National Chung Cheng University

Introduction to Deep Learning

Mid-Term Programming Exam (Take-Home)

2023/4/13 ~ 2022/4/30 (deadline: 23:59:59)   50% of total grade

You are provided a time-series forecasting problem centered around restaurant visitors. The data comes from two separate sites:

- Hot Pepper Gourmet (hpg): similar to Yelp, here users can search restaurants and also make a reservation online
- AirREGI / Restaurant Board (air): similar to Square, a reservation control and cash register system

You must use the reservations, visits, and other information from these sites to forecast future restaurant visitor totals on a given date. The dataset covers the dates from 2016 until April 2017. The dataset should be divided into training data (full year of 2016) and testing data (whatever is given for 2017).

There are days in the test set where the restaurant were closed and had no visitors. These are ignored in scoring. The training set omits days where the restaurants were closed.

File Descriptions

This is a relational dataset from two systems. Each file is prefaced with the source (either air_ or hpg_) to indicate its origin. Each restaurant has a unique `air_store_id` and `hpg_store_id`. Note that not all restaurants are covered by both systems, and that you have been provided data beyond the restaurants for which you must forecast. Latitudes and Longitudes are not exact to discourage de-identification of restaurants.

air_reserve.csv

This file contains reservations made in the air system. Note that the `reserve_datetime` indicates the time when the reservation was created, whereas the `visit_datetime` is the time in the future where the visit will occur.

- `air_store_id` - the restaurant's id in the air system
- `visit_datetime` - the time of the reservation
- `reserve_datetime` - the time the reservation was made
- `reserve_visitors` - the number of visitors for that reservation

hpg_reserve.csv

This file contains reservations made in the hpg system.

- `hpg_store_id` - the restaurant's id in the hpg system
- `visit_datetime` - the time of the reservation
- `reserve_datetime` - the time the reservation was made
- `reserve_visitors` - the number of visitors for that reservation

air_store_info.csv

This file contains information about select air restaurants. Column names and contents are self-explanatory.

- `air_store_id`
- `air_genre_name`
- `air_area_name`
- `latitude`
- `longitude`

Note: latitude and longitude are the latitude and longitude of the *area* to which the store belongs

hpg_store_info.csv

This file contains information about select hpg restaurants. Column names and contents are self-explanatory.

- `hpg_store_id`
- `hpg_genre_name`
- `hpg_area_name`
- `latitude`
- `longitude`

Note: latitude and longitude are the latitude and longitude of the *area* to which the store belongs

store_id_relation.csv

This file allows you to join select restaurants that have both the air and hpg system.

- `hpg_store_id`
- `air_store_id`

air_visit_data.csv

This file contains historical visit data for the air restaurants.

- `air_store_id`
- `visit_date` - the date
- `visitors` - the number of visitors to the restaurant on the date

Submission:

You need to submit the code of your model and fill in the following information:

Model Information:

- Number of layers: 加入 input layer 和 output layer 一起計算的話總共 4 層。
- Number of units in each layer:

  Input layer → 174

  Hidden layer 1 → 256

  Hidden layer 2 → 128

  Output layer → 1
- Activation functions used: relu
- Loss function: Root Mean Squared Log Error (RMSLELoss)
- Cost function: Root Mean Squared Log Error (RMSLELoss)

Training Epochs: 100

Training Accuracy: 0.77049 (RMSLELoss)

Testing Accuracy: 0.77890 (RMSLELoss)

Optimization techniques employed: RMSprop

Difference in accuracies after each optimization technique that you applied:

(1) Optimization technique name: Adam

Before optimization: Training/Testing Accuracies =

0.81019(RMSLELoss)/0.80749(RMSLELoss)

After optimization: Training/Testing Accuracies =

0.78469(RMSLELoss)/0.78696(RMSLELoss)

Any other changes: learning rate = 0.001

(2) Optimization technique name: SGD

Before optimization: Training/Testing Accuracies =

0.81019(RMSLELoss)/0.80749(RMSLELoss)

After optimization: Training/Testing Accuracies =

0.80358(RMSLELoss)/0.80769(RMSLELoss)

Any other changes: learning rate = 0.001

(3) Optimization technique name: RMSprop

Before optimization: Training/Testing Accuracies =

0.81019(RMSLELoss)/0.80749(RMSLELoss)

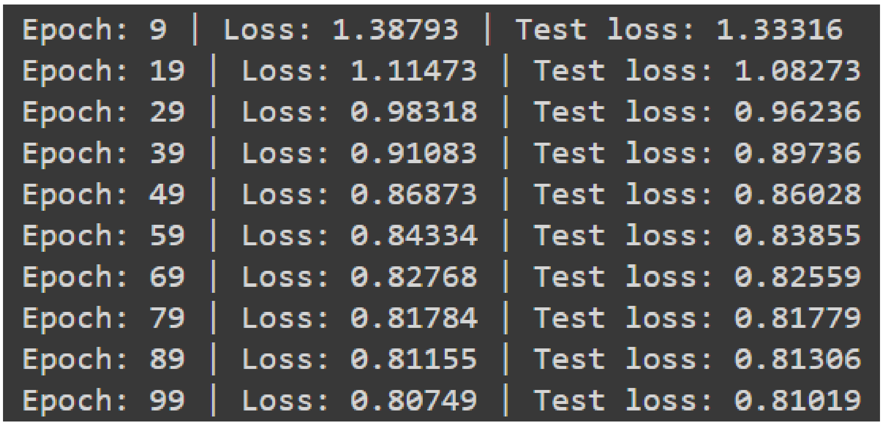
After optimization: Training/Testing Accuracies =

0.77049(RMSLELoss)/0.77890(RMSLELoss)

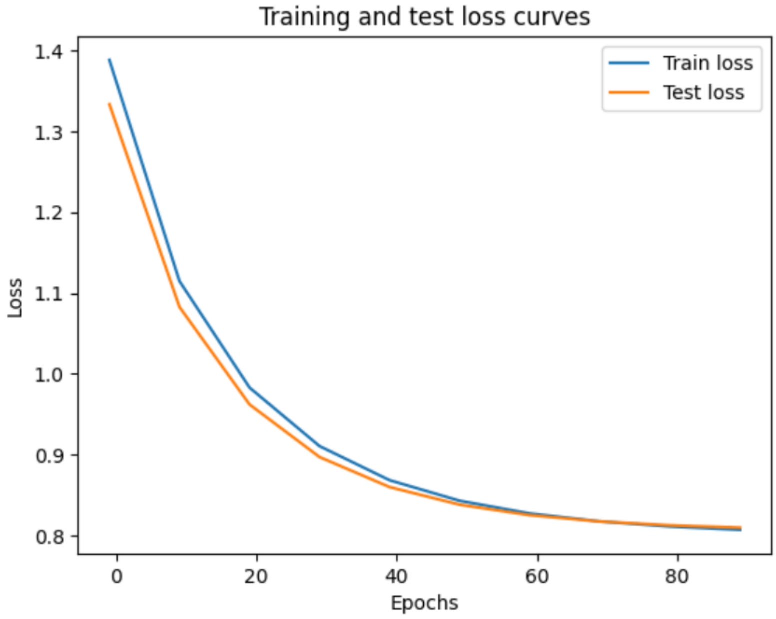
Any other changes: learning rate = 0.001

Anything special about your model:

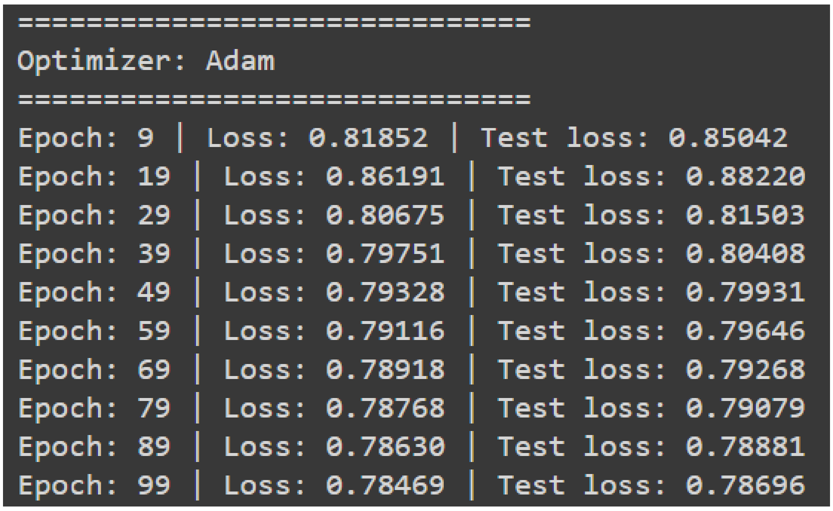
No optimizer's result and graph :



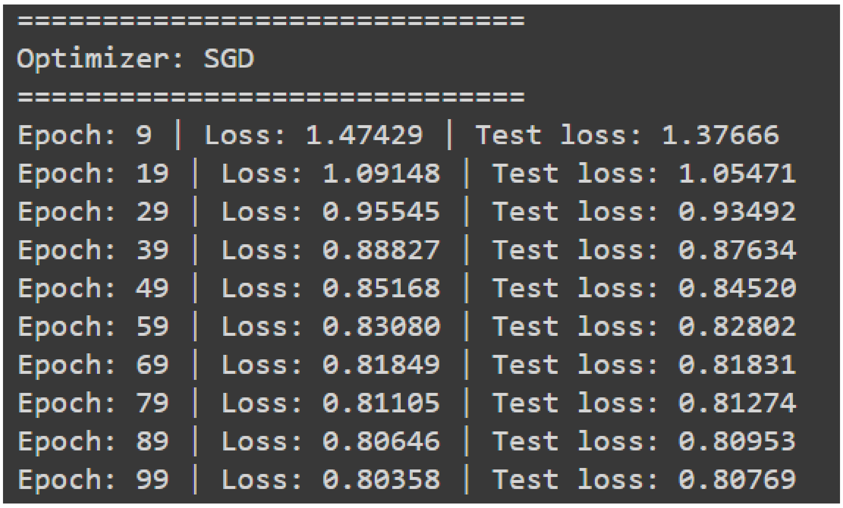
```
Epoch: 9 | Loss: 1.38793 | Test loss: 1.33316
Epoch: 19 | Loss: 1.11473 | Test loss: 1.08273
Epoch: 29 | Loss: 0.98318 | Test loss: 0.96236
Epoch: 39 | Loss: 0.91083 | Test loss: 0.89736
Epoch: 49 | Loss: 0.86873 | Test loss: 0.86028
Epoch: 59 | Loss: 0.84334 | Test loss: 0.83855
Epoch: 69 | Loss: 0.82768 | Test loss: 0.82559
Epoch: 79 | Loss: 0.81784 | Test loss: 0.81779
Epoch: 89 | Loss: 0.81155 | Test loss: 0.81306
Epoch: 99 | Loss: 0.80749 | Test loss: 0.81019
```



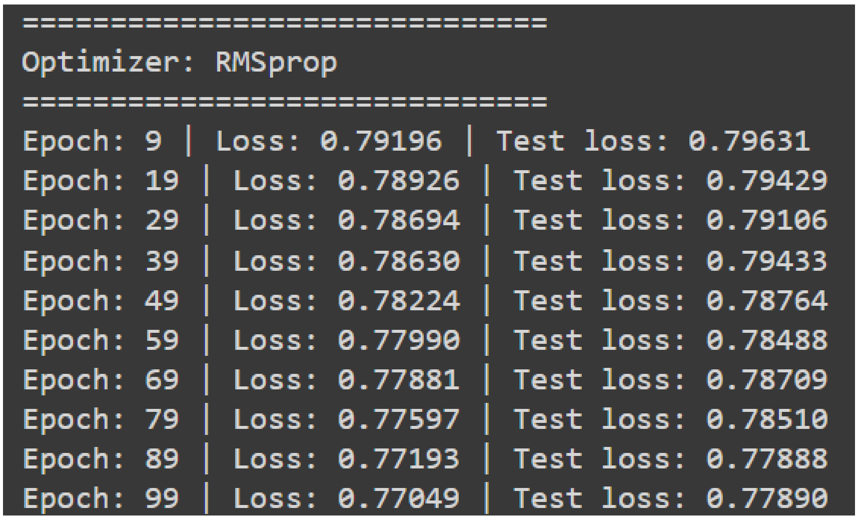
Different optimizers’ result and graph :



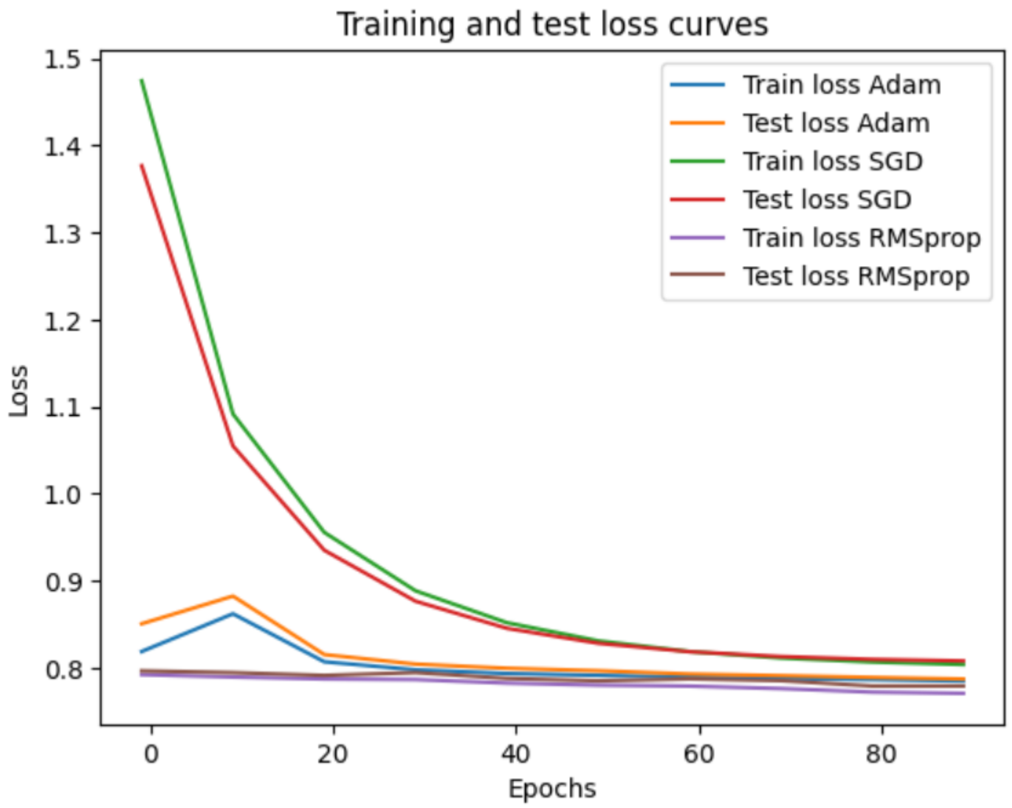
```
==============================
Optimizer: Adam
==============================
Epoch: 9 | Loss: 0.81852 | Test loss: 0.85042
Epoch: 19 | Loss: 0.86191 | Test loss: 0.88220
Epoch: 29 | Loss: 0.80675 | Test loss: 0.81503
Epoch: 39 | Loss: 0.79751 | Test loss: 0.80408
Epoch: 49 | Loss: 0.79328 | Test loss: 0.79931
Epoch: 59 | Loss: 0.79116 | Test loss: 0.79646
Epoch: 69 | Loss: 0.78918 | Test loss: 0.79268
Epoch: 79 | Loss: 0.78768 | Test loss: 0.79079
Epoch: 89 | Loss: 0.78630 | Test loss: 0.78881
Epoch: 99 | Loss: 0.78469 | Test loss: 0.78696
```



```
==============================
Optimizer: SGD
==============================
Epoch: 9 | Loss: 1.47429 | Test loss: 1.37666
Epoch: 19 | Loss: 1.09148 | Test loss: 1.05471
Epoch: 29 | Loss: 0.95545 | Test loss: 0.93492
Epoch: 39 | Loss: 0.88827 | Test loss: 0.87634
Epoch: 49 | Loss: 0.85168 | Test loss: 0.84520
Epoch: 59 | Loss: 0.83080 | Test loss: 0.82802
Epoch: 69 | Loss: 0.81849 | Test loss: 0.81831
Epoch: 79 | Loss: 0.81105 | Test loss: 0.81274
Epoch: 89 | Loss: 0.80646 | Test loss: 0.80953
Epoch: 99 | Loss: 0.80358 | Test loss: 0.80769
```



```
==============================
Optimizer: RMSprop
==============================
Epoch: 9 | Loss: 0.79196 | Test loss: 0.79631
Epoch: 19 | Loss: 0.78926 | Test loss: 0.79429
Epoch: 29 | Loss: 0.78694 | Test loss: 0.79106
Epoch: 39 | Loss: 0.78630 | Test loss: 0.79433
Epoch: 49 | Loss: 0.78224 | Test loss: 0.78764
Epoch: 59 | Loss: 0.77990 | Test loss: 0.78488
Epoch: 69 | Loss: 0.77881 | Test loss: 0.78709
Epoch: 79 | Loss: 0.77597 | Test loss: 0.78510
Epoch: 89 | Loss: 0.77193 | Test loss: 0.77888
Epoch: 99 | Loss: 0.77049 | Test loss: 0.77890
```



Comments on the course:

課程的內容非常深入淺出，簡明易懂，而且具有實用性，使我能夠更好地理解深度學習的核心概念和技術。

課程還提供了豐富的程式實作，並且讓我們運用深度學習技術來解決現實問題。作業雖然要求有一定的技術基礎，但是也非常有助於我們深入理解和掌握學習的內容。

助教在課程中也提供了非常好的支持和幫助，總是能夠及時回答我的問題和提供有用的建議和指導，使我更有信心和能力完成作業和項目。

還有，邀請執行長來演講開源專案也是一個非常不錯的安排，使我們能夠深入了解深度學習在實際項目中的應用，特別是在 kuberflow 和 kserve 相關的主題上，我獲得了豐富的知識和經驗。