**Lesson 19 (Neural Network Playground–3)**

**Under-Fitting Using Logistic Regression:** In this lesson we see what happens when logistic regression is used to solve a challenging classification task.

1. Go to playground.tensorflow.org and set-up configuration 3 shown in the table below.

| configuration | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| data | 2-cluster | 2-cluster | spiral | |
| train-test split | 50% | 50% | 50% | |
| noise | 0 | 50 | 0 | |
| batch size | 10 | 30 | 30 | |
| features | $x_1, x_2$ | $x_1, x_2$ | $x_1, x_2$ | |
| hidden layers | 0 | max | 0 | |
| nodes | 0 | max | 0 | |
| learning rate | 0.03 | 0.03 | 0.03 | |
| activation function | linear | ReLU | linear | |
| regularization | none | none | none | |
| regularization rate | 0 | 0 | 0 | |
| problem type | classification | classification | classification | |
| discretize output | yes | yes | yes | |

2. Click the REGENERATE button in the bottom left corner to generate a new data set. Then begin training the network by clicking the play button in the upper left corner.

   (a) What is the final training error?

   (b) What is the final testing error?

3. Click the rewinded button to reset the neural network. Repeat parts 2, but use the maximum number of hidden layers and maximum number of nodes per layer. Is it important to use nonlinear activiation functions? What happens when the activation functions are linear?