PARSHWANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Data Science

CSE DATA SCIENCE

Module 5

Truncated Backpropagation Through Time

Backpropagation is the training algorithm used to update the weights in a neural network in order to minimize the error between the expected output and the predicted output for a given input.

For sequence prediction problems where there is an order dependence between observations, recurrent neural networks are used instead of classical feed-forward neural networks. Recurrent neural networks are trained using a variation of the Backpropagation algorithm called Backpropagation Through Time, or BPTT for short.

In effect, BPTT unrolls the recurrent neural network and propagates the error backward over the entire input sequence, one timestep at a time. The weights are then updated with the accumulated gradients.

BPTT can be slow to train recurrent neural networks on problems with very long input sequences. In addition to speed, the accumulation of gradients over so many timesteps can result in a shrinking of values to zero, or a growth of values that eventually overflow, or explode.

A modification of BPTT is to limit the number of timesteps used on the backward pass and in effect estimate the gradient used to update the weights rather than calculate it fully.

This variation is called Truncated Backpropagation Through Time, or TBPTT.

The TBPTT training algorithm has two parameters:

- **k1**: Defines the number of timesteps shown to the network on the forward pass.
- **k2**: Defines the number of timesteps to look at when estimating the gradient on the backward pass.

As such, we can use the notation TBPTT(k1, k2) when considering how to configure the training algorithm, where k1 = k2 = n, where n is the input sequence length for classical non-truncated BPTT.

Module 5

Impact of TBPTT Configuration on the RNN Sequence Model

Modern recurrent neural networks like LSTMs can use their internal state to remember over very long input sequences. Such as over thousands of timesteps.

This means that the configuration of TBPTT does not necessarily define the memory of the network that you are optimizing with the choice of the number of timesteps. You can choose when the internal state of the network is reset separately from the regime used to update network weights.

Instead, the choice of TBPTT parameters influences how the network estimates the error gradient used to update the weights. More generally, the configuration defines the number of timesteps from which the network may be considered to model your sequence problem.

We can state this formally as something like:

```
1 yhat(t) = f(X(t), X(t-1), X(t-2), ... X(t-n))
```

Where yhat is the output for a specific timestep, f(…) is the relationship that the recurrent neural network is approximating, and X(t) are observations at specific timesteps.

It is conceptually similar (but quite different in practice) to the window size on Multilayer Perceptrons trained on time series problems or to the p and q parameters of linear time series models like ARIMA. The TBPTT defines the scope of the input sequence for the model during training.