

Sistemas Inteligentes – Aula 5

Hoje iremos criar uma rede neural simples para determinar a qual grupo uma fruta pertence, baseado em suas características.

Para facilitar o processo, iremos utilizar o Google Colab, um ambiente de desenvolvimento online que já apresenta as configurações necessárias.

Acesse <https://colab.research.google.com/> e clique no botão “+ Novo notebook”, no canto inferior esquerdo.

Para adicionar o código, clique no botão “+ Código” e cole o código disponibilizado no repositório

<https://github.com/Kobaldo/SistemasInteligentes/blob/main/Aula5/aula5.py>

Clique no botão de play, no lado esquerdo do seu código.

Vocês podem reparar que os resultados variam a cada execução do código, já que o treinamento de redes neurais envolve aleatoriedade em vários estágios do processo. Essa aleatoriedade afeta como os pesos iniciais são definidos e como os dados são processados.

Vamos entender um pouco do que está sendo feito:

Primeiro, importamos as bibliotecas (não se preocupe em decorar isso, é apenas para conhecimento mesmo): NumPy manipula arrays numéricos, TensorFlow e Keras criam e treinam a rede neural, Sequential gera o modelo linear de rede neural, Dense cria camadas densas da rede e MinMaxScaler normaliza os dados para que estejam na mesma escala, essencial para melhorar o desempenho do modelo.

Em seguida, colocamos os dados para treinar nossa rede, dizendo quais características correspondem a uma maçã, uma banana e uma laranja:

- dados: Contém as características (peso e cor) de cada fruta.
- Peso: Primeiro valor (ex.: 150 gramas).
- Cor: Segundo valor, codificado como:
 - 0 = Verde
 - 1 = Amarela
 - 2 = Laranja
- rotulos: As classes das frutas:
 - 0 = Maçã
 - 1 = Banana
 - 2 = Laranja

Na linha 15, fazemos a normalização, que é colocar tudo na mesma escala. O peso varia de 100 a 250, enquanto a cor varia de 0 a 2. Essa diferença pode dificultar o aprendizado da rede neural, por isso precisamos ajustar esses valores.

Da linha 19 a 23, criamos o modelo neural, que aprende a identificar as frutas por suas características. Nossa rede tem 3 “níveis”, com 8, 6 e 3 nós ou neurônios.

Na linha 26, fazemos os últimos ajustes de formato de dados e modelo de treinamento e na 27 rodamos a rede neural. O processo se repete 50 vezes, ou seja, nossa rede neural lê os dados e tenta identificar padrões 50 vezes. Essa quantidade de repetições é definida no parâmetro “epochs”.

Depois de gerar a rede neural, é hora de testar: Na linha 30, passamos as características de uma fruta e vemos se o resultado é o esperado. O restante do código é apenas para formatar os dados e deixá-los mais fáceis de ler.

Agora é com você:

1. Faça a rede neural repetir todo o procedimento 100, 200 e 300 vezes. Em qual delas a rede melhorou? Em qual piorou? Por que isso ocorreu?
2. Edite a rede neural para que ela tenha mais uma “camada” de nós. A precisão aumentou ou diminuiu? E com 2 camadas adicionais?
3. Crie novos conjuntos de teste, com valores diferentes para as duas frutas não previstas.
4. Na configuração e treinamento, altere o que está sendo colocado nos parâmetros optimizer, loss e metrics. Peça ajuda ao Gemini para ver as opções disponíveis e teste as opções, aprimorando para a mais eficiente.
5. Se baseando no código dado e nas explicações anteriores, crie uma rede neural que busca determinar qual linguagem de programação vai ser usada em um projeto, de acordo com parâmetros determinados por você. Eles podem envolver características da pessoa que está desenvolvendo, o objetivo do programa, a área de atuação ou qualquer outra questão que faça sentido para o caso estudado.

Caso tenha dificuldades, você pode clicar no canto superior direito do Google Colab, no botão “Gemini”, que te dará acesso à IA do Google. Não peça para ela fazer todo seu trabalho, lembre-se de que essa deve ser uma ferramenta de trabalho, não seu substituto. Bom trabalho!