Assignment 1

JD Gallego Posada, *University of Amsterdam* 23/11/2017

Exercise 1

We solve a more general case. We index the layers of the network from input ($l = 0$) to output ($l = L$) and denote the dimension of layer l by $d_l \in \mathbb{N}$. Let N be the number of examples in a batch, each example is a d_0-dimensional vector. If A is a matrix, $A_{i\cdot}$ and $A_{\cdot j}$ represent the i-th row and j-th column of A, respectively. If v and w are vectors, $v \cdot w$ represents their usual dot product. $\langle A, B\rangle_F$ represents the Frobenius inner product of A and B. $\odot$ is the usual Hadamard product. $\mathbf{1}_i^j$ is the indicator function of $i = j$. $\mathbb{1}_N$ is a N-dimensional vector of ones.

Consider a network with the following structure:

- $X = Z_0 \in \mathbb{R}^{d_0 \times N}$
- $S_i = W_i Z_{i-1} + b_i$, where $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, $b_i \in \mathbb{R}^{d_i}$ and $+$ is broadcast addition
- $Z_i = f_i(S_i)$, where $S_i \in \mathbb{R}^{d_i \times N}$, f_i is an elementwise applied non-linearity
- $Y_{out} = Z_L$, where L is the index of the last hidden layer and $Y_{gt}, Y_{out}, Z_L \in \mathbb{R}^{d_L \times N}$
- $\mathcal{L} = \frac{1}{2N} \|Y_{out} - Y_{gt}\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm.

We start by calculating the derivative of the loss wrt the output:

$$\frac{\partial \mathcal{L}}{\partial Z_L} = \frac{\partial \mathcal{L}}{\partial Y_{out}} = \frac{1}{2N} \frac{\partial \|Y_{out} - Y_{gt}\|_F^2}{\partial Y_{out}} = \frac{1}{2N} \frac{\partial \operatorname{tr}\left((Y_{out} - Y_{gt}) (Y_{out} - Y_{gt})^T \right)}{\partial Y_{out}} = \frac{1}{N} (Y_{out} - Y_{gt})$$

The multi-dimensional chain rule provides:

$$\frac{\partial \mathcal{L}}{\partial (W_l)_{ij}} = \left\langle \frac{\partial \mathcal{L}}{\partial Z_l}, \frac{\partial Z_l}{\partial (W_l)_{ij}} \right\rangle_F \qquad \frac{\partial \mathcal{L}}{\partial (b_l)_{ij}} = \left\langle \frac{\partial \mathcal{L}}{\partial Z_l}, \frac{\partial Z_l}{\partial (b_l)_{ij}} \right\rangle_F \qquad \frac{\partial \mathcal{L}}{\partial (Z_l)_{ij}} = \left\langle \frac{\partial \mathcal{L}}{\partial Z_{l+1}}, \frac{\partial Z_{l+1}}{\partial (Z_l)_{ij}} \right\rangle_F$$

Recall that the derivative of the loss wrt the activations of layer l can be written in matrix form as:

$$\frac{\partial \mathcal{L}}{\partial Z_l} = \begin{bmatrix} \ddots & & \frac{\partial \mathcal{L}}{\partial (Z_l)_{1n}} & & \ddots \\ & & \vdots & & \\ \frac{\partial \mathcal{L}}{\partial (Z_l)_{i1}} & \cdots & \frac{\partial \mathcal{L}}{\partial (Z_l)_{in}} & \cdots & \frac{\partial \mathcal{L}}{\partial (Z_l)_{iN}} \\ & & \vdots & & \\ \ddots & & \frac{\partial \mathcal{L}}{\partial (Z_l)_{d_l n}} & & \ddots \end{bmatrix}$$

Let us calculate the derivative of the activation of a layer with respect to the weights:

$$\frac{\partial (Z_l)_{kn}}{\partial (W_l)_{ij}} = \frac{\partial f_l(S_l)_{kn}}{\partial (W_l)_{ij}} = \frac{\partial f_l((W_l)_{k\cdot} \cdot (Z_{l-1})_{\cdot n} + (b_l)_k)}{\partial (W_l)_{ij}} = f_l'(S_l)_{kn} (Z_{l-1})_{jn} \mathbf{1}_i^k$$

Note how all the rows of the corresponding matrix vanish except for row i:

$$\frac{\partial Z_l}{\partial (W_l)_{ij}} = \begin{bmatrix} & & \mathbf{0} & & \\ f_l'(S_l)_{i1}(Z_{l-1})_{j1} & \cdots & f_l'(S_l)_{in}(Z_{l-1})_{jn} & \cdots & f_l'(S_l)_{iN}(Z_{l-1})_{jN} \\ & & \mathbf{0} & & \end{bmatrix}$$

So we see that upon calculating the Frobenius dot product from above we have:

$$\frac{\partial \mathcal{L}}{\partial (W_l)_{ij}} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial (Z_l)_{in}} f_l'(S_l)_{in}(Z_{l-1})_{jn} = \left(\frac{\partial \mathcal{L}}{\partial Z_l} \odot f_l'(S_l)\right)_{i\cdot} \cdot \left(Z_{l-1}^T\right)_{\cdot j}$$

Finally,

$$\frac{\partial \mathcal{L}}{\partial W_l} = \left(\frac{\partial \mathcal{L}}{\partial Z_l} \odot f_l'(S_l)\right) Z_{l-1}^T$$

A similar procedure for the biases yields:

$$\frac{\partial (Z_l)_{kn}}{\partial (b_l)_i} = \frac{\partial f_l(S_l)_{kn}}{\partial (b_l)_i} = \frac{\partial f_l((W_l)_{k\cdot} \cdot (Z_{l-1})_{\cdot n} + (b_l)_k)}{\partial (b_l)_i} = f_l'(S_l)_{kn}\mathbf{1}_i^k$$

$$\frac{\partial Z_l}{\partial (b_l)_i} = \begin{bmatrix} & & \mathbf{0} & & \\ f_l'(S_l)_{i1} & \cdots & f_l'(S_l)_{in} & \cdots & f_l'(S_l)_{iN} \\ & & \mathbf{0} & & \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial (b_l)_i} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial (Z_l)_{in}} f_l'(S_l)_{in}$$

$$\frac{\partial \mathcal{L}}{\partial b_l} = \left(\frac{\partial \mathcal{L}}{\partial Z_l} \odot f_l'(S_l)\right) \mathbb{1}_N$$

Let us conduct the analysis for the derivative activation of layer $l+1$ wrt the activation of layer l:

$$\frac{\partial (Z_{l+1})_{kn}}{\partial (Z_l)_{ij}} = \frac{\partial f_{l+1}(S_{l+1})_{kn}}{\partial (Z_l)_{ij}} = \frac{\partial f_l((W_{l+1})_{k\cdot} \cdot (Z_l)_{\cdot n} + (b_{l+1})_k)}{\partial (Z_l)_{ij}} = f_{l+1}'(S_{l+1})_{kn}(W_{l+1})_{ki}\mathbf{1}_j^n$$

Note how all the columns of the corresponding matrix vanish except for column j:

$$\frac{\partial Z_{l+1}}{\partial (Z_l)_{ij}} = \begin{bmatrix} & f_{l+1}'(S_{l+1})_{1j}(W_{l+1})_{1i} & \\ & \vdots & \\ \mathbf{0} & f_{l+1}'(S_{l+1})_{kj}(W_{l+1})_{ki} & \mathbf{0} \\ & \vdots & \\ & f_{l+1}'(S_{l+1})_{d_{l+1}j}(W_{l+1})_{d_{l+1}i} & \end{bmatrix}$$

Calculating the Frobenius dot product from above we have:

$$\frac{\partial \mathcal{L}}{\partial (Z_l)_{ij}} = \sum_{k=1}^{d_{l+1}} \frac{\partial \mathcal{L}}{\partial (Z_{l+1})_{kj}} f_{l+1}'(S_{l+1})_{kj}(W_{l+1})_{ki} = \left(W_{l+1}^T\right)_{i\cdot} \cdot \left(\frac{\partial \mathcal{L}}{\partial Z_{l+1}} \odot f_{l+1}'(S_{l+1})\right)_{\cdot j}$$

In matrix notation,

$$\frac{\partial \mathcal{L}}{\partial Z_l} = W_{l+1}^T \left(\frac{\partial \mathcal{L}}{\partial Z_{l+1}} \odot f_{l+1}'(S_{l+1})\right)$$

Note the recurrent pattern of a matrix of the form $\delta_l := \frac{\partial \mathcal{L}}{\partial Z_l} \odot f_l'(S_l)$. Note that $\delta_L = \frac{1}{N}\left(Y_{out} - Y_{gt}\right) \odot f_L'(S_L)$. Introducing this notation allows us to simplify the previous equations as:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \delta_l Z_{l-1}^T \qquad \frac{\partial \mathcal{L}}{\partial b_l} = \delta_l \mathbb{1}_N \qquad \frac{\partial \mathcal{L}}{\partial Z_l} = W_{l+1}^T \delta_{l+1} \qquad \delta_l = \left(W_{l+1}^T \delta_{l+1}\right) \odot f_l'(S_l)$$

Exercise 2

$$S_1 = W_1 X = \begin{bmatrix} 0.6 & 0.01 \\ 0.7 & 0.43 \\ 0 & 0.88 \end{bmatrix} \begin{bmatrix} 0.75 & 0.2 & -0.75 & 0.2 \\ 0.8 & 0.05 & 0.8 & -0.05 \end{bmatrix} = \begin{bmatrix} 0.458 & 0.1205 & -0.442 & 0.1195 \\ 0.869 & 0.1615 & -0.181 & 0.1185 \\ 0.704 & 0.044 & 0.704 & -0.044 \end{bmatrix}$$

$$Z_1 = \mathsf{relu}(S_1) = \begin{bmatrix} 0.458 & 0.1205 & 0 & 0.1195 \\ 0.869 & 0.1615 & 0 & 0.1185 \\ 0.704 & 0.044 & 0.704 & 0 \end{bmatrix}$$

$$Y_{out} = Z_2 = S_2 = W_2 Z_1 = \begin{bmatrix} 0.02 & 0.03 & 0.09 \end{bmatrix} \begin{bmatrix} 0.458 & 0.1205 & 0 & 0.1195 \\ 0.869 & 0.1615 & 0 & 0.1185 \\ 0.704 & 0.044 & 0.704 & 0 \end{bmatrix} = \begin{bmatrix} 0.0985 & 0.0112 & 0.0633 & 0.0059 \end{bmatrix}$$

$$\mathcal{L} = \frac{1}{2 \cdot 4}\left((0.0985 - 1)^2 + (0.0112 - 1)^2 + (0.0633 + 1)^2 + (0.0059 + 1)^2\right) = 0.49161$$

$$\delta_2 = \frac{1}{4}\begin{bmatrix} 0.0985 - 1 & 0.0112 - 1 & 0.0633 + 1 & 0.0059 + 1 \end{bmatrix} = \begin{bmatrix} -0.2253 & -0.2471 & 0.2658 & 0.2514 \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial W_2} = \delta_2 Z_1^T = \begin{bmatrix} -0.2253 & -0.2471 & 0.2658 & 0.2514 \end{bmatrix} \begin{bmatrix} 0.458 & 0.1205 & 0 & 0.1195 \\ 0.869 & 0.1615 & 0 & 0.1185 \\ 0.704 & 0.044 & 0.704 & 0 \end{bmatrix}^T = \begin{bmatrix} -0.1029 \\ -0.2059 \\ 0.0176 \end{bmatrix}^T$$

$$\delta_1 = (W_2^T \delta_2) \odot f_1'(S_1) = \left(\begin{bmatrix} 0.02 \\ 0.03 \\ 0.09 \end{bmatrix} \begin{bmatrix} -0.2253 \\ -0.2471 \\ 0.2658 \\ 0.2514 \end{bmatrix}^T\right) \odot \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -0.0045 & -0.0049 & 0 & 0.0050 \\ -0.0068 & -0.0074 & 0 & 0.0075 \\ -0.0203 & -0.0222 & 0.0239 & 0 \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = \delta_1 X^T = \begin{bmatrix} -0.0045 & -0.0049 & 0 & 0.0050 \\ -0.0068 & -0.0074 & 0 & 0.0075 \\ -0.0203 & -0.0222 & 0.0239 & 0 \end{bmatrix} \begin{bmatrix} 0.75 & 0.2 & -0.75 & 0.2 \\ 0.8 & 0.05 & 0.8 & -0.05 \end{bmatrix}^T = \begin{bmatrix} -0.0033 & -0.0041 \\ -0.0050 & -0.0061 \\ -0.0376 & 0.0018 \end{bmatrix}$$

$$W_2 \leftarrow W_2 - 0.5 \frac{\partial \mathcal{L}}{\partial W_2} = \begin{bmatrix} 0.0714 & 0.1329 & 0.0811 \end{bmatrix}$$

$$W_1 \leftarrow W_1 - 0.5 \frac{\partial \mathcal{L}}{\partial W_1} = \begin{bmatrix} 0.6016 & 0.0120 \\ 0.7025 & 0.4330 \\ 0.0188 & 0.8790 \end{bmatrix}$$

Exercise 3

(i) Tries to ensure that the probability assigned to the correct class is at least $margin$ larger than the probability assigned to each of the incorrect classes.

(ii)

$$
\begin{aligned}
\frac{\partial \mathcal{L}^{(i)}_{hinge}}{\partial o_j} &= \frac{\partial \sum_{l \neq y_i} \text{max}(0, p_l - p_{y_i} + margin)}{\partial o_j} \\
&= \sum_{l \neq y_i} \begin{cases} 0 \quad p_l - p_{y_i} + margin < 0 \\ \frac{\partial p_l - p_{y_i} + margin}{\partial o_j} \quad \text{else} \end{cases} \\
&= \sum_{l \neq y_i} \begin{cases} 0 \quad p_l - p_{y_i} + margin < 0 \\ \frac{\partial \frac{e^{o_l} - e^{o_{y_i}}}{\sum_r e^{o_r}}}{\partial o_j} \quad \text{else} \end{cases}
\end{aligned}
$$

$$
\frac{\partial \frac{e^{o_l} - e^{o_{y_i}}}{\sum_r e^{o_r}}}{\partial o_j} = \frac{e^{o_j}(\mathbf{1}^j_l - \mathbf{1}^j_{y_i})}{\sum_r e^{o_r}} - \frac{e^{o_j}(e^{o_l} - e^{o_{y_i}})}{(\sum_r e^{o_r})^2} = p_j(\mathbf{1}^j_l - \mathbf{1}^j_{y_i}) - p_l p_j + p_j p_{y_i} = p_j(\mathbf{1}^j_l - \mathbf{1}^j_{y_i} - p_l + p_{y_i})
$$

$$
\frac{\partial \mathcal{L}_{hinge}}{\partial o_j} = \sum_i \frac{\partial \mathcal{L}^{(i)}_{hinge}}{\partial o_j} = \sum_i \sum_{l \neq y_i} p_j(\mathbf{1}^j_l - \mathbf{1}^j_{y_i} - p_l + p_{y_i}) \mathbf{1}_{\{p_l - p_{y_i} + margin \geq 0\}}
$$

Exercise 4

The recurrent pattern observed in Exercise 1 showed a high dependence of the gradient step on the values computed in the forward pass. In the case we have memory issues, we could compromise some speed in the parameter updates by chaching only the preactivations (instead of the activations as well) and calculate the activations *in situ* during the gradient step.