UNIVERSITY OF CALIFORNIA, DAVIS
DEPARTMENT OF COMPUTER SCIENCE

ECS 171: Homework Set 2

Instructor: Ilias Tagkopoulos
TAs: Ameen Eetemadi, Jason Youn, ChengEn Tan
{eetemadi, jyoun, cetan}@ucdavis.edu

Homework is due on November 7, 2019

General Instructions: The homework should be submitted electronically through Canvas. Each submission should be a zip file that includes the following: (a) a report in pdf format ("report_HW2.pdf") that includes your answers to all questions, plots, figures and any instructions to run your code, (b) the python code files. Please note: (a) do not include any other files, for instance files that we have provided such as datasets, (b) each function should be written in a separate file, with the appropriate remarks in the code so it is generally understandable (what it does, how it does it), (c) do not use any toolbox unless is it explicitly allowed in the homework description. Shared/copied code from any source is not allowed, as it is considered plagiarism. There is a 20% penalty per day for a late submission.

1 WHERE DID THE BAKER GO? [100PT]

In this exercise, you will build a classifier that can find the localization site of a protein in yeast, based on 8 attributes (features). You will use the "Yeast" dataset ("yeast.data" file; 1484 proteins, 8 features, 10 different classes; no missing data) that is available in the UCI Machine Learning Repository:

https://archive.ics.uci.edu/ml/machine-learning-databases/yeast/

You can use scikit-learn, Keras and Tensorflow, we also recommend to use a notebook (for example Jupyter notebook). Report (code and results) the following:

1. Read about outlier detection algorithms (e.g. one-class SVM, LOF, Isolation Forest, etc.) and perform at least two of the methods to this dataset. Answer the following questions: (a) are there any outliers on the dataset?, (b) do the methods agree and why?, (c) what are the assumptions behind each method? Remove any outliers (using a single method) and then continue with this new, revised dataset. [10pt]

2. Construct a 4-layer artificial neural network (ANN) and specifically a feed-forward multilayer perceptron (with sigmoid activations and MSE loss function) to perform multi-class classification. The two hidden layers should have 3 nodes each. Split your data into a random set of 66% of the samples as the training set and the rest 34% as the testing set. Please note that you will never train with the testing set; the ANN will only take into account the training set for updating the weights. For the most popular class "CYT", provide 2 plots: (I) weight values per iteration for the last layer (3 weights and bias), (II) training and test error per iteration. Use stochastic gradient descent with back-propagation. When reporting error, use the ratio of misclassified samples [20pt]

3. Now re-train the ANN with all your data (all 1484 samples). What is your training error? Provide the final activation function formula for class "CYT" after training (this includes the functional form and corresponding weights from hidden layers necessary to calculate activation of "CYT" in output layer). [10pt]

4. For the ANN that you have built (4 layers, 2 hidden layers, 3 nodes per hidden layer) calculate the first round of weight updates with back-propagation with paper and pencil for the two final layers (output to 2nd hidden layer; 2nd hidden layer to 1st hidden layer) for only the first sample. Limit this calculation to only the weights corresponding to (a) a single output node and (b) a single hidden node from 2nd hidden layer. In other words, you need to calculate only 8 weights total (that includes the bias). You can initialize all weights to zero except the weights that you are calculating which can be initialized to 1. Confirm that the numbers you calculated are the same as those produced by the code and provide both your calculations and the code output. If your calculations do not agree, find out why. Provide both calculations made by hand (scanned image and using a calculator/computer to verify the results for each step is fine) and corresponding output from the program that shows that both are in agreement. [25pt]

5. Perform a parameter sweep (grid search) on the number of hidden layers (investigate 1,2 or 3) and number of nodes in each hidden layer (investigate 3,6,9,12). Create a 3x4 matrix with the number of hidden layers as rows and the number of hidden nodes per layer as columns, with each element (cell) of the matrix representing the testing set error for that specific combination of layers/nodes. What is the optimal configuration? What you find the relationship between these attributes (number of layers, number of nodes) and the generalization error (i.e. error in testing data) to be? [20pt]

6. Which class does the following sample belong to? [5pt]

   Unknown Sample 0.52 0.47 0.52 0.23 0.55 0.03 0.52 0.39

7. Change the hidden layer activation functions to ReLU, the output layer activation to softmax, and the loss function to cross-entropy. Is this a better choice? Use the same grid search methodology as in problem 5, and provide your justification with a plot of the training and test error vs. iteration (epochs) for the architecture that achieves the lowest error in either case. [10pt]

8. Can you come up with a quantitative measure of uncertainty for each classification? What is the uncertainty for the unknown sample of the previous question? Justify your assumptions and method [5pt bonus]

GOOD LUCK!