Stellenbosch
UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

Computer Science Division

Machine Learning
CS441/741

Assignment 2: Neural Networks

Due: 30 September 2024, 08:00

Instructions

For this assignment, you are going to implement and evaluate some neural networks. For the purposes of this assignment, please note the following:

- You may select any one of the options listed below.
- Please submit your own work.
- You may program in any language, provided that your code compiles and execute on Linux. While you are allowed to make use of machine learning libraries, note that you learn most about the machine learning algorithms when you implement these algorithms yourself.
- Submit one compressed archive (only zip files) to extract to the following folders:
  - `documents`: Extract your pdf report to this folder, as well as a readme document with instructions on how to compile and execute your code. Make sure to name your pdf file as `????????assignment2.pdf`, replacing the question numbers with your student number.
  - `data`: Extract the data files that you have used to this folder. This should be the data files in the format that you have used them.
  - `src`: All your source files to be extracted to this folder.

  Name this archive as `????????assignment2.zip`, replacing the question numbers with your student number.
- **NB**: You have to follow the file naming conventions as stated above. I make use of scripts to extract your zip files, and to pull out your pdf file. If you give your files arbitrary names, yours will not be pulled out and will therefore not be evaluated.

- As indicated above, note that you have to submit both a report (as a pdf document), your code and your data files. The report will be a formal technical report wherein you report all that you have done, your results, and a discussion of these results. For guidelines on the format of a technical report and a mark rubric, see the last sections.
- Note: No deadline extensions, and no late submissions will be accepted.

Option 1: Mini-Batch Training using Dynamic Meta-Heuristics

Consider this assignment only if you have done AI791 (Artificial Intelligence).

Recent research on fitness landscape analysis of the error surfaces produced by mini-batch training has shown that the error surfaces change over time (see `NNFLA.pdf` available on SUNLearn). Neural network training using mini-batches is therefore essentially a dynamic optimization problem. It is therefore hypothesized that it is best to make use of an optimization algorithm developed to solve dynamic optimization problems to train a neural network where mini-batching is employed. For this assignment, you are going to test this hypothesis.

For the purposes of this assignment, note the following:

- Select three classification problems. In your report, describe (under the appropriate section) these problems and also describe any data preprocessing that you had to do, with proper motivations.
- Decide on five different mini-batch sizes ranging from small to large. Remember to provide detail on the chosen mini-batch sizes in the appropriate section of your report.
- Use only one hidden layer, with an overestimate of the number of hidden units. In order to prevent overfitting, make use of a regularization approach such as weight decay. For this, you also have to consider that the penalty coefficient is problem dependent. Discuss the regularization approach and the approach that you have use to set the value of the penalty coefficient in your report. Make sure to provide detail on the activation functions used.
- Carefully decide on the performance measures to use to quantify the quality of the trained neural networks.
- Evaluate the performance of stochastic gradient descent for the different mini-batch sizes on the different classification problems. Make sure to discuss all control parameters and how you have decided on their values.
- Now, select a meta-heuristic that was developed to solve dynamic optimization problems, e.g. the quantum-inspired particle swarm optimization algorithm or the DynDE algorithm. Then evaluate the performance of this meta-heuristic as training algorithm for the different mini-batches on the different classification problems. Make sure to discuss all control parameters and how you have decided on their values.
- Also implement a standard version of the meta-heuristic, developed to solve static optimization problems, such as the standard inertia particle swarm optimizer. This is added as a control algorithm to see if the dynamic meta-heuristic actually contributed to differences in performance.
- Determine which of the approaches performed best, and provide an outcome to the stated hypothesis. Provide your detailed results and a discussion thereof in your report.

Option 2: Comparison of Feedforward Neural Network Training Algorithms

For this assignment you have to compare the performance of the following three neural network training algorithms:

- Stochastic gradient descent.
- Scaled conjugate gradient (see `SCG.pdf` available on SUNLearn).
- LeapFrog (see `LFROGa.pdf` and `LFROGb.pdf` available on SUNlearn).

For the purposes of this assignment, note the following:

- Select at least three classification and three function approximation problems of varying complexity. In your report, describe (under the appropriate section) these problems and also describe any data preprocessing that you had to do, with proper motivations.
- Use only one hidden layer. For each problem, you have to determine the optimal number of hidden units. For this purpose, you need to decide what is meant by optimality in this context, and design an appropriate empirical process to find the optimal number of hidden units. Provide sufficient empirical evidence to support your findings.
- Also find the best possible values for the control parameters of the training algorithms that you compare. As for the optimal number of hidden units, design an appropriate empirical process to find best values, and sufficient empirical evidence to support your findings.
- Decide on the performance criteria that you will use to compare the performance of the training algorithms. Define an appropriate empirical process that will aid in a statistically sound approach to determine which of the three training algorithms performed best.
- You may decide on any activation function and cost function.

Option 3: Active Learning in Neural Networks

For this assignment, you have to compare two approaches to active learning with standard passive learning using stochastic gradient descent. The purpose of the assignment is to determine which of the approaches provide the best performance, and to identify the advantages of active learning compared to passive learning.

The approaches to compare are:

- Passive learning using stochastic gradient descent.
- Active learning using neural network output sensitivity analysis (see `SASLA.pdf` available on SUNLearn).
- Active learning using uncertainty sampling (see `ALUS.pdf` available on SUNLearn).

For the purposes of this assignment, note the following:

- Select at least three classification and three function approximation problems of varying complexity. In your report, describe (under the appropriate section) these problems and also describe any data preprocessing that you had to do, with proper motivations.
- Use only one hidden layer, with an overestimate of the number of hidden units. In order to prevent overfitting, make use of a regularization approach such as weight decay. For this, you also have to consider that the penalty coefficient is problem dependent. Discuss the regularization approach and the approach to set the value of the penalty coefficient in your report.
- Also find the best possible values for the control parameters of the training algorithms that you compare.
- Decide on the performance criteria that you will use to compare the performance of the training algorithms. Define an appropriate empirical process that will aid in a statistically sound approach to determine which of the three training algorithms performed best.
- You may decide on any activation function and cost function.

Option 4: Time Series Forecasting using Recurrent Neural Networks

For this assignment, you are going to implement and compare various simple recurrent neural networks for time series prediction. Do this in the following steps:

- Find any five time series datasets. Describe these datasets. Make sure to indicate if the datasets are stationary or non-stationary.
- For self-study, find out how cross-validation should be applied to time series. As a starting point, refer to `https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4`. You have to make use of an appropriate cross-validation process in your analysis of the performance of the simple recurrent neural networks. Describe the process used in your report.

- Perform the necessary data pre-processing on the selected datasets. Describe what you have done and provide justifications.
- You are going to use an Elman recurrent neural network, a Jordan recurrent neural network, and a multi-recurrent neural network. Describe these neural networks in your report. You may select any optimization algorithm and appropriate loss function. Describe what you have used in your report.
- You have to make sure that the networks do not underfit or overfit. Describe the process that you have implemented to prevent undertitting and overfitting.
- Carefully define the empirical process that you have followed, and describe this process in your report. The process has to include settings for all hyperparameters, neural network architecture, performance measures, and the process followed to determine which simple recurrent neural network performed best for each of the datasets.
- Present and discuss your results, and conclude on which simple recurrent neural network performed best.

Option 5: Cooperative Particle Swarm Optimization Training Algorithm

For this assignment, you are going to explore different particle swarm optimization (PSO) algorithms as feedforward neural network training algorithms, specifically cooperative PSO algorithms.

For the purposes of this assignment, note the following:

- Select at least three classification and three function approximation problems of varying complexity. In your report, describe (under the appropriate section) these problems and also describe any data preprocessing that you had to do, with proper motivations.
- Use only one hidden layer. For each problem, you have to determine the optimal number of hidden units. For this purpose, you need to decide what is meant by optimality in this context, and design an appropriate empirical process to find the optimal number of hidden units. Provide sufficient empirical evidence to support your findings.
- For the PSO algorithms, you will make use of weight decay. So, remember to add the weight decay penalty function to the error function that you minimize.
- As a baseline, implement a global best inertia weight PSO algorithm to train the feedforward neural networks. State the values that you have used for the control parameters, with justificiation for why you have used these values.
- Select any one of the approaches in the paper `randomGrouping.pdf` available on SUNlearn, remembering to use weight decay, use the selected approach to train the feedforward neural networks.
- The last two algorithms that you will implement are the merging cooperative PSO (MCPSO) and the decomposition cooperative PSO (DCPSO), as described in the paper `mCPSOdCPSO.pdf` available on SUNlearn. Use these two algorithms to train the neural networks.
- Present the results for these algorithms, compare these results, and draw a conclusion on which approach performed best.

Technical Report writing

The following is a general guideline of how to structure your report.

Title Section

Provide your report with a title, and as author provide your initials, surname and student number. Also provide an email address.

Abstract

Provide a very concise summary of what this report provides. Provide some context, the goals, how these were achieved, and the main observation. The abstract should be short. No more than 300 words.

1. Introduction

The introduction sets the stage for the remainder of your report. You usually have very general statements here. The introduction prepares the reader for what to expect from reading your report. In general, the introduction should either contain or be a summary of your ENTIRE report.

2. Background

A very high level discussion on the problem domain and the algorithms and/or approaches that you have used. Do not be too specific on the algorithms and approaches. This section is typically where the "base cases" of concepts that appear throughout the remainder of your report are discussed. It is also an ideal place to refer a reader to other sources containing relevant information on the topic but which is outside the scope of your assignment. It is the perfect place for pseudo code. Remember to discuss very generally. After reading this section the marker should be able to determine whether or not you know what you're talking about.

3. Implementation

In this section you discuss how you approached, implemented and solved your assignment choice. You provide pseudo code where necessary and discussions of the solutions that you have implemented. This is also the section where your discussion specializes on the concepts mentioned in the background section. Be very specific in your discussions in this section.

4. Empirical Procedure

Here you describe the empirical procedure followed to apply your algorithms to obtain answers to the goals/hypothesis of the study. You elaborate on the performance measures used and provide the benchmark problems used. Provide all control parameter values with a motivation for why you have used these, and state the number of independent runs. After reading this section (in addition to the background) the reader should be able to duplicate your experiments to obtain similar results to those obtained by you.

5. Research Results:

This is the section where you report your results obtained from running the experiments as discussed in the implementation section. You have to give, at least, averages and standard deviations for the experiments/simulations. Thoroughly discuss the results that you have obtained and provide clear arguments in support of your results and observations from these results. Answer questions like "are these results to be expected?", "why did these results occur?" and "would different circumstances lead to different results?".

6. Conclusion(s):

Very general conclusions about the assignment that you have done. This section "answers" the questions and issues that you've raised and investigated. This section is, in general, a summary of what you have done, what the results where and finally what you concluded from these results. This is the final section in your document so be sure that all the issues raised up until now are answered here. This is also the perfect section to discuss what you have learnt in doing this assignment.

References

Provide all references that you have consulted.

Mark Rubric

Your report will be evaluated as follows:

Aspect	**Mark**
Abstract	5
Introduction	10
Background	20
Implementation	20
Empirical process	15
Results & discussion	50
Conclusions	5
References	5
Linguistic quality	20
Total	150

Note that if you did not submit your own code, the mark obtained above will be multiplied by a factor of 0.5.