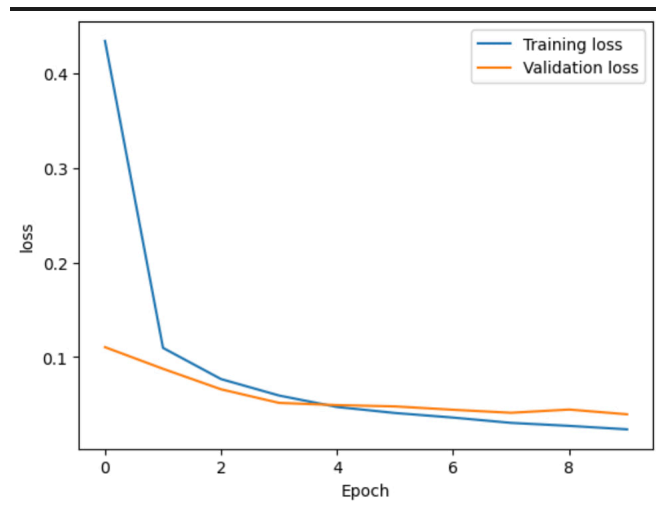
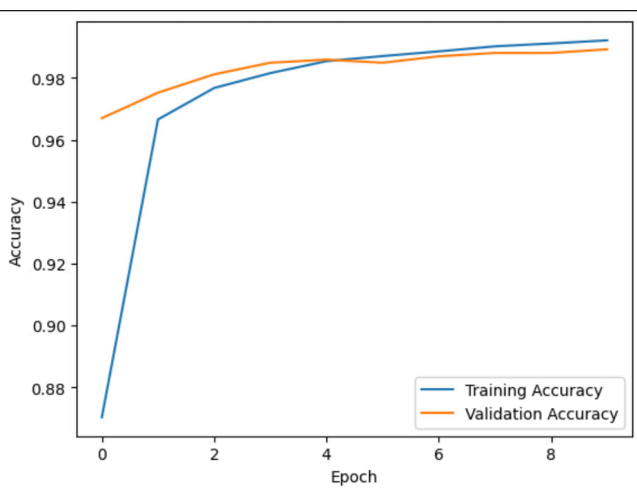
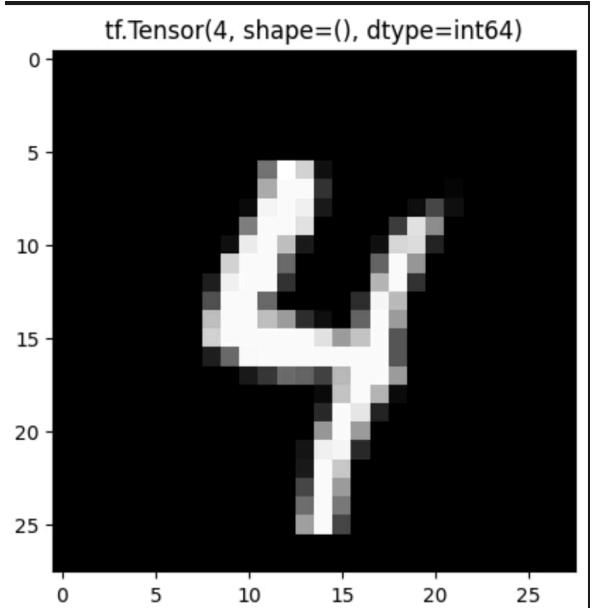


```

1 ~ # %%
2 ~ import tensorflow as tf
3   from datasets import load_dataset
4   import matplotlib.pyplot as plt
5   data = load_dataset('mnist', data_dir='./')
6   data = data.with_format("tf")
7   tdata = data['train']
8   vdata = data['test']
9   image = tdata['image']/255
10  label = tdata['label']
11 ~ model = tf.keras.Sequential([
12    tf.keras.layers.InputLayer(input_shape=(28, 28, 1)),
13    tf.keras.layers.Conv2D(8, 3, padding='SAME', activation='relu'),
14    tf.keras.layers.MaxPool2D(),
15    tf.keras.layers.Conv2D(16, 3, padding='same', activation='relu'),
16    tf.keras.layers.MaxPool2D(),
17    tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu'),
18    tf.keras.layers.Flatten(),
19    tf.keras.layers.Dense(128, activation='relu'),
20    tf.keras.layers.Dropout(0.2),
21    tf.keras.layers.Dense(10, activation='softmax')
22  ])
23
24 ~ model.compile(optimizer=tf.optimizers.Adam(),
25                loss=tf.losses.SparseCategoricalCrossentropy(),
26                metrics=['accuracy'])
27
28  model.summary()
29  Run Cell | Run Above | Debug Cell
30 ~ # %%
31  h = model.fit(image, label, epochs=10, batch_size=256, validation_split=0.2)

```



Five Activation functions:

Relu  
Elu  
LeakyRelu  
softmax  
sigmoid

categorical cross entropy is the sum over m sample in a batch  $y * \log y\_pred$  it is usually used for classification problems. y is one hot vector the output of the model but in this exercise, we don't have one hot label so we use sparse categorical cross entropy.

Adam is an optimizer like sgd or sgd with the momentum it has the adaptive learning rate for every weight.

epoch means we go through all batches and train the network for all batches then we say we trained it for one epoch. ten epochs means we went through all training batches ten times and computed loss and gradient to train the model