IFT6135 – Homework #2

Convolutional neural networks and regularization - Theory

Jean-Philippe Reid
701300

April 15, 2018

1 Convolutions

Compute the full, valid, and same convolution (with kernel flipping) for the following 1D matrices :

$$\big(1, 2, 3, 4\big) \circledast \big(1, 0, 2\big) \tag{1.1}$$

It is important to note that the kernel needs to be rotated in order to compute the convolution.

Valid convolution: padding $= 0$

$$\begin{aligned}[1, 2, 3, 4] \circledast [1, 0, 2] &= [2 \times 1 + 0 \times 2 + 1 \times 3, 2 \times 2 + 0 \times 3 + 4 \times 1] \\ &= [5, 8]\end{aligned} \tag{1.2}$$

Same convolution: padding $= 1$

$$[0, 1, 2, 3, 4, 0] \circledast [1, 0, 2] = [2, 5, 8, 6] \tag{1.3}$$

Full convolution: padding $= 2$

$$[0, 0, 1, 2, 3, 4, 0, 0] \circledast [1, 0, 2] = [1, 2, 5, 8, 6, 8] \tag{1.4}$$

2 Convolutional Neural Networks

Let's assume the input of size $3 \times 256 \times 256$*. The first layer convolves 64* 8×8 *kernels with the input, using a stride of 2 and no padding. The second layer subsamples the output of the first layer with a* 5×5 *non-overlapping max pooling. The third layer convolves 128* 4×4 *kernels with a stride of 1 and a zero-padding of size 1 on each border.*

(a)

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ &= \frac{256 - 8}{2} + 1 = 125 \end{aligned} \tag{2.1}$$

The size of the output of the first layer is $64 \times 125 \times 125$.

The second layer corresponds to a max pooling of 5×5 with non-overlapping. We find that,

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ &= \frac{125 - 5}{5} + 1 = 25 \end{aligned} \tag{2.2}$$

The size of the second layer output is $64 \times 25 \times 25$.

The third and last layer convolves 128 4×4 kernels with a stride of 1 and padding of 1. We find that,

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ &= \frac{25 + 2 - 4}{1} + 1 = 24 \end{aligned} \tag{2.3}$$

The size of the second layer output is $128 \times 24 \times 24 = 73728$ parameters.

(b) There are $128 \times 64 \times 4 \times 4$ weights in the last layer, i.e. 131072 weights.

3 Kernel configuration for CNNs

(a) *For the first convolutional layer, the input shape is* $3 \times 64 \times 64$ *and the output shape is* $64 \times 32 \times 32$. *Provide a configuration of the convolution operation that would match the input/output shapes (i.e. provide the kernel, stride, padding and dilatation between kernel elements).*

$$o = \frac{i + 2p - k}{s} + 1 \tag{3.1}$$

where o, i, p, and s are the output, the input, the padding, the kernel and and the stride parameters.

The dilatation has for main effect to increase the reach of the kernel filters, i.e.

$$\begin{aligned} \tilde{k} &= k + (k-1)(d-1) \\ &= k + kd - k - d + 1 \\ &= d(k-1) + 1 \end{aligned} \tag{3.2}$$

For $d = 1$, $\tilde{k} = k$, obviously. But for $d = 0$, $\tilde{k} = 1$.

(i) *Assuming no dilatation and kernel size of* 8×8.

For the problem in hand, we have that $d = 1$ and $k = 8$.

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ 32 &= \frac{64 + 2p - 8}{s} + 1 \\ 31 &= \frac{56 + 2p}{s} \\ 31s &= 56 + 2p \end{aligned} \tag{3.3}$$

If we set $s = 2$, we find that $p = 3$.

Answer:$k = 8, d = 1, s = 2$ and $p = 3$.

(ii) *Assuming dilatation of 6* $(d = 7)$ *between kernel elements and stride of 2* $(s = 2)$.

We have that $d = 7$ and $s = 2$

$$\begin{aligned} o &= \frac{i + 2p - \tilde{k}}{s} + 1 \\ &= \frac{i + 2p - d\,(k-1) - 1}{s} + 1 \\ 32 &= \frac{64 + 2p - 7\,(k-1) - 1}{2} + 1 \\ 32 &= \frac{64 + 2p - 7k + 7 - 1}{2} + 1 \\ 31 &= \frac{70 + 2p - 7k}{2} \\ 62 &= 70 + 2p - 7k \\ -8 &= 2p - 7k \end{aligned} \tag{3.4}$$

If we set $k = 2$, we find that $p = 3$.

Answer: $k = 2, d = 7, s = 2$ and $p = 3$.

(b) *For the second pooling layer, the input shape is* $64 \times 32 \times 32$ *and the output shape is* $64 \times 8 \times 8$. *Provide a configuration of the pooling operation that would match the input/output shapes, assuming no overlapping of pooling windows and no padding (i.e. what kernel size and stride to use).*

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ 8 &= \frac{32 - k}{s} + 1 \\ 7 &= \frac{32 - k}{s} \end{aligned} \tag{3.5}$$

Answer: $k = 4, s = 4, p = 0$

(c) *Without any padding and using the input from question 2 (*$64 \times 32 \times 32$*), what would have been the output size should the kernel be of size* 8×8, *but with the stride* 4×4 *?*

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ &= \frac{32 - 8}{4} + 1 \\ &= \frac{24}{4} + 1 = 7 \end{aligned} \tag{3.6}$$

Answer: output size $= 7 \times 7$.

(d) *In the last convolutional layer, the input shape is* $64 \times 8 \times 8$ *and the output shape is* $128 \times 4 \times 4$*. Provide a configuration of the convolution operation that would match the input/output shapes (i.e. provide the kernel, stride, padding and dilatation between kernel elements).*

(i) Let's assume that $p = 0$ and $d = 1$.

$$\begin{aligned} o &= \frac{i + 2p - k}{s} + 1 \\ 4 &= \frac{8 - k}{s} + 1 \\ 3 &= \frac{8 - k}{s} \end{aligned} \tag{3.7}$$

If we set $k = 2$, we find that $s = 2$.

Answer: $k = 2, d = 1, s = 2$ and $p = 0$.

(ii) Let's assume that $p = 2$ and $d = 2$.

$$\begin{aligned} o &= \frac{i + 2p - \tilde{k}}{s} + 1 \\ 4 &= \frac{8 + 2p - d\,(k - 1) - 1}{s} + 1 \\ 4 &= \frac{8 + 4 - 2\,(k - 1) - 1}{s} + 1 \\ 4 &= \frac{8 + 4 - 2k + 2 - 1}{s} + 1 \\ 3 &= \frac{13 - 2k}{s} \end{aligned} \tag{3.8}$$

If we set $s = 1$, we find that $k = 5$.

Answer: $k = 5, d = 2, s = 1$ and $p = 2$.

(iii) Let's assume that $p = 1$ and $d = 1$.

$$\begin{aligned} o &= \frac{i + 2p - \tilde{k}}{s} + 1 \\ 4 &= \frac{8 + 2p - d\,(k-1) - 1}{s} + 1 \\ 3 &= \frac{8 + 2 - k}{s} + 1 \\ 2 &= \frac{10 - k}{s} \\ 2 &= \frac{10 - k}{s} \end{aligned} \tag{3.9}$$

If we set $k = 4$, we find that $s = 3$.

Answer: $k = 4, d = 1, s = 3$ and $p = 1$.

4 Dropout as weight decay

(a) When the dropout is activated, each element of the input data is conserved with probability p. Therefore, the data can be expressed as $R \odot X$ where R is a matrix of the same dimension as that X, and where $R_{ij} \sim \text{Ber}(p)$. The reader may interpret R_{ij} as a delta kronecker.

(b) In this context, the loss is defined as

$$L(\Theta) = \mathbb{E}_{R\sim\text{Ber}(p)}\left[\|\mathbf{y} - (R \odot \mathbf{x})\,\mathrm{w}\|^2\right]. \tag{4.1}$$

(c)

$$\begin{aligned}\mathbb{E}_{R\sim\text{Ber}(p)}\left[\|\mathbf{y} - (R \odot \mathbf{x})\,\mathrm{w}\|^2\right] &= \dots \\ &= \mathbb{E}\left[\|\mathbf{y}\|^2 - 2y^\intercal R \odot \mathbf{x}\mathrm{w} + \mathrm{w}^\intercal (R \odot \mathbf{x})^\intercal (R \odot \mathbf{x})\,\mathrm{w}\right] \\ &= \|\mathbf{y}\|^2 - 2y^\intercal p\mathbf{x}\mathrm{w} + \mathrm{w}^\intercal \mathbb{E}\left[(R \odot \mathbf{x})^\intercal (R \odot \mathbf{x})\right]\mathrm{w}\end{aligned} \tag{4.2}$$

where

$$\begin{aligned}\mathbb{E}\left[(R \odot \mathbf{x})^\intercal (R \odot \mathbf{x})\right] &= \mathbb{E}\left[\sum_k b_{ik}\mathrm{x}_{ik}\mathrm{x}_{jk}b_{jk}\right] \\ &= \sum_k \mathbb{E}\left[b_{ik}\mathrm{x}_{ik}\mathrm{x}_{jk}b_{jk}\right]\end{aligned} \tag{4.3}$$

Let's consider the case where $i = j$,

$$\begin{aligned}\mathbb{E}\left[b_{ik}\mathrm{x}_{ik}\mathrm{x}_{ik}b_{ik}\right] &= \mathbb{E}\left[b_{ik}^2\mathrm{x}_{ik}^2\right] \\ &= \mathbb{E}\left[b_{ik}^2\right]\mathrm{x}_{ik}^2 \\ &= \left[\text{Var}\left[b_{ik}\right] + \mathbb{E}\left[b_{ik}\right]^2\right]\mathrm{x}_{ik}^2 \\ &= \left[p\,(1-p) + p^2\right]\mathrm{x}_{ik}^2.\end{aligned} \tag{4.4}$$

Then, the case where $i \neq j$,

$$\begin{aligned}\mathbb{E}\left[b_{ik}\mathrm{x}_{ik}\mathrm{x}_{ik}b_{ik}\right] &= \mathbb{E}\left[b_{ik}\right]\mathbb{E}\left[b_{jk}\right]\mathrm{x}_{ik}\mathrm{x}_{jk} \\ &= p^2\mathrm{x}_{ik}\mathrm{x}_{jk}\end{aligned} \tag{4.5}$$

Equation 4.2 then becomes

$$\begin{aligned}\mathbb{E}_{R\sim\text{Ber}(p)}\left[\|\mathbf{y} - (R \odot \mathbf{x})\,\mathrm{w}\|^2\right] &= \dots \\ &= \|\mathbf{y}\|^2 - 2y^\intercal p\mathbf{x}\mathrm{w} + \mathrm{w}^\intercal \mathbb{E}\left[(R \odot \mathbf{x})^\intercal (R \odot \mathbf{x})\right]\mathrm{w} \\ &= \|\mathbf{y}\|^2 - 2y^\intercal p\mathbf{x}\mathrm{w} + \mathrm{w}^\intercal\left[p\,(1-p)\,\text{diag}\left(\mathbf{x}^\intercal\mathbf{x}\right) + p^2\mathbf{x}^\intercal\mathbf{x}\right]\mathrm{w}\end{aligned} \tag{4.6}$$

We need to find the minimize the above to obtain the ideal w. We find that

$$\begin{aligned}\frac{\partial}{\partial \mathrm{w}} L(\Theta) &= -y^{\intercal} p\mathbf{x} + \mathrm{w} p(1-p)\operatorname{diag}(\mathbf{x}^{\intercal}\mathbf{x}) + p^2 \mathrm{w}\mathbf{x}^{\intercal}\mathbf{x} \\ 0 &= -y^{\intercal}\mathbf{x} + \mathrm{w}\left[(1-p)\operatorname{diag}(\mathbf{x}^{\intercal}\mathbf{x}) + p\mathbf{x}^{\intercal}\mathbf{x}\right] \\ y^{\intercal}\mathbf{x} &= \bar{\mathrm{w}} p\left[\frac{(1-p)}{p}\Gamma^2 + \mathbf{x}^{\intercal}\mathbf{x}\right]\end{aligned} \tag{4.7}$$

$$\boxed{\bar{\mathrm{w}}^{\star} = \left[\frac{(1-p)}{p}\Gamma^2 + \mathbf{x}^{\intercal}\mathbf{x}\right]^{-1} y\mathbf{x}^{\intercal}} \tag{4.8}$$

We defined in the above equations $\Gamma^2 = \operatorname{diag}(\mathbf{x}^{\intercal}\mathbf{x})$ and $\bar{\mathrm{w}} = p\mathrm{w}$.

5 Dropout as geometric ensemble

Note: for sake of transparency, we refer the reader to section 7.12 of I. Goodfellow *et al.* for more details.

Let's consider the case of a single linear layer model with softmax output,

$$P\left(\mathrm{y}=j \mid v\right)=\hat{\mathrm{y}}_j=\mathrm{softmax}\left(\mathbf{W}^{\intercal}\mathbf{v}+\mathbf{b}\right)_j, \tag{5.1}$$

where $\mathbf{v}$ is the input data.

The effect of dropout is to introduce a mask on the vector $\mathbf{W}$, transforming a significant fraction of the weights to zero. Since the mask is different for every evaluation of the model, we may represent mathematically this procedure by introducing a matrix mask $\mathbf{m}$ of the same dimension as $\mathbf{x}$ which set to zero some of the inputs,

$$\mathbf{x}\rightarrow\mathbf{m}\odot\mathbf{x}. \tag{5.2}$$

The reader may appreciate better the above expression by considering the non vectorize equation, given by $x_{ij}\rightarrow m_{ij}x_{i,j}$ where $m_{i,j}$ is equivalent to the kronecker operator $\delta_{i,j}$.

Thus, Eq. 5.1 becomes

$$P\left(\mathrm{y}=j \mid \mathbf{v},\mathbf{m}\right)=\mathrm{softmax}\left(\mathbf{W}^{\intercal}\left(\mathbf{m}\odot\mathbf{x}\right)+\mathbf{b}\right)_j. \tag{5.3}$$

The predictive distribution is defined as the normalized geometrical mean,

$$P_{\text{ensemble}}\left(\mathrm{y}=j \mid \mathbf{v}\right)=\frac{\tilde{P}_{\text{ensemble}}\left(\mathrm{y}=j \mid \mathbf{v}\right)}{\sum_{j'}\tilde{P}_{\text{ensemble}}\left(\mathrm{y}=j' \mid \mathbf{v}\right)} \tag{5.4}$$

where $\tilde{P}_{\text{ensemble}}\left(\mathrm{y}=j \mid v\right)$ is defined as the geometrical mean

$$\tilde{P}_{\text{ensemble}}\left(\mathrm{y}=j \mid \mathbf{v}\right)=\left[\prod_{\mathbf{m}}P\left(\mathrm{y}=j \mid \mathbf{v},\mathbf{m}\right)\right]^{1/2^d}. \tag{5.5}$$

In the above expression, d is the number of units that are put to zero due to dropout.

Putting Eqs.5.3 and 5.5 together, we find that

$$
\begin{aligned}
P_{\text{ensemble}}\left(\mathrm{y}=j \mid \mathbf{v}\right) &\propto \left[\prod_{\mathbf{m}} \operatorname{softmax}\left(\mathbf{W}^{\intercal}\left(\mathbf{m} \odot \mathbf{x}\right)+\mathbf{b}\right)_j\right]^{1/2^d} \\
&= \left[\prod_{\mathbf{m}} \exp\left(\mathbf{W}_{j,:}^{\intercal}\left(\mathbf{m} \odot \mathbf{x}\right)+\mathbf{b}_j\right)\right]^{1/2^d} \\
&= \exp\left(\sum_{\mathbf{m}} \mathbf{W}_{j,:}^{\intercal}\left(\mathbf{m} \odot \mathbf{x}\right)+\mathbf{b}_j\right)^{1/2^d} \\
&= \exp\left(\frac{1}{2^d}\sum_{\mathbf{m}} \mathbf{W}_{j,:}^{\intercal}\left(\mathbf{m} \odot \mathbf{x}\right)+\mathbf{b}_j\right) \\
&= \exp\left(\sum_{\mathbf{m}} \frac{1}{2}\mathbf{W}_{j,:}^{\intercal}\left(\mathbf{m} \odot \mathbf{x}\right)+\mathbf{b}'_j\right) \\
&= \exp\left(\frac{1}{2}\mathbf{W}_{j,:}^{\intercal}\mathbf{x}+\mathbf{b}'_j\right) \qquad (5.6)
\end{aligned}
$$

where $\mathbf{W}_{j,:}$ represent the j^{th} line of W. This result implies that the scaling with a factor of 0.5 is equivalent to the predictive probability distribution of the ensemble.

6 Normalization

(a) Let us suppose a vector x and y related through the affine relation

$$\mathrm{y} = W^{\intercal}\mathrm{x} + b. \tag{6.1}$$

The batch normalization is defined as

$$\mathrm{BN}\left(\mathrm{y}\right) = \gamma \odot \frac{\left(\mathrm{y} - \mathbb{E}\left[\mathrm{y}\right]\right)}{\mathrm{Var}\left[\mathrm{y}\right]} + \beta \tag{6.2}$$

For the problem in hand, we have that

$$\begin{aligned} \mathbb{E}\left[\mathrm{y}\right] &= \mathbb{E}\left[\left(W\mathrm{x} + b\right)\right] \\ &= W^{\intercal}\mathbb{E}\left[\mathrm{x}\right] + \mathbb{E}\left[b\right] \\ &= b \end{aligned} \tag{6.3}$$

$$\begin{aligned} \mathrm{Var}\left[\mathrm{y}\right] &= \mathrm{Var}\left[\left(W\mathrm{x} + b\right)\right] \\ &= W^{\intercal}\mathrm{Var}\left[\mathrm{x}\right]W + \mathrm{Var}\left[b\right] \\ &= W^{\intercal}W \\ &= \|W\| \end{aligned} \tag{6.4}$$

By combining Eqs. 6.2, 6.3 and 6.4, we find that

$$\begin{aligned} \mathrm{BN}\left(\mathrm{y}\right) &= \gamma \odot \frac{\left(\mathrm{y} - \mathbb{E}\left[\mathrm{y}\right]\right)}{\mathrm{Var}\left[\mathrm{y}\right]} + \beta \\ &= \gamma \odot \frac{\left(\mathrm{y} - b\right)}{\|W\|} + \beta \\ &= \gamma \odot \frac{W^{\intercal}}{\|W\|}\mathrm{x} + \beta \\ &= \gamma \odot \left(\frac{W}{\|W\|}\right)^{\intercal}\mathrm{x} + \beta \end{aligned} \tag{6.5}$$

which corresponds to the weight normalization.

(b)

$$\begin{aligned}
\nabla_{\mathbf{u}} L &= \frac{\partial L}{\partial \mathbf{u}} \\
&= \frac{\partial L}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{u}} \\
&= \frac{\partial L}{\partial \mathbf{w}} \frac{\partial}{\partial \mathbf{u}} \left(g \frac{\mathbf{u}}{\|\mathbf{u}\|} \right) \\
&= \frac{\partial L}{\partial \mathbf{w}} \left(\frac{g}{\|\mathbf{u}\|} - g \frac{\mathbf{u}}{\|\mathbf{u}\|^2} \frac{\mathbf{u}^\intercal}{\|\mathbf{u}\|} \right) \\
&= \frac{g}{\|\mathbf{u}\|} \left(1 - \frac{\mathbf{u}\mathbf{u}^\intercal}{\|\mathbf{u}\|^2} \right) \nabla_{\mathbf{w}} L
\end{aligned} \tag{6.6}$$

$1 - \frac{\mathbf{u}\mathbf{u}}{\|\mathbf{u}\|^2}$ is defined as the orthogonal complement projection matrix.

$$\begin{aligned}
1 - \frac{\mathbf{u}\mathbf{u}}{\|\mathbf{u}\|^2} &= 1 - \frac{\mathbf{u}\mathbf{u}^\intercal}{\mathbf{u}^\intercal \mathbf{u}} \\
&= 1 - \mathbf{u} \left(\mathbf{u}^\intercal \mathbf{u} \right)^{-1} \mathbf{u}^\intercal \\
&= 1 - \mathrm{P} \\
&= W^\star
\end{aligned} \tag{6.7}$$

where $\mathrm{P} = \mathbf{u} \left(\mathbf{u}^\intercal \mathbf{u} \right)^{-1} \mathbf{u}^\intercal$ is defined as the projection matrix. Therefore, we find that

$$\begin{aligned}
\nabla_{\mathbf{u}} L &= \frac{g}{\|\mathbf{u}\|} \left(1 - \frac{\mathbf{u}\mathbf{u}}{\|\mathbf{u}\|^2} \right) \nabla_{\mathbf{w}} L \\
&= \frac{g}{\|\mathbf{u}\|} W^\star \nabla_{\mathbf{w}} L
\end{aligned} \tag{6.8}$$

Taken from Tim Salimans *et al.* (2016) *Weight normalization has two effects: it scales the weight gradient, but it also projects the gradient away from the current weight vector. Both effects help to bring the covariance matrix of the gradient closer to identity and benefit optimization.*

(c) The update of $\mathbf{u}$ is given by

$$\mathbf{u}' \leftarrow \mathbf{u} + \alpha \nabla_{\mathbf{u}} L \tag{6.9}$$

where $\nabla_{\mathbf{u}} L \propto W^\star \nabla_{\mathbf{w}} L$.

It is clear that $\nabla_{\mathbf{u}} L$ is orthogonal to $\mathbf{w}$ since we project away from it due to the orthogonal complement projection matrix $W^\star$. Since $\mathbf{w}$ is proportional to $\mathbf{u}$, we find

that $\nabla_{\mathbf{u}} L \perp \mathbf{u}$. From the Pythagorean theorem, we find that

$$\begin{aligned} \mathbf{u}' &= \mathbf{u} + \alpha \nabla_{\mathbf{u}} L \\ \|\mathbf{u}'\| &= \left(\|\mathbf{u}\|^2 + \alpha \|\nabla_{\mathbf{u}} L\|^2\right)^{1/2} \\ &= \left(\|\mathbf{u}\|^2 + \alpha \|k\mathbf{u}\|^2\right)^{1/2} \\ &= \|\mathbf{u}\| \left(1 + \alpha k^2\right)^{1/2} > \|\mathbf{u}\| \end{aligned} \tag{6.10}$$

We see from the above equation that $\|\mathbf{u}'\|$ grows monotonically with the number of updates. Further, it also explain its evolution with the learning rate, i.e. the growth increases with α.

It is important to note that we assumed that$\nabla_{\mathbf{u}} L \propto k\mathbf{u}$ in the above equation. This is a consequence of Eq. 6.5. This is not a necessary condition, though.