Modern Education Society's Wadia College of Engineering, Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Assignment No: 4**

Laboratory Practice – V (Deep Learning)

ASSIGNMENT NO: 04

AIM: **Recurrent neural network (RNN):** Use the Google stock prices dataset and design a time series analysis and prediction system using RNN.

OBJECTIVES: Students should be able to design stock price prediction system by using Recurrent Neural Network using Google stock prices dataset.

PREREQUISITE:

1. Basic of programming language
2. Concept of RNN

THEORY:

What is RNN?

A recurrent neural network (RNN) is a deep learning model that is trained to process and convert a sequential data input into a specific sequential data output. Sequential data is data—such as words, sentences, or time-series data—where sequential components interrelate based on complex semantics and syntax rules. An RNN is a software system that consists of many interconnected components mimicking how humans perform sequential data conversions, such as translating text from one language to another. RNNs are largely being replaced by transformer-based artificial intelligence (AI) and large language models (LLM), which are much more efficient in sequential data processing.

How does a recurrent neural network work?

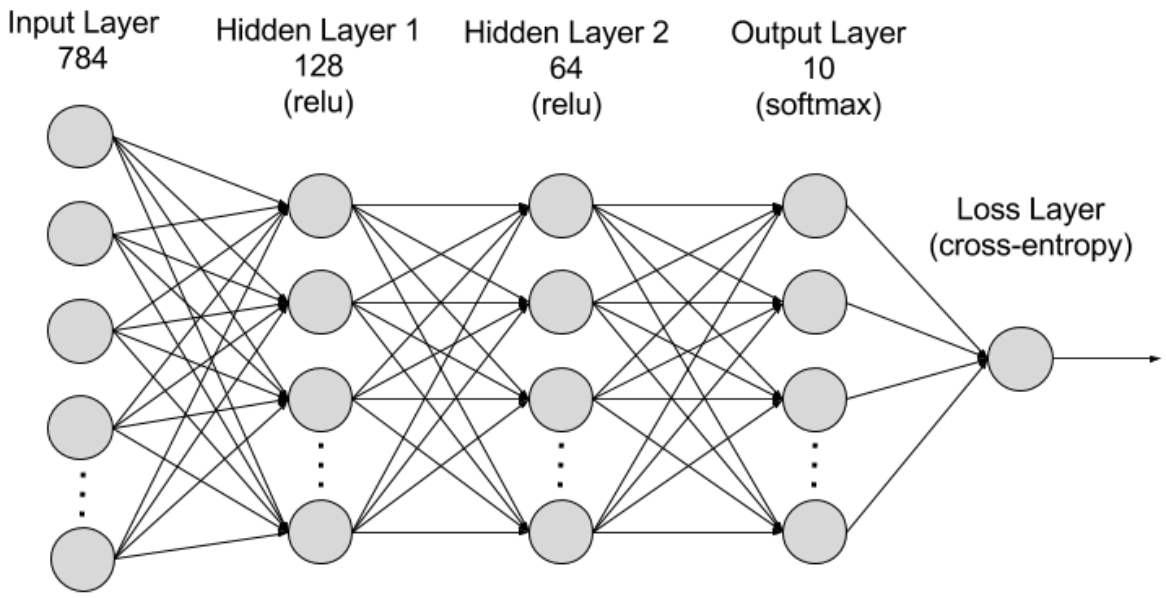

Figure A : Working of Recurrent Neural Network [Credit : AWS]

RNNs are made of neurons: data-processing nodes that work together to perform complex tasks. The neurons are organized as input, output, and hidden layers. The input layer receives the information to process, and the output layer provides the result. Data processing, analysis, and prediction take place in the hidden layer.

Hidden layer

RNNs work by passing the sequential data that they receive to the hidden layers one step at a time. However, they also have a self-looping or *recurrent* workflow: the hidden layer can remember and use previous inputs for future predictions in a short-term memory component. It uses the current input and the stored memory to predict the next sequence. This makes RNNs useful in speech recognition, machine translation, and other language modelling tasks.

Training

Machine learning (ML) engineers train deep neural networks like RNNs by feeding the model with training data and refining its performance. In ML, the neuron's weights are signals to determine how influential the information learned during training is when predicting the output. Each layer in an RNN shares the same weight.

ML engineers adjust weights to improve prediction accuracy. They use a technique called backpropagation through time (BPTT) to calculate model error and adjust its weight accordingly. BPTT rolls back the output to the previous time step and recalculates the error rate. This way, it can identify which hidden state in the sequence is causing a significant error and readjust the weight to reduce the error margin.

What are the types of recurrent neural networks?

RNNs are often characterized by one-to-one architecture: one input sequence is associated with one output. However, one can flexibly adjust them into various configurations for specific purposes. The following are several common RNN types.

1. **One-to-many:** This RNN type channels one input to several outputs. It enables linguistic applications like image captioning by generating a sentence from a single keyword.
2. **Many-to-many:** The model uses multiple inputs to predict multiple outputs. For example, you can create a language translator with an RNN, which analyzes a sentence and correctly structures the words in a different language.
3. **Many-to-one:** Several inputs are mapped to an output. This is helpful in applications like sentiment analysis, where the model predicts customers' sentiments like *positive*, *negative*, and *neutral* from input testimonials.

How do recurrent neural networks compare to other deep learning networks?

RNNs are one of several different neural network architectures.

Recurrent neural network vs. feed-forward neural network

Like RNNs, feed-forward neural networks are artificial neural networks that pass information from one end to the other end of the architecture. A feed-forward neural network can perform simple classification, regression, or recognition tasks, but it can't remember the previous input that it has processed. The RNN overcomes this memory limitation by including a hidden memory state in the neuron.

Recurrent neural network vs. convolutional neural networks

Convolutional neural networks are artificial neural networks that are designed to process spatial data. One can use convolutional neural networks to extract spatial information from videos and images by passing them through a series of convolutional and pooling layers in the neural network. RNNs are designed to capture long-term dependencies in sequential data

CONCLUSION:

We have successfully developed a time series analysis and prediction system using RNN on Google stock prices dataset.

QUESTIONS:

1. What are some variants of recurrent neural network architecture?
2. What are the limitations of recurrent neural networks?