

CSCI 447 — Machine Learning: Soft Computing

Extra Credit

Assigned: October 14, 2019

Design Document Due: November 1, 2019

Project Due: December 11, 2019

This assignment gives you the opportunity to earn up to 10% additional credit (equal to half of a programming assignment). There are some important conditions on obtaining the extra credit, however. First, you must complete both projects 3 and 4 successfully (i.e., implement and test all of the required elements). Second, you must submit a separate design document for this assignment, as well as the normal components of code, video, and paper.

As an opportunity to give you some experience with deep learning, this assignment asks you to implement various versions of a stacked auto-encoder to perform classification and regression on several data sets from the UCI Machine Learning Repository. In addition to implementing and testing the algorithms, you are required to write a research paper describing the results of your experiments.

For this assignment, you will use three classification datasets and three regression data sets that you will download from the UCI Machine Learning Repository, namely:

1. Abalone — <https://archive.ics.uci.edu/ml/datasets/Abalone>
[Classification] Predicting the age of abalone from physical measurements.
2. Car Evaluation — <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>
[Classification] The data is on evaluations of car acceptability based on price, comfort, and technical specifications.
3. Image Segmentation — <https://archive.ics.uci.edu/ml/datasets/Image+Segmentation>
[Classification] The instances were drawn randomly from a database of 7 outdoor images. The images were hand segmented to create a classification for every pixel.
4. Computer Hardware — <https://archive.ics.uci.edu/ml/datasets/Computer+Hardware>
[Regression] The estimated relative performance values were estimated by the authors using a linear regression method. The gives you a chance to see how well you can replicate the results with these two models.
5. Forest Fires — <https://archive.ics.uci.edu/ml/datasets/Forest+Fires>
[Regression] This is a difficult regression task, where the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data .
6. Wine Quality — <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>
[Regression] This contains two data sets, one for red wine and one for white. Either combine the data sets into a single set for the regression task or build separate regression trees. This is your choice; however, we expect the separate trees to be better. The objective is to learn a model to assess the quality of wine.

Some of the data sets may have missing attribute values. When this occurs in low numbers, you may simply edit the corresponding values out of the data sets. For more occurrences, you should do some kind of “data imputation” where, basically, you generate a value of some kind. This can be purely random, or it can be sampled according to the conditional probability of the values occurring, given the underlying class for that example. The choice is yours, but be sure to document your choice.

An important component in virtually any machine learning process is tuning. Most algorithms have a number of hyper-parameters that need to be tuned to get the best performance from the algorithm. You are expected to tune these hyper-parameters, and you need to explain your tuning process as part of the experimental design (in the design document) as well as when discussing your result (in the project report).

Your assignment consists of the following steps:

1. Prepare a design document addressing the design of the stacked auto-encoder neural network for performing classification and regression. Be sure to include an explanation of your experimental design as well.
2. Download the six (6) data sets from the UCI Machine Learning repository. You can find this repository at <http://archive.ics.uci.edu/ml/>.
3. Pre-process the data to ensure you are working with complete examples (i.e., no missing attribute values).
4. Implement a stacked auto-encoder with an arbitrary number of inputs, an arbitrary number of auto-encoder layers, an arbitrary number of outputs, and the ability to place an arbitrary feedforward neural network on the top for prediction. The SAE will be trained using unsupervised pre-training. The prediction layer will be trained using backpropagation. The entire SAE-prediction network will then be fine-tuned using backpropagation.
5. Develop a hypothesis focusing on convergence rate and final performance of each of the chosen architectures for each of the various problems.
6. Test the SAE algorithm with 1, 2, and 3 auto-encoder layers. The structure of the prediction layer is entirely up to you, but you should test a variety of architectures as part of a “tuning” process.
7. Write a paper that incorporates the following elements, summarizing the results of your experiments:
 - (a) Title and author name
 - (b) A brief, one paragraph abstract summarizing the results of the experiments
 - (c) Problem statement, including hypothesis
 - (d) Description of algorithms implemented
 - (e) Description of your experimental approach
 - (f) Presentation of the results of your experiments
 - (g) A discussion of the behavior of your algorithms, combined with any conclusions you can draw
 - (h) Summary
 - (i) References (You should have at least one reference related to each of the algorithms implemented and any other references you consider to be relevant. You should also provide a reference for each of the data sets. You may not reference websites except for the data sets.)
8. Create a video that is no longer than 5 minutes long demonstrating the functioning of your code. This video should focus on behavior and not on walking through the code. You need to show input, data structure, and output. How you do this is entirely up to you, but be sure it will convince the grader that your program works.
9. Submit your design document, fully documented code, results of the runs of your algorithm, video, and paper.