COMP3411/9814 Artificial Intelligence
Term 1, 2025

Assignment 2 – Neural Nets, Reinforcement Learning

Due: Tuesday 22 April, 10pm
Marks: 25% of final assessment

In this assignment you will be examining two machine learning methods. Part A requires you to implement a neural network for classification. You will also implement for Part B the method of Q-learning, with an extension to allow the learner to receive "advice" from a "teacher", actually another agent trained by reinforcement learning.

For both Part A and Part B you will implement and submit code plus a written report.

Note: You will make separate submissions for Parts A and B ! Submission details are at the end of this specification.

Part A: Neural Network Classification (11 marks)

Problem overview

Credit decisions are a long-standing application area for machine learning. If you have ever supplied information on your financial history when applying for a loan (or other credit facility) from a financial institution then your data has probably been run through a model trained by machine learning.

In this part of the assignment, you will train an artificial neural network on historical data to model the risk of default on loans to small businesses. The task is framed in terms of *classification*, i.e., predicting whether the outcome of a loan is likely to result in default, or not.

LendingClub was an early mover in peer-to-peer (P2P) marketplace lending. The idea behind P2P lending was to match borrowers with investors who fund loans. Loans could be made for various purposes. The company is no longer involved in P2P lending but has released some of its historical data for research. There are many versions of these datasets available online. Note: for this assignment we will use a version of a dataset for small business loans which is not generally available for download.

Dataset

The small business loan dataset used in this assignment has 17 variables (features):

Index	Variable	Description
1	term_36m	loan term of 36 months
2	term_60m	loan term of 60 months
3	grade_n	LendingClub assigned loan grade
4	sub_grade_n	LendingClub assigned loan sub-grade
5	revol_util_n	Sum of revolving credit utilisation
6	int_rate_n	Interest Rate on the loan
7	installment_n	Monthly payment owed
8	tot_hi_cred_lim_n	Total installment high credit/credit limit
9	emp_length_n	Employment length in years
10	dti_n	Calculated from total monthly debt payments
11	avg_cur_bal_n	Average current balance of all accounts
12	all_util_n	Balance to credit limit on all trades
13	acc_open_past_24mths_n	Number of trades opened in past 24 months
14	annual_inc_n	Self-reported annual income
15	loan_amnt_n	Listed amount of the loan
16	cover	Derived feature
17	Outcome	Loan outcome: default (1) or discharged (0)

Target variable

In this dataset the 'Outcome' feature is used as the target variable. Since Outcome has only two values the task is *binary* or *two-class* classification. Like many real-world datasets the class ratio is *unbalanced*. In this case, the majority of loans were discharged (paid back) with relatively few examples of loans that defaulted. In the dataset, default is represented as 1, with 0 meaning the loan was successfully discharged.

Classification task

In this classification task, the goal is for the neural network to predict whether the loan will default or not, based on the 16 features characterising each loan.

Solution overview

For Part A you will implement a neural network for the classification task, evaluate its performance, submit a short written report describing the network and its performance, and submit your neural network for automarking on a (random sample of) a hidden test set.

Getting started

Download the file `hw2nn.zip` from this directory:
`https://www.cse.unsw.edu.au/~cs3411/25T1/hw2/`

(or download it from here).

Unzip the file and change directory to `hw2nn`:

```
unzip hw2nn.zip
cd hw2nn
```

You should see in this directory the following files:

```
hw2main.py student.py hw2nn_data.csv
```

Model training

(a) The downloaded data you will use for training is in `hw2nn_data.csv`

(b) Implement your neural network using PyTorch in `student.py` by defining its architecture and hyperparameters. This includes: loss function, optimiser, batch size, learning rate, and number of epochs. It is recommended that your network has fewer parameters than the number of samples divided by 10.

(c) Run `hw2main.py` to train the neural network model. Monitor the evolution of the loss values during training.

Validation set to evaluate model performance

In `student.py` you can specify a split of the dataset `hw2nn_data.csv` into training and validation sets. The default is to use 80% of the data for training and 20% for validation, but you can change this.

(d) Train on the training set and use the validation set to estimate model accuracy.

(e) Repeat steps (b), (c) and (d) to see if you can improve the performance of your model.

Results from evaluating your classification model

When you think you have achieved the best accuracy you can through training, you need to collect the results as follows and include them in your report.

(f) Create a plot showing the accuracy (y-axis) versus the number of epochs (x-axis) on the training set. Save the plot for inclusion in the report.

(g) Use your model to predict the class on the validation set:

- Compute and plot a confusion matrix. Calculate this in terms of predictions where the positive class is 'Outcome' = 1, i.e., default.
- Note the "class ratio", i.e., the ratio of the number of positive examples to the size of the entire dataset.

Include your confusion matrix, together with a brief discussion of these results, in the report.

Additional notes

- You will need to set the random seed to ensure that your results are reproducible
- Note: the provided code will automatically save your model weights in the file `model.pth`
- In step (f), you only need to plot the result(s) for your best-performing neural network configuration. CODE IS NOT REQUIRED in the assignment pdf.

Questions These questions should be answered in your written report.

Question A1 [2 marks] Describe your neural network architecture, including, but not limited to: the number of inputs, number of outputs, number of hidden layer(s), if any, an all hyperparameters. Justify the choices made, and any experimentation that led to those.

Question A2 [1 mark] You will see in the file "`hw2main.py`" that scaling of dataset values is applied using the `sklearn` method StandardScaler. You can remove this scaling by setting `scale inputs` to `False` in `student.py` Try doing this, and compare what happens to performance with and without scaling. Explain the difference (if any) in outcomes you observe.

Marking outline:

Training and validation set performance [3 marks] These marks are given for the plot obtained in step (f), and the confusion matrix and discussion in step (g), above. These should be provided in the report.
Answers to questions [3 marks] Answers to Questions A1 and A2 above.