Using a Graph Database as a Neural Network

Exercise: Training a neural network to recognize handwritten digits

In this exercise, we take 5000 grayscale images of handwritten digits, each image 20 x 20 pixels and then linearized into a vector, to train a neural network to recognize digits. This exercise is taken directly from an exercise in Andrew Ng's Coursera online course on Machine Learning.

Data Set: 5000 records, where each record has 402 fields: 1 for the id, 400 for the pixels and 1 for the label. The label is the digit represented by the image: 0,1,2,3,4,5,6,7,8, or 9.

Neural Network structure and functions:
It is beyond the scope of this workshop to discuss the computational functions performed by the graph nodes (neurons). Interested parties can referred to technical details in the presentation slides.

Graph Schema:

- 3 vertex types:
  - inputLayer
  - hiddenLayer
  - outputLayer
- 2 edge typs:
  - Theta1 - connecting inputLayer to hiddenLayer
  - Theta2 - connecting hiddenLayer to outputLayer

Instructions:

1. In a Linux shell for your TigerGraph machine, change to the neural_network folder:
   `cd /home/tigergraph/usecases/neural_network`

2. Create the graph schema for the NeuralNetwork graph:
   `bash nn_schema.sh`

3. Load the graph data into the NeuralNetwork graph:
   `gsql -g NeuralNetwork nn_load.gsql`

4. Add the expression functions by copying the ExprFunctions.hpp file to the TigerGraph folder for query user-defined functions:
   `cp ExprFunctions.hpp /home/tigergraph/tigergraph/dev/gdk/gsql/src/QueryUdf`

5. Install the queries:

`bash nn_install_queries.sh`

6. To train the model, run the command below. This .sh script will run the backpropagation_validation.gsql query to train the neural network and also print the cost function for training and validation data for each iteration. It will also run the prediction_accuracy.gsql query to show the prediction accuracy on validation data for each iteration.
   `bash nn_train.sh`

7. To use our trained neural network to recognize a digit you draw, you can draw a digit yourself using an online sketchpad: https://sketch.io/sketchpad/. Create a new image which is at least 300 x 300. Use the pen and a line width in proportion to your canvas (at least 2% of the image size). After drawing the digit, you can export the drawing as a .jpeg file using the *Export* icon at the bottom left of the page. Save it under the `neural_network/data/prediction` directory. (You are recommended to draw your digit at the very center of the sketchpad for better prediction accuracy)

8. Run image2vec.py using the command below. This will convert a 1362 by 2454 jpeg image to a 20 by 20 grayscale image and save the result to Val.csv.
   `python3 image2vec.py data/prediction/<jpeg image name>`
   The output will be in the same folder as the jpeg file, but with the suffix changed to from .jpeg to .csv.

9. Run the command below to load the Val.csv into TigerGraph and run the prediction.gsql query. This will show a 0 to 1 value for each one of the digits (0~9). The higher the value is, the more likely the corresponding digit is.
   `bash nn_prediction.sh <path to csv file>`
   such as
   `bash nn_prediction_.sh data/prediction/test_3.csv`