**Subject: Deep Learning**

**Instructor Name: Masood Zamani**

**Final Project**

**Group:6**

**Team Members**

**Abhi Priyadarshi- n01513842**

**Akshay Gajera-n01515835**

**Utsav Jitendrabhai Patel-N01516259**

**Yatri Patel-n01511061**

**Summary:**

* **Requirements……………………………………………………3-5**
* **Dataset……………………………………………………………….6**
* **Implementation of sliding window…………………… 7**
* **RNN Model…………………………………………………………8**
* **Single Layer LSTM……………………………………………… 9**
* **Multi-Layer LSTM……………………………………………… 10**
* **Hyperparameter tuning……………………………………. 11**
* **Conclusion………………………………………………………… 12**

**Requirements:**

**(1)Dataset Description:**

A tabular time-series dataset (CSV file) with **290 monthly** records (from 1997-2021). Each monthly record (row) contains four features: date**, rate, ems, pkh**.

**(2)Model:**

Implementing a recurrent neural network by **Pytorch** based on **RNNs or LSTM.** The neural network **must be** trained with three columns **rate, ems, pkh as inputs,** and column **pkh as an output**. The column **date** shows only the time order of the monthly data.

**(3)Defining the training and test datasets:**

Select **the first 80%** (the first 232 rows), and **the rest of the 20%** (last 58 rows) of the CSV file for the training dataset and test dataset respectively. The two datasets at this phase must not be shuffled to preserve the order of the monthly data.

**(4)Creating training and test samples:**

Each sample in the train dataset should be a sequence with the **length of 231 elements**, and each

sample in the test dataset should be a sequence with **the length of 57 elements**. To make the lengths

of all samples equal, a zero padding technique is used at the beginning of each sample. The targets

(prediction values) for both training and test samples are selected from pkh column while creating

the samples as the following:

Initially, a **sliding time-window of length L=10** is defined which scans the data row by row and

moves **1 step every time when a new sample is created**. In this way, for two consecutive samples

X*i* and X*i+1*, the sample X*i+1* contains 1 more row of data compared to that of X*i* and, therefore, Xi+1

has 1 less row of zero padding. The target value (pkh) of each sample is selected from **the first row**

**that is not within the sliding window** and located **at the right side (head) of the sliding window**.

A screenshot of a data sheet

Description automatically generated

target of X1

NA NA **642**

X1 (training sample)

The 1st 221 rows

of X1 are set with

zero padding

r1, …, r10 from training dataset

The value of pkh column at r11

← t231

← t1

X2 (training sample):

features : [the 1st 220 rows with zero padding + r1,…,r11 ]

target : The value of column pkh at r12

.

.

.

Xn (last training sample):

features : [r1,…,r231] //no zero padding

target : The value of column pkh at r232

Create the test samples by the same technique using the test dataset.

**Dataset:**

A screenshot of a computer screen

Description automatically generated

**Implementation of sliding window:**

A computer screen shot of code

Description automatically generated

**RNN Model:**

**Defining and initializing RNN model**

A screen shot of a computer program

Description automatically generated

**Output:**

A graph of a training and test loss curves

Description automatically generated

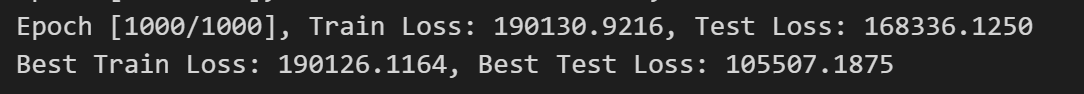
**Single Layer LSTM:**

**Defining and initializing single-layer LSTM model**

A computer screen shot of a program code

Description automatically generated

Output

 A graph of a training and test loss curves

Description automatically generated

**Multi-Layer LSTM:**

**Defining and initializing single-layer LSTM model**

A computer screen shot of a program code

Description automatically generated

Output:

A black screen with white text

Description automatically generated A graph of a training and test loss curves

Description automatically generated

**Hyperparameter tuning:**

1. **Number of Hidden Units**

* Hidden units control the model's capacity to learn complex patterns.
* Higher Units: More capacity to learn intricate patterns but the risk of overfitting.
* Lower Units: Simpler model with potential underfitting.
* We adjusted the number of hidden units to find an optimal trade-off between model complexity and generalization.

1. **Number of Layers**

* Number of layers influences the depth of our model.
* More Layers: Increased capacity to capture intricate relationships but the risk of vanishing/exploding gradients.
* Fewer Layers: Simpler model but may struggle with complex data.
* We experimented with different layer configurations to balance complexity and stability.

**Conclusion:**

* Among the three models evaluated – RNN, Single Layer LSTM, and Multi-layer LSTM – the LSTM-based models (both single and multi-layer) demonstrated enhanced performance compared to the basic RNN model.
* Both LSTM models exhibited similar test losses, implying their comparable ability to generalize to new, unseen data.
* Notably, the training losses for the LSTM models were considerably lower than those of the RNN model, indicating the LSTMs' effectiveness in capturing underlying patterns within the dataset.
* The Multi-layer LSTM, despite its added complexity, did not lead to a significant improvement in test performance compared to the Single-layer LSTM.
* To arrive at a well-informed decision about the most suitable model for your specific use case, it is advisable to conduct further analysis regarding prediction quality, potential overfitting, and practical implications.