

SCC0270 - Neural Networks and Deep Learning

Terceiro Projeto Prático

Esse projeto tem como objetivo ilustrar a construção de um modelo usando uma técnica de transfer learning. A base de dados usada neste projeto se encontra no arquivo **data.zip** e já foi dividida em treino e teste (ver subpastas).

Utilize a biblioteca Keras para criar os modelos pedidos. Você pode usar outras bibliotecas para fazer as demais análises =)

Para entender um pouco mais sobre o uso do Keras, veja estes tutoriais:

- https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator
- https://keras.io/guides/transfer_learning/

Todas as respostas devem ser justificadas com base em:

1. Código Python mostrando a(s) análise(s) e/ou o(s) modelo(s) feitos;
2. O resultado da(s) análise(s) e/ou do(s) modelo(s) e
3. Uma explicação textual (pode ser breve) da conclusão obtida.

Em caso de plágio (mesmo que parcial) o trabalho de todos os alunos envolvidos receberá nota ZERO.

NÃO enviar um link para o Google Colab como resposta/relatório do projeto.

Desorganização excessiva do código resultará em redução da nota do projeto.

Exemplos:

- Códigos que devem ser rodados de forma não sequencial;
- Projeto entregue em vários arquivos sem um README;
- ...

**Bom projeto,
Tiago.**

Questão 1 (valor 5 pontos)

Você deve criar a seguinte CNN usando o Keras:

- Parte convolucional da rede **MobileNetV2**;
 - Tal modelo deve se configurado para receber como entrada imagens RGB com dimensões espaciais 128x128
 - Todos os parâmetros (pesos e bias) dessa parte do modelo devem ser "congelados" (impedidos de atualizar);
 - Os parâmetros dessa parte do modelo devem ser obtidos do modelo pré-treinado na ImageNet .
- Uma camada densa com ativação **softmax**.

E configurar o seu processo de otimização para usar:

- Otimizador Adam;
- Cross-entropy como loss;
- Acurácia como métrica de sucesso.

Dicas:

1. Use uma camada do tipo **Flatten** para transformar o output da parte convolucional do modelo em um vetor;
2. Use o parâmetro **input_shape** para configurar o tamanho da imagem de input;
3. Use a propriedade **trainable** das camadas convolucionais para que elas sejam "congeladas";

Questão 2 (valor 5 pontos)

Treine o modelo usando o seguinte protocolo:

- Permita que os dados de treinamento (**somente os dados de treinamento**) sejam distorcidos usando as seguintes transformações:
 - `horizontal_flip=True`;
 - `zoom_range=0.2`;
 - `shear_range=10`;
 - `height_shift_range=0.1`;
 - `width_shift_range=0.1`.
- Usando mini-batch de 32 imagens;
- O modelo deve ser treinado por 10 épocas (ou seja, que ele deve processar durante o treinamento uma quantidade igual a 10x o tamanho da base de treinamento).

Acompanhe a acurácia do modelo nos dados de treinamento e na base de teste ao final de cada época.

Dicas:

1. Aplique a função de pré-processamento usada pela **MobileNetV2**;
2. Use dois **ImageDataGenerator** diferentes para que somente a base de treinamento tenha data augmentation;
3. Ao usar a função `flow_from_directory` na base de teste, use **`shuffle=False`**.