Question 2:

my gate is called Leaky Relu, which is defined as follows:

```
# leaky Relu
def my_gate(z):
  return tf.maximum(0.1*z,z)
```

(b)

For fully connected model:

Relu Gate does the best job. The train loss, validation error and test error gained by Relu Gate are the minimum values among five different gates. Leaky Relu Gate also does nice job, the result is just a little bit worse than Relu Gate, but still very good. Softplus Gate model is the worst model among these models. The test error gained by Softplus Gate model is significantly higher than other models.

Training and testing performance are almost monotonically related. Only one exceptional case is that the test error Sigmoid Gate gets finally is a little bit smaller than Softplus Gate, but its train loss is larger than the latter one.

(d)

For convolutional model:

Relu Gate and Leaky Relu Gate do equally best job. The train loss, validation error and test error gained by Relu Gate and Leaky Relu Gate are similar and the minimum values among five different gates. Tanh Gate also does nice job, the result is just a little bit worse than Relu Gate and Leaky Relu Gate, but still very good. Softplus Gate model and Sigmoid Gate model are the worst models among them. The test errors gained by these two model are significantly higher than the other three models.

Training and testing performance are perfectly monotonically related. The model who gets smaller train loss also gets smaller test error.

(e)

When compare the results between the fully connected and convolutional models, one interesting phenomena is that all convolutional models with different gates get better results (lower error) in validation and test data compared to fully connected models, even the worst model in convolutional models can beat the best model in fully connected models. However, some convolutional models get higher train loss than fully connected models, while others get lower train loss than fully connected models.