Page No.
Date

Assignment No. 4

Title: Recurrent Neural Network (RNN)

Objective: To study and understand RNN by doing analysis and designing predictions.

Problem Statement: Use the Google stock prices dataset and design a time series analysis and prediction system using RNN.

Theory:
RNN is a type of Neural Network where the output from the previous step is feed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence which solved this issue with the help of a Hidden Layer. The main and most important feature is Hidden state, which remembers some information about a sequence.

RNN have a "memory" which remembers all information about what has been calculated. This reduces the complexity of parameters, unlike other neural networks.

How RNN works.
The working of an RNN can be understood with the help of the below example.
Example: Suppose there is a deeper network with one input layer, three hidden layers and one output layer.

Page No.
Date

Then like other neural networks, each hidden layer will have its own set of weights and biases. let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for the second hidden layer and (w3, b3) for the third hidden layer. This means that each of these layers is independent of the other, i.e. they do not memorize the previous outputs

Now the RNN do the following:

- RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer.
- Hence these three layers can be joined together such that the weights and bias of all the hidden layers are the same, in a single recurrent layer.

The formula for calculating current state; where:

$$h_t = f(h_{t-1}, x_t)$$

$h_t \rightarrow$ current state $h_{t-1} \rightarrow$ previous state

$x_t \rightarrow$ input state

Formula for applying Activation function (tanh): where

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

$W_{hh} \rightarrow$ weight at recurrent neuron.

$W_{xh} \rightarrow$ weight at input neuron.

The formula for calculating output:

$$y_t = W_{hy} h_t$$

Training through RNN.

1. A single-time step of the internet is provided to the network.
2. Then calculate its current state using a set of current input and previous state.
3. The current ht becomes ht-1 for the next time step.
4. One can go as many time steps according to the problem and join the information from all the previous states.
5. Once all the time steps are completed the final current state is used to calculate the output.
6. The output is then compared to the actual output i.e. the target output and the error is generated.
7. The error is then back-propogated to the network to update the weights and hence the (RNN) is trained.

Advantages of RNN.

1) Remembers each and every piece of information. It is useful in time series prediction only because of feature.
2) Used for effective pixel neighborhood.

Disadvantages of RNN:

1) Gradient vanishing and exploding problems.
2) Training an RNN is a very difficult task.
3) Cannot process long sequences using tanh or relu.

Conclusion: We have successfully implemented a RNN to create a classifier.