CS6320 Assignment 3

https://github.com/gadarsh043/NLP_FFNN_RNN

Group06

Adarsh Gella	Akhila Susarla	Vijaya Sai Latha Pulipati	Abhiram Reddy Madapa
AXG240019	AXS240035	VXP230093	AXM240036

1 Introduction and Data (5pt)

In this assignment, we have performed a 5-class sentiment analysis using PyTorch with a Feed Forward Neural Network (FFNN) and a Recurrent Neural Network (RNN) on a data set of Yelp reviews.

The main task for this assignment is to perform sentimental analysis on a data set of Yelp reviews into 5 classes - [1, 2, 3, 4, 5] with both FFNN and RNN architectures. The data set consists of training, test and validation data in a json format with the number of examples in each file as shown in Table 1.

Data Type	No. of Examples
Training	16000
Test	800
Validation	800

Table 1: Data statistics

As a part of the pre-processing step, the text data is split into tokens with white space as delimiter and then were converted into vector embedding with the help of the word embedding pickle file. The words have been converted into lowercase. Also, the data is shuffled at every epoch.

2 Implementations (45pt)

2.1 FFNN (20pt)

```
def forward(self, input_vector):
    # [to fill] obtain first hidden layer representation
    out = self.W1(input_vector)
    out = self.activation(out)
    # [to fill] obtain output layer representation
    out = self.W2(out)
    # [to fill] obtain probability dist.
    predicted_vector = self.softmax(out)
    return predicted_vector
```

Figure 1: FFNN Code

The FFNN takes in a fixed amount of input data all at the same time and produces a fixed amount of output at the same time. Here in the forward propagation, the input vector at a particular layer is multiplied by weights at that layer. The result is passed through an activation function (such as ReLU) to introduce non-linearity. The output of this is multiplied by the weights of the next layer. Finally, a soft-max function is introduced at the end to provide normalized probability distribution over the output.

2.2 RNN (25pt)

The RNN takes in input one at a time and in a sequence. The output (hidden state) is combined with the following input in sequence to produce the following output. Here we are doing the sentiment analysis to predict the rating of a review which can be given by the last cell of RNN because the final output is dependent on all previous computations and inputs.

```
def forward(self, inputs):
    h0 = torch.zeros(self.numOfLayer, 1, self.h)
    # [to fill] obtain hidden layer representation
    # [to fill] obtain output layer representations
    _, hidden = self.rnn(inputs, h0)
    # [to fill] sum over output
    out = self.W(hidden[:, -1, :])
    # [to fill] obtain probability dist.
    predicted_vector = self.softmax(out)
    return predicted_vector
```

Figure 2: RNN Code

In the first step, a hidden state will usually be seeded as a matrix of zeros of dimensions (1, 1, h) where h is hidden dimension taken as input.

The result of this will then be passed through an activation function (such as a tanh function) to introduce non-linearity. Finally, a soft-max function is introduced to provide normalized probability distribution over the output.

3 Experiments and Results (25pt)

3.1 Evaluations (5pt)

Accuracy is used as the metric for the model evaluation which is a measure of the number of correctly predicted results over the total number of results.

3.2 Results (20pt)

Table 2 and Table 3 show the accuracy for different hyper parameters that we tested our implementation with for FFNN and RNN respectively.

Hidden Size	Learning Rate	Optimizer	Accuracy
10	0.01	SGD	0.49
10	0.1	SGD	0.008
10	0.001	SGD	0.51
10	0.01	Adam	0.53
20	0.01	SGD	0.6
20	0.001	Adam	0.52

Table 2: FFNN: Validation Accuracy with different parameters (5 epochs)

Hidden Size	Number of Layers	Optimizer	Learning Rate	Accuracy
32	1	Adam	0.01	0.35
32	2	Adam	0.01	0.118
50	1	Adam	0.01	0.38
10	1	Adam	0.001	0.336
32	1	SGD	0.01	0.26
32	1	Adam - LSTM	0.01	0.258
32	1	SGD - LSTM	0.01	0.348

Table 3: RNN: Validation Accuracy with different parameters

4 Analysis (20pt)

4.1 Plot of learning curve of the best system (10pt)

For the learning curve plots given below for our best system, the blue coloured lines represent the training accuracy and the orange lines represent the validation accuracy.

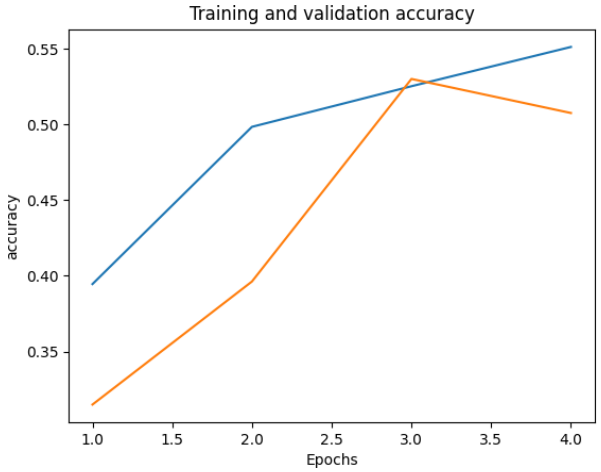

Figure 3: FFNN Training and Validation Accuracy

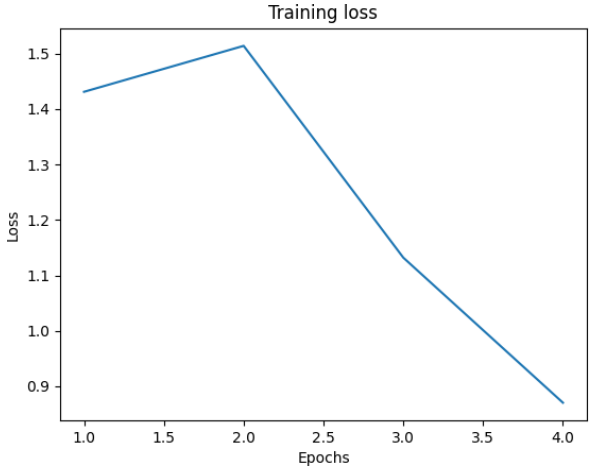

Figure 4: FFNN Training Loss

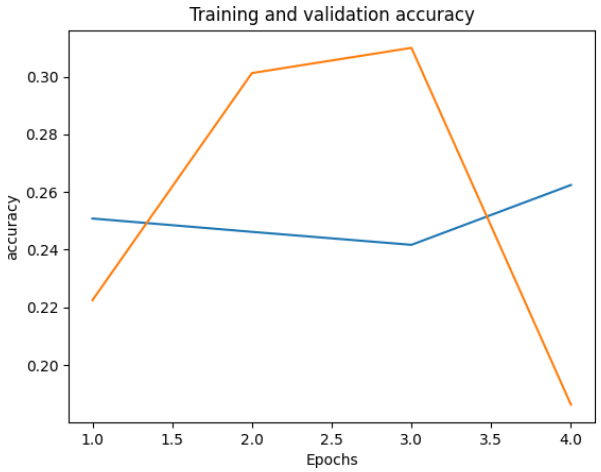

Figure 5: RNN Training and Validation Accuracy

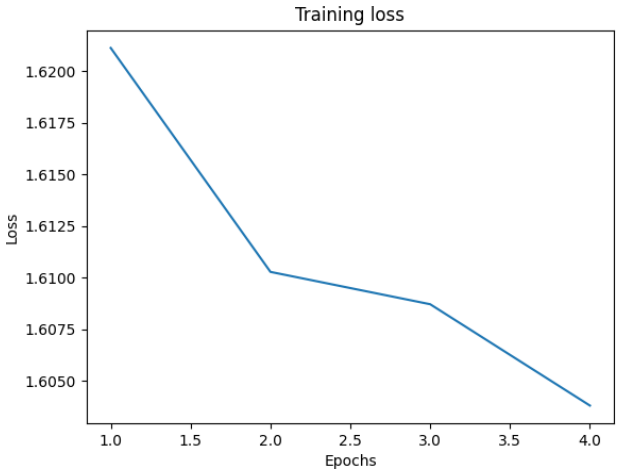

Figure 6: RNN Training Loss

4.2 Error analysis (10pt)

The FFNN gives bad results with a learning rate of 0.1 and performs better with a lower learning rate with the SGD and the Adam Optimizer.

The RNN is not consistent with the training and validation accuracy and changes depending upon the data at every epoch. To make it more consistent, we tried using an LSTM network of the RNN. The accuracy with LSTM is more consistent and performs similar to the RNN. Also, the runtime of the model is lot faster with LSTM implementation than the traditional RNN implementation.

5 Conclusion and Others (5pt)

In this assignment, we have successfully implemented sentiment analysis using FFNN and RNN along with trying different variations of hyper parameters for analysis and comparison.

The best accuracy we got for FFNN is 0.6 and for RNN is 0.38.

5.1 Individual Member Contribution

While all members of the team had to get together for initial planning and design of the workflow, once the layout was ready, we equally divided the implementation amongst the team members in the following ways:

- Adarsh Gella: RNN and report.
- Akhila Susarla: Experimentation and results.
- Vijaya Sai Latha Pulipati: FFNN and report.
- Abhiram Reddy Madapa: Analysis, Improvement and report .

5.2 Assignment Feedback

It was a great experience doing this assignment. We do feel the amount of coding we got to do was very less to make the implementation work. We request if there is a possibility to also add deadline on e-learning for ease of managing daily targets and keeping track of multiple deadlines with other courses.