Forward and Backward Propagation in Convolutional Neural Networks

Owen Diba

January 15, 2024

Contents

1 Introduction

In these notes, I derive the forward and backward propagation equations for a convolutional layer in a neural network. In section section 4, I outline one way of vectorizing the propagation steps to speed up computation. The motivation for these notes came whilst reading *Machine Learning - a First Course for Engineers and Scientists* [1]. Whilst it gives a good overview it does not provide a detailed explanation of the back-propagation algorithm in convolutional (CNNs). I use similar notation as Lindholm et al. and fig. 1 is adapted from figure 6.6 in their book.

2 Forward Propagation

For each layer we have an input fig. 1

$$\mathbf{q}^{(l-1)} \in \mathbb{R}^{\mathrm{ch}_{\mathrm{in}}^{(l)} \times h_{\mathrm{in}}^{(l)} \times h_{\mathrm{in}}^{(l)}}, \tag{1}$$

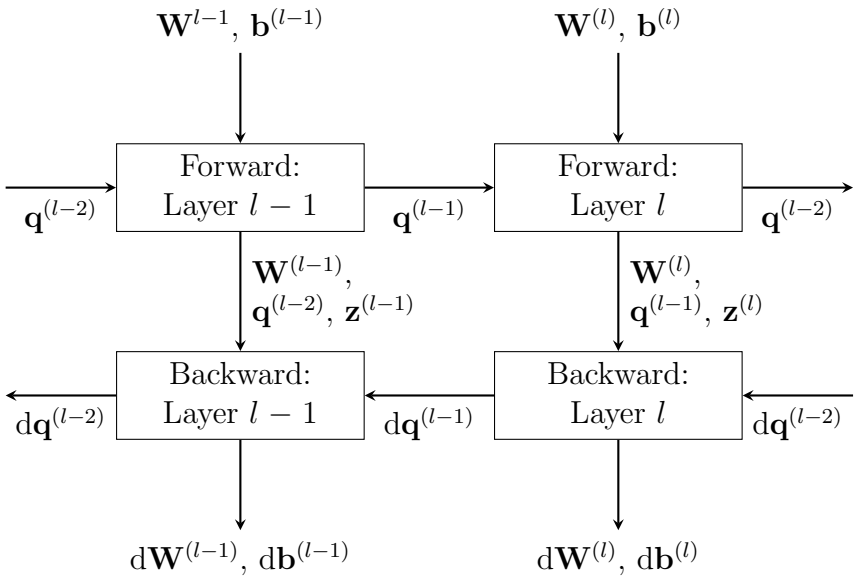

Figure 1: Computational graph of the back-propagation algorithm in two layers of a neural network. Adapted from fig. 6.6 in [1, p. 144].

where $\mathrm{ch}_{\mathrm{in}}^{(l)}$ is the number of input channels to the layer, and $h_{\mathrm{in}}^{(l)}$ is the height and width of each image. Each layer has a filter tensor

$$\mathbf{W}^{(l)} \in \mathbb{R}^{\mathrm{ch}_{\mathrm{in}}^{(l)} \times \mathrm{ch}_{\mathrm{out}}^{(l)} \times f_l \times f_l}, \tag{2}$$

where $\mathrm{ch}_{\mathrm{in(out)}}^{(l)}$ is the number of channels in (out), and f_l is the filter width, and a bias vector

$$\mathbf{b}^{(l)} \in \mathbb{R}^{\mathrm{ch}_{\mathrm{out}}^{(l)}}. \tag{3}$$

The input is cross-correlated with the input to give the pre-activation output

$$\mathbf{z}^{(l)} \in \mathbb{R}^{\mathrm{ch}_{\mathrm{out}}^{(l)} \times h_{\mathrm{out}}^{(l)} \times h_{\mathrm{out}}^{(l)}}, \tag{4}$$

where for the output channel ν the resulting image is given by

$$\mathbf{z}_{\nu}^{(l)} = \sum_{\mu=0}^{\mathrm{ch}_{\mathrm{in}}-1} \bar{\mathbf{q}}_{\mu}^{(l-1)} \underset{s_l}{\star} \mathbf{W}_{\mu\nu}^{(l)} + b_{\nu}^{(l)} \mathbb{1}^{h_{\mathrm{out}}^{(l)} \times h_{\mathrm{out}}^{(l)}}, \tag{5}$$

where $\underset{s_l}{\star}$ denotes the cross-correlation operation with stride s_l. More explicitly, for the element in the ith pixel row and jth pixel column of $\mathbf{z}_{\nu}^{(l)}$, we have

$$z_{\nu}^{(l)}(i,j) = \sum_{\mu=0}^{\mathrm{ch}_{\mathrm{in}}-1} \sum_{i'=0}^{f_l-1} \sum_{j'=0}^{f_l-1} \bar{q}_{\mu}^{(l-1)}(is_l + i', js_l + j') W_{\mu\nu}^{(l)}(i',j') + b_{\nu}^{(l)}. \tag{6}$$

Here, $\bar{q}^{(l-1)}$ is the input array padded with enough zeros to give an output array with image height $h_{\text{out}}^{(l)}$. If input and output array heights and filter sizes are fixed then the required amount of padding is given by

$$p_{\rightarrow}^{(l)} = s_l(h_{\text{out}}^{(l)} - 1) + f_l - h_{\text{in}}^{(l)}. \tag{7}$$

This is the number of zero pixels that the height and width of the image should be extended by. We try to pad as symmetrically as possible on all sides of the image. If $p_{\rightarrow}^{(l)}$ is even, then we can pad symmetrically with $p_{\rightarrow}^{(l)}/2$ zeros on all sides. If it is odd then we choose to pad with $\lceil p_{\rightarrow}^{(l)}/2 \rceil$ on the top left corner and $\lfloor p_{\rightarrow}^{(l)}/2 \rfloor$ on the bottom right corner. Then the (i,j)th element of $\bar{\mathbf{q}}_{\mu}^{(l-1)}$ is given by

$$\bar{\mathbf{q}}_{\mu}^{(l-1)}(i,j) = \mathbf{q}_{\mu}^{(l-1)}(i - \lceil p_{\rightarrow}^{(l-1)}/2 \rceil, j - \lceil p_{\rightarrow}^{(l-1)}/2 \rceil), \tag{8}$$

for $\lceil p_{\rightarrow}^{(l-1)}/2 \rceil \leq i,j \leq \lceil p_{\rightarrow}^{(l-1)}/2 \rceil + h_{\text{in}}^{(l)}$ and for all other indices is equal to zero.

The final step for the forward pass for each layer is the activation. Each pixel of each output channel of $\mathbf{z}^{(l)}$ is a hidden unit where the activation function $\sigma(\cdot)$ is applied. The final output from the layer is given by

$$\mathbf{q}^{(l)} = \sigma(\mathbf{z}^{(l)}). \tag{9}$$

3 Backward Propagation

3.1 Error on Output

We first consider the dependence of the pre-activation output of the convolutional layer to the post-activation differential. Clearly, only the input to each individual hidden neuron directly effects its output, hence

$$\mathrm{d}z_{\nu}^{(l)}(i,j) = \mathrm{d}q_{\nu}^{(l)}(i,j)\frac{\partial q_{\nu}^{(l)}(i,j)}{\partial z_{\nu}^{(l)}(i,j)} = \mathrm{d}q_{\nu}^{(l)}(i,j)\sigma'(z_{\nu}^{(l)}(i,j)), \tag{10}$$

and so the differential of the tensor is

$$\mathrm{d}\mathbf{z}_{\nu}^{(l)} = \mathrm{d}\mathbf{q}_{\nu}^{(l)} \circ \sigma(\mathbf{z}_{\nu}^{(l)}), \tag{11}$$

where $\circ$ is the Hadamard product of the two matrices.

3.2 Error on Filters and Biases

We next consider the differential of the filters. They directly effect the pre-activation output of the layer, and so

$$W_{\mu\nu}(i,j) = \sum_{\nu=0}^{\text{ch}_{\text{out}}^{(l)}-1} \sum_{i'=0}^{h_{\text{out}}^{(l)}-1} \sum_{j'=0}^{h_{\text{out}}^{(l)}-1} \mathrm{d}z_{\nu}^{(l)}(i',j')\frac{\partial z_{\nu}^{(l)}(i',j')}{\partial W_{\mu\nu}^{(l)}(i,j)}. \tag{12}$$

We have that

$$\frac{\partial z_\nu^{(l)}(i',j')}{\partial W_{\mu\nu}^{(l)}(i,j)} = \bar{q}_\mu^{(l-1)}(i's_l + i, j's_l + j) \tag{13}$$

and substituting this result into the previous equation we have

$$\mathrm{d}W_{\mu\nu}^{(l)}(i,j) = \sum_{\nu=0}^{\mathrm{ch}_{\mathrm{out}}^{(l)}-1} \sum_{i'=0}^{h_{\mathrm{out}}^{(l)}-1} \sum_{j'=0}^{h_{\mathrm{out}}^{(l)}-1} \mathrm{d}z_\nu^{(l)}(i',j')\bar{q}_\mu^{(l-1)}(i's_l + i, j's_l + j) \tag{14}$$

which we may see as a kind of cross correlation with $\mathrm{d}\mathbf{z}_\nu^{(l)}$ as the filter but instead of the stride effecting how the "sliding window" of the filter jumps along the elements of the image, instead it effectively causes a "dilation" of the filter. The simplest picture of this is that we have the standard cross-correlation but the filter matrix is filled with zeros horizontally, vertically, and diagonally between adjacent pixels of the original filter. The new effective window size is then $h_{\mathrm{out}}^{(l)}(s_l + 1) - 1$. So we can write

$$\mathrm{d}\mathbf{W}_{\mu\nu}^{(l)} = \sum_{\nu=0}^{\mathrm{ch}_{\mathrm{out}}^{(l)}-1} \bar{\mathbf{q}}_\mu^{(l-1)} \star \mathrm{d}\hat{\mathbf{z}}_\nu^{(l)} \tag{15}$$

where $\mathrm{d}\hat{\mathbf{z}}_\nu^{(l)}$ denotes the dilated matrix of $\mathrm{d}\mathbf{z}_\nu^{(l)}$. We also need the differential of the biases. The νth element of the bias vector effects every element of $\mathbf{z}(i,j)$ equally with linear dependence so we have

$$\mathrm{d}b_\nu^{(l)} = \sum_{i=0}^{h_{\mathrm{out}}^{(l)}-1} \sum_{j=0}^{h_{\mathrm{out}}^{(l)}-1} \mathrm{d}z_\nu^{(l)}(i,j). \tag{16}$$

3.3 Error on Input: stride $= 1$

Finally, we need the differential of the input to the layer with respect to the pre-activation output. Consider the element (i,j) of input. This corresponds to $(i + \lceil p_\rightarrow^{(l-1)}/2 \rceil, j + \lceil p_\rightarrow^{(l-1)}/2 \rceil)$ of the padded input. The element (i',j') of the output corresponds to a filter window with a bottom right hand corner that multiplies the element $(i' + f_l - 1, j' + f_l - 1)$ of the padded input. This corner of the window will be the first part of the filter that acts on element (i,j) of the input, and hence element (i',j') of the output is the first element to be effected by (i,j). So we solve for (i',j')

$$i' = i - (f_l - 1) + \lceil p_\rightarrow^{(l-1)}/2 \rceil), \tag{17}$$

$$j' = j - (f_l - 1) + \lceil p_\rightarrow^{(l-1)}/2 \rceil). \tag{18}$$

The last element of the output effected by input (i,j) corresponds to the filter window's top left hand corner multiplying element (i,j). So output coordinates will have advanced

by $k-1$ in both row and column indices. Therefore the set of all output pixels effected by input (i,j) is

$$i' - (f_l - 1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, \tag{19}$$

$$j' - (f_l - 1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b, \tag{20}$$

where $a, b \in \{0, \ldots, k-1\}$, and where we ignore indices that are outside the range $\{0, \ldots, h_{\text{in}}^{(l)}\}$. So by the chain rule we have

$$\begin{aligned} \mathrm{d}q^{(l-1)}(i,j) = \sum_{a=0}^{f_l-1} \sum_{b=0}^{f_l-1} \mathrm{d}z^{(l)}\left(i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b\right) \\ \times \frac{\partial z^{(l)}(i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b)}{\partial q^{(l-1)}(i,j)}. \end{aligned}$$

From eq. (6), the re-indexed output is

$$\begin{aligned} & z^{(l)}(i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b) \\ & \quad = \sum_{a'=0}^{f_l-1} \sum_{b'=0}^{f_l-1} q^{(l-1)}(i - (f_l-1) + a + a', i - (f_l-1) + b + b') W^{(l)}(a', b') + b, \end{aligned} \tag{21}$$

and so it is clear that the derivative is

$$\frac{\partial z^{(l)}(i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b)}{\partial q^{(l-1)}(i,j)} = W^{(l)}((k-1)-a, (k-1)-b).$$

it is easy to verify that

$$W^{(l)}((k-1)-a, (k-1)-b) = \underset{180^\circ}{\mathrm{rot}}(W^{(l)})(a,b) \tag{22}$$

where $\underset{180^\circ}{\mathrm{rot}}(\cdot)$ denotes that we rotate the argument matrix by 180° about its centre. Then

$$\begin{aligned} \mathrm{d}q^{(l-1)}(i,j) = \sum_{a=0}^{f_l-1} \sum_{b=0}^{f_l-1} \mathrm{d}z^{(l)}\left(i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + a, i - (f_l-1) + \left\lceil p_{\rightarrow}^{(l-1)}/2 \right\rceil + b\right) \\ \times \underset{180^\circ}{\mathrm{rot}}(W^{(l)})(a,b). \end{aligned} \tag{23}$$

Can we write this more compactly as a cross-correlation? We just need to adequately pad the input matrix $\mathrm{d}z^{(l)}$. The padding for any cross correlation is $p = d_{\text{out}} - 1 + d_{\text{filter}} - d_{\text{in}}$. In this particular case $d_{\text{out}} = h_{\text{in}}^{(l)}$, $d_{\text{in}} = p_{\rightarrow}^{(l)} + h_{\text{in}}^{(l)} + 1 - f_l$, and $d_{\text{filter}} = f_l$. So the padding is

$$p_{\leftarrow}^{(l)} = 2(f_l - 1) - p_{\rightarrow}^{(l)}. \tag{24}$$

Notice that, since $\lfloor -x \rfloor = \lceil x \rceil$,

$$\left\lfloor p_{\leftarrow}^{(l)}/2 \right\rfloor = f_l - 1 - \left\lceil p_{\rightarrow}^{(l)}/2 \right\rceil, \tag{25}$$

which is the shift we see in the argument of $\mathrm{d}z$ in eq. (23). So if we define $\mathrm{d}\bar{z}^{(l)} \in \mathbb{R}^{(h_{\text{out}}^{(l)}+p_{\leftarrow}^{(l)})\times(h_{\text{out}}^{(l)}+p_{\leftarrow}^{(l)})}$ with elements

$$\mathrm{d}\bar{z}^{(l)}\,(i,j) = \begin{cases} 0, & \text{for } 0 \le i,j < \left\lfloor p_{\rightarrow}^{(l)}/2 \right\rfloor, \\ \mathrm{d}z^{(l)}\,(i - \left\lfloor \frac{p_{\leftarrow}^{(l)}}{2} \right\rfloor, j - \left\lfloor \frac{p_{\leftarrow}^{(l)}}{2} \right\rfloor), & \text{for } \left\lfloor p_{\leftarrow}^{(l)}/2 \right\rfloor \le i,j \le \left\lfloor p_{\leftarrow}^{(l-1)}/2 \right\rfloor + h_{\text{out}}^{(l)}, \\ 0, & \text{for } \left\lfloor p_{\leftarrow}^{(l)}/2 \right\rfloor + h_{\text{out}}^{(l)} + 1 \le i,j \le h_{\text{out}}^{(l)} + p_{\leftarrow}^{(l)} - 1, \end{cases} \tag{26}$$

then

$$\mathrm{d}\mathbf{q}^{(l-1)} = \mathrm{d}\bar{\mathbf{z}}^{(l)} \star \mathrm{rot}(\mathbf{W}^{(l)}, 180^\circ), \tag{27}$$

and generalising to multiple channels

$$\mathrm{d}\mathbf{q}_{\nu}^{(l-1)} = \sum_{\nu=0}^{\mathrm{ch}_{\text{in}}^{(l)}-1} \mathrm{d}\bar{\mathbf{z}}_{\nu}^{(l)} \star \mathrm{rot}(\mathbf{W}_{\mu\nu}^{(l)}, 180^\circ). \tag{28}$$

3.4 Error on Input: stride > 1

This is hard to derive analytically, but reasonably easy to derive pictorially for particular cases, and then simple to generalise from there. We find that

$$\mathrm{d}\mathbf{q}_{\mu}^{(l-1)} = \sum_{\nu=0}^{\mathrm{ch}_{\text{out}}^{(l)}-1} \mathrm{d}\bar{\hat{\mathbf{z}}}_{\nu}^{(l)} \star \mathrm{rot}\left(W_{\mu\nu}^{(l)}, 180^\circ\right) \tag{29}$$

where $\mathrm{d}\bar{\hat{\mathbf{z}}}_{\nu}^{(l)}$ is the dilated matrix that appeared earlier but now padded on either side in order for the resulting dimension of the cross-correlation operation to match the input dimension to the layer $h_{\text{in}}^{(l)}$. The dimension of the dilated matrix $\mathrm{d}\hat{\mathbf{z}}_{\nu}^{(l)}$ is $h_{\text{out}}^{(l)}(s_l+1)-1$, and the padding is related to the forwards padding as in eq. (24).

4 Vectorization

To speed up both the forward and backward passes we will vectorize the convolution operations. We now consider an input

$$\mathbf{Q}^{(l-1)} \in \mathbb{R}^{n_b \times \mathrm{ch}_{\text{in}}^{(l)} \times h_{\text{in}}^{(l)} \times h_{\text{in}}^{(l)}}, \tag{30}$$

where n_b is the size of the current mini-batch. If we briefly consider just one input and output channel, and ignore the bias, the element $z^{(l)}(i,j)$ of the output is given by the

element wise multiplication of the filter $\mathbf{W}$ with the subarray $\bar{\mathbf{q}}^{(l-1)}[i,j]$

$$\bar{\mathbf{q}}^{(l-1)}[i,j] = \begin{bmatrix} \bar{q}^{(l-1)}(is_l, js_l) & \bar{q}^{(l-1)}(is_l, js_l+1) & \cdots & \bar{q}^{(l-1)}(is_l, js_l+f_l-1) \\ \bar{q}^{(l-1)}(is_l+1, js_l) & \bar{q}^{(l-1)}(is_l+1, js_l+1) & \cdots & \bar{q}^{(l-1)}(is_l+1, js_l+f_l-1) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{q}^{(l-1)}(is_l+f_l-1, js_l) & \bar{q}^{(l-1)}(is_l+f_l-1, js_l+1) & \cdots & \bar{q}^{(l-1)}(is_l+f_l-1, js_l+f_l-1) \end{bmatrix}. \tag{31}$$

We now vectorize this matrix

$$\text{vec}\left(\bar{\mathbf{q}}^{(l-1)}[i,j]\right) = \begin{bmatrix} \bar{q}^{(l-1)}(is_l, js_l) \\ \bar{q}^{(l-1)}(is_l, js_l+1) \\ \vdots \\ \bar{q}^{(l-1)}(is_l, js_l+f_l-1) \\ \bar{q}^{(l-1)}(is_l+1, js_l+f_l-1) \\ \vdots \\ \bar{q}^{(l-1)}(is_l+f_l-1, js_l+f_l-1) \end{bmatrix}. \tag{32}$$

We similarly vectorize the filter matrix, and thus we have

$$\mathbf{z}_\nu^{(l)}(i,j) = \sum_\mu \text{vec}\left(\bar{\mathbf{q}}_\mu^{(l-1)}[i,j]\right) \text{vec}\left(\mathbf{W}_{\mu\nu}^{(l)}\right)^T + b_\nu. \tag{33}$$

Now, given the input matrix $\mathbf{Q}^{(l-1)}$ we perform this partitioning into sub-arrays and vectorizing of the last two dimensions (image height and width) for each image in the mini-batch and each input channel (the first two dimensions). The result is the tensor

$$\mathbb{Q}^{(l-1)} \in \mathbb{R}^{n_b \times \text{ch}_{\text{in}}^{(l)} \times (h_{\text{out}}^{(l)})^2 \times f_l^2}. \tag{34}$$

We represent the reshaped filter tensor as $\mathbb{W}^{(l)} \in \mathbb{R}^{\text{ch}_{\text{in}}^{(l)} \times \text{ch}_{\text{out}}^{(l)} \times f_l^2}$, and then the cross correlation operation for the whole mini-batch is given by the tensor contraction over input channels and filter pixels (the last dimension in both tensors)

$$\mathbb{Z}_{i\nu k}^{(l-1)} = \mathbb{Q}_{i\mu kl}^{(l-1)} \mathbb{W}_{\mu\nu l}^{(l)}. \tag{35}$$

References

[1] Andreas Lindholm et al. *Machine Learning - a First Course for Engineers and Scientists*. Cambridge University Press, 2022. URL: `https://smlbook.org`.