

Tensor Flow API Reference

Hans C. Suganda

9th August 2021

Rationale

The goal of this document is to define basic tasks and execute said basic tasks. The tasks serve as the currently most efficient known way of familiarizing oneself with the api calls of the neural network library tensorflow.

Basic Terminology

Loss function refers uses the neural network outputs and supposed output to produce a loss score. The loss score is then fed to the optimizer which then changes all of the weights.

Loading Data

Mnist database is a large database of handwritten digits. Loading and initializing two variables, to the training data is shown below,

```
from keras.datasets import mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

train_images and test_images would be numpy arrays of 28×28 numpy arrays whose elements hold the pixel value.

Specifying Architecture

Specify the Architecture of the Neural Network: An example of of specifying how many hidden layers along with the various activation functions is shwon below,

```
from keras import models
from keras import layers

network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
```

Specifying Loss Function

Training and Evaluation

The api command that is used to train the neural network,

```
network.fit(train_images, train_labels, epochs=5, batch_size=128)
```

The api command that is used to find the accuracy of the neural network over a range of test cases,

```
test_loss, test_acc = network.evaluate(test_images, test_labels)
```

wherein the test_loss variable represents the test loss, and is somewhat arbitrary. The test_acc variable holds the value of the test accuracy and this is expressed in decimal points, ranging from 0 till 1.

$$0 = x^2 + x + 3$$