Report

Implementation of RNN Cell

Forward: The forward propagation takes in the 1) internal states from the previous time step and, 2) the inputs for the current time step to get the updated internal states.

Formula for that is $h_t = f(Ux_t + Vh_{t-1} + B)$

where h_t is the internal state at time t, x_t is the input at time t. U, V and B are the weights parameters that needs to be updated. f is an activation function which is tanh in this case.

Backward: For the backward pass, we need to calculate the derivatives with respect to the cost to update the parameters.

There are 5 gradients to calculate. They are for x, h, U, V and B from the forward pass.

For example $\frac{\partial L}{\partial V} = np.dot(1 - h_t^2, h_{t-1})$

We store the gradients for U, V and B. We output the gradients for x and h.

Implementation of RNN

Forward: For the forward pass, we need to repeat the forward pass for RNN cell T times.

For each time step, we use the internal state of the previous time step and the inputs of the current time step to calculate the internal state of the current time step. We store these internal states for calculate the gradients.

Backward: We start from the last time step to backpropagation through time. For each time step, we want to calculate the gradients. We sum up the contributions of each time step to the gradient.

Sentiment Analysis using RNN

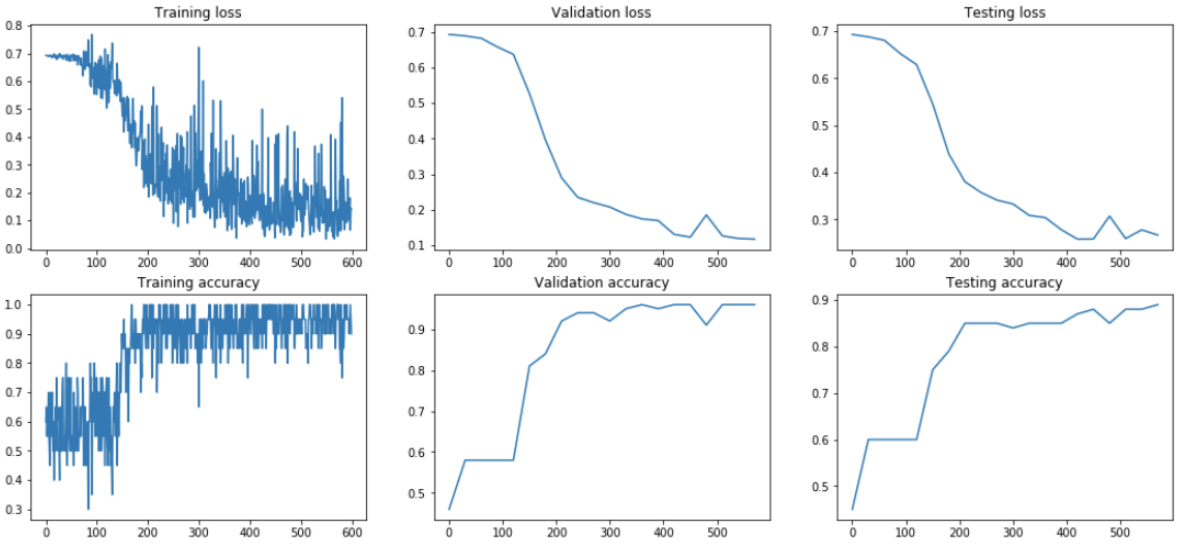
The model consists of a basic RNN that replaced the bidirectionalRNN with a vanilla RNN.

Initially I started off with just training in 5 epochs. However the model was unable to reach good results. So I used 20 epochs to train the model which gave me about 96% validation and 89% test accuracy which is quite high.

I observed that there are a lot of noise in the training, validation and testing loss which suggests that maybe the learning rate is too high. (There might also be a bug in my BPTT.) Nevertheless, lowering the learning rate gave me a much stable result but it didn't make a big difference eventually.

I have also tried to increase the number of hidden units in the RNN cell as well as the fully connected layers. It took way longer to train and did not have a lot of improvement on the results. I also added some dropout in between the fully connected layers to regularise the model.

We may also experiment with other RNN architectures such as the LSTM or the GRU which has become more popular nowadays.


Training loss
Validation loss
Testing loss
Training accuracy
Validation accuracy
Testing accuracy