Artificial Neural Networks

Task 1: Standard and Poor's (S&P) 500 Index prediction

The S&P 500 Index is an indicator of the state of the equity market in the USA. The Index value is based on the stock values of 500 companies listed in the New York stock exchange (NYSE) and NASDAQ.

We curated a dataset that consists of the daily S&P500 values and stock prices for 387 companies out of the 500. The data is listed in chronological order starting 7th October 2013 until 5th October 2018 and provided in csv format.

In this task, you will use a neural network to predict the value of the S&P Index based on company stock prices.

1. Start by finding out which stocks of the 387 are most correlated with the S&P index value. For the prediction task, you should use the stocks with (correlation ≥ 0.95 or correlation ≤ -0.95).

2. For the training, divide the dataset into training, testing and validation sets. Indicate the ratios you use for the division.

3. Indicate the scoring metric suitable for the task. Use this metric in evaluating the sgd and adam optimizers in the remaining steps.

4. Train the neural network with both the sgd and adam optimizers. Attempt different values for the learning rate with each optimizer to achieve the best result. You should indicate the values you used and the performance observed in each case. Justify your observations.

5. Indicate which regularization technique you use to avoid overfitting. Possible techniques are early stopping, the regularization factor (alpha) or using cross validation. Account for

your technique choice. Early stopping and cross validation automatically calculates and displays the validation score.

6. Which optimizer performs better in this task? Why?

7. Using the best performing neural network configuration, plot the predicted values for S&P versus the true values in the test set. Your plot should show 200 time points or more.

8. Comment on the points where the prediction and true values mismatch most. Why you think these particular points were difficult to predict?

Task 2: Stock prices time series prediction

For the stock that is most correlated with the S&P Index (as per your findings in step 1 above), you will train a neural network to predict the price of this stock for 4 days ahead based on today's value as well as the past 4 days.

To achieve this, you will use the original dataset to create a new data frame of the following format:

0	1	2	3	4	5	6	7	8
t-4	t-3	t-2	t-1	t	t+1	t+2	t+3	t+4

- The columns (0:4) are the inputs, prices for past 4 days and today.
- The columns (5:8) are the outputs, future prices for next 4 days.

Hints for implementation

- The shift operation in Pandas can be used to create the layout described above.
- Some rows will have undefined values (nan) after shifting. You should drop these rows.

1. Indicate the scoring metric suitable for this task.
2. Indicate the neural network architecture and configuration that achieve the best score.
3. Use the trained neural network to predict stock price for 3 instances from the test set. Display the output value versus the predicted value and justify the similarities and differences in values.

Task 3: Predicting product rating from review text

Reviews provide invaluable information to companies on which product features or shortcomings affect their clients most.

When a user provides a review as well as a product rating (1 to 5 stars), this is considered labeled data. The words in the review text are the features the numeric rating is the target.

A neural network trained on this dataset can then be used to assign rating to reviews that weren't combined with a product rating.

In this task you will build a classifier using Amazon product reviews for digital music. The dataset can be obtained from here: http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Digital_Music_5.json.gz

In this dataset, reviews have been processed to remove all products with less than 5 reviews (5-core). You can find out more about the rest of the dataset here: http://jmcauley.ucsd.edu/data/amazon/

1. Start by inspecting the dataset. You will notice that the dataset is not balanced. Display the count of reviews with overall rating =1, 2…5.

2. To train on a balanced dataset, select 2500 reviews for each of the overall ratings (2500 reviews with overall rating 1, 2500 reviews with overall rating 2, … etc).

3. As discussed in lecture, review words will have to be transformed to numeric features before they are fed to a neural network. The code for this transformation is provided in the attached notebook with explanation. Note that to cut down the training time, we limited the number of words to use to 40000.

   For more details you can use these two short tutorials:

   http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

   http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

4. As always, split your data into training, testing and validation. I recommend using early stopping as a regularization approach here.

5. Use the appropriate scoring metric for evaluating your neural networks. Justify your selection.

6. Attempt to train a neural network with 1, 2 or 3 layers and different number of nodes [128, 512, 1024]. Your notebook should show the best and worst architectures. Note that while you can achieve a high training score, the validation score on this dataset remains around 0.6!

7. Use the code available from the link below to plot the normalized confusion matrix for the best performing architecture. Comment on the ratios of misclassified instances for each class.
   http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

8. What is the activation function used at the output layer? Did you have to select this function yourself? How does SKLearn determines it?

Bonus

1. Print out the actual rating and predicted values for part of the test set.
2. Describe how the network fails, causing the low test and validation scores.
3. Suggest a modification in the network (type, dataset, output format, scoring metric) that best fits the network behavior you observed in the previous step. Your modification may improve testing and validation scores.
4. Implement your suggested modification.

Resources

Pandas data frames are a powerful data container.

For some examples on handling data frames, the course labs and solutions are very useful.

Additional short tutorials

- https://pandas.pydata.org/pandas-docs/stable/10min.html#min
- http://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

To join dataframes df.concat() is one possible method.

Submission

Your submission should consist of a well-formatted Jupyter notebook that explains the following point for each part of the code:

- reasoning (justification) for the step
- Interpretation of each result or visualization

The notebook should be clear and self-sufficient as it will replace a demo. Maintain the data files with the provided filenames to facilitate testing.

Please submit your Jupyter notebook as an attachment via email to guc.ml18@gmail.com by 11:59PM Thursday 18th October 2018.