Homework #1

Pranav Jangir
Deep Learning Spring' 23
NYU

12^{th} February 2023

Question 1.2

(a):
1. Forward Propagation
2. Loss computation
3. Clear Gradients (Reset to zero)
4. Backward propagation (Computing loss gradients for every layer wrt the weights)
5. Update weights

(b):
$Linear_1$ input $= x$ $\quad$ $Linear_1$ output $= W^{(1)}x + b^{(1)}$
f input $= W^{(1)}x + b^{(1)}$ $\quad$ f output $= 3(W^{(1)}x + b^{(1)})^+$
$Linear_2$ input $= 3(W^{(1)}x + b^{(1)})^+$ $\quad$ $Linear_2$ output $= 3W^{(2)}(W^{(1)}x + b^{(1)})^+ + b^{(2)}$
g input $= 3W^{(2)}(W^{(1)}x + b^{(1)})^+ + b^{(2)}$ $\quad$ g output $= 3W^{(2)}(W^{(1)}x + b^{(1)})^+ + b^{(2)}$

Loss input $= 3W^{(2)}(W^{(1)}x + b^{(1)})^+ + b^{(2)}$ and y
Loss output $= \parallel 3W^{(2)}(W^{(1)}x + b^{(1)})^+ + b^{(2)} - y \parallel^2$

(c):
We know that $x \in \mathbb{R}^n$ and $\tilde{y} \in \mathbb{R}^K$. If we assume that the first linear layer has a neurons, then $W^{(1)} \in \mathbb{R}^{a \times n}$ and $b^{(1)} \in \mathbb{R}^a$
also $W^{(2)} \in \mathbb{R}^{K \times a}$ and $b^{(2)} \in \mathbb{R}^K$
We have the following gradients that need to be computed :

$$\frac{\partial C}{\partial \tilde{y}} \in \mathbb{R}^{1 \times K}$$
$$\frac{\partial \tilde{y}}{\partial s_2} \in \mathbb{R}^{K \times K}$$
$$\frac{\partial s_2}{\partial a_1} \in \mathbb{R}^{K \times a}$$
$$\frac{\partial a_1}{\partial s_1} \in \mathbb{R}^{a \times a}$$

Also, partial derivatives of the linear outputs z_1 and z_3 wrt to their layer's weights and bias that need to be computed :

$$\begin{aligned}
\frac{\partial s_2}{\partial W^{(2)}} &\in \mathbb{R}^{K\times a\times K} \\
\frac{\partial s_2}{\partial b^{(2)}} &\in \mathbb{R}^{K\times K} \\
\frac{\partial s_1}{\partial W^{(1)}} &\in \mathbb{R}^{a\times n\times a} \\
\frac{\partial s_1}{\partial b^{(1)}} &\in \mathbb{R}^{a\times a}
\end{aligned}$$

Calculating the gradients via chain rule :
We are going to use the two terms repeatedly :

$$\begin{aligned}
\frac{\partial C}{\partial s_2} &= \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2} \\
\frac{\partial C}{\partial s_1} &= \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}\frac{\partial s_2}{\partial a_1}\frac{\partial a_1}{\partial s_1} \\
&= \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}
\end{aligned}$$

For biases :

$$\begin{aligned}
\frac{\partial C}{\partial b^{(2)}} &= \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial b^{(2)}} \\
&= \boxed{\frac{\partial C}{\partial s_2}} \\
\frac{\partial C}{\partial b^{(1)}} &= \frac{\partial C}{\partial s_1}\frac{\partial s_1}{\partial b^{(1)}} \\
&= \frac{\partial C}{\partial s_1} \\
&= \boxed{\frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}W^{(2)}\frac{\partial a_1}{\partial s_1}}
\end{aligned}$$

For weights we have :

$$\begin{aligned}
\frac{\partial C}{\partial W^{(2)}} &= \frac{\partial C}{\partial s_2}\frac{\partial s_2}{\partial W^{(2)}} \\
&= \sum_i \frac{\partial C}{\partial (s_2)_i} \times \frac{\partial (s_2)_i}{\partial W^{(2)}} \\
&= a_1 \frac{\partial C}{\partial s_2} \\
&= \boxed{3(W^{(1)}x + b^{(1)})^+ \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2}} \\
\frac{\partial C}{\partial W^{(1)}} &= \frac{\partial C}{\partial s_1}\frac{\partial s_1}{\partial W^{(1)}} \\
&= \sum_i \frac{\partial C}{\partial (s_1)_i} \times \frac{\partial (s_1)_i}{\partial W^{(1)}} \\
&= x \frac{\partial C}{\partial s_1} \\
&= \boxed{x \frac{\partial C}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}}
\end{aligned}$$

(d):

$\frac{\partial a_1}{\partial s_1}$ as noted before is a $a \times a$ matrix with non diagonal entries equal to zero.

$\boxed{(\frac{\partial a_1}{\partial s_1})_{ii} = 3\mathbb{1}_{(s_1)_i > 0}}$

for $i = 1 \cdots a$, where $\mathbb{1}$ is the indicator random variable.

Likewise, $\frac{\partial \tilde{y}}{\partial s_2}$ is a $K \times K$ matrix with non diagonal entries equal to zero.

$\boxed{(\frac{\partial \tilde{y}}{\partial s_2})_{ii} = 1}$

$\frac{\partial C}{\partial \tilde{y}}$ is a row vector of dimensions $1 \times K$

$\boxed{(\frac{\partial C}{\partial \tilde{y}})_i = 2(\tilde{y}_i - y_i)}$

Question 1.3

(a):

$Linear_1$ input $= x$ $\qquad$ $Linear_1$ output $= W^{(1)}x + b^{(1)}$
f input $= W^{(1)}x + b^{(1)}$ $\qquad$ f output $= \tanh\big(W^{(1)}x + b^{(1)}\big)$
$Linear_2$ input $= \tanh\big(W^{(1)}x + b^{(1)}\big)$
$Linear_2$ output $= W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}$
g input $= W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}$
g output $= \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})$

Loss input $= \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})$ and y
Loss output $= \parallel \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}) - y \parallel^2$

Gradients can be computed just as they were computed for 1.2 using chain rule. The new gradients are :

For biases :

$$\frac{\partial C}{\partial b^{(2)}} = \boxed{\frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}}$$
$$\frac{\partial C}{\partial b^{(1)}} = \boxed{\frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}}$$

For weights :

$$\frac{\partial C}{\partial W^{(2)}} = \boxed{\tanh\big(W^{(1)}x + b^{(1)}\big) \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}}$$
$$\frac{\partial C}{\partial W^{(1)}} = \boxed{x \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}}$$

The values of the gradients are :
$\frac{\partial a_1}{\partial s_1}$ is a $a \times a$ matrix with non diagonal entries equal to zero.
$\boxed{(\frac{\partial a_1}{\partial s_1})_{ii} = (1 - \tanh\left(s_1\right)_i^2)}$
for $i = 1 \cdots a$

Likewise, $\frac{\partial \tilde{y}}{\partial s_2}$ is a $K \times K$ matrix with non diagonal entries equal to zero.

$$\boxed{(\frac{\partial \tilde{y}}{\partial s_2})_{ii} = \sigma((s_2)_i)(1 - \sigma((s_2)_i))}$$

$\frac{\partial C}{\partial \tilde{y}}$ is a row vector of dimensions $1 \times K$

$$\boxed{(\frac{\partial C}{\partial \tilde{y}})_i = 2(\tilde{y}_i - y_i)}$$

(b):
Only the loss function has changed, therefore :

$Linear_1$ input $= x$ $\qquad$ $Linear_1$ output $= W^{(1)}x + b^{(1)}$
f input $= W^{(1)}x + b^{(1)}$ $\qquad$ f output $= \tanh\big(W^{(1)}x + b^{(1)}\big)$
$Linear_2$ input $= \tanh\big(W^{(1)}x + b^{(1)}\big)$
$Linear_2$ output $= W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}$
g input $= W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}$
g output $= \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})$

Loss input $= \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})$ and y
Loss output $=$

$$-\frac{1}{K}[y^T \log\big(\sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})\big) + (1-y)^T \log\big(1 - \sigma(W^{(2)} \tanh\left(W^{(1)}x + b^{(1)}\right) + b^{(2)})\big)]$$

The gradients are the same as the gradients for 1.3 part (a) :

For biases :

$$\frac{\partial C}{\partial b^{(2)}} = \boxed{\frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}}$$
$$\frac{\partial C}{\partial b^{(1)}} = \boxed{\frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}}$$

For weights :

$$\frac{\partial C}{\partial W^{(2)}} = \boxed{\tanh\big(W^{(1)}x + b^{(1)}\big) \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2}}$$
$$\frac{\partial C}{\partial W^{(1)}} = \boxed{x \frac{\partial C}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial s_2} W^{(2)} \frac{\partial a_1}{\partial s_1}}$$

The values of the gradients are :
$\frac{\partial a_1}{\partial s_1}$ is a $a \times a$ matrix with non diagonal entries equal to zero.

$$\boxed{(\frac{\partial a_1}{\partial s_1})_{ii} = (1 - \tanh{(s_1)_i^2})}$$

for $i = 1 \cdots a$

Likewise, $\frac{\partial \tilde{y}}{\partial s_2}$ is a $K \times K$ matrix with non diagonal entries equal to zero.

$$\boxed{(\frac{\partial \tilde{y}}{\partial s_2})_{ii} = \sigma((s_2)_i)(1 - \sigma((s_2)_i))}$$

$\frac{\partial C}{\partial \tilde{y}}$ is a row vector of dimensions $1 \times K$

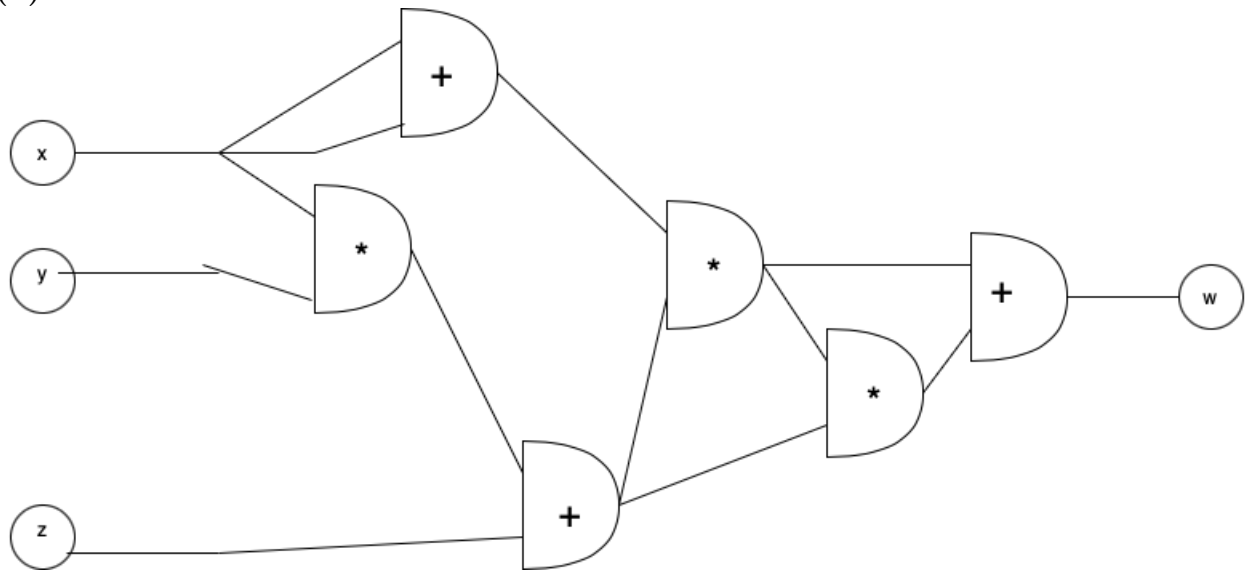
$$\boxed{(\frac{\partial C}{\partial \tilde{y}})_i = \frac{1}{K}\frac{y_i - \tilde{y}_i}{\tilde{y}_i - \tilde{y}_i^2}}$$

(c):
$f = ReLU$ is a much better choice than $f = \tanh$ and that is because tanh suffers from the problem of vanishing gradients. The range of tanh is [-1, 1] and therefore, its derivative $1 - \tanh^2(x)$ has values between [0, 1]. During the training phase, the gradients may reach close to zero and thus slow down the weight change during backward pass, therefore slowing down the learning. $ReLU$ does not have this problem as its derivative is either 0 or 1 and as a result learning is faster.

Question 1.4

(a):
Softmax is generally confused with Softargmax. What is colloquially called Softmax should be called Softargmax as it is a clearer name.
Softargmax gives a soft version of argmax. "Soft" here means diffrentiable everywhere. "Hard" functions like argmax are not diffrentiable everywhere. This makes it possible to give smooth gradients for the softargmax, which would be impossible with argmax.

(b):



(c):

(d):
Part (a) :

$$\frac{\partial f}{\partial x} = J_f = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{x}_1} & \frac{\partial f_1}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{x}_a} \\ \frac{\partial f_2}{\partial \boldsymbol{x}_1} & \ddots & \cdots & \frac{\partial f_2}{\partial \boldsymbol{x}_a} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_b}{\partial \boldsymbol{x}_1} & \frac{\partial f_b}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial f_b}{\partial \boldsymbol{x}_a} \end{bmatrix} = \boldsymbol{W}_1$$

$$\frac{\partial g}{\partial x} = J_g = \begin{bmatrix} \frac{\partial g_1}{\partial \boldsymbol{x}_1} & \frac{\partial g_1}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial g_1}{\partial \boldsymbol{x}_a} \\ \frac{\partial g_2}{\partial \boldsymbol{x}_1} & \ddots & \cdots & \frac{\partial g_2}{\partial \boldsymbol{x}_a} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_b}{\partial \boldsymbol{x}_1} & \frac{\partial g_b}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial g_b}{\partial \boldsymbol{x}_a} \end{bmatrix} = \boldsymbol{W}_2$$

Part (b):

$$J_h = \frac{\partial h}{\partial \boldsymbol{x}} = \frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{x}} = J_h = J_f + J_g = \boldsymbol{W_1} + \boldsymbol{W_2}$$

Part (c):
If $\boldsymbol{W_1} = \boldsymbol{W_2}$

$$J_h = \frac{\partial h}{\partial \boldsymbol{x}} = \frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{x}} = J_h = J_f + J_g = \boldsymbol{W_1} + \boldsymbol{W_2} = 2\boldsymbol{W_1}$$

Part (e):
(a)

The Jacobian of $f(\boldsymbol{x})$ is:

$$J_f = \frac{\partial f}{\partial \boldsymbol{x}} = J_f = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{x}_1} & \frac{\partial f_1}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{x}_a} \\ \frac{\partial f_2}{\partial \boldsymbol{x}_1} & \ddots & \cdots & \frac{\partial f_2}{\partial \boldsymbol{x}_a} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_b}{\partial \boldsymbol{x}_1} & \frac{\partial f_b}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial f_b}{\partial \boldsymbol{x}_a} \end{bmatrix} = \boldsymbol{W}_1$$

The Jacobian of $g(\boldsymbol{x})$ is:

$$J_g = \frac{\partial g}{\partial \boldsymbol{x}} = J_g = \begin{bmatrix} \frac{\partial g_1}{\partial \boldsymbol{x}_1} & \frac{\partial g_1}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial g_1}{\partial \boldsymbol{x}_b} \\ \frac{\partial g_2}{\partial \boldsymbol{x}_1} & \ddots & \cdots & \frac{\partial g_2}{\partial \boldsymbol{x}_b} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_c}{\partial \boldsymbol{x}_1} & \frac{\partial g_c}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial g_c}{\partial \boldsymbol{x}_b} \end{bmatrix} = \boldsymbol{W}_2$$

(b)

$$h(\boldsymbol{x}) = g(f(\boldsymbol{x})) = g(\boldsymbol{W_1 x}) = \boldsymbol{W_2 W_1 x}$$

Hence, $J_h =$

$$\frac{\partial h}{\partial \boldsymbol{x}} = \boldsymbol{W_2 W_1}$$

(c)

If $\boldsymbol{W_1} = \boldsymbol{W_2}$,

$$J_h = \frac{\partial h}{\partial \boldsymbol{x}} = \boldsymbol{W_2 W_1} = \boldsymbol{W_1^T W_1}$$