AI501 Homework 4

20200130 Yujun Kim

May 2023

1 Exercise #6.1

Consider

$$\frac{\sigma(x)}{x} = \frac{\partial\sigma(x)}{\partial(x)}$$

Let $y = \sigma(x)$, $\frac{\partial\sigma(x)}{\partial(x)} = y'$.

$$\begin{aligned} \frac{y}{x} = y' \Rightarrow \frac{1}{x} = \frac{y'}{y} \\ \Rightarrow ln(y) = c + ln(x) \\ y = e^c x = kx \text{ for some } k \end{aligned}$$

As ODE is defined for all points except at $x = 0$, given $y(1) = a, y(-1) = b$, y is uniquely determined on $(0, \infty)$ as $y(x) = ax$ and on $(-\infty, 0)$ as $y(x) = -bx$.

y naturally expand at $x = 0$ as $y(0) = 0$. Hence,

$$y(x) = \begin{cases} ax & \text{if } x \geq 0 \\ -bx & \text{if } x < 0 \end{cases}$$

Note that ReLU in particular is a solution of the ODE.

2 Exercise #6.2

Use induction on t to show $\Theta[t]$ obtained by (6.20) and $\tilde{\Theta}[t]$ obtained by (6.21) are same if started with the same initial value $\Theta[1] = \tilde{\Theta}[1]$.

By the initial condition, case $t = 1$ holds. Now, suppose $\Theta[i] = \tilde{\Theta}[i]$ for $i = 1, \cdots, t$. Then,

$$
\begin{aligned}
\tilde{\Theta}[t+1] &= \tilde{\Theta}[t] + V[t] && \text{by (6.21)} \\
&= \tilde{\Theta}[t] + \beta(\tilde{\Theta}[t] - \tilde{\Theta}[t1]) - \eta \frac{\partial c}{\partial \tilde{\Theta}}(\tilde{\Theta}[t]) && \text{by (6.21)} \\
&= \Theta[t] + \beta(\Theta[t] - \Theta[t-1]) - \eta \frac{\partial c}{\partial \Theta}(\Theta[t]) && \text{by induction hypothesis} \\
&= \Theta[t] - \beta(\sum_{s=1}^{t-1} \beta^{t-1-s} \frac{\partial c}{\partial \Theta}(\Theta[s])) - \eta \frac{\partial C}{\partial \Theta}(\Theta[t]) && \text{by (6.20)} \\
&= \Theta[t] - \sum_{s=1}^{t} \beta^{t-s} \frac{\partial c}{\partial \Theta}(\Theta[s]) && \text{by (6.20)} \\
&= \Theta[t+1]
\end{aligned}
$$

Hence, by induction, $\Theta[t] = \tilde{\Theta}[t]$ for all t.

3 Exercise #6.3

(a)

$$\frac{\partial l(y, o^L)}{\partial b^l} = \frac{\partial g^l}{\partial b^l} \frac{\partial l(y, o^L)}{\partial g^l} = I_{d(l)} \delta^l = \delta^l$$

Hence, with step size γ, we have update

$$\Delta b^l = -\gamma \delta^l$$

(b) As W^l are diagonal, forward and backward propagation happens as figure 1.

4 Exercise #6.4

(Iteration 1) To ease writing, I omit parenthesis for superscript. e.g. W^l instead of $W^{(l)}$. As $\eta = 1$,

$$\Delta W^1 = -\delta^1 o^{0T}, \Delta W^2 = -\delta^2 o^{1T}$$

and by Exercise #6.3,

$$\Delta b^1 = -\delta^1, \Delta b^2 = -\delta^2$$

Now for forward propagation,

$$
\begin{gathered}
o^0 = x, g^1 = W^1 o^0 + b^1 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, o^1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \\
g^2 = W^2 o^1 + b^2 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}, o^2 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}
\end{gathered}
$$

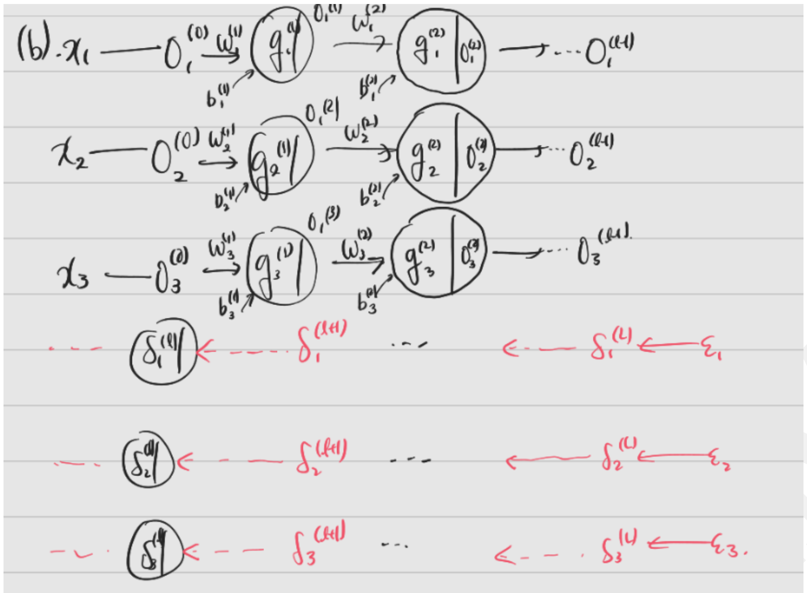

(b).
x1 — O1(0) w1(1) g1(1) O1(1) w1(2) g1(2) O1(2) → ··· O1(l-1)
b1(1)
b1(2)
x2 — O2(0) w2(1) g2(1) O1(2) w2(2) g2(2) O2(2) → ··· O2(l-1)
b2(1)
b2(2)
x3 — O3(0) w3(1) g3(1) O1(3) w3(2) g3(2) O3(2) → ··· O3(l-1)
b3(1)
b3(2)
δ1(1) ← ··· δ1(l-1) ··· ← δ1(l) ← ε1
δ2(1) ← ··· δ2(l-1) ··· ← δ2(l) ← ε2
δ3(1) ← ··· δ3(l-1) ··· ← δ3(l) ← ε3.

Figure 1: Forward and Backward propagation

For back propagation,

$$\epsilon = o^2 - y = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$\delta^2 = \Lambda^2 \epsilon = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$\delta^1 = \Lambda^1 W^{2T} \delta^2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \delta^2 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Hence,

$$\Delta W^1 = -\delta^1 o^{0T} = \begin{bmatrix} -3 & 3 \\ 0 & 3 \end{bmatrix}, \Delta W^2 = -\delta^2 o^{1T}, = \begin{bmatrix} -9 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Delta b^1 = -\delta^1 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \Delta b^2 = -\delta^2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$$

Resulting

$$W^1 = \begin{bmatrix} -2 & 2 \\ 0 & 1 \end{bmatrix}, W^2 = \begin{bmatrix} -8 & -1 \\ 0 & 1 \end{bmatrix}, b^1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, b^2 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

(Iteration 2) For forward propagation,

$$o^0 = x, g^1 = W^1 o^0 + b^1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, o^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$g^2 = W^2 o^1 + b^2 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, o^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For back propagation,

$$\epsilon = o^2 - y = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$
$$\delta^2 = \Lambda^2 \epsilon = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\delta^1 = \Lambda^1 W^{2T} \delta^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and thus,

$$\Delta W^1 = \Delta W^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \Delta b^1 = \Delta b^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Resulting

$$W^1, W^2, b^1, b^2$$

unchanged from iteration 1. Note that although the loss l is not 0(the global minima), the gradient with respect to trainable variables are 0 and thus we reached a critical point.

5 Exercise #6.5

(a) For a matrix $A \in \mathcal{M}_{m\times n}$,

$$||A||_F^2 = \sum_{j=1}^{n} ||a_j||^2 \text{ , where } A = [a_i \cdots a_n]$$

Thus,

$$|| - \Delta^{(l)} - WO^{(l-1)}||_F^2 = \sum_{n=1}^{N} || - \delta_n^{(l)} - W o_n^{(l-1)}||^2$$

as n-th column of $-\Delta^{(l)} - WO^{(l-1)}$ is $-\delta_n^{(l)} - W o_n^{(l-1)}$. (6.64) follows from above equality.

(b) By (6.37) of the textbook,

$$\frac{\partial c}{\partial W^{(l)}} = -\sum_{n=1}^{N} \delta^{(l)} o_n^{(l-1)T}$$

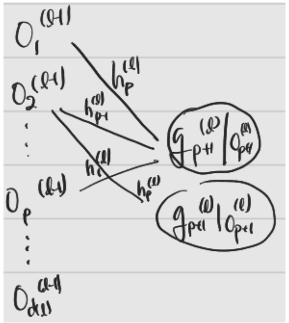


Figure 2: Convolutional Model

By Taylor theorem and assumption that $\frac{\partial c}{\partial W^{(l)}} \neq 0$, there is γ small enough so that changing $W^{(l)}$ by

$$\Delta W^{(l)} = -\gamma \sum_{n-1}^{N} \delta_n^{(l)} o_n^{(l-1)T}$$

reduce the loss function.

6 Exercise #6.7

In convolutional neural neural network model, we have relation

$$o_i^{(l)} = \sigma\left(\sum_{j=1}^{d(l)} h_{i-j}^{(l)} o_j^{(l-1)} + b_i^{(l)}\right)$$

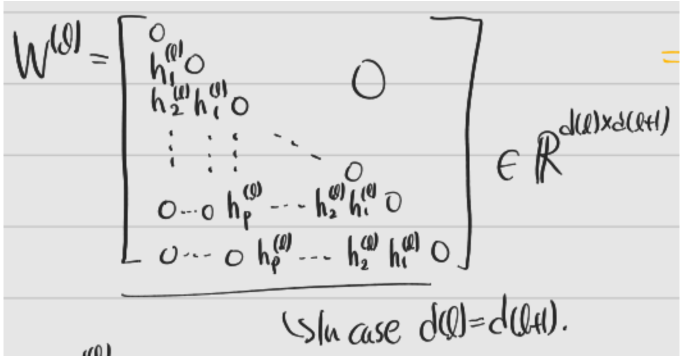
which is illustrated by figure 2. Considering matrix $W^{(l)}$ to have $o^{(l)} = \sigma(W^{(l)} o^{(l-1)} + b^{(l)}$, we have $W^{(l)}$ as in figure 2. Figure 3 typically shows the case $d(l) = d(l+1)$. If $d(l) < d(l+1)$, then we can cut some bottom rows or cut some rightmost colums if $d(l) > d(l+1)$.

(b) To derive back propagation algorithm, first note the forward propagating parts:

$$g^l = \sum_{j=1}^{d(l)} h_{i-j}^l o_j^{l=1} + b_i^l, o^l = \sigma(g^l)$$

Now,

$$\frac{\partial g^l}{\partial h^l} = \frac{\partial Vec(W^l)}{\partial h^l} \frac{\partial g^l}{\partial Vec(W^l)}$$



$$W^{(l)} = \begin{bmatrix} 0 & & & & & & \\ h_1^{(l)} & 0 & & & & O & \\ h_2^{(l)} & h_1^{(l)} & 0 & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ & & & & & 0 & \\ 0 \cdots 0 & h_p^{(l)} & \cdots & h_2^{(l)} & h_1^{(l)} & 0 & \\ 0 \cdots 0 & h_p^{(l)} & \cdots & h_2^{(l)} & h_1^{(l)} & 0 & \end{bmatrix} \in \mathbb{R}^{d(l)\times d(l+1)}$$

↳ In case $d(l)=d(l+1)$.

Figure 3: $W^{(l)}$

From textbook, $\frac{\partial g^l}{\partial Vec(W^l)} = o^{l-1} \bigotimes I_{d(l)}$. Also from (a), we have

$$\frac{\partial Vec(W^l)}{\partial h^l} = E := \begin{bmatrix} e_2^T & e_3^T & \cdots & e_{d(l)-1}^T & e_{d(l)}^T & 0 \\ e_3^T & e_4^T & \cdots & e_{d(l)}^T & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \\ e_{p+1}^T & e_{p+1}^T & \cdots & 0 & 0 & \end{bmatrix}$$

and so

$$\frac{\partial g^l}{\partial h^l} = E(o^{l-1} \otimes I_{d(l)})$$

Now,

$$\frac{\partial l(y, o^L)}{\partial h^l} = \frac{\partial g^l}{\partial h^l}\frac{\partial l}{\partial g^l} = E(o^{l-1} \otimes I_{d(l)})\delta^l = E(Vec(\delta^l o^{(l-1)T}))$$

Hence,

$$\Delta h^l = E(Vec(\delta^l o^{(l-1)T}))$$

7 Exercise #7.1

(a) VGG-Net in figure 7.2 consists of five blocks with each block connected by a pooling.

- Block 1 has two convolutional layer of 64 filters each
- Block 2 has three convolutional layer of 128 filters each.
- Block 3 has one convolutional layer of 64 filters and three convolutional layer of 256 filters each.
- Block 4 has three convolutional layer of 512 filters each.

- Block 2 has three convolutional layer of 512 filters each.

Hence, the total number of convolutional filters is

$$(64 \times 2) + (128 \times 3) + (64 + 256 \times 3) + (512 \times 3) + (512 \times 3) = 4416$$

(b) First, we count the number of trainable parameters in convolutional filters(assuming there is no bias term). Each filter has width and height as $3 = 9$. We count sum of depth $\times$ number of filters in each block and multiply them by 9.

- Block 1: $n_1 = 3 \times 64 + 64 \times 64$
- Block 2: $n_2 = 64 \times 128 + 128 \times 128$
- Block 3: $n_3 = 128 \times 64 + 64 \times 256 + 256 \times 256 + 256 \times 256$
- Block 4: $n_4 = 256 \times 512 + 512 \times 512 + 512 \times 512$
- Block 5: $n_5 = 512 \times 512 + 512 \times 512 + 512 \times 512 + 512 \times 512$

Hence, total number of convolutional parameters is $9 \times (n_1 + \cdots + n_5) = 9 \times 1,888,394 = 16,995,546$.

Next, we count the number of trainable parameters in fully connected parts. To do so, we have to know the dimension of output from block 5. As input width and heights are 224 and the data is passed through 5 pooling layers, the output of block 5 has width and height $224/2^5 = 7$.

1. Input dimension of FC1 is $7 \times 7 \times 512$ and output dimension of FC1 is 4096 having 102,760,448 parameters.
2. Input dimension of FC2 is 4096 and output dimension of FC1 is 4096 having 16,777,216 parameters.
3. Input dimension of FC3 is 4096 and output dimension of FC3 is number of image classes(typically 1000 for ImageNet) having 4,096,000 parameters.

Hence, total number of trainable parameters is **140,629,210**.

8 Exercise #7.3

Each convolutional layer increase width of effective receptive field(ERF) by +2. Pooling increase width of ERF by ×2. Unpooing decrease width of ERF by /2.

U-Net is composed of 9 blocks as in figure 4. Each block is connected by either pooling or unpooling. Consider figure 5, which calculates width of ERF starting from block9. The calculation shows that ERF of input twards output is 280×280

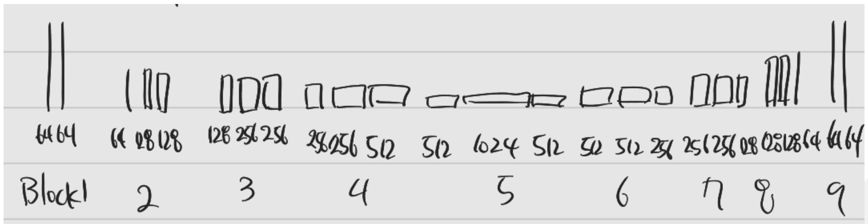
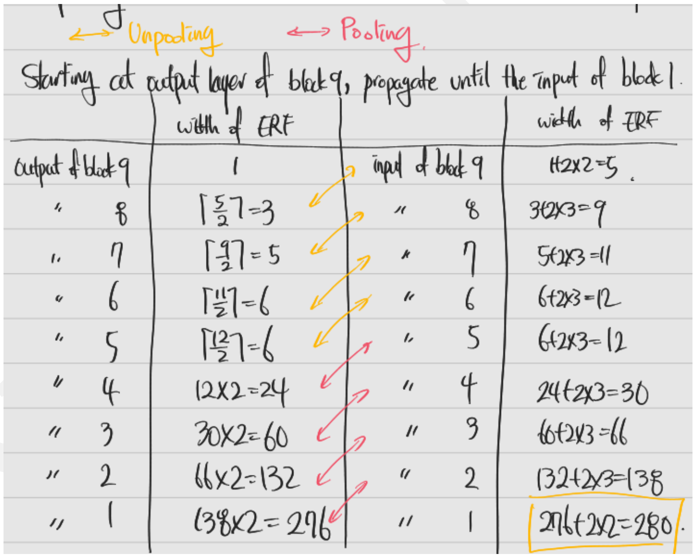


Figure 4: U-Net



↔ Unpooling ↔ Pooling.

Starting at output layer of block 9, propagate until the input of block 1.

	width of ERF		width of ERF
output of block 9	1	input of block 9	$1+2\times2=5$.
〃 8	$\lceil \frac{5}{2} \rceil = 3$	〃 8	$3+2\times3=9$
〃 7	$\lceil \frac{9}{2} \rceil = 5$	〃 7	$5+2\times3=11$
〃 6	$\lceil \frac{11}{2} \rceil = 6$	〃 6	$6+2\times3=12$
〃 5	$\lceil \frac{12}{2} \rceil = 6$	〃 5	$6+2\times3=12$
〃 4	$12\times2=24$	〃 4	$24+2\times3=30$
〃 3	$30\times2=60$	〃 3	$60+2\times3=66$
〃 2	$66\times2=132$	〃 2	$132+2\times3=138$
〃 1	$138\times2=276$	〃 1	$\boxed{276+2\times2=280}$.

Figure 5: Calculating ERF size