Computer Vision and Image Processing
Homework 4: Autoencoders for Image Classification

Instructor: Kevin R. Keane
TAs: Radhakrishna Dasari, Yuhao Du, Niyazi Sorkunlu

Due Date: *December 4, 2017*

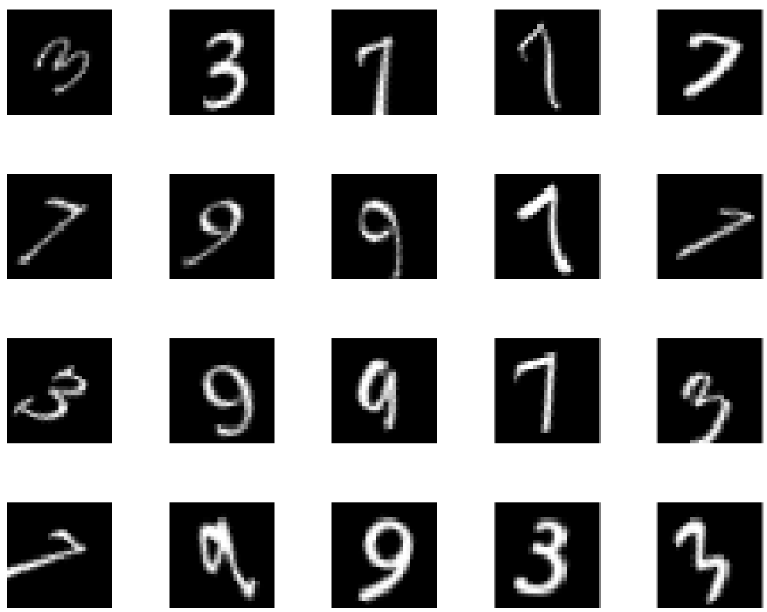

Figure 1: Synthetic images generated by applying random affine transformations to digit images created using different fonts (MathWorks, 2017).

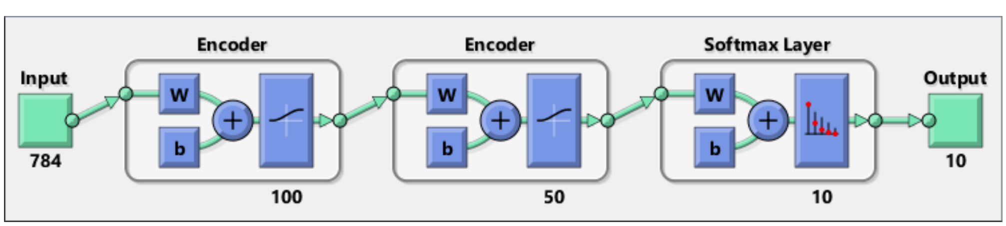

Figure 2: A neural network for classifying $28 \times 28 = 784$ pixel images containing a digit (MathWorks, 2017).

1 Overview

The goal of this assignment is to use the MATLAB Neural Network Toolbox autoencoders for training a deep neural network to classify images of digits. Deep learning has quickly become one of the most applied machine learning techniques in computer vision. Neural networks have been applied to many different computer vision problems such as image classification, recognition, and segmentation with great success. In this assignment, you will utilize a fully connected feed forward neural network for digit classification. Given an image of a digit, the network classifies the image into one of ten classes corresponding to the digits 0-9.

1.1 Reading assignments

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010

The tutorial may be found here.(MathWorks, 2017) You will need to install the Neural Network Toolkit if you have not done so already.

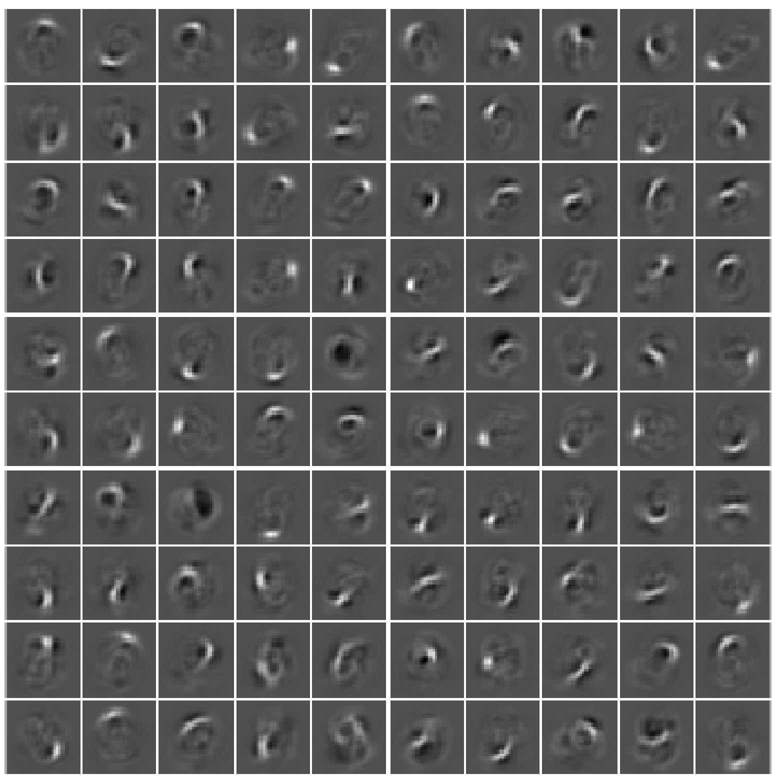

Figure 3: Features from hidden layer 1.

1.2 Questions

Q1 (8 points) How does an autoencoder detect errors?

Q2 (8 points) The network starts out with $28 \times 28 = 784$ inputs. Why do subsequent layers have fewer nodes?

Q3 (8 points) Why are autoencoders trained one hidden layer at a time?

Q4 (8 points) How were the features in Figure 3 obtained? Compare the method of identifying features here with the method of HW1. What are a few pros and cons of these methods?

Q5 (8 points) What does the function `plotconfusion` do?

1.2.1 Activation Functions

Q6 (8 points) What activation function is used in the hidden layers of the MATLAB tutorial?

Q7 (8 points) In training deep networks, the ReLU activation function is generally preferred to the sigmoid activation function. Why might this be the case?

1.2.2 Initialization

Q8 (8 points) The MATLAB demo uses a random number generator to initialize the network. Why is it not a good idea to initialize a network with all zeros? How about all ones, or some other constant value? (Hint: Consider what the gradients from backpropagation will look like.)

1.2.3 Training Loop

When training a neural network using Gradient Descent, there are two options one can take for the updates, batch or stochastic updates. In batch gradient descent, the gradient is computed for all the examples in the training set and then averaged before updating the weights. In stochastic, the gradient is computed with one sample and the weights are updated with that gradient. Each update is called a iteration, one complete pass through the entire dataset is called an epoch. In case of stochastic gradient descent, number of iterations in an epoch equals the number of training samples.

Q9 (8 points) Give pros and cons for both stochastic and batch gradient descent. In general, which one is faster to train in terms of number of epochs? Which one is faster in terms of number of iterations?

1.2.4 Exploration

Q10 (28 points) Try playing around with some of the parameters specified in the tutorial. Perhaps the sparsity parameters or the number of nodes; or number of layers. Report the impact of slightly modifying the parameters. Is the tutorial presentation robust or fragile with respect to parameter settings?

1.3 Extra Credit

Dowload and process the MNIST data as described here.(Fatahi, 2014) The data set is in a slightly unusual format. Loading the data to MATLAB is a good portion of the difficulty in this extra credit task.

Q (50 points) (Partial credit will be given. Report what you attempt, what you expected, and your actual results even if not fully "successful".)

- Load the data to MATLAB and inspect a few of your images using tools you learned about this semester. Upscale the images you inspect so that you can actually see them.
- Repeat the steps of the tutorial with "real" images.
- Compare the results of the synthetic images with the results of "real" images. Are the features similar in appearance? Is performance better or worse than observed with the synthetic dataset?
- Provide a confusion matrix for your final classifier.
- Document your network architecture, parameters settings, and number of training and test images used.

Grading checklist

Instructions for turning in the assignment

As usual, submit your PDF report named `personNumber.pdf` via UB*learns*. There is no requirement for code submission with this homework. If you try an unusual variant and choose to submit code, please submit a file named `personNumber.zip` via UB*learns*. Multiple attempts will be allowed but by default, only your last submission will be graded.

Late policy: You lose 50% of the points for every day the assignment is late (after expending your free late days). Any genuine emergency situations will be handled on an individual basis.

Academic integrity: Feel free to discuss the assignment with each other in general terms, and to search the Web for general guidance (not for complete solutions). The MATLAB tutorial, question responses, and extra credit should be done individually. If you make substantial use of some code snippets or information from outside sources, be sure to acknowledge the sources in your report.

References

Mazdak Fatahi. MNIST handwritten digits. Online., 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

MathWorks. Train stacked autoencoders for image classification. Online., 2017.