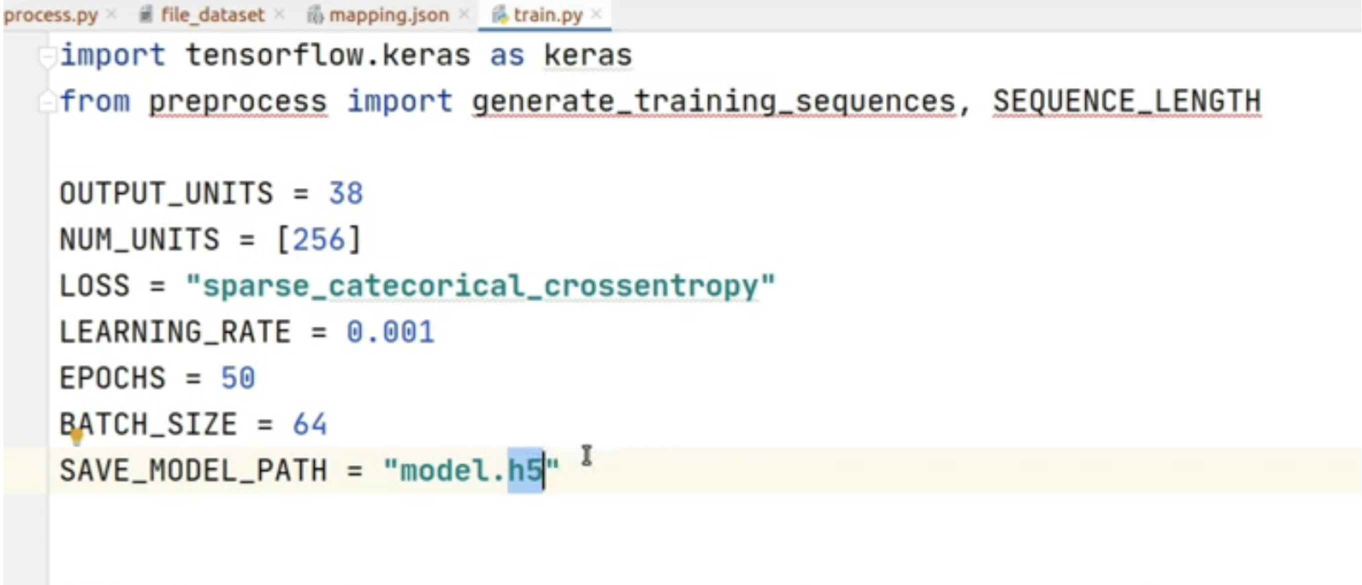
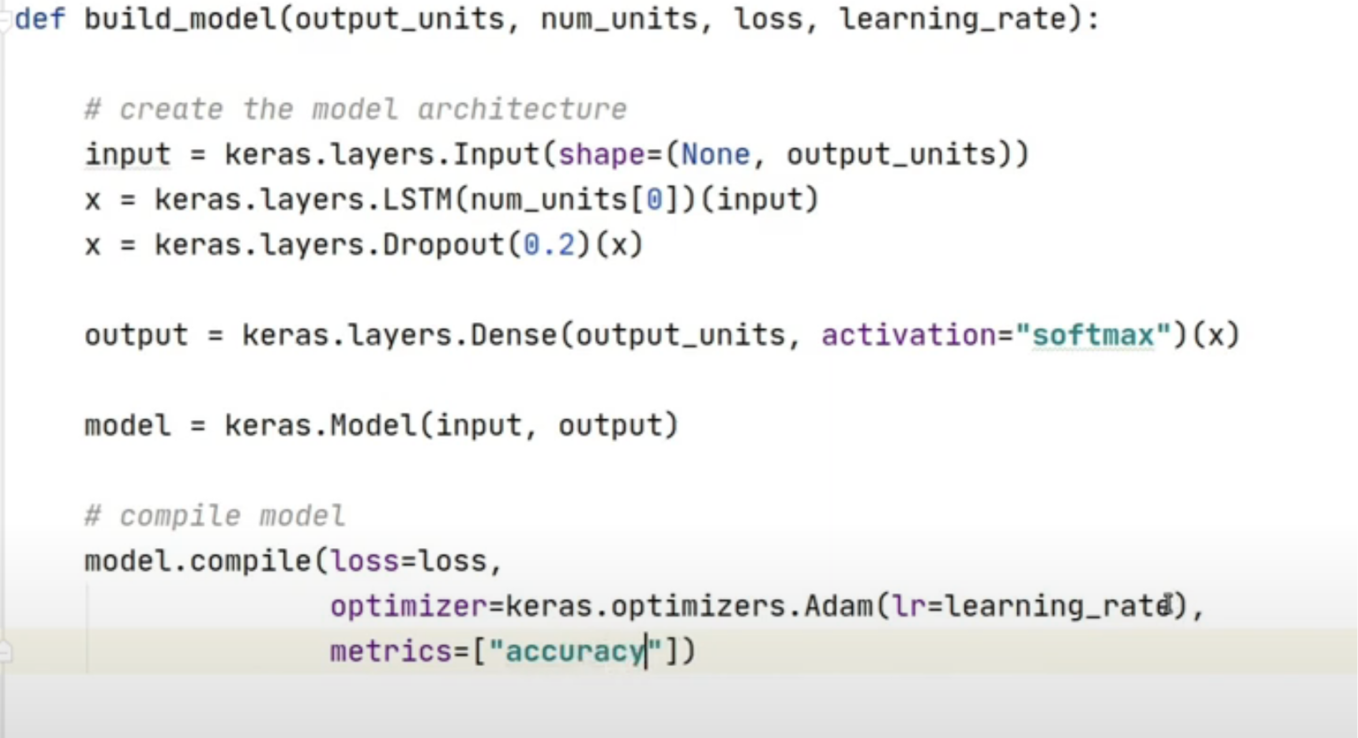
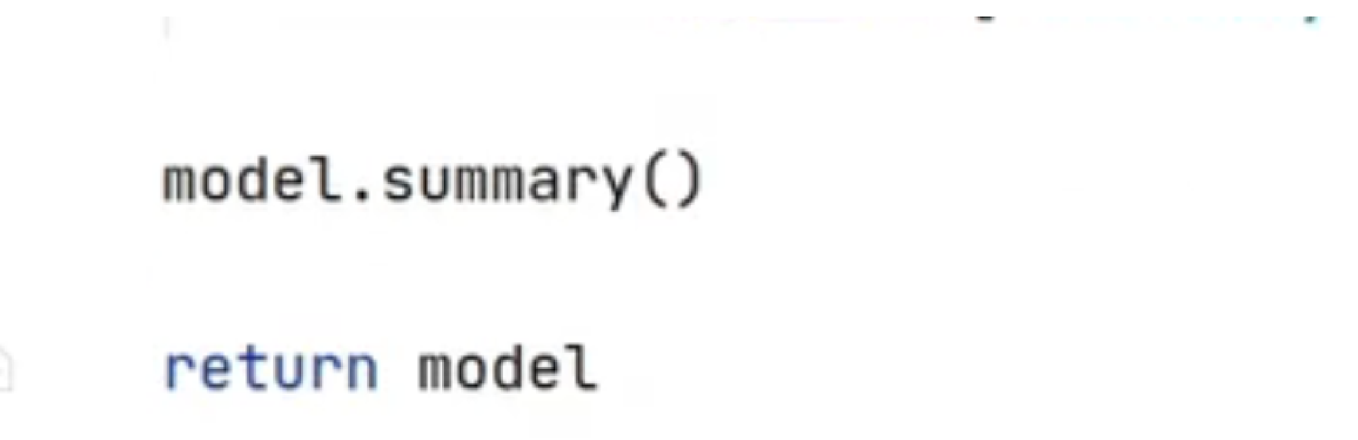
Training



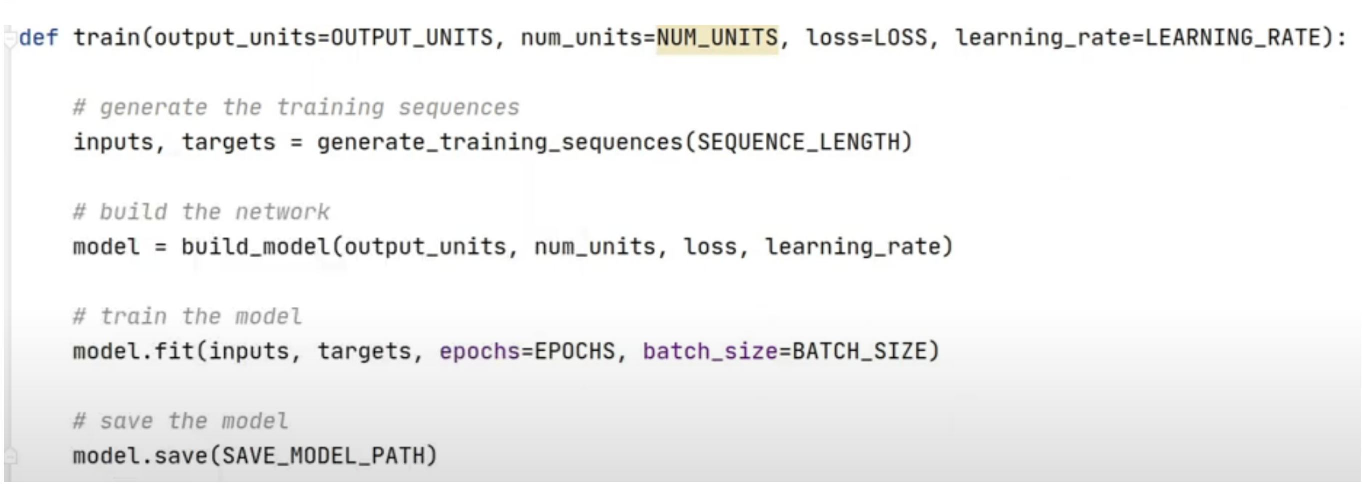
```python
import tensorflow.keras as keras
from preprocess import generate_training_sequences, SEQUENCE_LENGTH

OUTPUT_UNITS = 38
NUM_UNITS = [256]
LOSS = "sparse_catecorical_crossentropy"
LEARNING_RATE = 0.001
EPOCHS = 50
BATCH_SIZE = 64
SAVE_MODEL_PATH = "model.h5"
```



```python
def build_model(output_units, num_units, loss, learning_rate):

    # create the model architecture
    input = keras.layers.Input(shape=(None, output_units))
    x = keras.layers.LSTM(num_units[0])(input)
    x = keras.layers.Dropout(0.2)(x)

    output = keras.layers.Dense(output_units, activation="softmax")(x)

    model = keras.Model(input, output)

    # compile model
    model.compile(loss=loss,
                  optimizer=keras.optimizers.Adam(lr=learning_rate),
                  metrics=["accuracy"])
```



```
    model.summary()

    return model
```



```
def train(output_units=OUTPUT_UNITS, num_units=NUM_UNITS, loss=LOSS, learning_rate=LEARNING_RATE):

    # generate the training sequences
    inputs, targets = generate_training_sequences(SEQUENCE_LENGTH)

    # build the network
    model = build_model(output_units, num_units, loss, learning_rate)

    # train the model
    model.fit(inputs, targets, epochs=EPOCHS, batch_size=BATCH_SIZE)

    # save the model
    model.save(SAVE_MODEL_PATH)
```