HW1 - Theory

hhp9256 - Hoang Pham

1. (Regression Task)

   (a) There are 5 steps needed:
   - Forward pass the batch input to the model.
   - Compute the loss.
   - Clear the gradient.
   - Backward pass, computing the new gradient using SGD.
   - Update the model with the new gradient.

   (b) To make it simple, denote $\texttt{Linear}_i$ as L_i for $i = 1, 2$. The model can be written in short by:

$$\hat{\mathbf{y}} = g(L_2(f(L_1(\mathbf{x}))))$$

   - $L_1(x) = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$.
   - $f(L_1(\mathbf{x})) = 5\text{ReLU}(L_1(\mathbf{x})) = 5(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})^+$.
   - $L_2(f(L_1(\mathbf{x}))) = 5\mathbf{W}^{(2)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})^+ + \mathbf{b}^{(2)}$.
   - Lastly, since g is the identity function,

$$\hat{\mathbf{y}} = g(L_2(f(L_1(\mathbf{x})))) = 5\mathbf{W}^{(2)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})^+ + \mathbf{b}^{(2)}.$$

   - The loss function is calculated by

$$\ell = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \left\|\mathbf{y} - \left(5\mathbf{W}^{(2)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})^+ + \mathbf{b}^{(2)}\right)\right\|^2.$$

   (c) For some $m \in \mathbb{N}$, we have:
   - $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^K$.
   - $\mathbf{W}^{(1)} \in \mathbb{R}^{m\times n}$, $\mathbf{b}^{(1)} \in \mathbb{R}^m$.
   - $\mathbf{W}^{(2)} \in \mathbb{R}^{K\times m}$, $\mathbf{b}^{(2)} \in \mathbb{R}^k$.

   We have $\hat{\mathbf{y}} = g(\mathbf{z}_3) \in \mathbb{R}^K$, $\mathbf{z}_3 = L_2(\mathbf{z}_2) \in \mathbb{R}^K$, $\mathbf{z}_2 = f(\mathbf{z}_1) \in \mathbb{R}^m$, and $\mathbf{z}_1 = L_1(\mathbf{x}) \in \mathbb{R}^m$.
   - Several pre-calculation gradient of ℓ:,

$$\frac{\partial \ell}{\mathbf{z}_3} = \frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}$$

     and since $\mathbf{z}_3 = \mathbf{W}^{(2)}\mathbf{z}_2 + \mathbf{b}^{(2)}$,

$$\frac{\partial \ell}{\partial \mathbf{z}_1} = \left(\frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}\right)\frac{\partial \mathbf{z}_3}{\partial \mathbf{z}_2}\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}\right) = \left(\frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}\right)\mathbf{W}^{(2)}\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}\right)$$

   - First, we calculate the gradient for the bias terms:

$$\frac{\partial \ell}{\partial \mathbf{b}^{(2)}} = \frac{\partial \ell}{\mathbf{z}_3}\frac{\partial \mathbf{z}_3}{\partial \mathbf{b}^{(2)}} = \frac{\partial \ell}{\mathbf{z}_3} = \frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}.$$

     Similarly,

$$\frac{\partial \ell}{\partial \mathbf{b}^{(1)}} = \frac{\partial \ell}{\partial \mathbf{z}_1}\frac{\partial \mathbf{z}_1}{\mathbf{b}^{(1)}} = \frac{\partial \ell}{\partial \mathbf{z}_1} = \left(\frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}\right)\mathbf{W}^{(2)}\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}\right)$$

- We calculate the gradient for the matrix weights. By chain rules, using numerator layout:

$$\begin{aligned}
\frac{\partial \ell}{\partial \mathbf{W}^{(2)}} &= \frac{\partial \ell}{\partial \mathbf{z}_3}\frac{\partial \mathbf{z}_3}{\partial \mathbf{W}^{(2)}} = \begin{bmatrix} \frac{\partial \ell}{\partial \mathbf{z}_{3,1}} & \cdots & \frac{\partial \ell}{\partial \mathbf{z}_{3,K}} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{z}_{3,1}}{\partial \mathbf{W}^{(2)}} \\ \vdots \\ \frac{\partial \mathbf{z}_{3,K}}{\partial \mathbf{W}^{(2)}} \end{bmatrix} \\
&= \sum_{i=1}^{K} \frac{\partial \ell}{\partial \mathbf{z}_{3,i}}\frac{\partial \mathbf{z}_{3,i}}{\partial \mathbf{W}^{(2)}} \\
&= \sum_{i=1}^{K} \left(\frac{\partial \ell}{\partial \mathbf{z}_{3,i}} \begin{bmatrix} \mathbf{0}^\top \\ \vdots \\ \mathbf{z}_2^\top \ (i\text{−th index}) \\ \vdots \\ \mathbf{0}^\top \end{bmatrix} \right) \quad (\text{since } \mathbf{z}_3 = 5\mathbf{W}^{(2)}\mathbf{z}_2 + \mathbf{b}^{(2)}) \\
&= \frac{\partial \ell}{\partial \mathbf{z}_3}\mathbf{z}_2^\top = \mathbf{z}_2\frac{\partial \ell}{\partial \mathbf{z}_3} \\
&= \left(5(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})^+\right)\frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}
\end{aligned}$$

and similarly

$$\frac{\partial \ell}{\partial \mathbf{W}^{(1)}} = \frac{\partial \ell}{\partial \mathbf{z}_1}\frac{\partial \mathbf{z}_1}{\partial \mathbf{W}^{(1)}} = \sum_{j=1}^{m} \frac{\partial \ell}{\partial \mathbf{z}_{1,j}}\frac{\partial \mathbf{z}_{1,j}}{\partial \mathbf{W}^{(1)}},$$

where the tensor can be evaluated as

$$\frac{\partial \mathbf{z}_{1,j}}{\partial \mathbf{W}^{(1)}} = \begin{bmatrix} \mathbf{0}^\top \\ \vdots \\ \mathbf{x}^\top \ (j\text{−th index}) \\ \vdots \\ \mathbf{0}^\top \end{bmatrix}$$

since $\mathbf{z}_1 = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$. Therefore

$$\frac{\partial \ell}{\partial \mathbf{W}^{(1)}} = \frac{\partial \ell}{\partial \mathbf{z}_1}\mathbf{x}^\top = \mathbf{x}\frac{\partial \ell}{\partial \mathbf{z}_1} = \mathbf{x}\left(\frac{\partial \ell}{\partial \hat{\mathbf{y}}}\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}\right)\mathbf{W}^{(2)}\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}\right)$$

(d) Since $\mathbf{z}_2 = 5\text{ReLU}(\mathbf{z}_1)$, and the derivative of ReLU function is $\mathbb{1}_{x>0}$, we have

$$\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1} = \begin{bmatrix} 5\mathbb{1}_{\{\mathbf{z}_{1,1}>0\}} & & \\ & \ddots & \\ 0 & & 5\mathbb{1}_{\{\mathbf{z}_{1,K}>0\}} \end{bmatrix} \in \mathbb{R}^K,$$

where $\mathbf{z}_1 = [\mathbf{z}_{1,1} \dots \mathbf{z}_{1,K}]^\top$.

Additionally, since g is identity, $\hat{\mathbf{y}} = g(\mathbf{z}_3) = \mathbf{z}_3$, so $\dfrac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3} = \mathbf{I}_K$ the identity matrix in $\mathbb{R}^{K\times K}$.

Lastly, for $h : \mathbb{R}^K \to \mathbb{R}^K$, define $\ell(h) = \|h\|^2$ and $h(\hat{\mathbf{y}}) = \mathbf{y} - \hat{\mathbf{y}}$, we have:

$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}} = \frac{\partial \ell}{\partial h}\frac{\partial h}{\partial \hat{\mathbf{y}}} = -2h^\top\mathbf{I}_K = -2(\mathbf{y} - \hat{\mathbf{y}})^\top \in \mathbb{R}^{1\times K}$$

2. (Classification Task)

(a) Performing the same operation as the previous part, the forward output are

- $L_1(x) = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$.

- $f(L_1(\mathbf{x})) = \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$.
- $L_2(f(L_1(\mathbf{x}))) = \mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}$.
- $\hat{\mathbf{y}} = g(L_2(f(L_1(\mathbf{x})))) = \sigma(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$.
- The loss function is calculated by

$$\ell = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \left\|\mathbf{y} - \sigma(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})\right\|^2.$$

The gradients are:

$$\frac{\partial \ell}{\partial \mathbf{b}^{(2)}} = \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}$$

$$\frac{\partial \ell}{\partial \mathbf{W}^{(2)}} = \tanh\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}$$

$$\frac{\partial \ell}{\partial \mathbf{b}^{(1)}} = \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3} \mathbf{W}^{(2)} \frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}$$

$$\frac{\partial \ell}{\partial \mathbf{W}^{(1)}} = \mathbf{x} \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3} \mathbf{W}^{(2)} \frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}$$

And,

$$\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1} = \begin{bmatrix} 1 - \tanh(\mathbf{z}_{1,1}^2) & & \\ & \ddots & 0 \\ 0 & & 1 - \tanh(\mathbf{z}_{1,K}^2) \end{bmatrix} \in \mathbb{R}^K,$$

$$\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3} = \begin{bmatrix} \sigma(\mathbf{z}_{3,1})(1 - \sigma(\mathbf{z}_{3,1})) & & \\ & \ddots & 0 \\ 0 & & \sigma(\mathbf{z}_{3,K})(1 - \sigma(\mathbf{z}_{3,K})) \end{bmatrix} \in \mathbb{R}^K,$$

$\frac{\partial \ell}{\partial \hat{\mathbf{y}}}$ stays the same.

(b) The difference from the previous part and this part is the loss function, and its gradient to the weights. The output in each layer remains the same:

$$\begin{aligned} \ell &= \frac{1}{K} \sum_{i=1}^{K} -\left[y_i \log(\hat{y}_i) + (1 - y_i)(1 - \log(\hat{y}_i))\right] \\ &= -\frac{1}{K} \left[\mathbf{y}^T \log(\sigma(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})) + (\mathbf{1} - \mathbf{y}^T) \log(\mathbf{1} - \sigma(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}))\right] \end{aligned}$$

The gradients are calculated similarly, except for

$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}} = \begin{bmatrix} \frac{\partial \ell}{\partial \hat{\mathbf{y}}_1} & \cdots & \cdots & \frac{\partial \ell}{\partial \hat{\mathbf{y}}_K} \end{bmatrix}$$

where

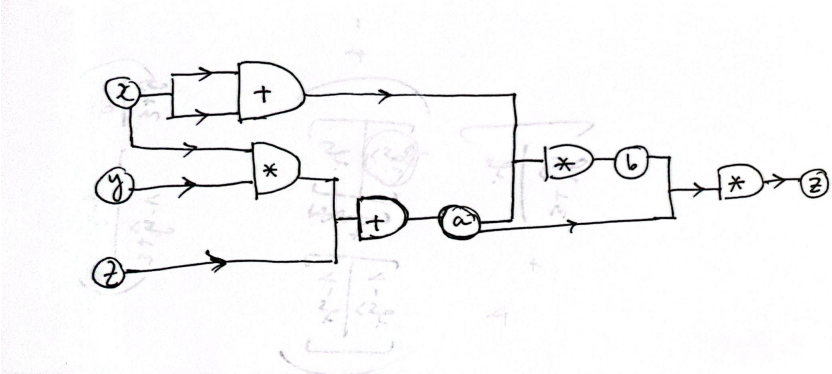
$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}_i} = -1 \left(\frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i} - \frac{1 - \mathbf{y}_i}{1 - \hat{\mathbf{y}}_i}\right).$$

(c) Using $f(.) = (.)^+$ makes the forwarding output of f non-negative, therefore backward pass in the gradient is not vanishing in a deeper network.

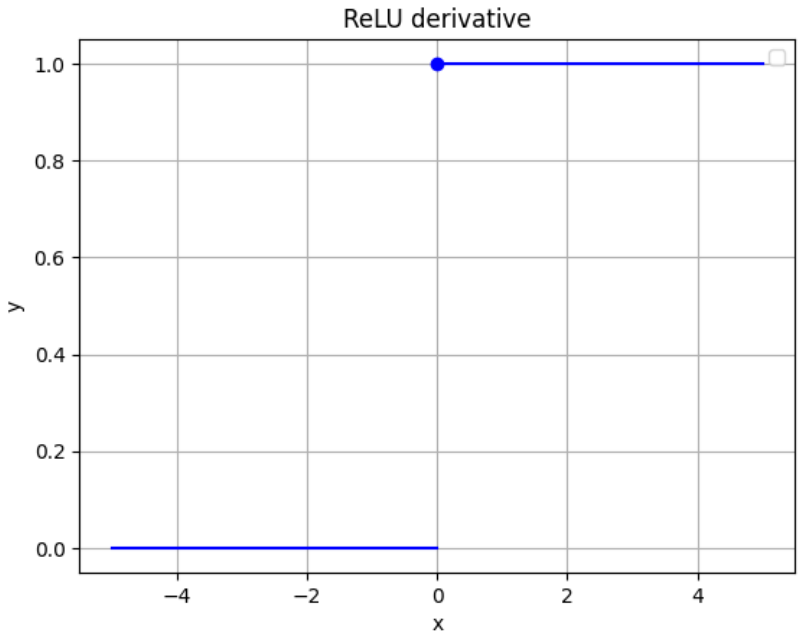
3. (Conceptual question)

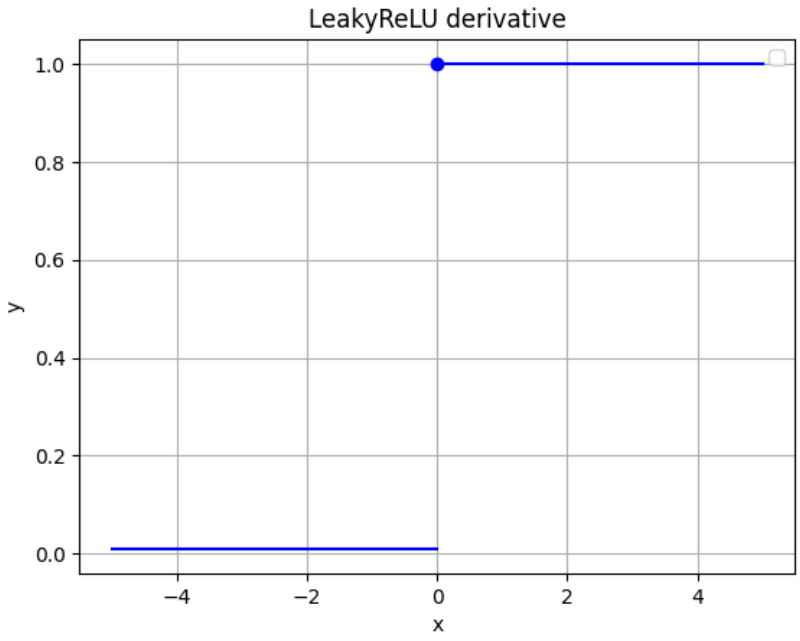
(a) softargmax is just taking the largest element out of softmax. With softmax, the output is a probability distribution, and using argmax to the output probability distribution you can yield softargmax.

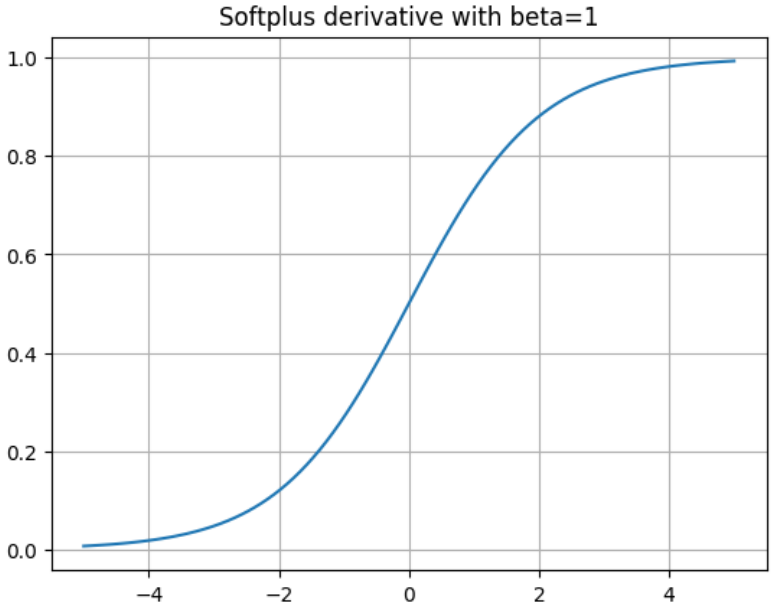
(b)

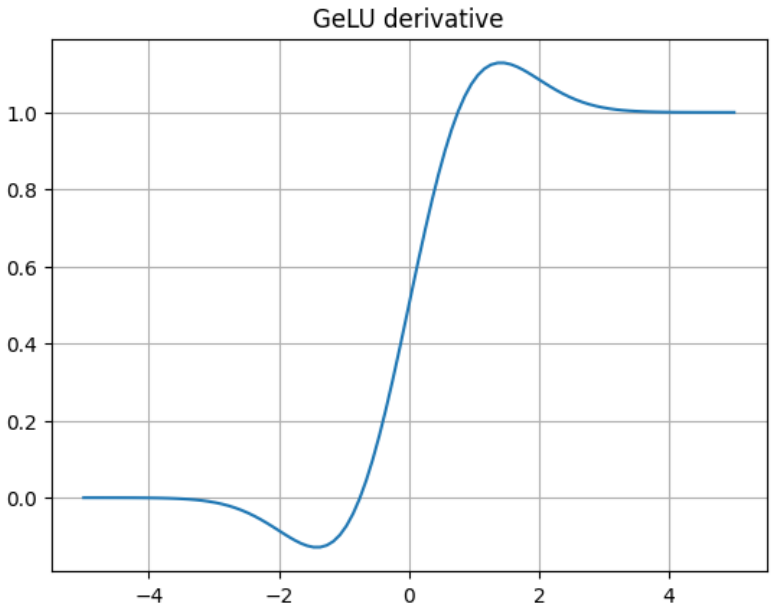


(c) Derivative plot:









(d) Linear transformations include rotation, translation, reflection, shearing. Without a non-linear activation function, composed linear transformations is just a single affine transformation. We need the non-linear transformations so that it can maps the inputs to linearly separable data points.

(e) We need to find θ such that it minimizes the MSE loss:

$$\arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \|F_i(\mathbf{x}_i) - \mathbf{y}_i\|^2 .$$