Machine Learning

Assignment 8.1

Submitted By: Ranji Raj

December 16, 2020

a) Problem with non-sequential data classifiers

- In Non-sequential data, every training or testing sample is independent. Meaning, the desired output is the function of only the *current* inputs and has no consistent relation to earlier inputs, outputs, or hidden state.
- **Example: Biomedical signal recording and analysis-** In this use case, for changes in biological signals for identifying temporal patterns over the period of time for span of 6-8 months are required. A range of input features are given:
  1. The admit time of the subject,
  2. Time the illness was developed,
  3. Existing co-morbidites.
- Now for handling real-time situations, where how the illness/disease has aggravated not only depends on only one input but a bunch of other time-related inputs and target variable can be influenced by past values of inputs which otherwise, cannot be modeled by classifiers running non-sequential data.

b) Difference of RNN to Feed-Forward NN

- Feed-forward neural networks pass the data forward from input to output, while recurrent networks have a **feedback loop** where data can be fed back into the input at some point before it is fed forward again for further processing and final output.
- Feed-forward networks do not capture possible **dependencies**.

c) Backpropogation Through Time-BPTT

- BPTT is applied specifically to train RNN networks to work on sequence data like time series, texts, etc.
- It works by unrolling all input timesteps.
- Each timestep has one input timestep, one copy of the network, and one output.
- Errors are calculated and collected for each timestep.
- The network is rolled-back up and all the weights are updated and the process is repeated.
- **Problem of vanishing and exploding gradient:** BPTT can be computationally expensive as the number of timesteps increases.
  In turn if the input sequences are consisted of 1000's of timesteps, then this much amount of derivatives must be required for a single weight update.
  This can cause the weights to vanish or explode (go to zero or over-shoot) leading to slow learning process.

d) LSTM for solving vanishing gradient problem

- Normal RNNs suffer from the vanishing/exploding gradient problem, which hinders their ability to learn long-range dependencies.
- The **forget gate** in LSTMs is the main reason why they are able to delay the above problem and hence, process longer input sequences.
- Also they posses a unique **additive** gradient structure that includes direct access to the forget gate's activations, enabling the network to encourage desired behaviour from the error gradient using frequent gates update on each time step of the learning process.

e) Summary take-away

- A classifier which only works on non-sequential data may not generalize for all the problems but need to work on sequential data as well.
- RNN have **feedback loop** due to which they can process long range sequences.
- For training RNNs, **BPTT** algorithms must be employed.
- LSTMs can solve the vanishing gradient problem by **simply turning multiplication into addition**.