Machine Learning Group
Faculty IV – Electrical Engineering and Computer Science
Technische Universität Berlin

Exercise Sheet 9

Exercise 1: Computing Gradients in RNNs ($5 \times 10 + 5 \times 10 = 100$ P)

We consider the task of binary classifying univariate time series (only two time steps for the purpose of the exercise) using a recurrent neural network. Let (x_1, x_2) be the time series given as input. The recurrent neural network is given by the equations:

$$\begin{aligned} h_1 &= w \cdot x_1 + \tanh(h_0) \\ h_2 &= w \cdot x_2 + \tanh(h_1) \\ y &= h_1 + h_2, \end{aligned}$$

and we assume that the neural network has initial state $h_0 = 0$. The variable y is the neural network output and w is the model parameter. We further assume that the univariate time series (x_1, x_2) comes with a binary target label $t \in \{-1, 1\}$ and the prediction error for this data point is modeled via the log-loss function

$$\mathcal{L}(y, t) = \log(1 + \exp(-yt)).$$

We would like to extract the gradient of the objective w.r.t. the parameter w.

(a) Draw the neural network graph, and annotate it with relevant variables (inputs, activations, and parameters).

(b) *Compute* $\partial \mathcal{L} / \partial y$.

(c) Assuming the last computation was stored in g, *compute* $\partial \mathcal{L} / \partial h_2$ as a function of g.

(d) Assuming the last computation was stored in δ_2, *compute* $\partial \mathcal{L} / \partial h_1$ as a function of g and δ_2.

(e) Assuming the last computation was stored in δ_1, *compute* $\partial \mathcal{L} / \partial w$ as a function of g, δ_2 and δ_1.

(f) Repeat the steps above (a–e) for the case where the recurrent neural network is given by the equations:

$$\begin{aligned} h_1 &= \tanh(x_1 + w + h_0) \\ h_2 &= \tanh(x_2 + w + h_1) \\ y &= h_1 + h_2, \end{aligned}$$

where the initial state is set to $h_0 = 0$, the target is real-valued ($t \in \mathbb{R}$), and the error function is given by

$$\mathcal{L}(y, t) = \log \cosh(y - t).$$

Exercise 1: Computing Gradients in RNNs ($5 \times 10 + 5 \times 10 = 100$ P)

We consider the task of binary classifying univariate time series (only two time steps for the purpose of the exercise) using a recurrent neural network. Let (x_1, x_2) be the time series given as input. The recurrent neural network is given by the equations:

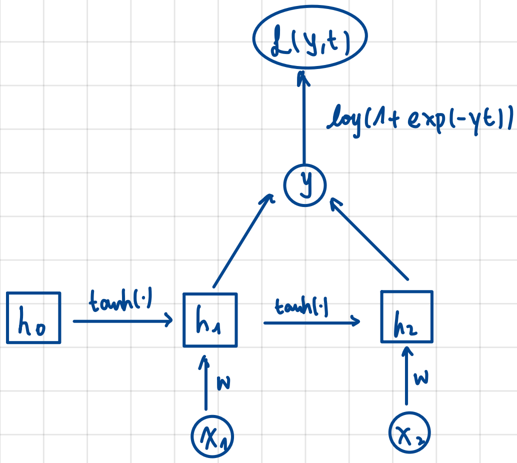
$$\begin{aligned} h_1 &= w \cdot x_1 + \tanh(h_0) \\ h_2 &= w \cdot x_2 + \tanh(h_1) \\ y &= h_1 + h_2, \end{aligned}$$

and we assume that the neural network has initial state $h_0 = 0$. The variable y is the neural network output and w is the model parameter. We further assume that the univariate time series (x_1, x_2) comes with a binary target label $t \in \{-1, 1\}$ and the prediction error for this data point is modeled via the log-loss function

$$\mathcal{L}(y,t) = \log(1 + \exp(-yt)).$$

We would like to extract the gradient of the objective w.r.t. the parameter w.

(a) Draw the neural network graph, and annotate it with relevant variables (inputs, activations, and parameters).



(b) *Compute* $\partial \mathcal{L} / \partial y$.

$$\frac{\partial \mathcal{L}}{\partial y} = \frac{1}{1+\exp(-yt)} \cdot \exp(-yt) \cdot (-t)$$

$$= - \frac{t \cdot \exp(-yt)}{1+\exp(-yt)} \quad (= g)$$

(c) Assuming the last computation was stored in g, *compute* $\partial \mathcal{L}/\partial h_2$ as a function of g.

$$\frac{\partial \mathcal{L}}{\partial h_2} = \frac{\partial \mathcal{L}}{\partial y}\frac{\partial y}{\partial h_2} = g \cdot \frac{\partial y}{\partial h_2} = g \cdot 1 = g \quad (= \delta_2)$$

$$= g \cdot \frac{\partial (h_1 + h_2)}{\partial h_2}$$

(d) Assuming the last computation was stored in δ_2, *compute* $\partial \mathcal{L}/\partial h_1$ as a function of g and δ_2.

$$\frac{\partial \mathcal{L}}{\partial h_1} = \frac{\partial \mathcal{L}}{\partial y} \cdot \frac{\partial y}{\partial h_1} + \frac{\partial \mathcal{L}}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1}$$

$$= g \cdot 1 + \delta_2 \cdot (1 - \tanh^2(h_1)) \quad (= \delta_1)$$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h_1} = \frac{\partial L}{\partial y} \cdot \frac{\partial (h_1 + h_2)}{\partial h_1} = \underbrace{\frac{\partial L}{\partial y}}_{=g} \cdot \left(\underbrace{\frac{\partial h_1}{\partial h_1}}_{=1} + \frac{\partial h_2}{\partial h_1} \right)$$

$$= g + g \cdot \frac{\partial (x_2 w + \tanh(h_1))}{\partial h_1} = g + g \cdot \left(\underbrace{\frac{\partial x_2 w}{\partial h_1}}_{=0} + \underbrace{\frac{\partial \tanh(h_1)}{\partial h_1}}_{=\tanh'(h_1)} \right)$$

$$= g + g \cdot \tanh'(h_1) = g(1 + \tanh'(h_1))$$

$$= \delta_2 (1 + \tanh'(h_1))$$

(e) Assuming the last computation was stored in δ_1, *compute* $\partial\mathcal{L}/\partial w$ as a function of g, δ_2 and δ_1.

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial h_1}\cdot\frac{\partial h_1}{\partial w} + \frac{\partial \mathcal{L}}{\partial h_2}\cdot\frac{\partial h_2^{+}}{\partial w}$$

$$= \delta_1 \cdot x_1 + \delta_2 \cdot x_2$$

$$\frac{\partial L}{\partial w} = \delta_2 \cdot \frac{\partial^+ h_2}{\partial w} + \delta_1 \cdot \frac{\partial^+ h_1}{\partial w}$$

$$= \delta_2 \cdot \frac{\partial^+ (x_2 w + \tanh(h_1))}{\partial w} + \delta_1 \cdot \frac{\partial^+ (x_1 w + \tanh(h_0))}{\partial w}$$

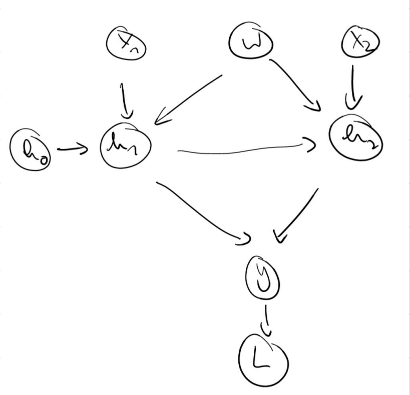
$$= \delta_2 \cdot x_2 + \delta_1 \cdot x_1$$

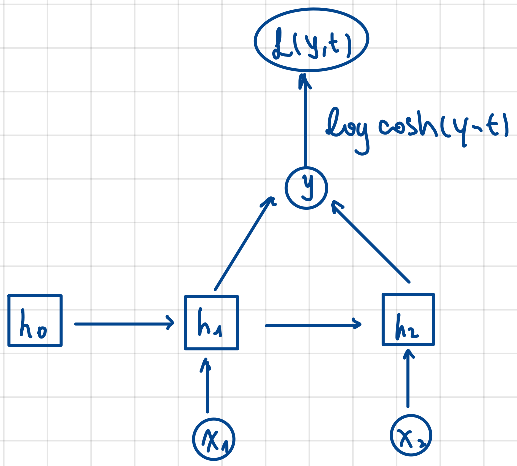
(f) Repeat the steps above (a–e) for the case where the recurrent neural network is given by the equations:

$$h_1 = \tanh(x_1 + w + h_0)$$
$$h_2 = \tanh(x_2 + w + h_1)$$
$$y = h_1 + h_2,$$

where the initial state is set to $h_0 = 0$, the target is real-valued ($t \in \mathbb{R}$), and the error function is given by

$$\mathcal{L}(y,t) = \log\cosh(y - t).$$





$$\frac{\partial \mathcal{L}}{\partial y} = \frac{\sinh(y-t)}{\cosh(y-t)}$$

$$= \tanh(y-t) \qquad (= g)$$

$$\frac{\partial \mathcal{L}}{\partial h_2} = \frac{\partial \mathcal{L}}{\partial y}\,\frac{\partial y}{\partial h_2} = g \cdot \frac{\partial y^{+}}{\partial h_2} = g \cdot 1 = g \quad (= \delta_2)$$

$$\frac{\partial \mathcal{L}}{\partial h_1} = \frac{\partial \mathcal{L}}{\partial y}\cdot\frac{\partial y^{+}}{\partial h_1} + \frac{\partial \mathcal{L}}{\partial h_2}\cdot\frac{\partial h_2^{+}}{\partial h_1}$$

$$= g \cdot 1 + \delta_2 \cdot (1 - \tanh^2(x_2 + w + h_1)) \qquad (= \delta_1)$$

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial h_1}\cdot\frac{\partial^{+} h_1}{\partial w} + \frac{\partial \mathcal{L}}{\partial h_2}\cdot\frac{\partial^{+} h_2}{\partial w}$$

$$= \delta_1 (1 - \tanh^2(x_1 + w + h_0)) + \delta_2 (1 - \tanh^2(x_2 + w + h_1))$$