SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

COIMBATORE – 35

Creating a Neural Network with TensorFlow

Creating a Neural Network using TensorFlow, one of the most popular machine learning libraries. TensorFlow is an open-source library developed by Google that provides various tools for creating and training machine learning models.

Step 1: Install TensorFlow

The first step is to install TensorFlow using pip. Open your terminal and execute the following command:

```
pip install tensorflow
```

Step 2: Import Required Libraries

Now, let's import the necessary libraries for our project.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
```

Step 3: Prepare the Dataset

In this tutorial, we will use the MNIST dataset, which consists of 70,000 images of handwritten digits. TensorFlow provides a convenient way to load this dataset.

```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()

# Normalize the data
x_train = x_train / 255.0
x_test = x_test / 255.0
```

Step 4: Create the Neural Network Model

Now, we will create a neural network model using the Sequential API from Keras. The Sequential API allows you to create models layer-by-layer. We will use two hidden layers with 128 nodes each and an output layer with 10 nodes (one for each digit).

```
model = tf.keras.Sequential([
    layers.Flatten(input_shape=(28, 28)),
    layers.Dense(128, activation="relu"),
    layers.Dense(128, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```

Step 5: Compile the Model

After creating the model, we need to compile it by specifying the optimizer, loss function, and evaluation metric.

```
model.compile(
    optimizer="adam",
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=["accuracy"]
)
```

Step 6: Train the Model

Now, we will train the model using the fit() method. We will use 10 epochs for training.

```
model.fit(x_train, y_train, epochs=10)
```

Step 7: Evaluate the Model

After training the model, we can evaluate its performance on the test dataset using the evaluate() method.

```
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=2)
print(f"Test accuracy: {test_acc}")
```

Conclusion

In this tutorial, we learned how to create a neural network with TensorFlow using the Sequential API. We also covered how to train and evaluate the model on the MNIST dataset. By following these steps, you can create your own neural network models and experiment with different architectures and parameters.