In [1]:

```
import tensorflow as tf
```

In [2]:

```
tf.test.is_gpu_available()
```

Out[2]:

```
True
```

In [3]:

```
tf.__version__
```

Out[3]:

```
'2.0.0'
```

In [4]:

```
from tensorflow.keras import layers, optimizers, metrics, datasets, Sequential
```

In [5]:

```
import os
```

In [7]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [8]:

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

In [9]:

```
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

Out[9]:

```
((60000, 28, 28), (60000,), (10000, 28, 28), (10000,))
```

In [10]:

```
x_train.min(), x_train.max(), y_train.min(), y_train.max()
```

Out[10]:

```
(0, 255, 0, 9)
```

In [13]:

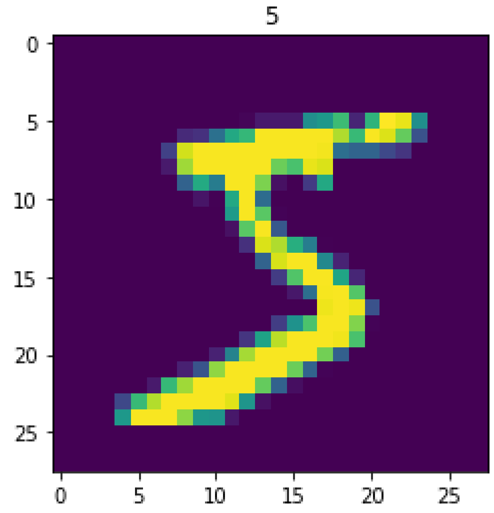
```
sample_img = x_train[0]
sample_label = y_train[0]
```

In [14]:

```
plt.imshow(sample_img)
plt.title(str(sample_label))
```

Out[14]:

```
Text(0.5, 1.0, '5')
```



In [18]:

```
x_train = tf.convert_to_tensor(x_train, dtype=tf.float32) / 255
x_test = tf.convert_to_tensor(x_test, dtype=tf.float32) / 255
```

In [19]:

```
y_train = tf.one_hot(y_train, depth=10)
```

In [20]:

```
y_test = tf.one_hot(y_test, depth=10)
```

In [22]:

```
x_train = tf.reshape(x_train, (-1, 28*28))
```

In [23]:

```
x_test = tf.reshape(x_test, (-1, 28*28))
```

In [24]:

```
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

Out[24]:

```
(TensorShape([60000, 784]),
 TensorShape([60000, 10]),
 TensorShape([10000, 784]),
 TensorShape([10000, 10]))
```

In [27]:

```
tf.reduce_min(x_train), tf.reduce_max(x_train)
```

Out[27]:

```
(<tf.Tensor: id=152, shape=(), dtype=float32, numpy=0.0>,
 <tf.Tensor: id=154, shape=(), dtype=float32, numpy=1.0>)
```

In [15]:

```
model = Sequential([
    layers.Dense(512, activation='relu', input_shape=(28*28, )),
    layers.Dense(10, activation='softmax')
])
```

In [16]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense (Dense)                (None, 512)               401920    
_________________________________________________________________
dense_1 (Dense)              (None, 10)                5130      
=================================================================
Total params: 407,050
Trainable params: 407,050
Non-trainable params: 0
_________________________________________________________________
```

In [17]:

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

In [28]:

```
model.fit(x_train, y_train, epochs=5, batch_size=128)
```

```
Train on 60000 samples
Epoch 1/5
60000/60000 [==============================] - 2s 27us/sample - loss: 0.2682
- accuracy: 0.9250
Epoch 2/5
60000/60000 [==============================] - 1s 14us/sample - loss: 0.1103
- accuracy: 0.9682
Epoch 3/5
60000/60000 [==============================] - 1s 14us/sample - loss: 0.0706
- accuracy: 0.9797
Epoch 4/5
60000/60000 [==============================] - 1s 14us/sample - loss: 0.0516
- accuracy: 0.9845
Epoch 5/5
60000/60000 [==============================] - 1s 14us/sample - loss: 0.0362
- accuracy: 0.9897
```

Out[28]:

```
<tensorflow.python.keras.callbacks.History at 0x1e6857ce5f8>
```

In [29]:

```
loss, acc = model.evaluate(x_test, y_test)
```

```
10000/1 [================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
=========================================================================
```

In [30]:

```
loss, acc
```

Out[30]:

```
(0.0645063339161221, 0.9804)
```

In [ ]: