MFDNN

Homework 5

24/04/04

1. (Working) (denoting loss function with $\mathcal{L}$)

$$\text{'}dy\text{'} := \frac{\partial \mathcal{L}}{\partial y_L} = \frac{\partial}{\partial y_L}\left(\frac{1}{2}(f_\theta(x) - y)^2\right) = \frac{\partial}{\partial y_L}\left(\frac{1}{2}(y_L - y)^2\right) = y_L - y \quad \text{(as given)}$$

$$z_\ell := A_\ell y_{\ell-1} + b_\ell$$

For $\ell = L$ ('ell' $= L-1$),
$$\frac{\partial \mathcal{L}}{\partial b_L} = \frac{\partial \mathcal{L}}{\partial y_L}\frac{\partial y_L}{\partial b_L} = dy \cdot 1 = dy \cdot \overset{\sigma'}{S(z)}$$

For $\ell < L$ ('ell' $< L-1$),
$$\frac{\partial \mathcal{L}}{\partial b_\ell} = \underbrace{\frac{\partial \mathcal{L}}{\partial y_L}\frac{\partial y_L}{\partial y_\ell}}_{=:\text{'}dy\text{'}}\frac{\partial y_\ell}{\partial b_\ell} = \overbrace{\frac{\partial \mathcal{L}}{\partial y_L}\left(\frac{\partial y_L}{\partial y_{L-1}} \cdots \frac{\partial y_{\ell+1}}{\partial y_\ell}\right)}\frac{\partial y_\ell}{\partial b_\ell} = dy \cdot \operatorname{diag}(S(z))$$

so 'add' to dy term to be reused next time

so $dy' := dy \cdot \operatorname{diag}(\sigma'(z))A_\ell$

i.e. $dy := \frac{\partial \mathcal{L}}{\partial y_{\ell-1}} = \frac{\partial \mathcal{L}}{\partial y_L}\frac{\partial y_L}{\partial y_{\ell-1}}$

$$\frac{\partial \mathcal{L}}{\partial A_\ell} = \frac{\partial \mathcal{L}}{\partial y_L}\frac{\partial y_L}{\partial A_\ell}$$

* $$= \frac{\partial \mathcal{L}}{\partial y_L}\operatorname{diag}(S(z))\left(\frac{\partial y_L}{\partial y_\ell}\right)^T (y_{\ell-1})^T$$

$$\underbrace{A_1, A_2, \dots, A_\ell}_{}, \overbrace{A_{\ell+1}, \dots, A_L}^{A_j \text{ small}}$$

2. $$\frac{\partial y_L}{\partial b_i} = \frac{\partial y_L}{\partial y_{L-1}} \frac{\partial y_{L-1}}{\partial y_{L-2}} \cdots \frac{\partial y_{i+1}}{\partial y_i} \frac{\partial y_i}{\partial b_i} \qquad \text{by chain rule}$$

$$= f(A_i, \dots, A_L) \operatorname{diag}(\sigma'(A_i y_{i-1} + b_i)) \qquad \text{by HW 4.6}$$

where f is a function involving the matrix multiplication of its args $A_j \in \{A_i, \dots, A_L\}$, i.e. A_j is a term in f

← inc. $\operatorname{diag}(\sigma'(A_\ell y_{\ell-1} + b_\ell))$

If A_j is small, then f will be a matrix multiplication of not too large matrices and a small matrix, A_j, so the result will be small and hence $\frac{\partial y_L}{\partial b_i}$ become small.

$$\frac{\partial y_L}{\partial A_i} = \operatorname{diag}(\sigma'(A_i y_{i-1} + b_i)) \underbrace{\left(\frac{\partial y_L}{\partial y_{L-1}} \frac{\partial y_{L-1}}{\partial y_{L-2}} \cdots \frac{\partial y_{i+1}}{\partial y_i} \right)^T}_{f(A_i, \dots, A_L)} (y_{i-1})^T$$

Hence, by a similar argument as above, $\frac{\partial y_L}{\partial A_i}$ become small (f will be small)

If $|\tilde{y}_j|$ is large then $\sigma'(\tilde{y}_j)$ is small (tending to 0) so now the $\operatorname{diag}(\sigma'(\tilde{y}_i))$ term will be the small one in a matrix multiplication of otherwise not too large matrices and so $\frac{\partial y_L}{\partial b_i}, \frac{\partial y_L}{\partial A_i}$ become small.

3. For $k=0$,
$$\begin{aligned}
\theta_I^1 &= \theta^0 - \alpha g^0 + \beta(\theta^0 - \theta^{-1}) \\
&= \theta^0 - \alpha g^0 \qquad \text{by Form I} \\
\theta_{II}^1 &= \theta^0 - \alpha v^1 \\
&= \theta^0 - \alpha(g^0 + \beta v^0) \\
&= \theta^0 - \alpha g^0 \qquad \text{by Form II} \qquad \text{So forms equivalent for } k=0.
\end{aligned}$$

For $k=1$,
$$\begin{aligned}
\theta_I^2 &= \theta^1 - \alpha g^1 + \beta(\theta^1 - \theta^0) = \theta^1 - \alpha g^1 + \beta(-\alpha v^1) \qquad \text{since } \theta^1 \text{ and } \theta^0 \text{ equivalent} \\
&= \theta^1 - \alpha g^1 - \alpha\beta v^1 \\
\theta_{II}^2 &= \theta^1 - \alpha v^2 \\
&= \theta^1 - \alpha(g^1 + \beta v^1) = \theta^1 - \alpha g^1 - \alpha\beta v^1 \qquad \text{So equivalent for } k=1.
\end{aligned}$$

Assume true for $k = n, n-1$ (i.e. $\theta_I^n = \theta_{II}^n = \theta^n$ and $\theta_I^{n-1} = \theta_{II}^{n-1} = \theta^{n-1}$)

← no need to distinguish since assume equal

Consider θ^{n+1} term. By form I, $\theta_I^{n+1} = \theta^n - \alpha g^n + \beta(\theta^n - \theta^{n-1})$

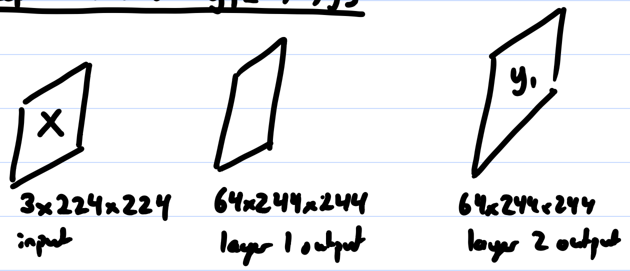
By form II, $\theta_{II}^{n+1} = \theta^n - \alpha(g^n + \beta v^n)$

By assumption,
$$\begin{aligned}
\theta_I^{n+1} &= (\theta_{II}^{n+1} + \alpha(g^n + \beta v^n)) - \alpha g^n + \beta(\theta^n - \theta^{n-1}) \\
&= \theta_{II}^{n+1} + \alpha\beta v^n + \beta(\theta^n - \theta^{n-1}) \\
&= \theta_{II}^{n+1} + \alpha\beta\left(\frac{\theta^n - \theta^{n-1}}{-\alpha}\right) + \beta(\theta^n - \theta^{n-1}) \qquad \text{since } \theta^k = \theta^{k-1} - \alpha v^k \text{ for } k \geq 1 \\
&= \theta_{II}^{n+1}
\end{aligned}$$
so forms are equivalent for $k = n+1$. Since true for $k = 0, 1$, true for all k by induction.

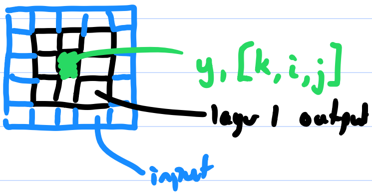
Reference consulted → Araujo, et al., "Computing Receptive Fields of Convolutional Neural Networks", Distill, 2019.

4. Receptive field of $y_1[k,i,j]$

NB: using 0-indexing for i, j, m, n



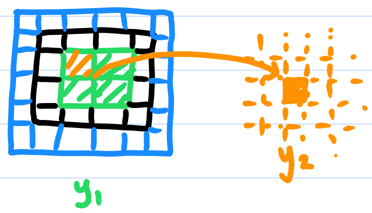
After first layer is applied, each new 'pixel' depends on the 8 surrounding pixels of input in all 3 channels. Applying another convolution each new 'pixel' depends again on the 8 surrounding it in all 64 channels.



So $y_1[k,i,j]$ depends on $X[c,m,n]$
for $1 \le c \le 3,\ i-2 \le m \le i+2,\ j-2 \le n \le j+2$
(just treat 'negative bounds' as 0 bounds)

Receptive field of y_2

Applying a max pool layer changes the diagram slightly since each y_2 pixel depends on 4 y_1 pixels



So $y_2[k,i,j]$ depends on $X[c,m,n]$
for $1 \le c \le 3,\ 2i-2 \le m \le 2i+3,\ 2j-2 \le n \le 2j+3$

Receptive field of y_3 – using recurrence relation equations

		k_i	s_i	p_i
input, y_0 →	$i=1$	3	1	1
	$i=2$	3	1	1
y_1 → max	$i=3$	2	2	0
y_2 →	$i=4$	3	1	1
	$i=5$	3	1	1
max	$i=6$	2	2	0
y_3 →				

Layer definitions ↑

$$r_0 = \sum_{l=1}^{L}\left((k_l-1)\prod_{i=1}^{l-1} s_i\right)+1$$ (equation 2 from reference)

(L=6)

$$=(2)(1)+(2)(1)+(1)(1\times1)+(2)(1\times1\times2)+(2)(1\times1\times2\times1)+(1)(1\times1\times2\times1\times1)+1$$

$$=2+2+1+4+4+2+1=16$$ ⇒ receptive field size of y_3 is 16×16

(also agrees with results for y_1 and y_2)

RF left-side index, $$u_0 = u_L\prod_{n=1}^{L} s_n - \sum_{l=1}^{L} p_l \prod_{n=1}^{l-1} s_n$$ (equation 5)

(L=6) ← or j, symmetric

$$=(i)(1\times1\times2\times1\times1\times2)-\left((1+1(1)+0+1(1\times1\times2)+1(1\times1\times2\times1)+0\right)$$

$$=4i-6$$

RF right-side index, $$v_0 = v_L\prod_{n=1}^{L} s_n - \sum_{l=1}^{L}(1+p_l-k_l)\prod_{n=1}^{l-1} s_n$$ (equation 6)

(L=6)

$$=(i)(1\times1\times2\times1\times1\times2)-\left(-1+(-1)(1)+(-1)(1\times1)+(-1)(1\times1\times2)+(-1)(1\times1\times2\times1)+(-1)(1\times1\times2\times1\times1)\right)$$

$$=4i-(-9)=4i+9$$

So $y_3[k,i,j]$ depends on $X[c,m,n]$ for $1 \le c \le 3,\ 4i-6 \le m \le 4i+9,\ 4j-6 \le n \le 4j+9$

5.

Naïve Inception | **With bottlenecks**

(i) Num trainable parameters

slide 12 $\left[k^2 C_{in} C_{out} + C_{out}\right]$

Naïve Inception:

$$(1^2 \times 256 \times 128 + 128) + (3^2 \times 256 \times 192 + 192) + (3^2 \times 256 \times 96 + 96)$$
$$= 696{,}736$$

With bottlenecks:

$$(1^2 \times 256 \times 128 + 128) + (1^2 \times 256 \times 64 + 64) + (3^2 \times 64 \times 192 + 192) + (1^2 \times 256 \times 64 + 64) + (5^2 \times 64 \times 96 + 96) + (1^2 \times 256 \times 64 + 64)$$
$$= 346{,}720$$

(ii) Each output element requires summing k^2 elements (k is filter size), so $k^2 - 1$ additions, over C_{in} layers so $C_{in}(k^2-1) + (C_{in}-1)$ additions. This is performed for each output layer and a bias is added per layer so: $C_{out}\left(C_{in}(k^2-1) + (C_{in}-1) + 1\right) = C_{out} C_{in} k^2 = C_{in} \times C_{out} \times k^2$. This is for each window location. So repeat for each 'pixel' of output i.e. $m \times n$ (since all padded to maintain dimension)

slide 7 $\left[Y_{\ell,i,j} = \sum_{\gamma}^{C_{in}} \sum_{\alpha}^{k} \sum_{\beta}^{k} (\dots) + b_\ell\right]$

$\ell = 1, \dots, C_{out}$

Additions:

Naïve Inception:

$$\left((256 \times 128 \times 1^2) + (256 \times 192 \times 3^2) + (256 \times 96 \times 5^2)\right) \times 32^2$$
$$= 1{,}115{,}684{,}864$$

With bottlenecks:

$$\left((256 \times 128 \times 1^2) + (256 \times 64 \times 1^2) + (64 \times 192 \times 3^2) + (256 \times 64 \times 1^2) + (64 \times 96 \times 5^2) + (256 \times 64 \times 1^2)\right) \times 32^2$$
$$= 354{,}418{,}688$$

Multiplications are simpler to count since it's merely counting $(\dots)$ term $\Rightarrow C_{out} C_{in} k^2$ (coincidentally equal!)

So same num of multiplications as additions.

The activation function is evaluated on each element of a layer, so on $C_{out} \times m \times n$ elements.

Naïve Inception:

$$(128 \times 32^2) + (192 \times 32^2) + (96 \times 32^2)$$
$$= 425{,}984$$

With bottlenecks:

$$(128 \times 32^2) + (64 \times 32^2) + (192 \times 32^2) + (64 \times 32^2) + (96 \times 32^2) + (64 \times 32^2)$$
$$= 622{,}592$$

$\sum_{x}^{3} x = 1 + 2 + 3$ — 2 addition

$\sum_{k}^{3} \sum_{x}^{3} \sum_{y}^{3} xy = (1 + 2 + 3) + (2 + 4 + 6) + (3 + 4 + 9) + \dots + \dots$ — 8 additions (4 per group), 8 additions, 8 additions

k sets, sum terms, 'join' additions between sets

$k(k-1) + (k-1)$

$= (k-1)(k+1) = (k^2 - 1)$