**SWS3009 Summer Workshop**

**Lab 4 – Answer Book**

**SUBMISSION DEADLINE: Friday 5 JULY 2024, 11.59 PM**

Name 1: YANG RUNKANG

Name 2: WEN SIJIE

**QUESTION 1.**

The main difference between the Sequential and Model APIs in Keras lies in how they define the structure of a neural network. The Sequential API is used to build simple models where layers are stacked one after another in a linear fashion, making it easier to use and understand for straightforward neural networks with a single input and output. It is suitable when the model is a plain stack of layers, each with exactly one input tensor and one output tensor, and there is no need to share layers or create a complex structure.

The Model class, on the other hand, is part of the Functional API, which is more flexible and allows for the creation of more complex models. It can handle models with multiple inputs and outputs, layers that share weights, and models with non-linear topology, such as residual connections or multi-branch models. The Functional API is used when creating models with shared layers or multiple inputs/outputs, when the model architecture is complex, or when more control over the model’s architecture is desired.

**QUESTION 2.**

The performance of deep learning models often relies on having a large amount of training data. In many cases, the labeled dataset we have may not be sufficiently large. By using data augmentation techniques, we can effectively expand the dataset, improving the model's generalization ability and reducing the risk of overfitting. These transformations simulate various scenarios that might be encountered in real environments, making the model more robust to different image variations.Two other transformations in ImageDataGenerator:

Rotation: Rotating images by a certain degree helps the model recognize objects irrespective of their orientation. This transformation simulates scenarios where the camera angle changes.

Zoom: Zooming into images helps the model focus on different parts of the image and learn to recognize objects at various scales. This transformation simulates scenarios where the distance from the camera to the object varies.

**QUESTION 3.**

The array returned by model.predict represents the probabilities assigned by the model to each class for a given input image. Each number in the array corresponds to the likelihood that the input image belongs to a particular class. In this example, the model is predicting the probabilities for five classes (daisy, dandelion, roses, sunflowers, tulips). The numbers in the array can be interpreted as follows:

0.231701: Probability that the image is a daisy.

0.6328776: Probability that the image is a dandelion.

0.01869423: Probability that the image is a rose.

0.0764939: Probability that the image is a sunflower.

0.04023323: Probability that the image is a tulip.

The model predicts that the image is most likely a dandelion because it has the highest probability (0.6328776).

**QUESTION 4.**

np.argmax is a function in NumPy that returns the index of the maximum value in an array. np.argmax(result) returns the index of the highest probability in the prediction array. This index corresponds to the class with the highest predicted probability, indicating the model's most likely classification for the input image.

We call np.argmax here to determine which class the model believes the input image belongs to with the highest confidence.

The last line:

return (dict[themax], result[0][themax], themax)

returns a tuple containing three elements:

dict[themax]: The class label corresponding to the highest probability.

result[0][themax]: The highest probability value from the prediction array.

themax: The index of the highest probability in the prediction array.

This provides the class label, the confidence score, and the index of the predicted class.