

Basic Image Classification with TensorFlow

Python Sample Code

Libraries

Load the TensorFlow library.

```
import tensorflow as tf
```

Dataset

Load the MNIST dataset and scale the pixel values to the range of [0,1].

```
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

Artificial Neural Network Model

Define a fully-connected feedforward artificial neural network architecture with a single hidden layer composed of 128 neurons. Set the dropout rate at 20%.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
```

Define the loss function, the optimizer, and the evaluation metric for the training.

```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
model.compile(optimizer='adam',
              loss=loss_fn,
              metrics=['accuracy'])
```

Model Training and Evaluation

Fit the model on the training set.

```
model.fit(x_train, y_train, epochs=5, verbose=2)
```

```
## Epoch 1/5
## 1875/1875 - 2s - loss: 0.2946 - accuracy: 0.9138 - 2s/epoch - 1ms/step
## Epoch 2/5
## 1875/1875 - 2s - loss: 0.1422 - accuracy: 0.9585 - 2s/epoch - 1ms/step
## Epoch 3/5
## 1875/1875 - 2s - loss: 0.1066 - accuracy: 0.9684 - 2s/epoch - 1ms/step
## Epoch 4/5
## 1875/1875 - 2s - loss: 0.0873 - accuracy: 0.9732 - 2s/epoch - 1ms/step
## Epoch 5/5
## 1875/1875 - 2s - loss: 0.0734 - accuracy: 0.9772 - 2s/epoch - 1ms/step
## <keras.callbacks.History object at 0x000001DA389145E0>
```

Evaluate the classification performance of the model on the test set.

```
model.evaluate(x_test, y_test, verbose=2)
```

```
## 313/313 - 0s - loss: 0.0735 - accuracy: 0.9764 - 320ms/epoch - 1ms/step
## [0.0735146775841713, 0.9764000177383423]
```