1 Layers

1.1 Convolution

Let X, Y denote the input and output images (maps) respectively. Let F denote the filters. We assume that X and Y have been reshaped into two matrices, with the first index spanning spatial dimensions and the second spanning feature channels. Thus

$$X \in \mathbb{R}^{(wh)\times d}, \qquad Y \in \mathbb{R}^{(w'h')\times k}, \qquad F \in \mathbb{R}^{(w_f h_f d)\times k} \qquad w' = w - w_f + 1, \qquad h'' = h - h_f + 1,$$

where (w, h, d) is the size of the input image X, (w', h', k) is the size of the output image, (w_f, h_f, d) is the size of a filter, and there are k filters. The output image Y is a function of X and F and is connected to the output energy by a function f:

$$Y = g(X, F), \qquad z = f(Y) \in \mathbb{R}.$$

The operation g is a linear filter, applying each of the filters in F to produce each of the channels in Y. Up to a rearrangement of the elements of X, this can be written as a matrix multiplication. In particular, let $\phi(X)$ be the `im2row` operator, which extracts from X patches of the same volumes as the filters, placing them as columns of a matrix:

$$Y = g(X, F) = \phi(X)F, \qquad \phi : \mathbb{R}^{(wh)\times d} \to \mathbb{R}^{(w'h')\times(w_f h_f d)}.$$

Note that ϕ simply rearranges the elements of X and is, therefore, a linear operator. In particular we can rewrite it as

$$\operatorname{vec}(\phi(X)) = H \operatorname{vec}(X), \qquad H \in \mathbb{R}^{(w'h'w_f h_f d)\times(whd)}$$

for a suitable matrix H. The derivative of the function $f(g(X, F))$ are then given by

$$\boxed{\frac{dz}{dF} = \phi(X)^\top \frac{df}{dY}, \qquad \frac{dz}{d\operatorname{vec}(X)} = H^\top \operatorname{vec}\left(\frac{df}{dY} F^\top\right) = \phi^*\left(\frac{df}{dY} F^\top\right).}$$

Here we define the `row2im` operator $H^\top$ as the dual of `im2row`:

$$\operatorname{vec}(\phi^*(Y)) = H^\top \operatorname{vec}(Y).$$

Let (l, m, k, p, d) be an index in the `im2row` output $\phi(X)$ and (i, j, d) an index in the input X. Here indexes are mapped as $(l, m), (k, p, d)$ to the first and second index of $\phi(X)$ and to $(i, j), d$ of X. With slight abuse of notation one has:

$$[\phi(X)]_{(l,m,k,p,d)} = X_{(i,j,d)}, \qquad i = l + k, \quad j = m + p.$$

Likewise for the dual operator `row2im`:

$$[\phi^*(Y)]_{(i,j,d)} = \sum_{k=0}^{w_f-1} \sum_{p=0}^{h_f-1} Y_{(i,j,k,p,d)}.$$

Sizes, strides, and padding. Suppose we have w pixels in the x direction and a filter of size w_f. Then the filter is contained in the signal

$$w' = w - w_f + 1$$

times (for all possible translations), provided that $w \geq w_f$. If the signal is padded with p pixels to the left and to the right, then

$$w' = w + 2p - w_f + 1.$$

If the filter output is subsampled every δ steps, then samples are at $i = \delta i'$. We must have

$$0 \leq i = \delta i' \leq w' - 1 \qquad \Rightarrow \qquad 0 \leq i' \leq \lfloor \frac{w + 2p - w_f}{\delta} \rfloor.$$

1.2 Max pooling

Similarly to the convolution case, we define a function:

$$Y = g(X, \wedge), \quad z = f(Y) \in \mathbb{R}.$$

Where

$$\boxed{Y = g(X, \wedge) = \operatorname{maxrow} \phi(X).}$$

and $\phi(X)$ is the `im2row` operator defined above. In order to write more compact formulas for the derivative, we introduce the matrix $S(X) \in \mathbb{R}^{(w'h') \times (w_f h_f d)}$ which selects the maximal element in each row of $\phi(X)$:

$$Y = \phi(X) S, \qquad S(X) = \operatorname*{argmax}_{S \geq 0,\ \mathbf{1}^\top S \leq \mathbf{1}^\top} \phi(X) S.$$

Then the derivative is

$$\boxed{\frac{dz}{d\operatorname{vec}(X)} = H^\top \operatorname{vec}\left(\frac{df}{dY} S^\top\right) = \phi^* \left(\frac{df}{dY} S^\top\right).}$$

1.3 Normalization

The normalisation operation normalises the feature channels at any given spatial location (i, j):

$$Y_{(i,j,k)} = X_{(i,j,k)} \left(\kappa + \alpha \sum_{t \in G(k)} X^2_{(i,j,t)}\right)^{-\beta}, \qquad z = f(Y),$$

where $G(k) \subset \{1, 2, \ldots, D\}$ is a subset of the input channels. Note that input X and output Y have the same dimensions. The derivative is easily computed as:

$$\frac{dz}{dX_{(i,j,d)}} = \frac{dz}{dY_{(i,j,d)}} L(i,j,d|X)^{-\beta} - 2\alpha\beta \sum_{k: d \in G(k)} \frac{dz}{dY_{(i,j,k)}} L(i,j,k|X)^{-\beta-1} X_{(i,j,ki)} X_{(i,j,d)}$$

where

$$L(i,j,k|X) = \kappa + \alpha \sum_{t \in G(k)} X^2_{(i,j,t)}.$$

1.4 Vectorisation

Vectorisation (utilised between convolutional and fully connected layers):

$$Y = \operatorname{vec} X, \qquad z = f(Y).$$

The derivative is also a rearrangement of terms:

$$\frac{dz}{dX} = \operatorname{reshape} \frac{dz}{dY}.$$

1.5 ReLU

Rectified linear unit:

$$Y_k = \max\{0, X_k\}, \qquad z = f(Y).$$

Derivative:

$$\frac{dz}{dX_k} = \frac{dz}{dY_k} \delta_{\{X_k > 0\}}.$$

1.6 Fully connected layer

A fully connected layer is simply a matrix multiplication:

$$\operatorname{vec} Y = W \operatorname{vec} X, \qquad z = f(Y).$$

The derivatives w.r.t. input X and parameters W are:

$$\frac{dz}{d \operatorname{vec}(X)^\top} = \frac{dz}{d(\operatorname{vec} Y)^\top} W, \qquad \frac{dz}{dW} = \frac{df}{d \operatorname{vec} Y} (\operatorname{vec} X)^\top.$$

1.7 Softmax

Softmax:

$$Y_k = \frac{e^{X_i}}{\sum_{t=1}^{D} e^{X_t}}, \qquad z = f(Y).$$

Derivative

$$\frac{dz}{dX_d} = \sum_k \frac{dz}{dY_k} \left(e^{X_d} L(X)^{-1} \delta_{\{k=d\}} - e^{X_d} e^{X_k} L(X)^{-2} \right), \quad L(X) = \sum_{t=1}^{D} e^{X_t}.$$

Simplifyng

$$\frac{dz}{dX_d} = Y_d \left(\frac{dz}{dY_d} - \sum_{k=1}^{K} \frac{dz}{dY_k} Y_k. \right).$$

1.8 Log-loss

The log loss is:

$$y = \ell(X, c) = -\log X_c, \qquad z = f(y) = y,$$

where $c \in \{1, 2, \dots, D\}$ is the g.t. class of the image and, this being the output of the network, has $z = y$. The derivative is

$$\frac{dz}{dX_c} = -\frac{1}{X_c}\delta_{\{k=c\}}.$$

Note that one takes the average loss on all the training data.

A Proofs

$$\begin{aligned}
\frac{dz}{d\,\mathrm{vec}(F)^\top} &= \frac{df}{d\,\mathrm{vec}(Y)^\top}\frac{d[\phi(X)F]}{d\,\mathrm{vec}(F)^\top} \\
&= \frac{df}{d\,\mathrm{vec}(Y)^\top}\frac{d[(I_k \otimes \phi(X))\,\mathrm{vec}(F)]}{d\,\mathrm{vec}(F)^\top} \\
&= \mathrm{vec}\left(\frac{df}{dY}\right)^\top (I_k \otimes \phi(X)) \\
&= \mathrm{vec}\left(\phi(X)^\top I_k \frac{df}{dY}\right)^\top \\
&= \mathrm{vec}\left(\phi(X)^\top \frac{df}{dY}\right)^\top
\end{aligned}$$

From which

$$\frac{dz}{dF} = \phi(X)^\top \frac{df}{dY}.$$

Also

$$\begin{aligned}
\frac{dz}{d\,\mathrm{vec}(X)^\top} &= \frac{df}{d\,\mathrm{vec}(Y)^\top}\frac{d\,\mathrm{vec}[\phi(X)F]}{d\,\mathrm{vec}(X)^\top} \\
&= \frac{df}{d\,\mathrm{vec}(Y)^\top}\frac{d[(F^\top \otimes I_{w'h'})\,\mathrm{vec}(\phi(X))]}{d\,\mathrm{vec}(X)^\top} \\
&= \frac{df}{d\,\mathrm{vec}(Y)^\top}\frac{d[(F^\top \otimes I)H\,\mathrm{vec}(X)]}{d\,\mathrm{vec}(X)^\top} \\
&= \frac{df}{d\,\mathrm{vec}(Y)^\top}(F^\top \otimes I)H \\
&= \mathrm{vec}\left(\frac{df}{dY}F^\top\right)^\top H
\end{aligned}$$