19CST302-NEURAL NETWORKS AND DEEP LEARNING

UNIT-IV DEEP LEARNING ARCHITECTURES

QUESTIONS

LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail.

It fails to store information for a longer period of time. At times, a reference to certain information stored quite a long time ago is required to predict the current output. But RNNs are absolutely incapable of handling such “long-term dependencies”.

There is no finer control over which part of the context needs to be carried forward and how much of the past needs to be ‘forgotten’.

Other issues with RNNs are exploding and vanishing gradients (explained later) which occur during the training process of a network through backtracking.

Thus, Long Short-Term Memory (LSTM) was brought into the picture. It has been so designed that the vanishing gradient problem is almost completely removed, while the training model is left unaltered. Long-time lags in certain problems are bridged using LSTMs which also handle noise, distributed representations, and continuous values. With LSTMs, there is no need to keep a finite number of states from beforehand as required in the hidden Markov model (HMM). LSTMs provide us with a large range of parameters such as learning rates, and input and output biases.

Structure of LSTM

The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got only a single neural net layer of tanh, LSTMs comprise three logistic sigmoid gates and one tanh layer. Gates have been introduced

in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'. Information is retained by the cells and the memory manipulations are done by the gates. There are three gates which are explained below

Forget Gate

The information that is no longer useful in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_t-1 (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state, the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

Input gate

The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_t-1 and x_t. Then, a vector is created using the tanh function that gives an output from -1 to +1, which contains all the possible values from h_t-1 and x_t. At last, the values of the vector and the regulated values are multiplied to obtain useful information.

Output gate

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying the tanh function on the cell. Then, the information is regulated using the sigmoid function and filtered by the values to be remembered using inputs h_t-1 and x_t. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

Variations in LSTM Networks

With the increasing popularity of LSTMs, various alterations have been tried on the conventional LSTM architecture to simplify the internal design of cells to make them work in a more efficient way and to reduce computational complexity. Gers and Schmidhuber

introduced peephole connections which allowed gate layers to have knowledge about the cell state at every instant. Some LSTMs also made use of a coupled input and forget gate instead of two separate gates which helped in making both decisions simultaneously. Another variation was the use of the Gated Recurrent Unit(GRU) which improved the design complexity by reducing the number of gates. It uses a combination of the cell state and hidden state and also an update gate which has forgotten and input gates merged into it.