Deep Learning Assignment 2

Jochem Loedeman
12995282

1 Recurrent Neural Networks

1.1 Vanilla RNNs

Question 1.1

(a)

$$\frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{W}_{ph}} = \frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{p}^{(T)}} \frac{\partial \boldsymbol{p}^{(T)}}{\partial \boldsymbol{W}_{ph}}$$

In the computational graph, there is no path from outputs of recurrent hidden states to the loss at T. Therefore, no sum over previous states is present in the expression for the gradient.

(b)

$$\frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{W}_{hh}} = \frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{p}^{(T)}} \frac{\partial \boldsymbol{p}^{(T)}}{\partial \boldsymbol{h}^{(T)}} \frac{\partial \boldsymbol{h}^{(T)}}{\partial \boldsymbol{W}_{hh}} = \sum_{i=0}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{p}^{(T)}} \frac{\partial \boldsymbol{p}^{(T)}}{\partial \boldsymbol{h}^{(T)}} \frac{\partial \boldsymbol{h}^{(T)}}{\partial \boldsymbol{h}^{(i)}} \frac{\partial \boldsymbol{h}^{(i)}}{\partial \boldsymbol{W}_{hh}}$$

$$= \sum_{i=0}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial \boldsymbol{p}^{(T)}} \frac{\partial \boldsymbol{p}^{(T)}}{\partial \boldsymbol{h}^{(T)}} \left(\prod_{j=i+1}^{T} \frac{\partial \boldsymbol{h}^{(j)}}{\partial \boldsymbol{h}^{(j-1)}} \right) \frac{\partial \boldsymbol{h}^{(i)}}{\partial \boldsymbol{W}_{hh}}$$

In this case, there are paths from nodes in previous timesteps computed with $\boldsymbol{W}_{hh}$ to the loss at T. Therefore, we should account for these dependencies through the chain rule, giving rise to sums and products within the expression for the gradient.

(c) The first and second gradients are respectively independent and dependent on previous timesteps, as was already explained through the connectivity of the computational graph. When training this network for a large number of timesteps, we have to account for very long-term dependencies of the recurrent states through the product in the expression for the second gradient. This product of Jacobians can cause the total gradient to become very small or very large, making learning difficult.

Question 1.2

(a)
- *Input modulation gate:*
  Proposes an updated cell state. This gate has a tanh activation, which is important to keep the activations of the internal state zero-centered. If for example we used the sigmoid, these states would increase over time.

- *Input gate:*
  Regulates how much of the new value proposed by the input modulation gate is actually used in the updated cell state. Uses a sigmoid to obtain a gating value between 0 and 1.

- *Forget gate:*
  Regulates how much of the previous cell state is retained in the updated cell state. Uses a sigmoid to obtain a gating value between 0 and 1, which corresponds to completely forgetting or strongly remembering the previous state.

- *Output gate:*
  Regulates how much of the nonlinearized current cell state is passed as output of the cell. Uses a sigmoid to obtain a gating value between 0 and 1.

(b)

$$N_{\text{total}} = 4\underbrace{\left(N_{\text{hidden}} \cdot N_{\text{input}} + N_{\text{hidden}} \cdot N_{\text{hidden}} + N_{\text{hidden}}\right)}_{\text{gate input weights, recurrent weights and biases}} + \underbrace{N_{\text{output}} \cdot N_{\text{hidden}}}_{\text{output weights}} + \underbrace{N_{\text{output}}}_{\text{output bias}}$$