BIOEN 498J Final Report
MATLAB Character Classification

Parker Ruth

Background

I began my project with the goal of building an automated handwritten digit classifier. Rather than attempt to code explicit instructions for recognizing specific digits, I chose to use machine learning (ML). Specifically, I chose a feedforward neural network as my ML architecture. A feedforward neural network is a directed acyclic graph with weighted connections between its nodes; the nodes are arranged in layers (1 input layer, any number of hidden layers, and one output layer). Every node is associated with an activation level (a real number). The output layer activation is a function of the input layer activation calculated by setting the activation of each layer to the weighted sum of its inputs[1].

Neural networks are universal function approximators [1], meaning that with sufficiently many nodes, a neural network can approximate any function of its inputs to arbitrary precision. However, this requires assigning weights to the layer connections. The most general method for assigning weights is backpropagation. In this method, weights are initially randomized, and known input output pairs are presented to the network; differences between the network's output and the target value are propagated backwards as small changes to the weights. In principle, this method uses gradient descent to minimize the training error with respect to the weights. Repeating this process with many examples until the weights converge is called "trainings" the network. The accuracy of the network can be evaluated by testing its output with a separate training dataset. In general, training a neural network requires a volume of data commensurate with the complexity of the target function.

Handwritten Character Classification

To classify images of handwritten characters, I chose to use a specialized network architecture called a convolutional neural network (CNN). CNNs operate on the same principles as above, with the additional constraint that the first several layers have consolidated weights that compute banks of convolutions over the input image. This reduces the dimensionality of the input and captures the spatial relationships between adjacent pixels. Applying this convolutional neural network to the MNIST dataset, a collection of over 70,000 labeled images of handwritten digits, yielded 99.8% accuracy on the testing data. I also applied the same neural network to the extended MNIST (EMNIST) database [2], including upper and lowercase letters as well as digits; this led to 81.65% accuracy on testing data, which I

[1] Since any composition of weighted sums is a linear function of the inputs, the activation of each neuron is normalized by a nonlinear activation function (e.g. a sigmoid function $\sigma(x) = \frac{1}{1-e^{-x}}$) to allow the network to learn nonlinear models; otherwise, the network is equivalent to a single layer and is called a perceptron.

estimate to be close to human performance, since upper and lower case ‘O’, and zero can all be easily confused with each other.

Star Wars Character Classification

The encouraging results from the MNIST dataset encouraged me to progress to a more challenging problem, namely that of classifying images of Star Wars characters. To prepare my testing and training data, I used a Google Chrome extension to download the top image search results for given search terms; the dataset consisted of over 300 labeled images of Star Wars characters according to five classes: vader, chewy, yoda, r2d2, and c3po. However, 300 images were insufficient to train an image classification model from scratch. Thus, I repurposed a pre-trained neural network called AlexNet[3], which was originally trained with the massive online ImageNet dataset [4], consisting of over 14 million labeled images with over 1000 different classes (most animals, foods, vehicles, and common household objects). To adapt ImageNet to classify Star Wars characters, I replaced the last dense fully connected layer from AlexNet, while retaining all of the remaining architecture and trained weights, and fine tuned the network by training with my Star Wars character dataset. The resulting CNN, which I call VaderNet, has an accuracy of 98%.

Discussion

Convolutional neural networks are a powerful mechanism for building and deploying image classification programs, and the Neural Networks Toolbox is a powerful library for prototyping neural network models. However, the usability comes at the price of flexibility; custom neural network architectures and training methods are difficult if not impossible to implement without starting from scratch. Nonetheless, the MATLAB ecosystem is well suited for most mainstream applications of neural networks.

References

[1] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.

[2] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.