Homework 3: Recurrent Neural Networks

Student Name: Student ID:

Sun Yat-sen University

1 Exercise 1: Backpropagation through Time

Consider the RNN (Recurrent Neural Network) in Figure 1:

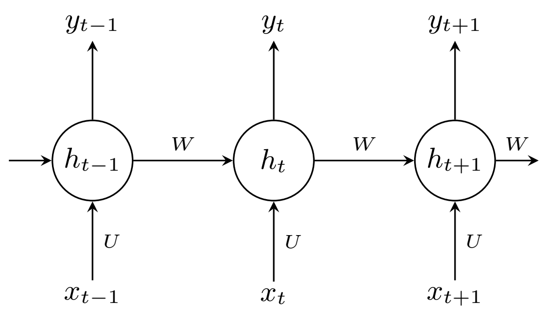


Figure 1: A recurrent neural network.

Each state h_t is given by:

$$h_t = \sigma(Wh_{t-1} + Ux_t), \text{where } \sigma(z) = \frac{1}{1+\exp(-z)}.$$

Let L be a loss function defined as the sum over the losses L_t at every time step until time T: $L = \sum_{t=0}^{T} L_t$, where L_t is a scalar loss depending on h_t.

In the following, we want to derive the gradient of this loss function with respect to the parameter W.

(a) Suppose we have $y = \sigma(Wx)$ where $y \in \mathbb{R}^n, x \in \mathbb{R}^d$ and $W \in \mathbb{R}^{n\times d}$. Derive the Jacobian $\frac{\partial y}{\partial x} = \operatorname{diag}(\sigma')W \in \mathbb{R}^{n\times d}$.

(b) Derive the quantity $\frac{\partial L}{\partial W} = \sum_{t=0}^{T}\sum_{k=1}^{t} \frac{\partial L_t}{\partial h_t}\frac{\partial h_t}{\partial h_k}\frac{\partial h_k}{\partial W}$.

2 Exercise 2: Vanishing/Exploding Gradients in RNNs

In this exercise, we want to understand why RNNs (Recurrent Neural Networks) are especially prone to the Vanishing/Exploding Gradients problem and what role the eigenvalues of the weight matrix play. Consider part (b) of exercise 1 again.

(a) Write down $\frac{\partial L}{\partial W}$ as expanded sum for $T = 3$. You should see that if we want to backpropagate through n timesteps, we have to multiply the matrix $\text{diag}(\sigma')W$ n times with itself.

(b) Remember that any diagonalizable (square) matrix M can be represented by its eigendecomposition $M = Q\Lambda Q^{-1}$ where Q is a matrix whose i-th column corresponds to the i-th eigenvector of M and Λ is a diagonal matrix with the corresponding eigenvalues placed on the diagonals. Recall that every eigenvector v_i satisfies this linear equation $Mv_i = \lambda_i v_i$, where $\lambda_i = \Lambda_{ii}$ is an eigenvalue of M. Proof by induction that for such a matrix the product $\prod_{i=1}^{n} M$ can be represented as: $M^n = Q\Lambda^n Q^{-1}$.

(c) Consider the weight matrix $\begin{bmatrix} 0.58 & 0.24 \\ 0.24 & 0.72 \end{bmatrix}$. Its eigendecomposition is:

$$W = Q\Lambda Q^{-1} = \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 0.9 & 0 \\ 0 & 0.4 \end{bmatrix} \begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix}$$

Calculate W^{30}. What do you observe? What happens in general if the absolute value of all eigenvalues of W is smaller than 1? What happens if the absolute value of any eigenvalue of W is larger than 1? What if all eigenvalues are 1?

3 Exercise 3: LSTMs

Recall the elements of a module in an LSTM and the corresponding computations, where $\odot$ stands for pointwise multiplication. For a good explanation on LSTMs you can refer to `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Consider the LSTM in Figure 2.

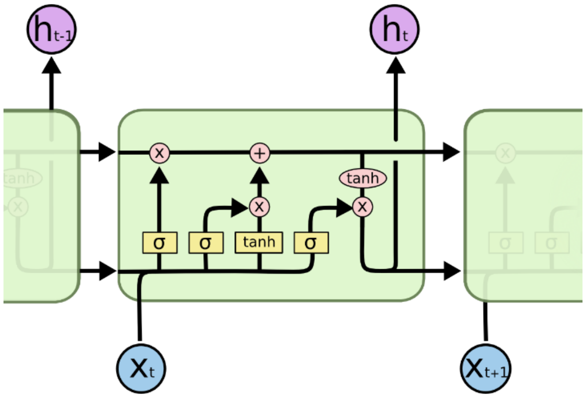
(a) What do the gates f_t, i_t and o_t do?

(b) Which of the quantities next to the figure are always positive?

Let's now try to understand how this architecture approaches the vanishing gradients problem. To calculate the gradient $\frac{\partial L}{\partial \theta}$, where θ stands for the parameters (W_f, W_o, W_i, W_c), we now have to consider the cell state C_t instead of h_t. Like h_t in normal RNNs, C_t will also depend on the previous cell states $C_{t-1}, \ldots, C_0$, so we get a formula of the form:

$$\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \sum_{k=1}^{t} \frac{\partial L}{\partial C_t} \frac{\partial C_t}{\partial C_k} \frac{\partial C_k}{\partial W},$$

where note that the real formula is a bit more complicated since C_t also depends on f_t, i_t and $\widetilde{C}_t$, which in turn all depend on W, but this can be neglected.



$$
\begin{aligned}
f_t &= \sigma(W_f h_{t-1} + U_f x_t) \\
i_t &= \sigma(W_i h_{t-1} + U_i x_t) \\
o_t &= \sigma(W_o h_{t-1} + U_o x_t) \\
\tilde{C}_t &= \tanh(W_c h_{t-1} + U_c x_t) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

Figure 2: A Long Short Term Memory network.

(c) We know that $\frac{\partial C_t}{\partial C_k} = \prod_{i=k+1}^{t} \frac{\partial C_t}{\partial C_{t-1}}$. Let $f_t = 1$ and $i_t = 0$ such that $C_t = C_{t-1}$ for all t. What is the gradient $\frac{\partial C_t}{\partial C_k}$ in this case?