Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Matthias Schubert
Maximilian Bernhard, Niklas Strauß

Munich, 18.11.2024

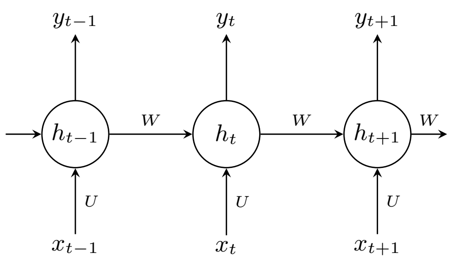
Deep Learning and Artificial Intelligence

WS 2024/25

Exercise 5: Recurrent Neural Networks

Exercise 5-1 Backpropagation through Time

Consider the following RNN:



Each state h_t is given by:

$$h_t = \sigma(Wh_{t-1} + Ux_t), \qquad \sigma(z) = \frac{1}{1+e^{-z}}$$

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) Given $\mathbf{y} = \sigma(W\mathbf{x})$ where $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^d$ and $W \in R^{n \times d}$. Derive the Jacobian $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$!

$\Big[$Solution: $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \text{diag}(\sigma'(\cdot))\, W\Big]$

Detailed Solution

We have $y_i = \sigma(\sum_{k=1}^{d} W_{ik}x_k) = \sigma(\mathbf{w}_i^T\mathbf{x})$, where $\mathbf{w}_i$ denotes the vector corresponding to the i-th row of W. Thus: $\frac{\partial y_i}{\partial x_j} = \sigma'(\mathbf{w}_i^T\mathbf{x})W_{ij}$.

With that, the Jacobian is given as:

$$\frac{\partial y}{\partial x} = \begin{pmatrix} \sigma'(\mathbf{w}_1^T\mathbf{x})W_{11} & \dots & \sigma'(\mathbf{w}_1^T\mathbf{x})W_{1d} \\ \vdots & \ddots & \vdots \\ \sigma'(\mathbf{w}_n^T\mathbf{x})W_{n1} & \dots & \sigma'(\mathbf{w}_n^T\mathbf{x})W_{nd} \end{pmatrix}$$

$$= \begin{pmatrix} \sigma'(\mathbf{w}_1^T\mathbf{x}) & 0 & \cdots & 0 \\ 0 & \sigma'(\mathbf{w}_2^T\mathbf{x}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma'(\mathbf{w}_n^T\mathbf{x}) \end{pmatrix} \begin{pmatrix} W_{11} & \dots & W_{1d} \\ \vdots & \ddots & \vdots \\ W_{n1} & \dots & W_{nd} \end{pmatrix}$$

$$= \mathrm{diag}(\sigma'(W\mathbf{x}))W = \mathrm{diag}(\sigma')\ W (\text{short notation}).$$

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

Detailed Solution

We have

$$\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \frac{\partial L_t}{\partial W}.$$

For each L_t, the state h_t depends on every (previous) state h_k with $k \leq t$. Each of those h_k in turn depends on W. Thus we have to sum over all of those states:

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial W} = \sum_{k=0}^{t} \frac{\partial L_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

where

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{i=k+1}^{t} diag(\sigma'(\cdot))W. \qquad \text{(see part a))}$$

Thus:

$$\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \sum_{k=0}^{t} \frac{\partial L_t}{\partial h_t} \left(\prod_{i=k+1}^{t} diag(\sigma')W \right) \frac{\partial h_k}{\partial W}.$$

Note that h_k again depends on all previous hidden states:

$$\frac{\partial h_k}{\partial W} = \sigma'(Wh_{k-1} + ...)(1 \cdot h_{k-1} + W\frac{\partial h_{k-1}}{\partial W})$$

Exercise 5-2 Vanishing/Exploding Gradients in RNNs

In this exercise, we want to understand why RNNs are especially prone to the Vannishing/Exploding Gradients problem and what role the eigenvalues of the weight matrix play. Consider part b) of exercise 5-1 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 $diag(\sigma')W$ n times with itself.

Detailed Solution

$$
\begin{aligned}
\frac{\partial L}{\partial W} = & \frac{\partial L_0}{\partial h_0}\frac{\partial h_0}{\partial h_0}\frac{\partial h_0}{\partial W} \\
& + \frac{\partial L_1}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_1}{\partial h_1}\frac{\partial h_1}{\partial h_1}\frac{\partial h_1}{\partial W} \\
& + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W} + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial h_2}\frac{\partial h_2}{\partial W} \\
& + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_1}\frac{\partial h_1}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_3}\frac{\partial h_3}{\partial W} \\
& = \frac{\partial L_0}{\partial h_0}\frac{\partial h_0}{\partial W} \\
& + \frac{\partial L_1}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_1}{\partial h_1}\frac{\partial h_1}{\partial W} \\
& + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W} + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial W} \\
& + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial W} \\
& = \frac{\partial L_0}{\partial h_0}\frac{\partial h_0}{\partial W} \\
& + \frac{\partial L_1}{\partial h_1}\, diag(\sigma')W\, \frac{\partial h_0}{\partial W} + \frac{\partial L_1}{\partial h_1}\frac{\partial h_1}{\partial W} \\
& + \frac{\partial L_2}{\partial h_2}(diag(\sigma')W)^2\, \frac{\partial h_0}{\partial W} + \frac{\partial L_2}{\partial h_2} diag(\sigma')W \frac{\partial h_1}{\partial W} + \frac{\partial L_2}{\partial h_2}\frac{\partial h_2}{\partial W} \\
& + \frac{\partial L_3}{\partial h_3}(diag(\sigma')W)^3\, \frac{\partial h_0}{\partial W} + \frac{\partial L_3}{\partial h_3}(diag(\sigma')W)^2\, \frac{\partial h_1}{\partial W} + \frac{\partial L_3}{\partial h_3} diag(\sigma')W\, \frac{\partial h_2}{\partial W} + \frac{\partial L_3}{\partial h_3}\frac{\partial h_3}{\partial W}
\end{aligned}
$$

If we want to backpropagate through n timesteps, we have calculate the n-th power of $(diag(\sigma')W)$. Note that we used $(diag(\sigma')W)^n$ to denote the matrix product $\prod_{i=1}^{n} diag(\sigma')W$ (not elementwise product).

(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.[1].

Proof by induction that for such a matrix the product $\prod_{i=1}^{n} M$ can be written as: $M^n = Q\Lambda^n Q^{-1}$!

Detailed Solution

Induction start $n = 1$:

$M^1 = (Q\Lambda^1 Q^{-1})$ (given)

[1] Every eigenvector v_i satisfies the linear equation $Mv_i = \lambda_i v_i$ where $\lambda_i = \Lambda_{ii}$

Let's try $n = 2$:

$$\begin{aligned} M^2 = (Q\Lambda Q^{-1})^2 &= (Q\Lambda Q^{-1})(Q\Lambda Q^{-1}) \\ &= Q\Lambda(Q^{-1}Q)\Lambda Q^{-1} \\ &= Q\Lambda^2 Q^{-1}. \end{aligned}$$

Induction hypothesis: $M^n = Q\Lambda^n Q^{-1}$.

Inductive step:

$$M^{n+1} = M^n M = (Q\Lambda^n Q^{-1})(Q\Lambda Q^{-1}) = Q\Lambda^{n+1} Q^{-1}.$$

$\square$

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

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

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?

Detailed Solution

$W^{30} = Q\Lambda^{30}Q^{-1} = \begin{pmatrix} 0.0153 & 0.0203 \\ 0.0203 & 0.02713 \end{pmatrix}$.

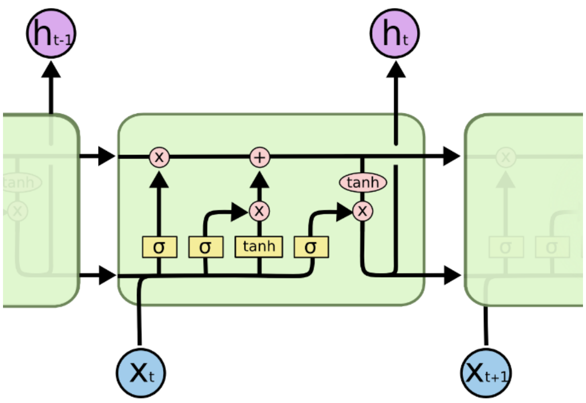
The values get very small $\Rightarrow$ vanishing gradients! This happens for all cases in which all absolute eigenvalues $|\lambda| < 1$.

If there was one eigenvalue with $|\lambda| > 1$, the values would get very large $\Rightarrow$ exploding gradients!

If all eigenvalues were 1, the values would remain constant.

Exercise 5-3 LSTMs

Recall the elements of a module in an LSTM and the corresponding computations, where $\odot$ stands for pointwise multiplication. [2]



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

[2] For a good explanation on LSTMs you can refer to http://colah.github.io/posts/2015-08-Understanding-LSTMs/

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

Detailed Solution

(i) f_t ”Forget Gate”: Decide what old information we’are going to delete out of the cell state

(ii) i_t ”Update Gate”: Decide what new information we’are using in the new cell state

(iii) o_t ”Output Gate”: Decide what information we’are going to output.

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

Detailed Solution

The gates f_t, i_t and o_t are always positive since the sigmoid activation function only outputs values between 0 and 1.

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 $C_{t-1}, ...C_0$, so we get a formula of the form:

$$\frac{\partial L}{\partial W} = \sum_{t=0}^{T} \sum_{k=1}^{t} \frac{\partial L}{\partial C_t} \frac{\partial C_t}{\partial C_k} \frac{\partial C_k}{\partial W}. \quad {}^3$$

(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?

Detailed Solution

We have $\frac{\partial C_t}{\partial C_{t-1}} \approx 1$, thus $\frac{\partial C_t}{\partial C_k} \approx 1$.

[3] The real formula is a bit more complicated since C_t also depends on f_t, i_t and $\tilde{C}_t$, which in turn all depend on W, but this can be neglected.