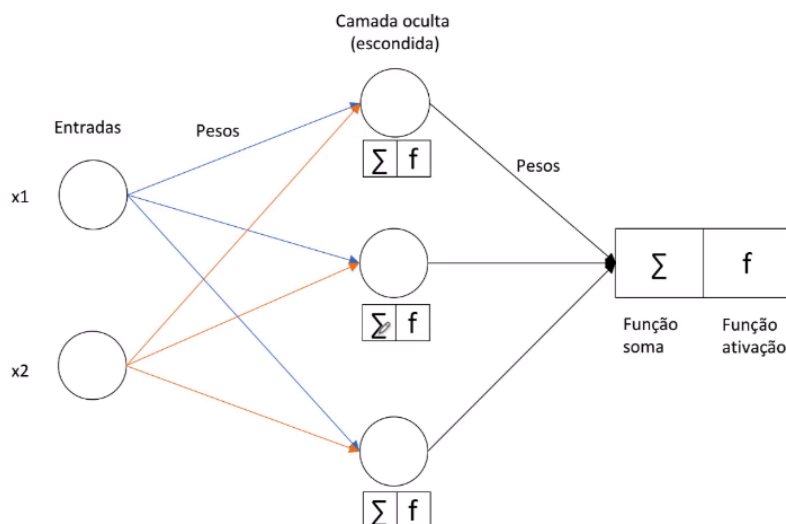
$$\text{soma} = (0 \times 0.5) + (1 \times 0.5) = 0.5 \Rightarrow \text{saída} = 0$$

$$\text{soma} = (1 \times 0.5) + (1 \times 0.5) = 1 \Rightarrow \text{saída} = 1$$

Redes Neurais Artificiais

- **Funções**

Para formar uma Rede Neural, é necessário organizar múltiplos neurônios artificiais em diversas camadas. Cada camada atua como um grupo de neurônios interconectados, seguindo os mesmos passos do neurônio artificial mas aumentando a complexidade adicionando mais camadas.



Que se utiliza da função sigmóide, invés da Step Function. Sendo ela:

$$y = \frac{1}{1+e^{-x}}$$

Usando os valores resultantes da soma no lugar do x. Que caso tendo um alto valor para x, o resultado será próximo a 1, e sendo baixo, próximo a 0.

- **Cálculo do Erro**

Após obter a resposta calculada pelo neurônio, é necessário determinar o erro. O cálculo do erro é feito da seguinte forma:

$$erro = Correto - Calculado$$

Ao calcular os erros para todas as saídas e determinar a média desses erros, obtemos a porcentagem de erro do algoritmo. Usando o exemplo do vídeo, temos:

Correto	Calculado	Erro
0	0.406	-0.406
1	0.432	0.568
1	0.437	0.563
0	0.458	-0.458

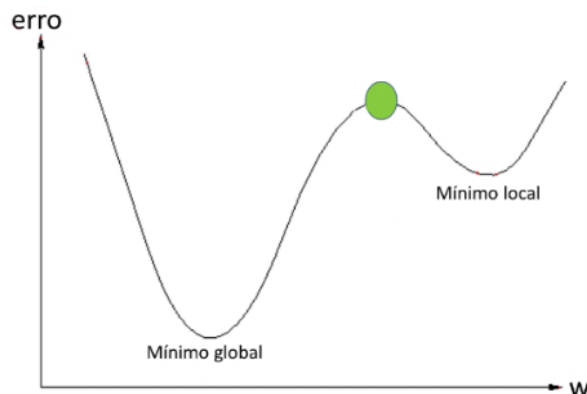
Tabela 1 - Erros Calculados

Média absoluta do Erro: 0.49

Isso significa que o algoritmo tem uma precisão de apenas 51%, indicando que é necessário ajustar os pesos para melhorar seu desempenho. Essa é a principal função de uma Rede Neural: ajustar os pesos e calcular os erros, repetindo esse ciclo até alcançar uma boa precisão.

- **Descida do gradiente**

O objetivo do gradiente é encontrar a combinação de pesos que minimize o erro. Isso é feito ajustando os pesos com base no cálculo das derivadas parciais da função de erro em relação aos pesos.



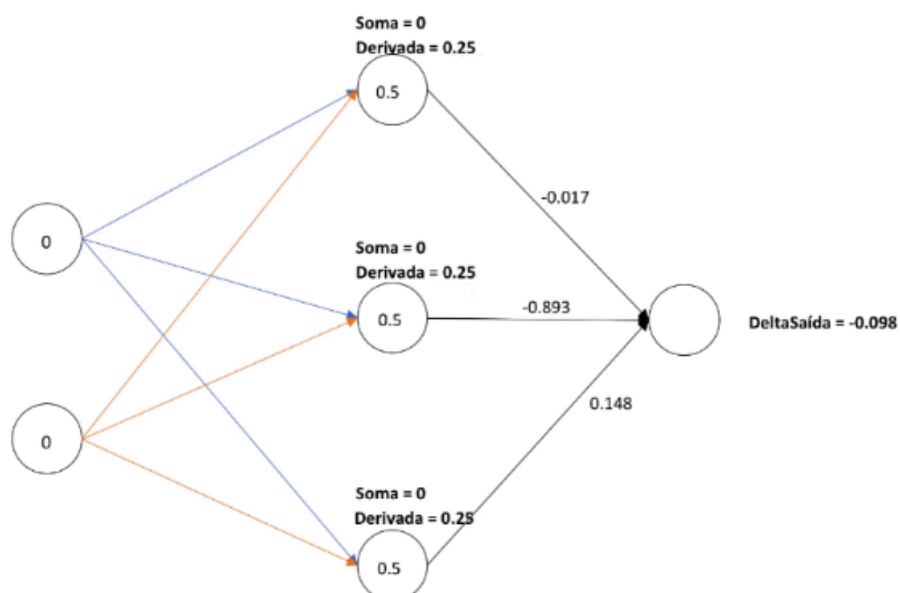
$$y = \frac{1}{1+e^{-x}} \implies d = y * (1 - y)$$

Com o valor resultante da derivada (d) da sigmóide, será possível determinar a direção que será percorrida no gradiente para encontrar o erro mínimo.

- **Cálculo do parâmetro delta**

Para calcular o delta de saída, multiplicamos cada erro pelo resultado da derivada da função sigmóide. Por exemplo, no caso 1 da Tabela 1:

$$\text{DeltaSaída} = -0.406 * 0.241 = -0.098$$



Com o delta de saída calculado, o próximo passo é calcular o delta da camada escondida, multiplicando a derivada da sigmóide pelo peso correspondente e pelo delta de saída.

1. Primeiro Neurônio

$$\Delta = 0.25 \times (-0.017) \times (-0.098) = 0.000$$

2. Segundo Neurônio

$$\Delta = 0.25 \times (-0.893) \times (-0.098) = 0.022$$

3. Terceiro Neurônio

$$\Delta = 0.25 \times 0.148 \times (-0.098) = -0.004$$

Cada neurônio na camada oculta possui seu próprio delta, calculado a partir de seus próprios pesos e delta de saída. Esse processo é repetido para todos os neurônios na rede, ajustando os pesos de forma iterativa até que o erro seja minimizado.

● Ajuste dos Pesos

Para realizar o ajuste dos pesos é necessário usar a seguinte equação:

$$peso_{n+1} = (peso_n * momento) + (entrada * delta * taxa\ de\ aprendizagem)$$

Essa equação permite que os pesos sejam atualizados de maneira iterativa, sendo os seus componentes:

1. *Peso atual* ($peso_n$): O valor do peso da iteração atual;
2. *Peso ajustado* ($peso_{n+1}$): Novo peso;
3. *momento*: Suaviza as atualizações de peso;
4. *entrada*: Valor de entrada associado ao peso;
5. *delta*: Valor delta calculado;
6. *taxa de aprendizagem*: Determina o quanto os pesos são ajustados em cada iteração

● Bias

O Bias é um parâmetro adicional na rede neural, ele é adicionado na rede antes de aplicar a função de ativação. Seu objetivo é permitir que o modelo se ajuste melhor aos dados, deslocando a função de ativação para que ela não passe necessariamente pela origem. Durante o treinamento da rede neural, os valores do Bias e dos pesos são ajustados para minimizar os erros.

● Validação Cruzada

É uma técnica estatística utilizada para avaliar o desempenho de um modelo de aprendizado de máquina. O principal objetivo da validação cruzada é garantir que o modelo não apenas memorize os dados de treinamento, mas também seja capaz de performar bem em novos dados.

Uma das abordagens mais comuns é a k-fold cross validation. Os dados são divididos em k folds aproximadamente do mesmo tamanho. O modelo é então treinado k vezes, cada vez utilizando k-1 folds como dados de treinamento e o fold restante como dados de validação, e a média dos desempenhos em cada uma das k interações é utilizada como estimativa da performance do modelo.

- **Underfitting**

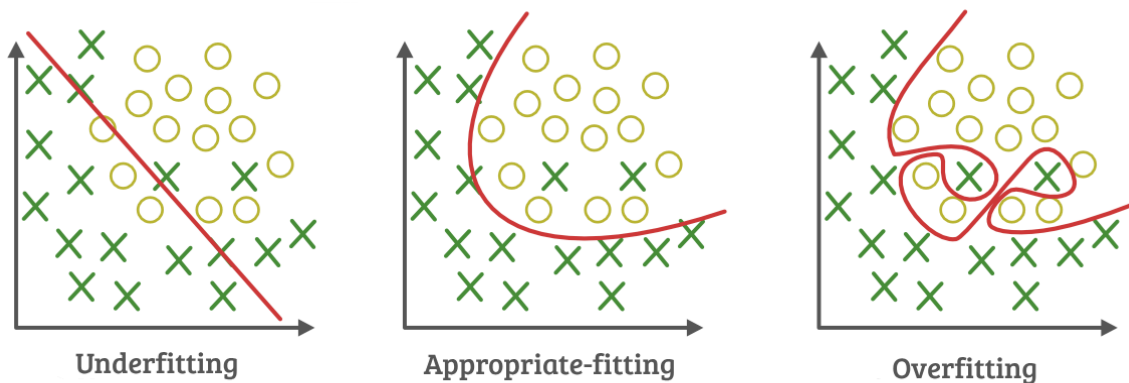
O underfitting ocorre quando um modelo é muito simples para capturar os padrões dos dados. Isso geralmente acontece quando o modelo subestima a complexidade do problema. O resultado é um modelo com desempenho ruim tanto nos dados de treinamento quanto nos dados de teste

- **Overfitting**

O overfitting ocorre quando um modelo é excessivamente complexo e se ajusta muito bem aos dados de treinamento, mas não generaliza bem para dados novos. Como resultado, ele tem um desempenho muito bom nos dados de treinamento, mas um desempenho ruim nos dados de teste.

- **Appropriate-fitting**

O fitting apropriado é o equilíbrio entre underfitting e overfitting. Resultando em um modelo que generaliza bem para novos dados, mantendo um bom desempenho tanto nos dados de treinamento quanto nos dados de teste.



Desenvolvimento do Modelo de Redes Neurais para Previsão da final da Eurocopa 2024

Neste relatório, foi desenvolvido um modelo de redes neurais para prever o resultado da final da Eurocopa 2024. Os dados foram retirados do arquivo [‘Euro_2024_Matches.csv’](#) contendo as informações detalhadas de cada partida.

- **Seleção dos dados**

Foram selecionadas colunas relevantes para a análise, como gols marcados, chutes, passes e expectativa de gols para ambas equipes.

- **Codificação dos Resultados**

Cada partida foi codificada com base no desempenho das equipes. Os resultados foram codificados como 1 para vitória e 0 para derrota ou empate, tanto para a Espanha quanto para a Inglaterra.

- **Modelagem**

Utilizei uma arquitetura de rede neural sequencial com duas camadas densas e uma camada de dropout para evitar o overfitting. Foi feito para as duas seleções, para haver a comparação.

- **Acurácia**

O modelo treinado alcançou 100% de acerto nos jogos da Espanha, que venceu todos os jogos. Já para a Inglaterra, que teve alguns empates, fazendo alguns testes chegou a 100% de acerto, porém na maioria das vezes 83% de acerto.

- **Previsão**

Com o modelo treinado, previu-se que a Espanha iria vencer a final da Eurocopa com base nos dados extraídos. A final, realizada em 14/07, terminou com a vitória da Espanha por 2x1 sobre a Inglaterra, confirmando a previsão feita pelo modelo. É importante pontuar que todos os testes foram realizados antes da final, e a previsão correta destaca a eficácia do modelo.



- **Considerações Finais**

Embora o modelo tenha apresentado alta acurácia nos testes realizados e tenha acertado o resultado final, é importante destacar que não é possível prever perfeitamente quem ganhará uma partida de futebol, pois este é um esporte muito imprevisível.

No entanto, foi interessante realizar essa previsão, interligando os dados das partidas com o desempenho das equipes. A análise de algumas estatísticas básicas de cada equipe forneceu insights valiosos que contribuíram para a previsão de que a Espanha iria ganhar a final da Eurocopa.

Conclusão

Este estudo demonstrou a eficácia das redes neurais na previsão de resultados, especialmente quando adaptadas e treinadas com dados específicos. Utilizando uma base de dados sobre a Eurocopa obtive uma visão detalhada do desempenho das seleções da Inglaterra e da Espanha na Eurocopa 2024, destacando a capacidade do modelo em antecipar resultados.