Deep Learning 1
Assignment 1

Tijs Wiegman, 13865617

2 november 2024

Question 1: Consider a linear module as described above. The input and output features are labeled as $\mathbf{X}$ and $\mathbf{Y}$, respectively. Find closed form expressions for

a) $\dfrac{\partial L}{\partial \mathbf{W}}$

b) $\dfrac{\partial L}{\partial \mathbf{b}}$

c) $\dfrac{\partial L}{\partial \mathbf{X}}$

in terms of the gradients of the loss with respect to the output features $\frac{\partial L}{\partial \mathbf{Y}}$ provided by the next module during backpropagation. Assume the gradients have the same shape as the object with respect to which is being differentiated. E.g. $\frac{\partial L}{\partial \mathbf{W}}$ should have the same shape as $\mathbf{W}$, $\frac{\partial L}{\partial \mathbf{b}}$ should be a row-vector just like $\mathbf{b}$ etc.

Solution

To start, note that for the elements of $\mathbf{Y}$ we can write

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top + \mathbf{B} \implies Y_{ij} = \sum_{k=1}^{M} X_{ik} W_{jk} + B_{ij}$$

$$\frac{\partial L}{\partial W_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial W_{nm}}$$

$$\begin{aligned}
\frac{\partial Y_{ij}}{\partial W_{nm}} &= \frac{\partial}{\partial W_{nm}} \left[\sum_{k=1}^{M} X_{ik} W_{jk} + B_{ij} \right] \\
&= \sum_{k=1}^{M} X_{ik} \frac{\partial W_{jk}}{\partial W_{nm}} \\
&= \sum_{k=1}^{M} X_{ik} \delta_{jn} \delta_{km} \\
&= X_{im} \delta_{jn}
\end{aligned}$$

$$\implies \frac{\partial L}{\partial W_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} X_{im} \delta_{jn} = \sum_i \frac{\partial L}{\partial Y_{in}} X_{im} \implies \frac{\partial L}{\partial \mathbf{W}} = \left(\frac{\partial L}{\partial \mathbf{Y}}\right)^\top \mathbf{X}$$

$$\frac{\partial L}{\partial b_\ell} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial b_\ell}$$

$$\frac{\partial Y_{ij}}{\partial b_\ell} = \frac{\partial}{\partial b_\ell}\left[\sum_{k=1}^{M} X_{ik} W_{jk} + B_{ij}\right] = \frac{\partial b_j}{\partial b_\ell} = \frac{\partial B_{ij}}{\partial b_\ell} = \delta_{j\ell}$$

$$\implies \frac{\partial L}{\partial b_\ell} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \delta_{j\ell} = \sum_i \frac{\partial L}{\partial Y_{i\ell}} \implies \frac{\partial L}{\partial \mathbf{b}} = \mathbf{1}^\top \frac{\partial L}{\partial \mathbf{Y}}$$

$$\frac{\partial L}{\partial X_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial X_{nm}}$$

$$\begin{aligned}
\frac{\partial Y_{ij}}{\partial X_{nm}} &= \frac{\partial}{\partial X_{nm}}\left[\sum_{k=1}^{M} X_{ik} W_{jk} + B_{ij}\right] \\
&= \sum_{k=1}^{M} \frac{\partial X_{ik}}{\partial X_{nm}} W_{jk} \\
&= \sum_{k=1}^{M} \delta_{in}\delta_{km} W_{jk} \\
&= \delta_{in} W_{jm}
\end{aligned}$$

$$\implies \frac{\partial L}{\partial X_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \delta_{in} W_{jm} = \sum_j \frac{\partial L}{\partial Y_{nj}} W_{jm} \implies \frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \mathbf{W}$$

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top + \mathbf{B} \implies Y_{ij} = \sum_{k=1}^{M} X_{ik}W_{jk} + B_{ij}$$

$$\frac{\partial L}{\partial W_{nm}} = \sum_i \frac{\partial L}{\partial Y_{ni}} \frac{\partial Y_{ni}}{\partial W_{nm}}$$

$$\begin{aligned}
\frac{\partial Y_{ni}}{\partial W_{nm}} &= \frac{\partial}{\partial W_{nm}} \left[\sum_{k=1}^{M} X_{nk}W_{ik} + B_{ni} \right] \\
&= \sum_{k=1}^{M} X_{nk} \frac{\partial W_{ik}}{\partial W_{nm}} \\
&= \sum_{k=1}^{M} X_{nk}\delta_{in}\delta_{km} \\
&= X_{nm}\delta_{in}
\end{aligned}$$

$$\implies \frac{\partial L}{\partial W_{nm}} = \sum_i \frac{\partial L}{\partial Y_{ni}} X_{nm}\delta_{in}$$

Consider an element-wise activation function h. The activation module has input and output features labelled by $\mathbf{X}$ and $\mathbf{Y}$, respectively. I.e. $\mathbf{Y} = h(\mathbf{X}) \implies Y_{ij} = h(X_{ij})$. Find a closed-form expression for

$$\frac{\partial L}{\partial \mathbf{X}}$$

in terms of the gradient of the loss with respect to the output features $\frac{\partial L}{\partial \mathbf{Y}}$ provided by the next module. Assume the gradient has the same shape as $\mathbf{X}$.

Solution

$$\frac{\partial L}{\partial X_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial X_{nm}}$$

$$\frac{\partial Y_{ij}}{\partial X_{nm}} = \frac{\partial h(X_{ij})}{\partial X_{nm}} = \delta_{in}\delta_{jm} h'(X_{ij})$$

$$\implies \frac{\partial L}{\partial X_{nm}} = \sum_{i,j} \frac{\partial L}{\partial Y_{ij}} \delta_{in}\delta_{jm} h'(X_{ij}) = \frac{\partial L}{\partial Y_{nm}} h'(X_{nm})$$

$$\implies \frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{Y}} \circ h'(\mathbf{X})$$

e) Let $\mathbf{Z} \in \mathbb{R}^{S \times C}$ be a feature matrix with S samples at the end of a deep neural network. Consider a softmax layer $Y_{ij} = \frac{e^{Z_{ij}}}{\sum_k e^{Z_{ik}}}$ followed by a categorical cross-entropy loss. The final scalar loss L is the arithmetic mean of $L_i = -\sum_k T_{ik} \log(Y_{ik})$ over all samples i in the batch. Targets are collected in $\mathbf{T} \in \mathbb{R}^{S \times C}$ and the elements of each row sum to 1. It can be shown that the gradients of these modules have the following closed form:

$$\frac{\partial L}{\partial \mathbf{Z}} = \mathbf{Y} \circ \left(\frac{\partial L}{\partial \mathbf{Y}} - \left(\frac{\partial L}{\partial \mathbf{Y}} \circ \mathbf{Y} \right) \mathbf{1}\mathbf{1}^\top \right)$$

$$\frac{\partial L}{\partial \mathbf{Y}} = -\frac{1}{S} \frac{\mathbf{T}}{\mathbf{Y}}$$

The Hadamard product is defined by $[\mathbf{A} \circ \mathbf{B}]_{ij} = A_{ij} B_{ij}$ and the division of the two matrices is also element-wise. The ones vector is denoted by $\mathbf{1}$ and its size is such that the matrix multiplication in the expression above is well-defined.

All gradients of the loss have the shape of the object with respect to which is being differentiated. One can combine these into a single module with the following gradient:

$$\frac{\partial L}{\partial \mathbf{Z}} = \alpha \mathbf{M}$$

Find expressions for the positive scalar $\alpha \in \mathbb{R}^+$ and the matrix $\mathbf{M} \in \mathbb{R}^{S \times C}$ in terms of $\mathbf{Y}$, $\mathbf{T}$, and S.

Since the division of two matrices is element-wise, we can write

$$\left(\frac{\mathbf{T}}{\mathbf{Y}} \right)_{ij} = \frac{T_{ij}}{Y_{ij}} \implies \left(\frac{\mathbf{T}}{\mathbf{Y}} \circ \mathbf{Y} \right)_{ij} = \frac{T_{ij}}{Y_{ij}} Y_{ij} = T_{ij} \implies \frac{\mathbf{T}}{\mathbf{Y}} \circ \mathbf{Y} = \mathbf{T}$$

We also know the rows of $\mathbf{T}$ sum to 1, i.e. $\sum_j T_{ij} = 1$. We get

$$(\mathbf{T1})_i = \sum_j \mathbf{T}_{ij}\mathbf{1}_j = \sum_j T_{ij} = 1 \implies \mathbf{T1} = \mathbf{1}$$

Lastly, note that for any matrix $\mathbf{A}$, we get

$$(\mathbf{A} \circ \mathbf{11}^\top)_{ij} = \mathbf{A}_{ij}(\mathbf{11}^\top)_{ij} = A_{ij} \cdot 1 = A_{ij} \implies \mathbf{A} \circ \mathbf{11}^\top = \mathbf{A}$$

Putting this together, we find

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{Z}} &= \mathbf{Y} \circ \left(-\frac{1}{S}\frac{\mathbf{T}}{\mathbf{Y}} - \left(-\frac{1}{S}\frac{\mathbf{T}}{\mathbf{Y}} \circ \mathbf{Y} \right) \mathbf{11}^\top \right) \\
&= \mathbf{Y} \circ \left(-\frac{1}{S}\frac{\mathbf{T}}{\mathbf{Y}} + \frac{1}{S}\mathbf{T11}^\top \right) \\
&= \frac{1}{S}\left(-\mathbf{Y} \circ \frac{\mathbf{T}}{\mathbf{Y}} + \mathbf{Y} \circ (\mathbf{11}^\top) \right) \\
&= \frac{1}{S}\left(-\mathbf{T} + \mathbf{Y} \right) \\
&\implies \alpha = \frac{1}{S}, \qquad \mathbf{M} = \mathbf{Y} - \mathbf{T}
\end{aligned}$$

Question 4: Consider point x_p where $\nabla_{\mathbf{x}} f(\mathbf{x}_p) = \mathbf{0}$, we call this point a critical or stationary point (the p is to represent the critical point in $\mathbf{x}$). If a critical point is not a local maximum or minimum, it will be classified as a saddle point. To determine if a critical point in a higher dimension is a local minimum or maximum, we can use the Hessian matrix check. Applying the Hessian matrix to a critical point $H(\mathbf{x}_p)$ captures how the function curves around the critical point in a higher dimension, similar to how the derivative captures how a quadratic function curves around the critical point in 2 dimensions.

For continuously differentiable function f and real non-singular (invertible) Hessian matrix H at point $\mathbf{x}_p$, if H is positive definite we have a strictly local minimum, and if it is negative definite we have a strictly local maximum.

a) Show that the eigenvalues for the Hessian matrix in a strictly local minimum are all positive.

Suppose $\mathbf{x}_p$ is a stictly local minimum, so that $\nabla_{\mathbf{x}} f(\mathbf{x}_p) = \mathbf{0}$. We will assume f is **twice** continuously differentiable, so that its Hessian exists and the parital derivatives commute:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \implies H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} = H_{ji}$$

Thus, under these circumstances, H is symmetric, meaning its eigenvalues are all real, at any point including $\mathbf{x}_p$. We consider the second-order Taylor expansion of $f(\mathbf{x})$ around

$\mathbf{x}_p$, with $\mathbf{h} = \mathbf{x} - \mathbf{x}_p$:

$$f(\mathbf{x}) \approx f(\mathbf{x}_p) + \nabla_{\mathbf{x}} f(\mathbf{x}_p)^\top (\mathbf{x} - \mathbf{x}_p) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_p)^\top H(\mathbf{x}_p)(\mathbf{x} - \mathbf{x}_p)$$
$$\implies f(\mathbf{h} + \mathbf{x}_p) \approx f(\mathbf{x}_p) + \frac{1}{2}\mathbf{h}^\top H(\mathbf{x}_p)\mathbf{h}$$

Now to prove H is positive definite, let $\mathbf{x}$ sufficiently close to $\mathbf{x}_p$. As $\mathbf{x}_p$ is a strictly local minimum we have $f(\mathbf{x}) > f(\mathbf{x}_p)$, which means we require $(\mathbf{x} - \mathbf{x}_p)^\top H(\mathbf{x}_p)(\mathbf{x} - \mathbf{x}_p) > 0$ **Can I use the fact that at in a strictly local minimum, the Hessian matrix is positive definite?

b) If some of the eigenvalues of the Hessian matrix at point p are positive and some are negative, this point would be a saddle point; intuitively explain why the number of saddle points is exponentially larger than the number of local minima for higher dimensions?
Hint: Think of the eigenvalue sign as flipping a coin with probability (1/2) for a head coming up (positive sign).

Following the hint, note that for each eigenvalue, the sign has probability 1/2 of being positive and probability 1/2 of being negative. For a local minimum we need all eigenvalues to be positive, for a local maximum we need all eigenvalues to be negative, and for a saddle point the signs of the eigenvalues need to be mixed, i.e. atleast one is positive and atleast one is negative. If we consider $\mathbb{R}^n$, then $H \in \mathbb{R}^{n \times n}$ and thus has n eigenvalues. The probability that all are postitve is $(\frac{1}{2})^n$ and the probability that all are negative is $(\frac{1}{2})^n$. This means the probability that the signs are mixed is $1 - 2 \cdot (\frac{1}{2})^n = 1 - (\frac{1}{2})^{n-1}$, which increases exponentially with the number of dimensions n.

c) By using the update formula of gradient descent around saddle point p, show why saddle points can be harmful to training.

For weights $\mathbf{w}$ and loss function L, gradient descent in training is used to iteratively used to update the weights to decrease the loss. The update rule of gradient descent at iteration τ is

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \cdot \nabla_{\mathbf{w}} L(\mathbf{w}^{(\tau)})$$

At a sadde point, the first derivative can be 0 as the area is nearly flat in all direction