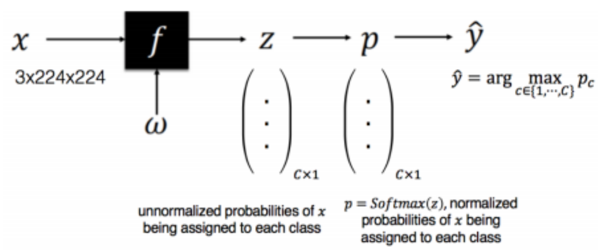

1. Consider a convolutional neural network with input images sized $3 \times 224 \times 224$

    (a) Define the loss function $Loss(y, p)$ for a random sample (x, y) where y is the ground-truth one-hot label vector

    For this problem, we will simply be using cross-entropy for our definition of the loss function $Loss(y, p)$, which also happens to be the negative log-likelihood function. For this problem, we will be looking at an individual input x with one respective label y

    $$Loss(y, p) = -\sum_{i=1}^{C} y_i \log p_i \tag{1}$$

    As one can see, the one-hot labels y are multiplied element-wise by the natural log of the final layer. Since only the positive results remain from y, the error term is negated to indicate a smaller error for a larger value of p. Alternatively, this can be thought of as an intuitively equivalent term to $\prod_{i=1}^{C} p_i^{y_i}$, which is just the likelihood of the function

    (b) Derive the gradient $\frac{\partial Loss(y,p)}{\partial z_c}$ where $c \in [1, C]$

    The gradient of the loss function w.r.t. each individual node of layer z, indicated by z_c, is calculated by first knowing that

    $$p = Softmax(z) = \frac{e^z}{\sum_{k=1}^{C} e^{z_k}} \tag{2}$$

    It should be noted that the summation term in the denominator term from hereon out will be denoted by the scalar Z, as it is easier for calculation down the line. This can be substituted for in the loss function to get

    $$Loss(y, p) = -\sum_{i=1}^{C} y_i \log\left(\frac{e^{z_i}}{Z}\right) \tag{3}$$

    which becomes

    $$\frac{\partial Loss(y, p)}{\partial z_c} = -\sum_{i=1}^{C} y_i \frac{\partial \log\left(\frac{e^{z_i}}{Z}\right)}{\partial z_c} \tag{4}$$

    when calculating the gradient. In order to calculate the derivative of this summation, we have to consider when $i = c$ and when $i \neq c$ and add them together

    $$\left.\frac{\partial Loss(y, p)}{\partial z_c}\right|_{i=c} = -\frac{Zy_i}{e^{z_i}}\left(\frac{\partial}{\partial z_i}\frac{e^{z_i}}{Z}\right) = -\frac{Zy_i}{e^{z_i}}\left(\frac{e^{z_i}}{Z} \cdot \frac{Z - e^{z_i}}{Z}\right) \tag{5}$$

where
$$\frac{e^{z_i}}{Z} \cdot \frac{Z - e^{z_i}}{Z} = \frac{e^{z_i}}{Z} \cdot \left(\frac{Z}{Z} - \frac{e^{z_i}}{Z} \right) = p_i \left(1 - p_i\right) \tag{6}$$
so that
$$\frac{\partial Loss(y,p)}{\partial z_c}_{i=c} = -\frac{y_i}{p_i} p_i \left(1 - p_i\right) \tag{7}$$
additionally for $i \neq c$
$$\frac{\partial Loss(y,p)}{\partial z_c}_{i \neq c} = -\sum_{i \neq c} \frac{Z y_c}{e^{z_c}} \left(\frac{\partial}{\partial z_c} \frac{e^{z_i}}{Z} \right) = -\sum_{i \neq c} \frac{Z y_c}{e^{z_c}} \left(-\frac{e^{z_i}}{Z} \cdot \frac{e^{z_c}}{Z} \right) \tag{8}$$
where
$$-\frac{e^{z_i}}{Z} \cdot \frac{e^{z_c}}{Z} = -p_i p_c \tag{9}$$
so that
$$\frac{\partial Loss(y,p)}{\partial z_c}_{i \neq c} = -\sum_{i \neq c} \frac{y_c}{p_c} \left(-p_i p_c\right) \tag{10}$$

The difference between these gradient terms comes from the simple quotient rule. In the first, since $i = c$, then the derivative is always w.r.t. the current node, meaning that the first term of the quotient rule exists. Otherwise, when $i \neq c$, the first term of the quotient rule will be some node at position i w.r.t. to index c, which has no relation and thus is 0

so overall
$$\frac{\partial Loss(y,p)}{\partial z_c} = \frac{\partial Loss(y,p)}{\partial z_c}_{i=c} + \frac{\partial Loss(y,p)}{\partial z_c}_{i \neq c} \tag{11}$$
$$\frac{\partial Loss(y,p)}{\partial z_c} = -\frac{y_i}{p_i} p_i \left(1 - p_i\right) - \sum_{i \neq c} \frac{y_c}{p_c} \left(-p_i p_c\right) \tag{12}$$
$$\frac{\partial Loss(y,p)}{\partial z_c} = -y_i + y_i p_i + \sum_{i \neq c} y_c p_i \tag{13}$$
which ends up being
$$\frac{\partial Loss(y,p)}{\partial z_c} = p_i - y_i \tag{14}$$

2. Understand and derive the gradient of convolution. First consider the toy example, let X be a 3×3 data and W a 2×2 filter kernel

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}, W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

(a) Show the convolution output $Z = X \cdot W$ (i.e. forward propagation)

The kernel W is applied onto X and swept across all rows and columns. In this example, we will assume there is no padding. The result will be an element-wise multiplication and summation for all regions the kernel can be swept to. A general mathematical form is shown in (c), but here is the toy example result with no padding

$$
\begin{aligned}
z_{11} &= x_{11}w_{11} + x_{12}w_{12} + x_{21}w_{21} + x_{22}w_{22} \\
z_{12} &= x_{12}w_{11} + x_{13}w_{12} + x_{22}w_{21} + x_{23}w_{22} \\
z_{21} &= x_{21}w_{11} + x_{22}w_{12} + x_{31}w_{21} + x_{32}w_{22} \\
z_{22} &= x_{22}w_{11} + x_{23}w_{12} + x_{32}w_{21} + x_{33}w_{22}
\end{aligned} \tag{15}
$$

and

$$Z = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}$$

In order to use padding, it would be more effective to create an X_{pad}

$$X_{\text{pad}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & x_{11} & x_{12} & x_{13} & 0 \\ 0 & x_{21} & x_{22} & x_{23} & 0 \\ 0 & x_{31} & x_{32} & x_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where the number of padded zeros is one less than the the largest dimension of kernel W, and Z is redefined as $Z = W \cdot X_{\text{pad}}$

(b) Show the gradient $\frac{\partial Z}{\partial W}$ (i.e. back propagation)

We can put each index of Z into vector form by defining a vectorized W as well as a vectorized subset of X. For example, z_{11}:

$$z_{11} = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{12} \\ x_{22} \end{bmatrix}^\top \begin{bmatrix} w_{11} \\ w_{21} \\ w_{12} \\ w_{22} \end{bmatrix} = vec(X_{[1,2;1,2]})^\top vec(W) \tag{16}$$

$$\frac{\partial z_{kl}}{\partial w_{ij}} = \sum_{k,l} 1_{i=k,j=l} \times x_{(i+k-1)(j+l-1)} \tag{17}$$

And this occurs for all k, l of Z.

(c) Now consider the general case: let $X_{m \times n}$ be an $m \times n$ matrix and let W be a $k \times k$ kernel, define the convolution output $Z = X \cdot W$ and derive the gradient $\frac{\partial Z}{\partial W}$

A more general form of the output Z can be calculated using what we found above in (15), where for all rows and columns of Z we calculate

$$z_{ij} = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w_{st} x_{(i+s)(j+t)} \tag{18}$$

where

$$a = b = \left\lfloor \frac{k}{2} \right\rfloor$$

as for the gradient $\frac{\partial Z}{\partial W}$, we can use the same notation used in (17) $\forall$ dimensions in Z. Again, that result would be

$$\frac{\partial z_{kl}}{\partial w_{ij}} = \sum_{k,l} 1_{i=k,j=l} \times x_{(i+k-1)(j+l-1)} \tag{19}$$

This notation is unlike I have seen in other references, but essentially what (19) is saying is the final gradient $\frac{\partial Z}{\partial W} \in \mathbb{R}^{k \times l}$ and only the elements of X remain for whichever node of W the gradient is w.r.t.