EEE 443/543: Neural Networks

Mini Project

(Due 2 May 2022, 23:59PM)

Instructions:

1. Prepare a report (including your answers/plots) to be uploaded on Moodle.
2. The report should be typeset (no handwriting allowed except for lengthy derivations, which may be scanned and embedded into the report).
3. Show all steps of your work clearly.
4. Unclear presentation of results will be penalized heavily.
5. No partial credits for unjustified answers.
6. **Use of any toolbox or library for neural networks is prohibited**.
7. Return all Matlab/Python code that you wrote in a single `.m/.py` file.
8. Code should be commented, code for different HW questions should be clearly separated.
9. The code file should NOT return an error during runtime.
10. If the code returns an error at any point, the remaining part of your code will not be evaluated (i.e., 0 points).

Question	Points	Your Score
Q1	35	
Q2	30	
Q3	35	
TOTAL	100	

Question 1. [35 points]

In this question you will implement an autoencoder neural network with a single hidden layer for unsupervised feature extraction from natural images. The following cost function will be minimized:

$$J_{ae} = \frac{1}{2N}\sum_{i=1}^{N}\|d(m) - o(m)\|^2 + \frac{\lambda}{2}\left[\sum_{b=1}^{L_{hid}}\sum_{a=1}^{L_{in}}\left(W_{a,b}^{(1)}\right)^2 + \sum_{c=1}^{L_{out}}\sum_{b=1}^{L_{hid}}\left(W_{b,c}^{(2)}\right)^2\right] + \beta\sum_{b=1}^{L_{hid}} KL(\rho|\hat{\rho}_b) \tag{1}$$

The first term is the average squared-error between the desired response and the network output across training samples. Note that the desired output is the same as the input. The second term enforces Tykhonov regularization on the connection weights with parameter λ. The last term enforces that the hidden unit activations are sparse with parameter β for controlling the relative weighting of this term. The level of sparsity is tuned via ρ in the KL term (Kullback-Leibler divergence) between a Bernouilli variable with mean ρ and another with mean $\hat{\rho}_b$. $\hat{\rho}_b$ is the average activation of hidden unit b across training samples.

a) The file `data1.h5` contains a collection of 16×16 RGB patches extracted from various natural images in `data`. Preprocess the data by first converting the images to graysale using a luminosity model: $Y = 0.2126 * R + 0.7152 * G + 0.0722 * B$. To normalize the data, first remove the mean pixel intensity of each image from itself, and then clip the data range at ±3 standard deviations (measured across all pixels in the data). To prevent saturation of the activation function, map the ±3 std. data range to [0.1 0.9]. Display 200 random sample patches in RGB format, and separately display the normalized versions of the same patches. Comment on your results.

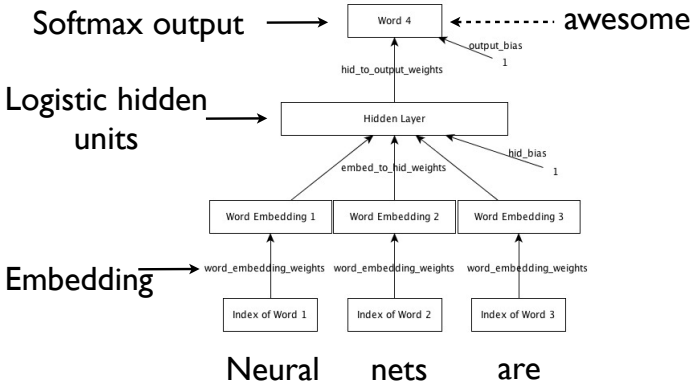
b) Prior to training, initialize the weights and the bias terms as uniform random numbers from the interval $[-w_o, w_o]$, where $w_o = sqrt(\frac{6}{Lpre+L_{post}})$ and $L_{pre,post}$ are the number of neurons on either side of the connection weights. Write a cost function for the network $[J, J_{grad}] = aeCost(W_e, data, params)$ that calculates the cost and its partial derivatives. $W_e = [W_1\ W_2\ b_1\ b_2]$, a vector containing the weights for the first and second layers followed by the bias terms; *data* is of size $L_{in} \times N$; *params* is a structure with the following fields *Lin* (L_{in}), *Lhid* (L_{hid}), *lambda* (λ), *beta* (β), *rho* (ρ). Use J and J_{grad} as inputs to a gradient-descent solver to minimize the cost. Assuming $L_{hid} = 64$, $\lambda = 5 \times 10^{-4}$, experiment with β, ρ to find parameters that work well. Note that performance here is defined based on the 'quality' of the features extracted by the network.

c) The solver will return the trained network parameters. Display the first layer of connection weights as a separate image for each neuron in the hidden layer. What do the hidden-layer features look like? Are these features representative of natural images?

d) Retrain the network for 3 different values (low, medium, high) of $L_{hid} \in [10\ 100]$, of $\lambda \in [0\ 10^{-3}]$, while keeping β, ρ fixed. Display the hidden-layer features as separate images. Comparatively discuss the results you obtained for different combinations of training parameters.

Question 2. [30 points]

Neural network architectures can produce powerful computational models for natural language processing. Here, you will consider one particular model for examining sequences of words. The task is to predict the fourth word in sequence given the preceding trigram, e.g., trigram: 'Neural nets are', fourth word: 'awesome'. A database of articles were parsed to store sample fourgrams restricted to a vocabulary size of 250 words. The file `data2.h5` contains training samples for input and output (`trainx`, `traind`), for validation (`valx`, `vald`), and for testing (`testx`, `testd`). Using these samples, the following network should be trained via backpropagation:



The input layer has 3 neurons corresponding to the trigram entries. An embedding matrix R (250×D) is used to linearly map each single word onto a vector representation of length D. The same embedding matrix is used for each input word in the trigram, without considering the sequence order. The hidden layer uses a sigmoidal activation function on each of P hidden-layer neurons. The output layer predicts a separate response z_i for each of 250 vocabulary words, and the probability of each word is estimated via a soft-max operation ($o_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$).

a) Assume the following parameters: a stochastic gradient descent algorithm, a mini-batch size of 200 samples, a learning rate of $\eta = 0.15$, a momentum rate of $\alpha = 0.85$, a maximum of 50 epochs, and weights and biases initialized as random Gaussian variables of std 0.01. If necessary, adjust these parameters to improve network performance. The algorithm should be stopped based on the cross-entropy error on the validation data. Experiment with different D and P values, (D,P) = (32,256), (16,128), (8,64) and discuss your results.

b) Pick some sample trigrams from the test data, and generate predictions for the fourth word via the trained neural network. Store the the predicted probability for each of the 250 words. For each of 5 sample trigrams, list the top 10 candidates for the fourth word. Are the network predictions sensible?

Question 3. [35 points]

In this question we will consider classifying human activity (downstairs=1, jogging=2, sitting=3, standing=4, upstairs=5, walking=6) from movement signals measured with three sensors simultaneously. The file `data3.h5` contains time series of training and testing data (trX and tstX), and their corresponding labels (trY and tstY). The length of each time series is 150 units. The training set consists of 3000 samples, and the test set consists of 600 samples. You are going to implement fundamental recurrent neural network architectures, trained with back propagation through time to solve a multi-class time series classification problem.

a) Using the back propagation through time algorithm, implement a single layer recurrent neural network with 128 neurons and hyperbolic tangent activation function, followed by a multi-layer perceptron with a softmax function for classification. Use: a stochastic gradient descent algorithm, mini-batch size of 32 samples, learning rate of $\eta = 0.1$, momentum rate of $\alpha = 0.85$, maximum of 50 epochs, and weights/biases initialized with Xavier Uniform distribution. Adjust the parameters, and number of hidden layers of the classification neural network to improve network performance. The algorithm should be stopped based on the categorical cross-entropy error on a validation data (10% samples selected from the training data). Report the following: Validation error as a function of epoch number, accuracy measured over the test dataset, confusion matrix for the training and test set, and discussion of your results.

b) For the time-series data, it is vital to summarize the past observations in the hidden state and to control this information. For this reason, we consider a better alternative which is a long-short term memory or LSTM neural network. Repeat part a for LSTM. Report the following: Validation error as a function of epoch number, accuracy measured over the test set, confusion matrix for the training and test set, discussion of your results, and comparison with the performance in part a?

c) Finally, we consider an alternative to LSTM neural networks, called gated recurrent units (GRU in short). Repeat part a for GRU. Report the following: Validation error as a function of epoch number, accuracy measured over the test set, confusion matrix for the training and test set, discussion of your results, and comparison with the performance in parts a and b?