EE 511 – Assignment 4

Due: Feb. 18th, 2019

1. **Neural Network Warm-up Problem**

   In this problem we will generate some synthetic data according to some simple rules and then fit a neural network to the data.

   To generate the data sample points x_1 and x_2 uniformly from the range $x_1 \in [-1.0, 1.0]$ and $x_2 \in [-1.0, 1.0]$. The labels can be assigned as follows:

   - If $0.4 < |x_1| + |x_2| < 0.7$ assign label 0.
   - Else, if $\sqrt{x_1^2 + x_2^2} < 0.3$ assign label 1.
   - Else, if $\sin(10.0x_2) < 0$ assign label 2.
   - Else, if $\sin(5.0x_1) > 0$ assign label 3.
   - Else, discard this point.

   Note that the above criteria are not linear functions, so you need a non-linear function to classify the data.

   (a) Generate 500 samples each for training and validation sets and an additional 1,000 points to use for the test set. What is the relative frequency of each class in the training data?

   (b) Train a neural network with one hidden layer to classify the data. You can use the validation data for parameter selection, such as choosing the learning rate, the size of the hidden layer, regularization, etc. You don't have to exhaustively explore all the options if you are able to find a good model without doing that. Make sure to describe in your report what options you explored and what model you ended up choosing.

   (c) Report the overall accuracy of the classifier on the test data.

   (d) Sample points on a grid over the space and make two plots: one showing the true lable at each point and a second showing the label predicted at each point. Comment on what you observe and whether it matched or was different from what you expected.

2. **Autoencoder**

   An autoencoder is a neural network used for non-linear dimensionality reduction. The network is trained to minimize the error between the input and its output, $||X - f(X)||^2$. The key to the dimensionality reduction is to have a hidden layer, known as the bottleneck, with a small number of dimensions. After training the model, if we throw out all of the layers after the bottleneck then we are left with a network that outputs a reduced representation of the input. The bottleneck vectors can be used for data compression or for unsupervised feature extraction.

(a) For the problem we will be using images of galaxies from the Galaxy Zoo project. Download the images from Canvas. You should be able to see subfolders that contain train, validation and test sets, respectively.

(b) Load the images. Resize them to 20x20 pixels and convert them to grayscale. If you are using Python, some code will be posted to Canvas to help you with this part.

(c) You will be comparing your autoencoder against principal component analysis (PCA), a linear dimensionality reduction technique. There's a function in sklearn that you can use – no need to implement it yourself. So, let's do this part first. Use PCA to reduce the images down to 25 dimensions and then reconstruct them again. Compute the reconstruction error in terms of the squared error per pixel on the validation and the test set.

(d) Train an autoencoder with a 25-dimensional bottleneck layer. You can try using different activation functions, learning rates, regularization, number of hidden layers, etc. Measure the performance of your model using the average squared error per pixel on the validation set.

Often, the sizes of the hidden layers will be symmetric. For example, if the input is size 400 and you want to use three hidden layers, then their sizes might be 100, 25, and 100. Note: it might help to divide by 255 to scale the input and then invert the scaling after you reconstruct the image.

(e) Using the test data, compare the error from the autoencoder with the error from using PCA. Note that PCA tends to work really well. It's great if your autoencoder beats PCA. If not, it should be somewhat close in performance.

(f) Find an image in the test set where the autoencoder beats PCA. Show the original image, the PCA reconstruction, and the output from the autoencoder side-by-side. Do the same for an image where PCA gives a better reconstruction than the autoencoder. Comment on whether there are qualities of an image that are better suited to one technique or the other.