SWS3009 Summer Workshop

Lab 4 – Answer Book

SUBMISSION DEADLINE: SUNDAY 9 JULY 2023, 11.59 PM

Name 1: Tao YiCheng

Name 2: Jia ShiAn

Q1

`Sequential` is a simple way to build a linear stack of layers, where each layer has exactly one input tensor and one output tensor. This makes it easy to define a neural network model as a sequence of layers, without worrying about the inputs and outputs of each layer. `Sequential` is a good choice for simple models that have a linear structure, such as feedforward neural networks.

`Model` is a more flexible way to define neural network models, where the layers can be connected in arbitrary ways. With `Model`, we can define multiple input tensors and output tensors, and we can connect layers in complex ways using the functional API. This makes it possible to define more complex models, such as models with multiple inputs or outputs, or models with shared layers.

In general, we use `Sequential` when we have a simple, linear model structure, and we use `Model` when we have a more complex model structure that requires more flexibility.

Q2

Deep learning models require large amounts of data to learn effectively. However, in many cases, we may not have enough data to train our models effectively. By using data augmentation techniques like ImageDataGenerator, we can generate new data from our existing data, which can help to increase the size of our training set and improve the performance of our models.

Two other transformations are:

1. Rotation: We can use the `rotation_range` parameter to randomly rotate the images in our training set by a specified number of degrees.

Vertical flipping. We can use the `vertical_flip` parameter to randomly flip the images in our training set vertically.

Q3

Each number in the array represents the predicted probability for the flowers. The order of the classes is exactly the order in which they were defined when the model was compiled, i.e., 0: 'daisy', 1:'dandelion', 2: 'roses', 3:'sunflowers', 4: 'tulips'. In this example, the predicted probabilities for the five kind of flowers are `[0.231701, 0.6328776, 0.01869423, 0.0764939, 0.04023323]`.

Q4

np.argmax` returns the index of the maximum value in the array.

In this code, we call `np.argmax` to find the index of the maximum value in the `result` array, which represents the predicted probabilities for each flower. Then, we can determine which class has the highest predicted probability and make a prediction based on this information.

The last line of the code returns a tuple that contains three values: the name of the predicted flower, the predicted probability for the flower, and the index of the predicted flower.