QUESTION

How to write the following code with Sequential model?

inputs = keras.Input(shape=(sequence\_length, raw\_data.shape[-1]))

x = layers.Flatten()(inputs)

x = layers.Dense(16, activation="relu")(x)

outputs = layers.Dense(1)(x)

model = keras.Model(inputs, outputs)

ANSWER

from tensorflow import keras

from tensorflow.keras import layers

# Define the Sequential model

model = keras.Sequential()

# Add layers sequentially

model.add(layers.Flatten(input\_shape=(sequence\_length, raw\_data.shape[-1])))

model.add(layers.Dense(16, activation="relu"))

model.add(layers.Dense(1))

# Compile the model if needed

# model.compile(optimizer='adam', loss='mean\_squared\_error')

# Summary of the model

model.summary()

QUESTION

What is the technical difference between functional API and Sequential model? Why people want to use functional API when we can do the same with Sequential model?

ANSWER

The main technical difference between the Functional API and the Sequential model lies in their flexibility and complexity.

Sequential Model:

The Sequential model is the simplest way to build a neural network in Keras. It allows you to create models layer by layer in a linear fashion, where each layer has exactly one input tensor and one output tensor.

It's suitable for building simple architectures where the data flows sequentially through the layers.

It's limited to single-input, single-output models, which makes it less flexible for more complex architectures, such as models with multiple inputs or outputs, shared layers, or branching architectures.

Functional API:

The Functional API provides more flexibility and allows for building complex architectures, including models with multiple inputs, multiple outputs, shared layers, and branched architectures.

It allows you to define a computational graph by explicitly connecting the layers to each other, which gives you more control over the data flow.

With the Functional API, you can create models with loops, conditional branching, and other dynamic behaviors, which are not possible with the Sequential model.

It's more suitable for advanced use cases, such as building custom architectures, implementing complex network designs, or models requiring non-linear data flow.

Reasons for using the Functional API over the Sequential model:

Flexibility: The Functional API allows for building more complex architectures that are not possible with the Sequential model.

Customization: It provides greater control over the connections between layers, enabling the creation of custom architectures.

Reusability: The Functional API allows you to easily reuse and share layers or submodels across different parts of the network.

Complex data flows: For models with multiple inputs, multiple outputs, or complex data flow requirements, the Functional API is necessary.

In summary, while the Sequential model is simple and suitable for many basic use cases, the Functional API offers more flexibility and control, making it preferred for building complex and customized neural network architectures.