Calculation of derivatives of cost function for Convolutional Neural Networks

Alex Grishin

September 23, 2021

This exercise was driven by pure curiosity rather than practical need. I was curious whether I can take a relatively simple CNN, calculate partial derivatives of cost function with respect to all learnable parameters analytically by hand, and then program it without using any ML framework or autograd library. I did exactly that, trained an image classifier, and got the same accuracy as in TensorFlow. I'm publishing the calculations here in case if anybody, for whatever reason, would be interested to follow.

Contents

Part I
Layout

For the calculations we will be using a specific example of a Convolutional Neural Network. The example we intend to use is generic enough, and those who can follow the calculations presented in this article will have no difficulties in performing similar calculations for any Convolutional Neural Network.

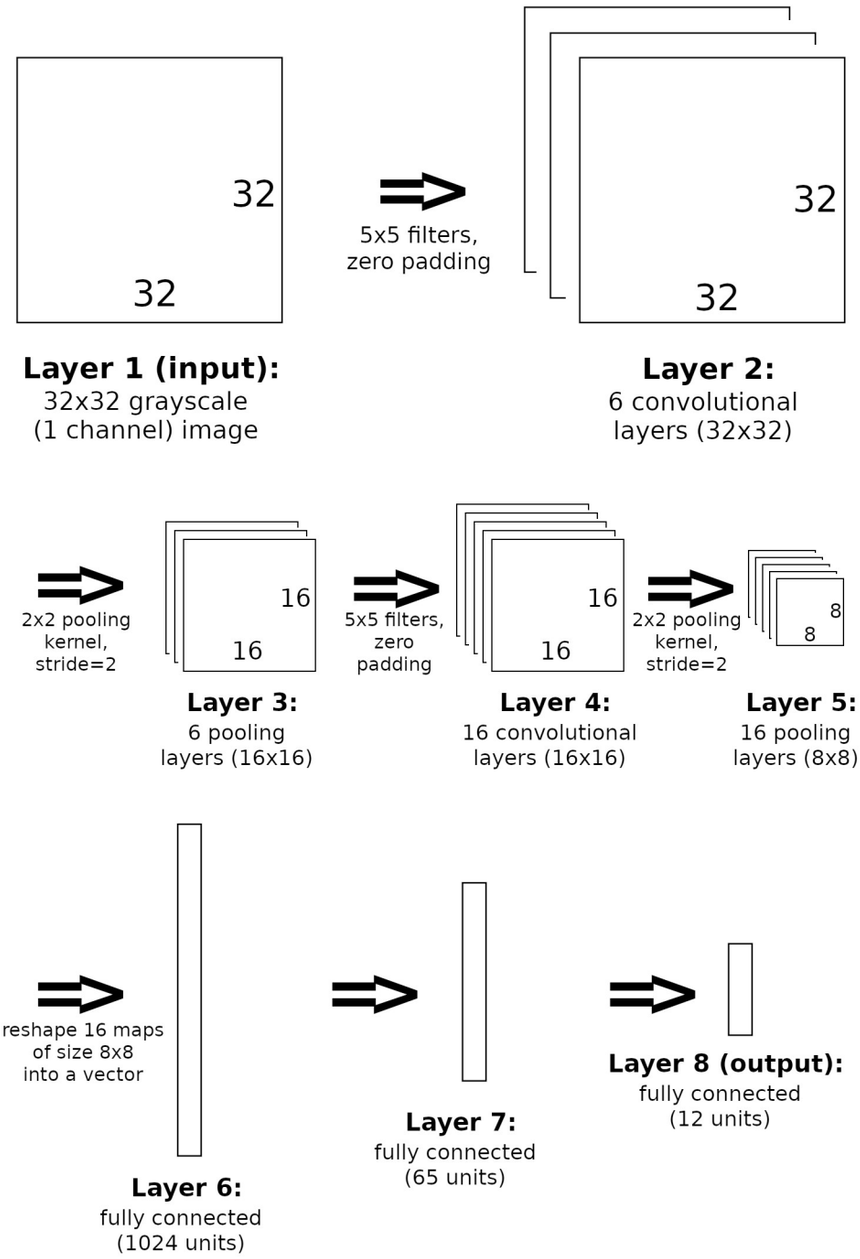

Figure 1: CNN example used for calculations in this tutorial. It consists of eight layers with the first layer being an input 32x32 grayscale (one channel) image, and the last layer being an output classifier layer (in our example the CNN distinguishes beigtween 12 different classes)

Fig 1. shows the architecture of our example CNN. Note that we use specific numbers literally everywhere. We say that the second layer is of 32x32 size and has 6 maps (instead of specifying its size as $n2$x$m2$ and number of maps as $d2$), filters are 5x5 (instead of fxf), seventh layer has 65 units (instead of $n7$), etc... Although in a proper scientific paper this ostensibly unwarranted de-generalization would be (very much) frowned upon, we believe that use of specific numbers in this tutorial makes following the calculations simpler while not actually sacrificing anything. The reader who followed the derivations will easily generalize the formulae in a matter of minutes.

Layer 1 The input layer is a 32x32 grayscale image (and as such consists of one map).

Layer 2 By applying six 5x5 filters to the input layer we get six maps in the second convolutional layer. We are using stride equal to 1 and zero padding of width 2 around the original image, and therefore the convolutional maps in the second layer are also of size 32x32. Our choice of activation function is ELU (Exponential Linear Unit).

Layer 3 Then we do 2x2 subsampling with stride equal to 2 and max as pooling function, which brings us to the third pooling layer with six 16x16 maps.

Layer 4 At the next step we apply sixteen 5x5x6 filters (those filters have depth equal to 6 as the previous layer consists of 6 maps). Again we use zero padding of width 2, stride equal to 1, and ELU as activation. As the result we get sixteen maps of size 16x16 in the fourth convolutional layer.

Layer 5 Then again we do the same subsampling as at Layer 2 $\Rightarrow$ Layer 3 step and end up with sixteen 8x8 maps in the fifth pooling layer.

Layer 6 Layer 6 is just a reshape of Layer 5 - instead of 3D matrix of size 8x8x16 we have the same units in a vector of length 1024. Layer 6 is introduced merely for notational convenience, it is exactly the same (apart from its shape) as Layer 5.

Layer 7 Layers 6 and 7 are fully connected, Layer 7 has 65 units, activation function is ELU.

Layer 8 Layers 7 and 8 are fully connected, Layer 8 has 12 units (it is the output layer and classifies the input images into 12 categories), activation function is sigmoid.

Part II
Forward pass

1 Layer 1 to Layer 2

Transition from Layer 1 to Layer 2 is governed by six 5x5 filters, and six bias terms (one per each map). Before applying filters to the input layer we need to prepare it by adding zero padding of width 2. Let $a^{(1)}_{u_1,v_1}$ be input values (here the superscript (1) refers to the first layer, indices u_1 and v_1 run from 1 to 32). After padding is added we have a 36x36 matrix (let's call it $aa^{(1)}$) with non-zero values only in rows and columns from 3 to 34. In pseudo code it would look like that:

$$aa^{(1)} = zeros(36, 36);$$

$$aa^{(1)}(3:34, 3:34) = a^{(1)}(:, :);$$

Applying filters to the padded 36x36 matrix $aa^{(1)}$ gives us six maps in the convolutional Layer 2:

$$z^{(2)k_2}_{u_2,v_2} = b^{(1)k_2} + \sum_{o_1,e_1=1}^{5} w^{(1)k_2}_{o_1,e_1} aa^{(1)}_{u_2+o_1-1,v_2+e_1-1};$$

Here k_2 is a map index, which is a number from 1 to 6, u_2 and v_2 are horizontal indices, which run from 1 to 32. For each map k_2, filter $w^{(1)k_2}_{o_1,e_1}$ is a 5x5 matrix (indices o_1 and e_1 run from 1 to 5). $b^{(1)k_2}$ is a bias term for map k_2. Indices of $aa^{(1)}$ (which are equal to $u_2 + o_1 - 1$ and $v_2 + e_1 - 1$) can take values from 1 to 36. If for a particular map the bias term is zero, and the 5x5 filter $w^{(1)k_2}_{o_1,e_1}$ has zero values everywhere apart from the center (position (3,3)), where it is equal to 1, then the transformation above is the identity transformation for this map.

After using filters and getting values of $z^{(2)k_2}_{u_2,v_2}$ we need to apply the activation function:

$$a^{(2)k_2}_{u_2,v_2} = ELU(z^{(2)k_2}_{u_2,v_2});$$

Note that in this tutorial when calculating values of units (neurons) $a^{(n+1)}$ for the layer $n+1$ based on values $a^{(n)}$ in the layer n, we use notation $z^{(n+1)}$ for the result before application of activation function and notation $a^{(n+1)}$ for the final result for the layer $n+1$ after activation function. If for a transition *Layer* $n \Rightarrow$ *Layer* $n+1$ we do not need activation function, then we only use notation $a^{(n+1)}$. In other words $z^{(n+1)}$ is always an intermediate result (if there is such), $a^{(n+1)}$ is always the final result (activation) for the layer $n+1$.

As our activation function we use ELU (Exponential Linear Unit), which is defined as:

$$ELU(z) = \begin{cases} z & for\ z \geq 0; \\ e^z - 1 & for\ z < 0; \end{cases}$$

2 Layer 2 to Layer 3

When transiting from Layer 2 to Layer 3 we just do subsampling (pooling). We substitute 2x2 squares (4 values) in Layer 2 with one value in Layer 3 thus reducing the dimensions from 32x32 to 16x16. Number of maps, of course, stays the same. Our choice of pooling function is max. Only the maximum value from each 2x2 square in Layer 2 makes it to Layer 3:

$$a^{(3)k_3}_{u_3,v_3} = max\left\{a^{(2)k_3}_{2u_3-1,2v_3-1};\ a^{(2)k_3}_{2u_3-1,2v_3};\ a^{(2)k_3}_{2u_3,2v_3-1};\ a^{(2)k_3}_{2u_3,2v_3}\right\};$$

Here indices u_3 and v_3 run from 1 to 16. Index k_3 numerates maps and changes from 1 to 6.

When doing Layer 2 to Layer 3 forward pass it is a good idea to also get a position of maximum within each 2x2 pooling area. Let's call this position matrix $p^{(3)k_3}_{u_3,v_3}$ where values of it can be 1,2,3 or 4. We will be needing this $p^{(3)}$ matrix later for the Layer 3 $\rightarrow$ Layer 2 back propagation step.

3 Layer 3 to Layer 4

Transition from Layer 3 to Layer 4 is very similar to that between Layers 1 and 2. First we need to add zero padding to all six 16x16 maps in Layer 3. For this we introduce an auxiliary 3D structure $aa^{(3)}$, whose horizontal indices run from 1 to 20 with non-zero values being only in rows 3 to 18, and columns 3 to 18:

$$aa^{(3)} = zeros(20, 20, 6);$$

$$aa^{(3)}(3:18, 3:18, :) = a^{(3)}(:, :, :);$$

Then we apply filters:

$$z^{(4)k_4}_{u_4,v_4} = b^{(3)k_4} + \sum_{k_3=1}^{6} \sum_{o_3,e_3=1}^{5} w^{(3)k_4,k_3}_{o_3,e_3} aa^{(3)k_3}_{u_4+o_3-1,v_4+e_3-1};$$

The upper index k_4 runs from 1 to 16 (16 being number of maps in Layer 4), the horizontal indices u_4 and v_4 run from 1 to 16 (as each map is of 16x16 size).

The difference between the formula above and the one for Layer 1 $\Rightarrow$ Layer 2 transition is that the layer to which filters are applied is now itself a multi-map layer and therefore the filters $w^{(3)k_4,k_3}_{o_3,e_3}$ have four indices instead of three. In other words we have 16 filters of 5x5x6 size, and each of those filters is applied to a 5x5 horizontal square in all six maps in Layer 3 (thus additional summation over k_3).

The last step for this transition is application of activation function:

$$a^{(4)k_4}_{u_4,v_4} = ELU(z^{(4)k_4}_{u_4,v_4});$$

4 Layer 4 to Layer 5

This pooling transition is fully analogous to Layer 2 $\Rightarrow$ Layer 2 transition:

$$a^{(5)k_5}_{u_5,v_5} = max\left\{a^{(4)k_5}_{2u_5-1,2v_5-1};\ a^{(4)k_5}_{2u_5-1,2v_5};\ a^{(4)k_5}_{2u_5,2v_5-1};\ a^{(4)k_5}_{2u_5,2v_5}\right\};$$

Index k_5 runs from 1 to 16, indices u_5 and v_5 from 1 to 8.

Again, for later use at this step we should get a position matrix $p^{(5)k_5}_{u_5,v_5}$ with 1,2,3 or 4 values which indicate the position of the maximum within each 2x2 pooling area.

5 Layer 5 to Layer 6

As we said above, we introduce Layer 6 as a separate entity for notational convenience only. In reality Layer 6 is the same thing as Layer 5. As the next layers (Layer 7 and Layer 8) are vector-like strings of neurons, and as we need to fully connect our Layer 5 to Layer 7, it is convenient to first reshape Layer 5 from 3D matrix of size 8x8x16 into a vector of length 1024 (and call the result of this procedure Layer 6):

$$a^{(6)}_{s_6} = a^{(5)k_5}_{u_5,v_5};$$

The index s_6 runs from 1 to 1024. Here are the rules to convert the combination of u_5, v_5, k_5 into s_6 and back:

$$s_6 = 64(k_5 - 1) + 8(v_5 - 1) + u_5;$$

$$\begin{cases} k_5 = ceil\left(s_6/64\right); \\ v_5 = ceil\left(\left[s_6 - 64(k_5)\right]/8\right]; \\ u_5 = s_6 - 64(k_5 - 1) - 8(v_5 - 1); \end{cases}$$

In real life the formulae above are likely to not be used. Your programming language will have a reshape (or whatever it's called in your language) function, which will take care of the needed conversions.

6 Layer 6 to Layer 7

This transition is a straightforward linear transformation followed by the application of activation function:

$$z_{s_7}^{(7)} = b_{s_7}^{(6)} + \sum_{s_6=1}^{1024} \Theta_{s_7,s_6}^{(6)} a_{s_6}^{(6)};$$

$$a_{s_7}^{(7)} = ELU(z_{s_7}^{(7)});$$

Vector $b^{(6)}$ of length 65, and matrix $\Theta^{(6)}$ of size 65x1024 are parameters of the transformation.

7 Layer 7 to Layer 8

The same as the previous one, with the only difference that activation function is now sigmoid: $g(z) = 1/(1 + e^{-z})$.

$$z_{s_8}^{(8)} = b_{s_8}^{(7)} + \sum_{s_7=1}^{65} \Theta_{s_8,s_7}^{(7)} a_{s_7}^{(7)};$$

$$a_{s_8}^{(8)} = g(z_{s_8}^{(8)});$$

Vector $b^{(7)}$ has length 12, and matrix $\Theta^{(7)}$ has size 12x65.

Part III
Back propagation

8 Learnable parameters

Transitions Layer 2 ⇒ Layer 3, Layer 4 ⇒ Layer 5, and Layer 5 ⇒ Layer 6 do not have any parameters. The other four transitions have learnable parameters. Here is the table with their dimensions and sizes:

Learnable parameter	**Dimensions**	**Size**
$b^{(1)k_2}$	1	6 $(k_2 = 1:6)$
$w^{(1)k_2}_{o_1,e_1}$	3	5x5x6 $(o_1 = 1:5, e_1 = 1:5, k_2 = 1:6)$
$b^{(3)k_4}$	1	16 $(k_4 = 1:16)$
$w^{(3)k_4,k_3}_{o_3,e_3}$	4	5x5x16x6 $(o_3 = 1:5, e_3 = 1:5, k_4 = 1:16, k_3 = 1:6)$
$b^{(6)}_{s_7}$	1	65 $(s_7 = 1:65)$
$\Theta^{(6)}_{s_7,s_6}$	2	65x1024 $(s_7 = 1:65, s_6 = 1:1024)$
$b^{(7)}_{s_8}$	1	12 $(s_8 = 1:12)$
$\Theta^{(7)}_{s_8,s_7}$	2	12x65 $(s_8 = 1:12, s_7 = 1:65)$

The task of training the network is to get the values of the parameters in the table, which would produce as low value of the cost function as possible. To achieve that one needs to feed partial derivatives of the cost function with respect to all of the above parameters to whatever flavor of gradient descent procedure he or she uses. We, therefore, need to get those partial derivatives calculated. This is what we will be doing for the rest of this tutorial. What we did up to this point had introductory meaning, we simply described our CNN and introduced the notation. From this point on we mean business.

9 Starting with the last layer

Let x be a parameter with respect to which we want to calculate a partial derivative of the cost function J. Here x could be an element of the 4D structure $w^{(3)k_4,k_3}_{o_3,e_3}$, or an element of $\Theta^{(6)}_{s_7,s_6}$, or of $b^{(1)k_2}$, or anything else from the table above, we do not want to specify it for the time being.

Let's start. For convenience we will first do all the calculations for one training example, and at the end we will (trivially) generalize our formulae for a batch. Our choice of cost function is

log loss:

$$J = -\sum_{s_8=1}^{12} \left[y_{s_8} ln\left(a_{s_8}^{(8)}\right) + (1 - y_{s_8})\, ln\left(1 - a_{s_8}^{(8)}\right) \right];$$

Here vector y is a known outcome for our training example. If we have a mutually excluding classifier, and our training example belongs to one of the 12 classes, then y would have 11 zero components, and one component equal to 1. The output layer values $a_{s_8}^{(8)}$ depend on all learnable parameters from the table, with the dependancy set by the forward pass procedure. Vector y, of course, doesn't depend on anything, it's given, and therefore not touched by the partial derivative:

$$\begin{aligned}
\frac{\partial}{\partial x} J &= -\sum_{s_8=1}^{12} \left[y_{s_8} \frac{\partial}{\partial x} ln\left(a_{s_8}^{(8)}\right) + (1 - y_{s_8}) \frac{\partial}{\partial x} ln\left(1 - a_{s_8}^{(8)}\right) \right] \\
&= -\sum_{s_8=1}^{12} \left(\frac{y_{s_8}}{a_{s_8}^{(8)}} - \frac{1 - y_{s_8}}{1 - a_{s_8}^{(8)}} \right) \frac{\partial}{\partial x} a_{s_8}^{(8)} \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8}\left(1 - a_{s_8}^{(8)}\right) - a_{s_8}^{(8)}\left(1 - y_{s_8}\right)}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \frac{\partial}{\partial x} g\left(z_{s_8}^{(8)}\right) \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8} - a_{s_8}^{(8)}}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \frac{\partial}{\partial x} \frac{1}{1 + e^{-z_{s_8}^{(8)}}} \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8} - a_{s_8}^{(8)}}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \frac{e^{-z_{s_8}^{(8)}}}{\left(1 + e^{-z_{s_8}^{(8)}}\right)^2} \frac{\partial}{\partial x} z_{s_8}^{(8)} \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8} - a_{s_8}^{(8)}}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \frac{1 + e^{-z_{s_8}^{(8)}} - 1}{\left(1 + e^{-z_{s_8}^{(8)}}\right)^2} \frac{\partial}{\partial x} z_{s_8}^{(8)} \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8} - a_{s_8}^{(8)}}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \left[\frac{1}{1 + e^{-z_{s_8}^{(8)}}} - \frac{1}{\left(1 + e^{-z_{s_8}^{(8)}}\right)^2} \right] \frac{\partial}{\partial x} z_{s_8}^{(8)} \\
&= -\sum_{s_8=1}^{12} \frac{y_{s_8} - a_{s_8}^{(8)}}{a_{s_8}^{(8)}\left(1 - a_{s_8}^{(8)}\right)} \left[a_{s_8}^{(8)} - a_{s_8}^{(8)^2} \right] \frac{\partial}{\partial x} z_{s_8}^{(8)} \\
&= \sum_{s_8=1}^{12} \left(a_{s_8}^{(8)} - y_{s_8} \right) \frac{\partial}{\partial x} z_{s_8}^{(8)};
\end{aligned}$$

Now let's introduce a vector $t^{(8)}$:

$$t^{(8)} = a^{(8)} - y;$$

In terms of $t^{(8)}$ the partial derivative equation will be written as:

$$\frac{\partial}{\partial x} J = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \frac{\partial}{\partial x} z_{s_8}^{(8)};$$

$z^{(8)}$ depends on learnable parameters, which govern transition from Layer 7 to Layer 8, and it also depends on activations $a^{(7)}$ of the seventh layer. How we approach the partial derivative $\partial z^{(8)} / \partial x$ depends on whether x is a Layer 7 $\Rightarrow$ Layer 8 parameter or anything else. In the former case the calculation is trivial and stops here:

$$\frac{\partial}{\partial b_i^{(7)}} J = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \frac{\partial}{\partial b_i^{(7)}} z_{s_8}^{(8)} = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \delta_{i,s_8} = t_i^{(8)};$$

$$\frac{\partial}{\partial \Theta_{i,j}^{(7)}} J = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \frac{\partial}{\partial \Theta_{i,j}^{(7)}} z_{s_8}^{(8)} = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \sum_{s_7=1}^{65} a_{s_7}^{(7)} \delta_{i,s_8} \delta_{j,s_7} = t_i^{(8)} a_j^{(7)};$$

In the latter case (when x is a parameter from one of the previous layers) we need to go deeper. In the next sections (named like Layer 8⇒Layer 7 or Layer 5⇒Layer 4 or similar) we will pretend that x is always the deepest parameter (a parameter with the upper index (1)). Later, when we need a formula for let's say $\partial J/\partial w_{s,s'}^{(3)k,k'}$ we will simply go to the section Layer 4⇒Layer 3, start with the formula there and get $\partial J/\partial w_{s,s'}^{(3)k,k'}$ in the same trivial manner we've just done.

10 Layer 8 ⇒ Layer 7

We know that

$$\frac{\partial}{\partial x}J = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \frac{\partial}{\partial x} z_{s_8}^{(8)};$$

and, as agreed upon, we pretend that x is the deepest parameter. In this case $z^{(8)}$ depends on x only through its dependance on activations $a^{(7)}$:

$$\frac{\partial}{\partial x}J = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \sum_{s_7=1}^{65} \Theta_{s_8,s_7}^{(7)} \frac{\partial}{\partial x} a_{s_7}^{(7)};$$

Here $\Theta^{(7)}$ is a matrix. $t^{(8)}$ is a vector, and for notational convenience let's declare that it is a row vector (actually, we want $a^{(8)}$, $z^{(8)}$, y, $a^{(7)}$, $z^{(7)}$, $a^{(6)}$ all to be row vectors). Let's introduce another row vector $\mu^{(7)}$, which we define as matrix multiplication of row vector $t^{(8)}$ by matrix $\Theta^{(7)}$:

$$\mu^{(7)} = t^{(8)} * \Theta^{(7)};$$

In index notation that would read:

$$\mu_{s_7}^{(7)} = \sum_{s_8=1}^{12} t_{s_8}^{(8)} \Theta_{s_8,s_7}^{(7)};$$

And the formula for partial derivative takes the form:

$$\frac{\partial}{\partial x}J = \sum_{s_7=1}^{65} \mu_{s_7}^{(7)} \frac{\partial}{\partial x} a_{s_7}^{(7)} = \sum_{s_7=1}^{65} \mu_{s_7}^{(7)} \frac{\partial}{\partial x} ELU\left(z_{s_7}^{(7)}\right) = \sum_{s_7=1}^{65} \mu_{s_7}^{(7)} ELUD\left(z_{s_7}^{(7)}\right) \frac{\partial}{\partial x} z_{s_7}^{(7)};$$

Where we introduced a new function $ELUD$, which is a derivative of ELU:

$$ELUD(z) = \begin{cases} 1 & for\ z \geq 0; \\ e^z & for\ z < 0; \end{cases}$$

Let now a row vector $t^{(7)}$ be the result of an element wise multiplication of row vector $\mu^{(7)}$ and row vector $ELUD\left(z^{(7)}\right)$:

$$t^{(7)} = \mu^{(7)} .* ELUD\left(z^{(7)}\right);$$

The partial derivative formula is then simplified to:

$$\frac{\partial}{\partial x}J = \sum_{s_7=1}^{65} t_{s_7}^{(7)} \frac{\partial}{\partial x} z_{s_7}^{(7)};$$

11 Layer 7 ⇒ Layer 6

Substituting the forward pass expression for $z_{s_7}^{(7)}$ in the above formula, we get:

$$\frac{\partial}{\partial x}J = \sum_{s_7=1}^{65} t_{s_7}^{(7)} \sum_{s_6=1}^{1024} \Theta_{s_7,s_6}^{(6)} \frac{\partial}{\partial x} a_{s_6}^{(6)} = \sum_{s_6=1}^{1024} t_{s_6}^{(6)} \frac{\partial}{\partial x} a_{s_6}^{(6)};$$

where we introduced a row vector $t^{(6)}$, which is a matrix product of row vector $t^{(7)}$ by matrix $\Theta^{(6)}$:

$$t^{(6)} = t^{(7)} * \Theta^{(6)};$$

12 Layer 6 ⇒ Layer 5

As mentioned (and not once) above, Layer 5 and Layer 6 are the same. The only difference is in shape: $a^{(6)}$ is a vector, while $a^{(5)}$ is a 3D structure, but they are equal to each other component by component: $a^{(6)}_{s_6} = a^{(5)k_5}_{u_5,v_5}$. As we have one to one correspondence between s_6 and the combination of u_5, v_5, k_5 (see section Layer 5 to Layer 6 in Forward pass), instead of summation over s_6 we can do summation over three indices u_5, v_5, k_5 in our partial derivative formula:

$$\frac{\partial}{\partial x}J = \sum_{k_5=1}^{16}\sum_{u_5=1}^{8}\sum_{v_5=1}^{8} t^{(6)}_{s_6(u_5,v_5,k_5)}\frac{\partial}{\partial x}a^{(5)k_5}_{u_5,v_5};$$

where s_6 (the index of $t^{(6)}$) is treated as a function of u_5, v_5, k_5:

$$s_6(u_5, v_5, k_5) = 64(k_5 - 1) + 8(v_5 - 1) + u_5;$$

Let's introduce a new 3D structure $t^{(5)k_5}_{u_5,v_5}$:

$$t^{(5)k_5}_{u_5,v_5} = t^{(6)}_{s_6(u_5,v_5,k_5)};$$

$t^{(5)}$ is a reshape of the vector $t^{(6)}$ of length 1024 into a 8x8x16 3D structure. In pseudo code :

$$t^{(5)} = reshape(t^{(6)} \Rightarrow 8 \times 8 \times 16);$$

The partial derivative formula now takes the form:

$$\frac{\partial}{\partial x}J = \sum_{k_5=1}^{16}\sum_{u_5=1}^{8}\sum_{v_5=1}^{8} t^{(5)k_5}_{u_5,v_5}\frac{\partial}{\partial x}a^{(5)k_5}_{u_5,v_5};$$

13 Layer 5 ⇒ Layer 4

When we do a forward pass from Layer 4 to Layer 5 we calculate the value of $a^{(5)k_5}_{u_5,v_5}$ by taking the maximum from the four corresponding values in Layer 4. Here is our formula from Forward pass part:

$$a^{(5)k_5}_{u_5,v_5} = max\left\{a^{(4)k_5}_{2u_5-1,2v_5-1};\ a^{(4)k_5}_{2u_5-1,2v_5};\ a^{(4)k_5}_{2u_5,2v_5-1};\ a^{(4)k_5}_{2u_5,2v_5}\right\};$$

We can always say which indices u_5 and v_5 from Layer 5 correspond to the particular pair u_4, v_4 in Layer 4, but not vice versa:

$$u_5 = ceil(u_4/2);$$

$$v_5 = ceil(v_4/2);$$

u_4 could be $(2u_5 - 1)$ or it could be $2u_5$;

v_4 could be $(2v_5 - 1)$ or it could be $2v_5$;

Let's introduce a new 3D structure $tt^{(4)}$ using the following rule:

$$tt^{(4)k_4}_{u_4,v_4} = \begin{cases} t^{(5)k_4}_{ceil(u_4/2),ceil(v_4/2)} & \text{if this pair } u_4 \text{ and } v_4 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$

With $tt^{(4)}$ we can re-write the partial derivative formula in the following form:

$$\frac{\partial}{\partial x}J = \sum_{k_4=1}^{16}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} tt^{(4)k_4}_{u_4,v_4}\frac{\partial}{\partial x}a^{(4)k_4}_{u_4,v_4};$$

Only one quarter of $tt^{(4)}$ components are not zero, which ensures that the above summation over 4 times larger space of indices (u_4 and v_4 run from 1 to 16, while u_5 and v_5 ran from 1 to 8) produces exactly the same result as the previous version at the end of Layer 6 ⇒ Layer 5 section.

To produce $tt^{(4)}$ we suggest to first build an auxiliary structure $ttt^{(4)}$:

$$ttt^{(4)k_4}_{u_4,v_4} = t^{(5)k_4}_{ceil(u_4/2),ceil(v_4/2)};$$

and then set "wrong" components to zero:

$$tt^{(4)} = ttt^{(4)}. * p4;$$

where $.*$ stands for an element wise multiplication, and 3D structure $p4$ is defined as:

$$p4^{k_4}_{u_4,v_4} = \begin{cases} 1 & \text{if this pair } u_4 \text{ and } v_4 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$

When writing a machine learning code it is always preferable to avoid loops as much as possible and use vectorized representation instead. Vectorized code can be effectively parallelized when run on GPUs, while loops cannot. Granted, loops (when all loop steps are independent of each other) can be run in parallel on multiple CPUs, but 1) GPU parallelization is generally much better than that on CPU 2) CPU parallelization works ok on the very external loops (e.g. when you loop over different initializations or over different values of your hyper parameters), not in the middle of back propagation procedure. Rationale - avoid loops if you can.

Well, for calculating $tt^{(4)}$ we can't do that and here we are forced to use loops. Below is our pseudo code. Note that $p^{(5)k_5}_{u_5,v_5}$ is a 8x8x16 position matrix with values 1,2,3 or 4 which indicate the position of maximum within each 2x2 pooling area. This $p^{(5)}$ matrix was supposed to be calculated earlier at the forward pass step.

$ttt4 = zeros(16, 16, 16);$
$tt4 = zeros(16, 16, 16);$
$p4 = zeros(16, 16, 16);$

$p5aux = zeros(1, 16);$
$p4aux1 = zeros(1, 16);$
$p4aux2 = zeros(1, 16);$
$p4aux3 = zeros(1, 16);$
$p4aux4 = zeros(1, 16);$

$for\ i = 1 : 8$
$for\ j = 1 : 8$

$ttt4(2 * i - 1, 2 * j - 1, :) = t5(i, j, :);$
$ttt4(2 * i - 1, 2 * j, :) = t5(i, j, :);$
$ttt4(2 * i, 2 * j - 1, :) = t5(i, j, :);$
$ttt4(2 * i, 2 * j, :) = t5(i, j, :);$

$p5aux(1, :) = uint8(p5(i, j, :));$

$p4aux1(1, :) = p5aux(1, :);$
$p4aux2(1, :) = p5aux(1, :)/2;$
$p4aux3(1, :) = p5aux(1, :)/3;$
$p4aux4(1, :) = p5aux(1, :)/4;$

$p4aux1(p4aux1 \sim= 1) = 0;$
$p4aux2(p4aux2 \sim= 1) = 0;$
$p4aux3(p4aux3 \sim= 1) = 0;$
$p4aux4(p4aux4 \sim= 1) = 0;$

$p4(2 * i - 1, 2 * j - 1, :) = p4aux1(1, :);$
$p4(2 * i - 1, 2 * j, :) = p4aux2(1, :);$
$p4(2 * i, 2 * j - 1, :) = p4aux3(1, :);$
$p4(2 * i, 2 * j, :) = p4aux4(1, :);$

end

end

$tt4 = ttt4. * p4;$

Now we recall that $a^{(4)}$ is the result of application of our activation function, and proceed further:

$$\frac{\partial}{\partial x} J = \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} tt_{u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} ELU\left(z_{u_4,v_4}^{(4)k_4}\right) = \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} tt_{u_4,v_4}^{(4)k_4} ELUD\left(z_{u_4,v_4}^{(4)k_4}\right) \frac{\partial}{\partial x} z_{u_4,v_4}^{(4)k_4};$$

Then we introduce a new structure $t^{(4)}$:

$t^{(4)} = tt^{(4)}. * ELUD(z^{(4)});$

and re-write the partial derivative formula:

$$\frac{\partial}{\partial x} J = \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} t_{u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} z_{u_4,v_4}^{(4)k_4};$$

14 Layer 4 ⇒ Layer 3

We substitute into our last version of the partial derivative formula the expression for $z_{u_4,v_4}^{(4)k_4}$ from the Forward pass part:

$$\frac{\partial}{\partial x} J = \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} \sum_{k_3=1}^{6} \sum_{o_3=1}^{5} \sum_{e_3=1}^{5} t_{u_4,v_4}^{(4)k_4} w_{o_3,e_3}^{(3)k_4,k_3} \frac{\partial}{\partial x} aa_{u_4+o_3-1,v_4+e_3-1}^{(3)k_3};$$

Six summation symbols above can be arranged in any order as limits of summations are just numbers and independent of each other. Let's do a change of variables:

$u_3 = u_4 + o_3 - 1;$
$v_3 = v_4 + e_3 - 1;$

which transforms sums over u_4 and v_4 within fixed limits to sums over u_3 and v_3 now running from o_3 to $15 + o_3$ and from e_3 to $15 + e_3$ respectively. Those summations over u_3 and v_3 are dependant on o_3 and e_3 and thus must be internal with respect to summations over o_3 and e_3:

$$\frac{\partial}{\partial x} J = \sum_{k_3=1}^{6} \sum_{k_4=1}^{16} \sum_{o_3=1}^{5} \sum_{e_3=1}^{5} \sum_{u_3=o_3}^{15+o_3} \sum_{v_3=e_3}^{15+e_3} t_{u_3-o_3+1,v_3-e_3+1}^{(4)k_4} w_{o_3,e_3}^{(3)k_4,k_3} \frac{\partial}{\partial x} aa_{u_3,v_3}^{(3)k_3};$$

Horizontal indices of $t^{(4)}$ change from 1 to 16. Let's introduce its padded version $\gamma^{(4)}$ with zero padding of width 4. Horizontal indices of $\gamma^{(4)}$ will run from 1 to 24 and non-zero values will only be in rows and columns 5 to 20:

$\gamma^{(4)} = zeros(24, 24, 16);$
$\gamma^{(4)}(5 : 20, 5 : 20, :) = t^{(4)}(:, :, :);$

Summations over u_3 and v_3 restrict those indices to the region where $t^{(4)}$ equal to $\gamma^{(4)}$ with horizontal indices shifted by 4. We can therefore re-write the partial derivative formula as:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{k_4=1}^{16}\sum_{o_3=1}^{5}\sum_{e_3=1}^{5}\sum_{u_3=o_3}^{15+o_3}\sum_{v_3=e_3}^{15+e_3}\gamma^{(4)k_4}_{u_3-o_3+5,v_3-e_3+5}w^{(3)k_4,k_3}_{o_3,e_3}\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3,v_3};$$

note that the horizontal indices of $\gamma^{(4)}$ are $u_3 - o_3 + 5$ and $v_3 - e_3 + 5$ (as compared to $u_3 - o_3 + 1$ and $v_3 - e_3 + 1$ for $t^{(4)}$).

Now let's have a close look at the summation over v_3, which runs from e_3 to $15 + e_3$. First we make a trivial statement that

$$\sum_{v_3=e_3}^{15+e_3}... = \sum_{v_3=1}^{20}... \; - \; \sum_{v_3=1}^{e_3-1}... \; - \; \sum_{v_3=16+e_3}^{20}... \; ;$$

and then examine the last two terms.

For all the terms (if there are any) in the summation from 1 to $e_3 - 1$ the inequality $v_3 < e_3$ holds true. If $v_3 < e_3$, then the second horizontal index $v_3 - e_3 + 5$ of $\gamma^{(4)}$ is less than 5, which means that for all those indices $\gamma^{(4)}$ is zero. Which in turn means that in our case the sum from 1 to $e_3 - 1$ is zero. Similarly, in the summation from $16 + e_3$ to 20 the inequality $v_3 > 15 + e_3$ holds true, which translates into $v_3 - e_3 + 5 > 20$, which ensures that all those components of $\gamma^{(4)}$ are zero, and as the result the sum from $16 + e_3$ to 20 is zero. Therefore, in our case it turns out that summation from e_3 to $15 + e_3$ is equivalent to summation from 1 to 20. Absolutely the same is true for the summation over u_3. The partial derivative formula then becomes:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{k_4=1}^{16}\sum_{o_3=1}^{5}\sum_{e_3=1}^{5}\sum_{u_3=1}^{20}\sum_{v_3=1}^{20}\gamma^{(4)k_4}_{u_3-o_3+5,v_3-e_3+5}w^{(3)k_4,k_3}_{o_3,e_3}\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3,v_3};$$

As all the summation limits are now independent of each other, we can re-arrange the summation symbols in any order we want. We want it this way:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{u_3=1}^{20}\sum_{v_3=1}^{20}\left(\sum_{k_4=1}^{16}\sum_{o_3=1}^{5}\sum_{e_3=1}^{5}\gamma^{(4)k_4}_{u_3-o_3+5,v_3-e_3+5}w^{(3)k_4,k_3}_{o_3,e_3}\right)\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3,v_3};$$

Now we introduce a new 3D (20x20x6) structure $\gamma^{(3)}$:

$$\gamma^{(3)k_3}_{u3,v3} = \sum_{k_4=1}^{16}\sum_{o_3=1}^{5}\sum_{e_3=1}^{5}\gamma^{(4)k_4}_{u_3-o_3+5,v_3-e_3+5}w^{(3)k_4,k_3}_{o_3,e_3};$$

and with its help re-write the partial derivative formula:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{u_3=1}^{20}\sum_{v_3=1}^{20}\gamma^{(3)k_3}_{u3,v3}\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3,v_3};$$

Now we recall (see section Layer 3 to Layer 4 in the Forward pass part) that $aa^{(3)k_3}_{u_3,v_3}$ is zero outside rows and columns from 3 to 18 (and so will be its derivative). Therefore we can shrink u_3 and v_3 summations to those limits:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{u_3=3}^{18}\sum_{v_3=3}^{18}\gamma^{(3)k_3}_{u3,v3}\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3,v_3};$$

which is fully equvalent to:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{u_3=1}^{16}\sum_{v_3=1}^{16}\gamma^{(3)k_3}_{u3+2,v3+2}\frac{\partial}{\partial x}aa^{(3)k_3}_{u_3+2,v_3+2};$$

Recalling that $aa^{(3)}(3:18, 3:18, :) = a^{(3)}(:, :, :)$ (again, see section Layer 3 to Layer 4 in the Forward pass part), we have:

$$\frac{\partial}{\partial x}J = \sum_{k_3=1}^{6}\sum_{u_3=1}^{16}\sum_{v_3=1}^{16}\gamma^{(3)k_3}_{u3+2,v3+2}\frac{\partial}{\partial x}a^{(3)k_3}_{u_3,v_3};$$

Finally, we introduce a structure $t^{(3)}$ defined as:

$$t^{(3)k_3}_{u_3,v_3} = \gamma^{(3)k_3}_{u3+2,v3+2};$$

and using this definition we re-write the partial derivative formula:

$$\frac{\partial}{\partial x} J = \sum_{k_3=1}^{6} \sum_{u_3=1}^{16} \sum_{v_3=1}^{16} t^{(3)k_3}_{u3,v3} \frac{\partial}{\partial x} a^{(3)k_3}_{u_3,v_3};$$

15 Layer 3 ⇒ Layer 2

This step is fully analogous to Layer 5 ⇒ Layer 4 step. First we calculate an auxiliary structure $tt^{(2)}$, which is defined as:

$$tt^{(2)k_2}_{u_2,v_2} = \begin{cases} t^{(3)k_2}_{ceil(u_2/2),ceil(v_2/2)} & \text{if this pair } u_2 \text{ and } v_2 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$

and produced by the following piece of pseudo code:

$ttt2 = zeros(32, 32, 6);$
$tt2 = zeros(32, 32, 6);$
$p2 = zeros(32, 32, 6);$

$p3aux = zeros(1, 6);$
$p2aux1 = zeros(1, 6);$
$p2aux2 = zeros(1, 6);$
$p2aux3 = zeros(1, 6);$
$p2aux4 = zeros(1, 6);$

$for\ i = 1 : 16$
$for\ j = 1 : 16$

$ttt2(2 * i - 1, 2 * j - 1, :) = t3(i, j, :);$
$ttt2(2 * i - 1, 2 * j, :) = t3(i, j, :);$
$ttt2(2 * i, 2 * j - 1, :) = t3(i, j, :);$
$ttt2(2 * i, 2 * j, :) = t3(i, j, :);$

$p3aux(1, :) = uint8(p3(i, j, :));$

$p2aux1(1, :) = p3aux(1, :);$
$p2aux2(1, :) = p3aux(1, :)/2;$
$p2aux3(1, :) = p3aux(1, :)/3;$
$p2aux4(1, :) = p3aux(1, :)/4;$

$p2aux1(p2aux1 \sim= 1) = 0;$
$p2aux2(p2aux2 \sim= 1) = 0;$
$p2aux3(p2aux3 \sim= 1) = 0;$
$p2aux4(p2aux4 \sim= 1) = 0;$

$p2(2*i-1,2*j-1,:) = p2aux1(1,:);$
$p2(2*i-1,2*j,:) = p2aux2(1,:);$
$p2(2*i,2*j-1,:) = p2aux3(1,:);$
$p2(2*i,2*j,:) = p2aux4(1,:);$

end

end

$tt2 = ttt2. * p2;$

With $tt^{(2)}$ the partial derivative formula takes the form:

$$\frac{\partial}{\partial x}J = \sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} tt_{u_2,v_2}^{(2)k_2}\frac{\partial}{\partial x}a_{u_2,v_2}^{(2)k_2};$$

or after substituting the activation function:

$$\frac{\partial}{\partial x}J = \sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} tt_{u_2,v_2}^{(2)k_2}\frac{\partial}{\partial x}ELU\left(z_{u_2,v_2}^{(2)k_2}\right) = \sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} tt_{u_2,v_2}^{(2)k_2}ELUD\left(z_{u_2,v_2}^{(2)k_2}\right)\frac{\partial}{\partial x}z_{u_2,v_2}^{(2)k_2};$$

We then introduce a new structure $t^{(2)}$:

$t^{(2)} = tt^{(2)}. * ELUD(z^{(2)});$

and re-write the partial derivative formula:

$$\frac{\partial}{\partial x}J = \sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} t_{u_2,v_2}^{(2)k_2}\frac{\partial}{\partial x}z_{u_2,v_2}^{(2)k_2};$$

16 Generalization for batch of training examples

What we did so far we did for the case of one training example. Now we need to generalize for a batch of examples of size N. Let's do it first for the forward pass.

16.1 Forward pass

The task is trivial to say the very least. All what we need to do is to add an extra index (let's call it e) to all structures $a^{(l)}$ and $z^{(l)}$ for all layers l to acknowledge the fact that those structures are different for different training examples. Thus row vectors become matrices with each row corresponding to one example (N rows altogether). 3D structures become 4D structures with the extra dimension index e numerating different examples. For all new structures with index e we will be using capital letters.

Layer 1 to Layer 2

$A_{e|u_1,v_1}^{(1)}$ - input image;
$AA^{(1)} = zeros(N,36,36);$
$AA^{(1)}(:,3:34,3:34) = A^{(1)}(:,:,:);$

$$Z_{e|u_2,v_2}^{(2)k_2} = b^{(1)k_2} + \sum_{o_1,e_1=1}^{5} w_{o_1,e_1}^{(1)k_2} AA_{e|u_2+o_1-1,v_2+e_1-1}^{(1)};$$

$$A_{e|u_2,v_2}^{(2)k_2} = ELU\left(Z_{e|u_2,v_2}^{(2)k_2}\right);$$

Layer 2 to Layer 3

$$\left[A_{e|u_3,v_3}^{(3)k_3}, P_{e|u_3,v_3}^{(3)k_3}\right] = max\left\{A_{e|2u_3-1,2v_3-1}^{(2)k_3};\ A_{e|2u_3-1,2v_3}^{(2)k_3};\ A_{e|2u_3,2v_3-1}^{(2)k_3};\ A_{e|2u_3,2v_3}^{(2)k_3}\right\};$$

$A^{(3)}$ - matrix with max values, $P^{(3)}$ - matrix with positions (1,2,3 or 4) of these max values.

Layer 3 to Layer 4

$$AA^{(3)} = zeros(N, 20, 20, 6);$$
$$AA^{(3)}(:, 3:18, 3:18, :) = A^{(3)}(:, :, :, :);$$
$$Z^{(4)k_4}_{e|u_4,v_4} = b^{(3)k_4} + \sum_{k_3=1}^{6} \sum_{o_3,e_3=1}^{5} w^{(3)k_4,k_3}_{o_3,e_3} AA^{(3)k_3}_{e|u_4+o_3-1,v_4+e_3-1};$$
$$A^{(4)k_4}_{e|u_4,v_4} = ELU\left(Z^{(4)k_4}_{e|u_4,v_4}\right);$$

Layer 4 to Layer 5

$$\left[A^{(5)k_5}_{e|u_5,v_5}, P^{(5)k_5}_{e|u_5,v_5}\right] = max\left\{A^{(4)k_5}_{e|2u_5-1,2v_5-1};\ A^{(4)k_5}_{e|2u_5-1,2v_5};\ A^{(4)k_5}_{e|2u_5,2v_5-1};\ A^{(4)k_5}_{e|2u_5,2v_5}\right\};$$

$A^{(5)}$ - matrix with max values, $P^{(5)}$ - matrix with positions (1,2,3 or 4) of these max values.

Layer 5 to Layer 6

$$A^{(6)}_{e|s_6} = A^{(5)k_5}_{e|u_5,v_5};$$

Layer 6 to Layer 7

$$Z^{(7)}_{e|s_7} = b^{(6)}_{s_7} + \sum_{s_6=1}^{1024} \Theta^{(6)}_{s_7,s_6} A^{(6)}_{e|s_6};$$
$$A^{(7)}_{e|s_7} = ELU\left(Z^{(7)}_{e|s_7}\right);$$

Layer 7 to Layer 8

$$Z^{(8)}_{e|s_8} = b^{(7)}_{s_8} + \sum_{s_7=1}^{65} \Theta^{(7)}_{s_8,s_7} A^{(7)}_{e|s_7};$$
$$A^{(8)}_{e|s_8} = g\left(Z^{(8)}_{e|s_8}\right);$$

16.2 Back propagation

All our old formulae for partial derivatives of cost function have a form:

$$\frac{\partial}{\partial x} J \ = \sum_{\substack{\text{sum over indices} \\ \text{of t and z}}} t^{(l)} \frac{\partial}{\partial x} z^{(l)};$$

where (l) is a layer number, explicit indices for $t^{(l)}$ and $z^{(l)}$ are omitted, and instead of $z^{(l)}$ in a particular layer we may have $a^{(l)}$ if this layer does not use activation function.

The cost function is a sum of individual contributions from training examples and so are its partial derivatives. So, we will need to add summation over e, replace $z^{(l)}$ with $Z^{(l)}_e$ (or $a^{(l)}$ with $A^{(l)}_e$), and replace $t^{(l)}$ with its multi-example version $T^{(l)}_e$.

$$\frac{\partial}{\partial x} J \ = \sum_{e=1}^{N} \sum_{\substack{\text{sum over indices} \\ \text{of t and z}}} T^{(l)}_e \frac{\partial}{\partial x} Z^{(l)}_e;$$

$Z^{(l)}_e$ (and/or $A^{(l)}_e$) are calculated during forward pass, and we already have formulae for them (see the above subsection). All what is left is to figure out $T^{(l)}_e$ for each layer.

The last layer

For $t^{(8)}$ in component notation we had:

$$t^{(8)}_{s_8} = a^{(8)}_{s_8} - y_{s_8};$$

Let's now introduce matrices $T^{(8)}$, $A^{(8)}$, Y each row of which are row vectors $t^{(8)}$, $a^{(8)}$, y corresponding to one example (strictly speaking we've already introduced $A^{(8)}$ above):

$$T^{(8)}_{e|s_8} = A^{(8)}_{e|s_8} - Y_{e|s_8};$$

Those matrices have as many rows as we have training examples in our batch.

Partial derivative formula for the last layer level is now written as:

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{s_8=1}^{12} T^{(8)}_{e|s_8}\frac{\partial}{\partial x}Z^{(8)}_{e|s_8};$$

Layer 8 $\Rightarrow$ Layer 7

$$M^{(7)} = T^{(8)} * \Theta^{(7)}, \quad \text{in index notation that would read:} \quad M^{(7)}_{e|s_7} = \sum_{s_8=1}^{12} T^{(8)}_{e|s_8}\Theta^{(7)}_{s_8,s_7};$$

$$T^{(7)} = M^{(7)}.*ELUD\left(Z^{(7)}\right);$$

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{s_7=1}^{65} T^{(7)}_{e|s_7}\frac{\partial}{\partial x}Z^{(7)}_{e|s_7};$$

Layer 7 $\Rightarrow$ Layer 6

$$T^{(6)} = T^{(7)} * \Theta^{(6)}, \quad \text{in index notation that would read:} \quad T^{(6)}_{e|s_6} = \sum_{s_7=1}^{65} T^{(7)}_{e|s_7}\Theta^{(6)}_{s_7,s_6};$$

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{s_6=1}^{1024} T^{(6)}_{e|s_6}\frac{\partial}{\partial x}A^{(6)}_{e|s_6};$$

Layer 6 $\Rightarrow$ Layer 5

$$T^{(5)k_5}_{e|u_5,v_5} = T^{(6)}_{e|s_6(u_5,v_5,k_5)}, \quad \text{where } s_6(u_5,v_5,k_5) = 64(k_5-1) + 8(v_5-1) + u_5;$$

$T^{(5)}$ is a reshape of Nx1024 matrix $T^{(6)}$ into a Nx8x8x16 4D structure. In pseudo code :

$$T^{(5)} = reshape(T^{(6)} \Rightarrow N \times 8 \times 8 \times 16);$$

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{k_5=1}^{16}\sum_{u_5=1}^{8}\sum_{v_5=1}^{8} T^{(5)k_5}_{e|u_5,v_5}\frac{\partial}{\partial x}A^{(5)k_5}_{e|u_5,v_5};$$

Layer 5 $\Rightarrow$ Layer 4

$$TTT^{(4)k_4}_{e|u_4,v_4} = T^{(5)k_4}_{e|ceil(u_4/2),ceil(v_4/2)};$$

$$P4^{k_4}_{e|u_4,v_4} = \begin{cases} 1 & \text{if this pair } u_4 \text{ and } v_4 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$

$$TT^{(4)} = TTT^{(4)}.*P4;$$

$$T^{(4)} = TT^{(4)}.*ELUD(Z^{(4)});$$

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{k_4=1}^{16}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4}\frac{\partial}{\partial x}Z^{(4)k_4}_{e|u_4,v_4};$$

And here is pseudo code to calculate $TT^{(4)}$:

$$TTT4 = zeros(N,16,16,16);$$

$$TT4 = zeros(N,16,16,16);$$

$$P4 = zeros(N,16,16,16);$$

$P5aux = zeros(N, 16);$
$P4aux1 = zeros(N, 16);$
$P4aux2 = zeros(N, 16);$
$P4aux3 = zeros(N, 16);$
$P4aux4 = zeros(N, 16);$

$for\ i = 1 : 8$
$for\ j = 1 : 8$

$TTT4(:, 2 * i - 1, 2 * j - 1, :) = T5(:, i, j, :);$
$TTT4(:, 2 * i - 1, 2 * j, :) = T5(:, i, j, :);$
$TTT4(:, 2 * i, 2 * j - 1, :) = T5(:, i, j, :);$
$TTT4(:, 2 * i, 2 * j, :) = T5(:, i, j, :);$

$P5aux(:, :) = uint8(P5(:, i, j, :));$

$P4aux1(:, :) = P5aux(:, :);$
$P4aux2(:, :) = P5aux(:, :)/2;$
$P4aux3(:, :) = P5aux(:, :)/3;$
$P4aux4(:, :) = P5aux(:, :)/4;$

$P4aux1(P4aux1 \ = 1) = 0;$
$P4aux2(P4aux2 \ = 1) = 0;$
$P4aux3(P4aux3 \ = 1) = 0;$
$P4aux4(P4aux4 \ = 1) = 0;$

$P4(:, 2 * i - 1, 2 * j - 1, :) = P4aux1(:, :);$
$P4(:, 2 * i - 1, 2 * j, :) = P4aux2(:, :);$
$P4(:, 2 * i, 2 * j - 1, :) = P4aux3(:, :);$
$P4(:, 2 * i, 2 * j, :) = P4aux4(:, :);$

end
end

$TT4 = TTT4. * P4;$

Layer 4 $\Rightarrow$ Layer 3

$\Gamma^{(4)} = zeros(N, 24, 24, 16);$
$\Gamma^{(4)}(:, 5 : 20, 5 : 20, :) = T^{(4)}(:, :, :, :);$
$\Gamma^{(3)k_3}_{e|u3,v3} = \sum_{k_4=1}^{16} \sum_{o_3=1}^{5} \sum_{e_3=1}^{5} \Gamma^{(4)k_4}_{e|u_3-o_3+5,v_3-e_3+5} w^{(3)k_4,k_3}_{o_3,e_3};$
$T^{(3)k_3}_{e|u_3,v_3} = \Gamma^{(3)k_3}_{e|u3+2,v3+2};$
$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_3=1}^{6} \sum_{u_3=1}^{16} \sum_{v_3=1}^{16} T^{(3)k_3}_{e|u3,v3} \frac{\partial}{\partial x} A^{(3)k_3}_{e|u_3,v_3};$

Layer 3 $\Rightarrow$ Layer 2

$$TTT^{(2)k_2}_{e|u_2,v_2} = T^{(3)k_2}_{e|ceil(u_2/2),ceil(v_2/2)};$$

$$P2^{k_2}_{u_2,v_2} = \begin{cases} 1 & \text{if this pair } u_2 \text{ and } v_2 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$

$$TT^{(2)} = TTT^{(2)}. * P2;$$

$$T^{(2)} = TT^{(2)}. * ELUD(Z^{(2)});$$

$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial x} Z^{(2)k_2}_{e|u_2,v_2};$$

And here is pseudo code to calculate $TT^{(2)}$:

$TTT2 = zeros(N, 32, 32, 6);$
$TT2 = zeros(N, 32, 32, 6);$
$P2 = zeros(N, 32, 32, 6);$

$P3aux = zeros(N, 6);$
$P2aux1 = zeros(N, 6);$
$P2aux2 = zeros(N, 6);$
$P2aux3 = zeros(N, 6);$
$P2aux4 = zeros(N, 6);$

for $i = 1 : 16$
for $j = 1 : 16$

$TTT2(:, 2 * i - 1, 2 * j - 1, :) = T3(:, i, j, :);$
$TTT2(:, 2 * i - 1, 2 * j, :) = T3(:, i, j, :);$
$TTT2(:, 2 * i, 2 * j - 1, :) = T3(:, i, j, :);$
$TTT2(:, 2 * i, 2 * j, :) = T3(:, i, j, :);$

$P3aux(:, :) = uint8(P3(:, i, j, :));$

$P2aux1(:, :) = P3aux(:, :);$
$P2aux2(:, :) = P3aux(:, :)/2;$
$P2aux3(:, :) = P3aux(:, :)/3;$
$P2aux4(:, :) = P3aux(:, :)/4;$

$P2aux1(P2aux1 \sim= 1) = 0;$
$P2aux2(P2aux2 \sim= 1) = 0;$
$P2aux3(P2aux3 \sim= 1) = 0;$
$P2aux4(P2aux4 \sim= 1) = 0;$

$P2(:, 2 * i - 1, 2 * j - 1, :) = P2aux1(:, :);$
$P2(:, 2 * i - 1, 2 * j, :) = P2aux2(:, :);$
$P2(:, 2 * i, 2 * j - 1, :) = P2aux3(:, :);$
$P2(:, 2 * i, 2 * j, :) = P2aux4(:, :);$

end
end

$TT2 = TTT2. * P2;$

17 Final formulae for derivatives

At the end of the day we need expressions for partial derivatives of cost function with respect to all learnable parameters. Let's start from the top of our parameters table at the very beginning of Back propagation part, and calculate $\partial J/\partial b^{(1)k}$. For this we will be using our 2nd layer level formula:

$$\begin{aligned}
\frac{\partial}{\partial b^{(1)k}} J &= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial b^{(1)k}} Z^{(2)k_2}_{e|u_2,v_2} \\
&= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial b^{(1)k}} \left(b^{(1)k_2} + \sum_{o_1,e_1=1}^{5} w^{(1)k_2}_{o_1,e_1} AA^{(1)}_{e|u_2+o_1-1,v_2+e_1-1} \right) \\
&= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \delta_{k,k_2} \\
&= \sum_{e=1}^{N} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k}_{e|u_2,v_2};
\end{aligned}$$

The next is $\partial J/\partial w^{(1)k}_{s,s'}$:

$$\begin{aligned}
\frac{\partial}{\partial w^{(1)k}_{s,s'}} J &= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial w^{(1)k}_{s,s'}} Z^{(2)k_2}_{e|u_2,v_2} \\
&= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial w^{(1)k}_{s,s'}} \left(b^{(1)k_2} + \sum_{o_1,e_1=1}^{5} w^{(1)k_2}_{o_1,e_1} AA^{(1)}_{e|u_2+o_1-1,v_2+e_1-1} \right) \\
&= \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} \sum_{o_1,e_1=1}^{5} T^{(2)k_2}_{e|u_2,v_2} \delta_{k,k_2} \delta_{s,o_1} \delta_{s',e_1} AA^{(1)}_{e|u_2+o_1-1,v_2+e_1-1} \\
&= \sum_{e=1}^{N} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} T^{(2)k}_{e|u_2,v_2} AA^{(1)}_{e|u_2+s-1,v_2+s'-1};
\end{aligned}$$

To calculate $\partial J/\partial b^{(3)k}$ we will need the 4th layer level formula:

$$\begin{aligned}
\frac{\partial}{\partial b^{(3)k}} J &= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4} \frac{\partial}{\partial b^{(3)k}} Z^{(4)k_4}_{e|u_4,v_4} \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4} \frac{\partial}{\partial b^{(3)k}} \left(b^{(3)k_4} + \sum_{k_3=1}^{6} \sum_{o_3,e_3=1}^{5} w^{(3)k_4,k_3}_{o_3,e_3} AA^{(3)k_3}_{e|u_4+o_3-1,v_4+e_3-1} \right) \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4} \delta_{k,k_4} \\
&= \sum_{e=1}^{N} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k}_{e|u_4,v_4};
\end{aligned}$$

For $\partial J/\partial w^{(3)k,k'}_{s,s'}$:

$$\begin{aligned}
\frac{\partial}{\partial w^{(3)k,k'}_{s,s'}} J &= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4} \frac{\partial}{\partial w^{(3)k,k'}_{s,s'}} Z^{(4)k_4}_{e|u_4,v_4} \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k_4}_{e|u_4,v_4} \frac{\partial}{\partial w^{(3)k,k'}_{s,s'}} \left(b^{(3)k_4} + \sum_{k_3=1}^{6} \sum_{o_3,e_3=1}^{5} w^{(3)k_4,k_3}_{o_3,e_3} AA^{(3)k_3}_{e|u_4+o_3-1,v_4+e_3-1} \right) \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} \sum_{k_3=1}^{6} \sum_{o_3,e_3=1}^{5} T^{(4)k_4}_{e|u_4,v_4} \delta_{k,k_4} \delta_{k',k_3} \delta_{s,o_3} \delta_{s',e_3} AA^{(3)k_3}_{e|u_4+o_3-1,v_4+e_3-1} \\
&= \sum_{e=1}^{N} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T^{(4)k}_{e|u_4,v_4} AA^{(3)k'}_{e|u_4+s-1,v_4+s'-1};
\end{aligned}$$

To calculate $\partial J/\partial b_i^{(6)}$ we will need the 7th layer level formula:

$$\begin{aligned}\frac{\partial}{\partial b_i^{(6)}}J &= \sum_{e=1}^{N}\sum_{s_7=1}^{65} T_{e|s_7}^{(7)}\frac{\partial}{\partial b_i^{(6)}}Z_{e|s_7}^{(7)} \\ &= \sum_{e=1}^{N}\sum_{s_7=1}^{65} T_{e|s_7}^{(7)}\frac{\partial}{\partial b_i^{(6)}}\left(b_{s_7}^{(6)} + \sum_{s_6=1}^{1024}\Theta_{s_7,s_6}^{(6)}A_{e|s_6}^{(6)}\right) \\ &= \sum_{e=1}^{N}\sum_{s_7=1}^{65} T_{e|s_7}^{(7)}\delta_{i,s_7} \\ &= \sum_{e=1}^{N} T_{e|i}^{(7)};\end{aligned}$$

For $\partial J/\partial \Theta_{i,j}^{(6)}$:

$$\begin{aligned}\frac{\partial}{\partial \Theta_{i,j}^{(6)}}J &= \sum_{e=1}^{N}\sum_{s_7=1}^{65} T_{e|s_7}^{(7)}\frac{\partial}{\partial \Theta_{i,j}^{(6)}}Z_{e|s_7}^{(7)} \\ &= \sum_{e=1}^{N}\sum_{s_7=1}^{65} T_{e|s_7}^{(7)}\frac{\partial}{\partial \Theta_{i,j}^{(6)}}\left(b_{s_7}^{(6)} + \sum_{s_6=1}^{1024}\Theta_{s_7,s_6}^{(6)}A_{e|s_6}^{(6)}\right) \\ &= \sum_{e=1}^{N}\sum_{s_7=1}^{65}\sum_{s_6=1}^{1024} T_{e|s_7}^{(7)}\delta_{i,s_7}\delta_{j,s_6}A_{e|s_6}^{(6)} \\ &= \sum_{e=1}^{N} T_{e|i}^{(7)}A_{e|j}^{(6)} = \left(T^{(7)^T} * A^{(6)}\right)_{i,j};\end{aligned}$$

To calculate $\partial J/\partial b_i^{(7)}$ we will need the 8th layer level formula:

$$\begin{aligned}\frac{\partial}{\partial b_i^{(7)}}J &= \sum_{e=1}^{N}\sum_{s_8=1}^{12} T_{e|s_8}^{(8)}\frac{\partial}{\partial b_i^{(7)}}Z_{e|s_8}^{(8)} \\ &= \sum_{e=1}^{N}\sum_{s_8=1}^{12} T_{e|s_8}^{(8)}\frac{\partial}{\partial b_i^{(7)}}\left(b_{s_8}^{(7)} + \sum_{s_7=1}^{65}\Theta_{s_8,s_7}^{(7)}A_{e|s_7}^{(7)}\right) \\ &= \sum_{e=1}^{N}\sum_{s_8=1}^{12} T_{e|s_8}^{(8)}\delta_{i,s_8} \\ &= \sum_{e=1}^{N} T_{e|i}^{(8)};\end{aligned}$$

For $\partial J/\partial \Theta_{i,j}^{(7)}$:

$$\begin{aligned}\frac{\partial}{\partial \Theta_{i,j}^{(7)}}J &= \sum_{e=1}^{N}\sum_{s_8=1}^{12} T_{e|s_8}^{(8)}\frac{\partial}{\partial \Theta_{i,j}^{(7)}}Z_{e|s_8}^{(8)} \\ &= \sum_{e=1}^{N}\sum_{s_8=1}^{12} T_{e|s_8}^{(8)}\frac{\partial}{\partial \Theta_{i,j}^{(7)}}\left(b_{s_8}^{(7)} + \sum_{s_7=1}^{65}\Theta_{s_8,s_7}^{(7)}A_{e|s_7}^{(7)}\right) \\ &= \sum_{e=1}^{N}\sum_{s_8=1}^{12}\sum_{s_7=1}^{65} T_{e|s_8}^{(8)}\delta_{i,s_8}\delta_{j,s_7}A_{e|s_7}^{(7)} \\ &= \sum_{e=1}^{N} T_{e|i}^{(8)}A_{e|j}^{(7)} = \left(T^{(8)^T} * A^{(7)}\right)_{i,j}.\end{aligned}$$

Part IV
Adding batch normalization

Now we will take our results generalized for a batch of training examples and will add batch normalization to the procedure.

As batch normalization step occurs after calculation of z-inputs and before application of activation function (i.e. before calculation of a-inputs), from now on this would be a good idea to separate z-input calculation and application of activation function into different layers. Here is what we mean. If previously Layer 1 $\rightarrow$ Layer 2 transition involved both convolution (i.e. calculation of z-inputs) and application of activation function (i.e. calculation of a-inputs), then now instead of one transition we will have three:

$$\begin{aligned} &Layer\ 1 \ \rightarrow Layer\ 21 && - \ convolution \\ &Layer\ 21 \rightarrow Layer\ 22 && - \ batch\ normalization \\ &Layer\ 22 \rightarrow Layer\ 23 && - \ application\ of\ activation\ function \end{aligned}$$

and instead of one layer (Layer 2) we will have three layers (Layer 21, Layer 22 and Layer 23).

The original layout diagram changes accordingly:

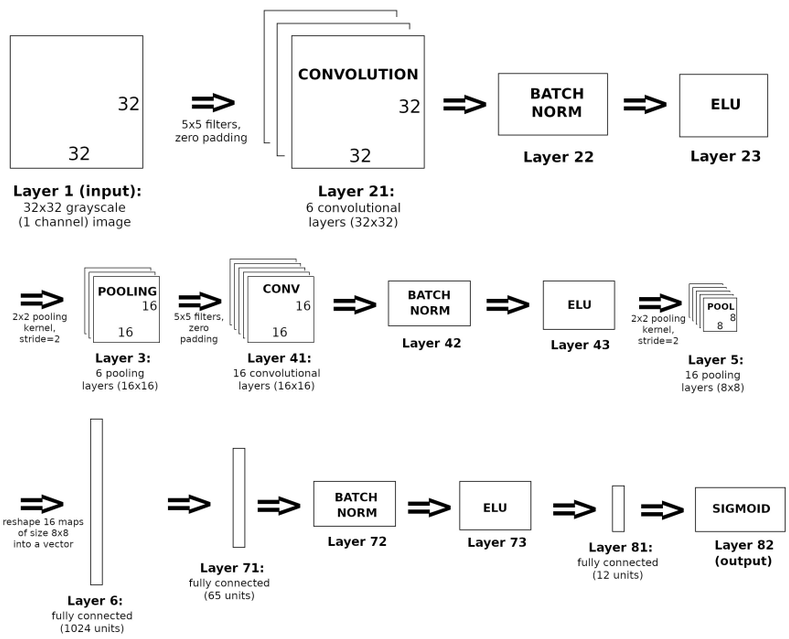

Figure 2: Original layout is modified to add batch normalization steps (Layer 22, Layer 42, and Layer 72).

Below is a batch normalized version of Section 16. Many of the transitions are unchanged and we re-write the same formulae just for convenience so the readers can have all what they need in one place.

18 Forward pass with batch normalization

Layer 1 to Layer 21

$A^{(1)}_{e|u_1,v_1}$ - input image;

$AA^{(1)} = zeros(N, 36, 36);$

$AA^{(1)}(:, 3:34, 3:34) = A^{(1)}(:, :, :);$

$$Z^{(21)k_2}_{e|u_2,v_2} = b^{(1)k_2} + \sum_{o_1,e_1=1}^{5} w^{(1)k_2}_{o_1,e_1} AA^{(1)}_{e|u_2+o_1-1,v_2+e_1-1};$$

Layer 21 to Layer 22

$$\mu_{u_2,v_2}^{(21)k_2} = \frac{1}{N}\sum_{e=1}^{N} Z_{e|u_2,v_2}^{(21)k_2};$$

$$V_{u_2,v_2}^{(21)k_2} = \frac{1}{N}\sum_{e=1}^{N}\left(Z_{e|u_2,v_2}^{(21)k_2} - \mu_{u_2,v_2}^{(21)k_2}\right)^2;$$

$$Z_{e|u_2,v_2}^{(22)k_2} = \frac{Z_{e|u_2,v_2}^{(21)k_2} - \mu_{u_2,v_2}^{(21)k_2}}{\sqrt{V_{u_2,v_2}^{(21)k_2} + \epsilon}};$$

$$Y_{e|u_2,v_2}^{(22)k_2} = \alpha_{u_2,v_2}^{(2)k_2} Z_{e|u_2,v_2}^{(22)k_2} + \beta_{u_2,v_2}^{(2)k_2};$$

Where $\alpha^{(2)}$ and $\beta^{(2)}$ are learnable 1x32x32x6 structures, and ϵ is a small number added for numerical stability.

Layer 22 to Layer 23

$$A_{e|u_2,v_2}^{(23)k_2} = ELU\left(Y_{e|u_2,v_2}^{(22)k_2}\right);$$

Layer 23 to Layer 3

$$\left[A_{e|u_3,v_3}^{(3)k_3}, P_{e|u_3,v_3}^{(3)k_3}\right] = max\left\{A_{e|2u_3-1,2v_3-1}^{(23)k_3};\ A_{e|2u_3-1,2v_3}^{(23)k_3};\ A_{e|2u_3,2v_3-1}^{(23)k_3};\ A_{e|2u_3,2v_3}^{(23)k_3}\right\};$$

$A^{(3)}$ - matrix with max values, $P^{(3)}$ - matrix with positions (1,2,3 or 4) of these max values.

Layer 3 to Layer 41

$$AA^{(3)} = zeros(N, 20, 20, 6);$$

$$AA^{(3)}(:, 3:18, 3:18, :) = A^{(3)}(:, :, :, :);$$

$$Z_{e|u_4,v_4}^{(41)k_4} = b^{(3)k_4} + \sum_{k_3=1}^{6}\sum_{o_3,e_3=1}^{5} w_{o_3,e_3}^{(3)k_4,k_3} AA_{e|u_4+o_3-1,v_4+e_3-1}^{(3)k_3};$$

Layer 41 to Layer 42

$$\mu_{u_4,v_4}^{(41)k_4} = \frac{1}{N}\sum_{e=1}^{N} Z_{e|u_4,v_4}^{(41)k_4};$$

$$V_{u_4,v_4}^{(41)k_4} = \frac{1}{N}\sum_{e=1}^{N}\left(Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}\right)^2;$$

$$Z_{e|u_4,v_4}^{(42)k_4} = \frac{Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}}{\sqrt{V_{u_4,v_4}^{(41)k_4} + \epsilon}};$$

$$Y_{e|u_4,v_4}^{(42)k_4} = \alpha_{u_4,v_4}^{(4)k_4} Z_{e|u_4,v_4}^{(42)k_4} + \beta_{u_4,v_4}^{(4)k_4};$$

Where $\alpha^{(4)}$ and $\beta^{(4)}$ are learnable 1x16x16x16 structures.

Layer 42 to Layer 43

$$A_{e|u_4,v_4}^{(43)k_4} = ELU\left(Y_{e|u_4,v_4}^{(42)k_4}\right);$$

Layer 43 to Layer 5

$$\left[A_{e|u_5,v_5}^{(5)k_5}, P_{e|u_5,v_5}^{(5)k_5}\right] = max\left\{A_{e|2u_5-1,2v_5-1}^{(43)k_5};\ A_{e|2u_5-1,2v_5}^{(43)k_5};\ A_{e|2u_5,2v_5-1}^{(43)k_5};\ A_{e|2u_5,2v_5}^{(43)k_5}\right\};$$

$A^{(5)}$ - matrix with max values, $P^{(5)}$ - matrix with positions (1,2,3 or 4) of these max values.

Layer 5 to Layer 6

$$A^{(6)}_{e|s_6} = A^{(5)k_5}_{e|u_5,v_5};$$

Layer 6 to Layer 71

$$Z^{(71)}_{e|s_7} = b^{(6)}_{s_7} + \sum_{s_6=1}^{1024} \Theta^{(6)}_{s_7,s_6} A^{(6)}_{e|s_6};$$

Layer 71 to Layer 72

$$\mu^{(71)}_{s_7} = \frac{1}{N}\sum_{e=1}^{N} Z^{(71)}_{e|s_7};$$
$$V^{(71)}_{s_7} = \frac{1}{N}\sum_{e=1}^{N} \left(Z^{(71)}_{e|s_7} - \mu^{(71)}_{s_7}\right)^2;$$
$$Z^{(72)}_{e|s_7} = \frac{Z^{(71)}_{e|s_7} - \mu^{(71)}_{s_7}}{\sqrt{V^{(71)}_{s_7} + \epsilon}};$$
$$Y^{(72)}_{e|s_7} = \alpha^{(7)}_{s_7} Z^{(72)}_{e|s_7} + \beta^{(7)}_{s_7};$$

Where $\alpha^{(7)}$ and $\beta^{(7)}$ are learnable 1x65 row vectors.

Layer 72 to Layer 73

$$A^{(73)}_{e|s_7} = ELU\left(Y^{(72)}_{e|s_7}\right);$$

Layer 73 to Layer 81

$$Z^{(81)}_{e|s_8} = b^{(7)}_{s_8} + \sum_{s_7=1}^{65} \Theta^{(7)}_{s_8,s_7} A^{(73)}_{e|s_7};$$

Layer 81 to Layer 82

$$A^{(82)}_{e|s_8} = g\left(Z^{(81)}_{e|s_8}\right);$$

19 Back propagation with batch normalization

The last layer

$$T^{(8)}_{e|s_8} = A^{(82)}_{e|s_8} - Y_{e|s_8};$$

$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N}\sum_{s_8=1}^{12} T^{(8)}_{e|s_8} \frac{\partial}{\partial x} Z^{(81)}_{e|s_8};$$

Layer 81 $\Rightarrow$ Layer 73

$$M^{(7)} = T^{(8)} * \Theta^{(7)};$$
$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N}\sum_{s_7=1}^{65} M^{(7)}_{e|s_7} \frac{\partial}{\partial x} A^{(73)}_{e|s_7};$$

Layer 73 $\Rightarrow$ Layer 72

$$H^{(7)} = M^{(7)} .* ELUD\left(Y^{(72)}\right);$$
$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e|s_7} \frac{\partial}{\partial x} Y^{(72)}_{e|s_7};$$

Layer 72 $\Rightarrow$ Layer 71

$$
\begin{aligned}
\frac{\partial}{\partial x}J =& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\frac{\partial}{\partial x}Y^{(72)}_{e'|s_7} \\
=& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\frac{\partial}{\partial x}Z^{(72)}_{e'|s_7} \\
=& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}Z^{(71)}_{e'|s_7} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\mu^{(71)}_{s_7} \\
& - \frac{1}{2}\frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\frac{\partial}{\partial x}V^{(71)}_{s_7}\Bigg] \\
=& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\sum_{e=1}^{N}\delta_{ee'}Z^{(71)}_{e|s_7} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\frac{1}{N}\sum_{e=1}^{N}Z^{(71)}_{e|s_7} \\
& - \frac{1}{2}\frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\frac{\partial}{\partial x}\frac{1}{N}\sum_{e=1}^{N}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)^2\Bigg] \\
=& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\sum_{e=1}^{N}\delta_{ee'}Z^{(71)}_{e|s_7} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\frac{1}{N}\sum_{e=1}^{N}Z^{(71)}_{e|s_7} \\
& - \frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\frac{1}{N}\sum_{e=1}^{N}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\frac{\partial}{\partial x}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\Bigg] \\
=& \sum_{e'=1}^{N}\sum_{s_7=1}^{65} H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\sum_{e=1}^{N}\delta_{ee'}Z^{(71)}_{e|s_7} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\frac{\partial}{\partial x}\frac{1}{N}\sum_{e=1}^{N}Z^{(71)}_{e|s_7} \\
& - \frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\frac{1}{N}\sum_{e=1}^{N}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\frac{\partial}{\partial x}Z^{(71)}_{e|s_7} \\
& + \frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\frac{1}{N}\left(\frac{\partial\mu^{(71)}_{s_7}}{\partial x}\right)\sum_{e=1}^{N}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\Bigg]
\end{aligned}
$$

The very last term in the square brackets produces zero. After discarding this term and changing summation order for indices e' and e we get:

$$
\begin{aligned}
\frac{\partial}{\partial x}J =& \sum_{e=1}^{N}\sum_{s_7=1}^{65}\Bigg(\frac{1}{N}\sum_{e'=1}^{N}H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{N\delta_{ee'}}{\sqrt{V^{(71)}_{s_7}+\epsilon}} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}} \\
& - \frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\Bigg]\Bigg)\frac{\partial}{\partial x}Z^{(71)}_{e|s_7}; \\
\frac{\partial}{\partial x}J =& \sum_{e=1}^{N}\sum_{s_7=1}^{65} T^{(7)}_{e|s_7}\frac{\partial}{\partial x}Z^{(71)}_{e|s_7}, \text{ where matrix } T^{(7)} \text{ is given by:} \\
T^{(7)}_{e|s_7} =& \frac{1}{N}\sum_{e'=1}^{N}H^{(7)}_{e'|s_7}\alpha^{(7)}_{s_7}\Bigg[\frac{N\delta_{ee'}}{\sqrt{V^{(71)}_{s_7}+\epsilon}} - \frac{1}{\sqrt{V^{(71)}_{s_7}+\epsilon}} - \frac{Z^{(71)}_{e'|s_7}-\mu^{(71)}_{s_7}}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)\Bigg] \\
=& H^{(7)}_{e|s_7}\frac{\alpha^{(7)}_{s_7}}{\sqrt{V^{(71)}_{s_7}+\epsilon}} - \frac{1}{N}\frac{\alpha^{(7)}_{s_7}}{\sqrt{V^{(71)}_{s_7}+\epsilon}}\sum_{e'=1}^{N}H^{(7)}_{e'|s_7} - \frac{1}{N}\frac{\alpha^{(7)}_{s_7}\left(Z^{(71)}_{e|s_7}-\mu^{(71)}_{s_7}\right)}{\left(V^{(71)}_{s_7}+\epsilon\right)^{3/2}}\Bigg(\sum_{e'=1}^{N}H^{(7)}_{e'|s_7}Z^{(71)}_{e'|s_7} \\
& - \mu^{(71)}_{s_7}\sum_{e'=1}^{N}H^{(7)}_{e'|s_7}\Bigg).
\end{aligned}
$$

Layer 71 ⇒ Layer 6

$$T^{(6)} = T^{(7)} * \Theta^{(6)};$$
$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{s_6=1}^{1024} T^{(6)}_{e|s_6} \frac{\partial}{\partial x} A^{(6)}_{e|s_6};$$

Layer 6 ⇒ Layer 5

$$T^{(5)k_5}_{e|u_5,v_5} = T^{(6)}_{e|s_6(u_5,v_5,k_5)}, \quad \text{where } s_6(u_5,v_5,k_5) = 64(k_5-1) + 8(v_5-1) + u_5;$$

$T^{(5)}$ is a reshape of Nx1024 matrix $T^{(6)}$ into a Nx8x8x16 4D structure. In pseudo code :

$$T^{(5)} = reshape(T^{(6)} \Rightarrow N \times 8 \times 8 \times 16);$$

$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_5=1}^{16} \sum_{u_5=1}^{8} \sum_{v_5=1}^{8} T^{(5)k_5}_{e|u_5,v_5} \frac{\partial}{\partial x} A^{(5)k_5}_{e|u_5,v_5};$$

Layer 5 ⇒ Layer 43

$$TTTT^{(4)k_4}_{e|u_4,v_4} = T^{(5)k_4}_{e|ceil(u_4/2),ceil(v_4/2)};$$
$$P4^{k_4}_{e|u_4,v_4} = \begin{cases} 1 & \text{if this pair } u_4 \text{ and } v_4 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases}$$
$$TTT^{(4)} = TTTT^{(4)} .* P4;$$
$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TTT^{(4)k_4}_{e|u_4,v_4} \frac{\partial}{\partial x} A^{(43)k_4}_{e|u_4,v_4};$$

And here is pseudo code to calculate $TTT^{(4)}$:

$$TTTT4 = zeros(N,16,16,16);$$
$$TTT4 = zeros(N,16,16,16);$$
$$P4 = zeros(N,16,16,16);$$

$$P5aux = zeros(N,16);$$
$$P4aux1 = zeros(N,16);$$
$$P4aux2 = zeros(N,16);$$
$$P4aux3 = zeros(N,16);$$
$$P4aux4 = zeros(N,16);$$

$for\ i = 1:8$

$for\ j = 1:8$

$$TTTT4(:,2*i-1,2*j-1,:) = T5(:,i,j,:);$$
$$TTTT4(:,2*i-1,2*j,:) = T5(:,i,j,:);$$
$$TTTT4(:,2*i,2*j-1,:) = T5(:,i,j,:);$$
$$TTTT4(:,2*i,2*j,:) = T5(:,i,j,:);$$

$$P5aux(:,:) = uint8(P5(:,i,j,:));$$

$$P4aux1(:,:) = P5aux(:,:);$$
$$P4aux2(:,:) = P5aux(:,:)/2;$$
$$P4aux3(:,:) = P5aux(:,:)/3;$$
$$P4aux4(:,:) = P5aux(:,:)/4;$$

$P4aux1(P4aux1 \ = 1) = 0;$
$P4aux2(P4aux2 \ = 1) = 0;$
$P4aux3(P4aux3 \ = 1) = 0;$
$P4aux4(P4aux4 \ = 1) = 0;$

$P4(:, 2 * i - 1, 2 * j - 1, :) = P4aux1(:, :);$
$P4(:, 2 * i - 1, 2 * j, :) = P4aux2(:, :);$
$P4(:, 2 * i, 2 * j - 1, :) = P4aux3(:, :);$
$P4(:, 2 * i, 2 * j, :) = P4aux4(:, :);$

end

end

$TTT4 = TTTT4. * P4;$

Layer 43 ⇒ Layer 42

$$TT^{(4)} = TTT^{(4)}. * ELUD(Y^{(42)});$$
$$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e|u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} Y_{e|u_4,v_4}^{(42)k_4};$$

Layer 42 ⇒ Layer 41

$$\begin{aligned}
\frac{\partial}{\partial x} J &= \sum_{e'=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e'|u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} Y_{e'|u_4,v_4}^{(42)k_4} \\
&= \sum_{e'=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e'|u_4,v_4}^{(4)k_4} \alpha_{u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} Z_{e'|u_4,v_4}^{(42)k_4} \\
&= \sum_{e'=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e'|u_4,v_4}^{(4)k_4} \alpha_{u_4,v_4}^{(4)k_4} \left[\frac{1}{\sqrt{V_{u_4,v_4}^{(41)k_4} + \epsilon}} \frac{\partial}{\partial x} Z_{e'|u_4,v_4}^{(41)k_4} - \frac{1}{\sqrt{V_{u_4,v_4}^{(41)k_4} + \epsilon}} \frac{\partial}{\partial x} \frac{1}{N} \sum_{e=1}^{N} Z_{e|u_4,v_4}^{(41)k_4} \right. \\
&\left. - \frac{1}{2} \frac{Z_{e'|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}}{\left(V_{u_4,v_4}^{(41)k_4} + \epsilon \right)^{3/2}} \frac{\partial}{\partial x} \frac{1}{N} \sum_{e=1}^{N} \left(Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4} \right)^2 \right] \\
&= \sum_{e'=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e'|u_4,v_4}^{(4)k_4} \alpha_{u_4,v_4}^{(4)k_4} \left[\frac{1}{\sqrt{V_{u_4,v_4}^{(41)k_4} + \epsilon}} \sum_{e=1}^{N} \left(\delta_{ee'} - \frac{1}{N} \right) \frac{\partial}{\partial x} Z_{e|u_4,v_4}^{(41)k_4} \right. \\
&- \frac{Z_{e'|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}}{\left(V_{u_4,v_4}^{(41)k_4} + \epsilon \right)^{3/2}} \frac{1}{N} \sum_{e=1}^{N} \left(Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4} \right) \frac{\partial}{\partial x} Z_{e|u_4,v_4}^{(41)k_4} \\
&\left. + \frac{Z_{e'|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}}{\left(V_{u_4,v_4}^{(41)k_4} + \epsilon \right)^{3/2}} \frac{1}{N} \sum_{e=1}^{N} \left(Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4} \right) \frac{\partial}{\partial x} \mu_{u_4,v_4}^{(41)k_4} \right]
\end{aligned}$$

Summation over e in the last term produces zero. After discarding this term and changing summation order over e and e' indices we have:

$$\begin{aligned}
\frac{\partial}{\partial x} J &= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} \sum_{e'=1}^{N} TT_{e'|u_4,v_4}^{(4)k_4} \alpha_{u_4,v_4}^{(4)k_4} \left[\frac{1}{\sqrt{V_{u_4,v_4}^{(41)k_4} + \epsilon}} \left(\delta_{ee'} - \frac{1}{N} \right) \right. \\
&\left. - \frac{Z_{e'|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4}}{\left(V_{u_4,v_4}^{(41)k_4} + \epsilon \right)^{3/2}} \frac{1}{N} \left(Z_{e|u_4,v_4}^{(41)k_4} - \mu_{u_4,v_4}^{(41)k_4} \right) \right] \frac{\partial}{\partial x} Z_{e|u_4,v_4}^{(41)k_4};
\end{aligned}$$

$\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} T_{e|u_4,v_4}^{(4)k_4} \frac{\partial}{\partial x} Z_{e|u_4,v_4}^{(41)k_4}$, where $T^{(4)}$ is defined as:

$$
\begin{aligned}
T^{(4)k_4}_{e|u_4,v_4} &= \sum_{e'=1}^{N} TT^{(4)k_4}_{e'|u_4,v_4} \alpha^{(4)k_4}_{u_4,v_4} \Bigg[\frac{1}{\sqrt{V^{(41)k_4}_{u_4,v_4} + \epsilon}} \left(\delta_{ee'} - \frac{1}{N} \right) \\
&- \frac{Z^{(41)k_4}_{e'|u_4,v_4} - \mu^{(41)k_4}_{u_4,v_4}}{\left(V^{(41)k_4}_{u_4,v_4} + \epsilon \right)^{3/2}} \frac{1}{N} \left(Z^{(41)k_4}_{e|u_4,v_4} - \mu^{(41)k_4}_{u_4,v_4} \right) \Bigg] \\
&= TT^{(4)k_4}_{e|u_4,v_4} \frac{\alpha^{(4)k_4}_{u_4,v_4}}{\sqrt{V^{(41)k_4}_{u_4,v_4} + \epsilon}} - \frac{1}{N} \frac{\alpha^{(4)k_4}_{u_4,v_4}}{\sqrt{V^{(41)k_4}_{u_4,v_4} + \epsilon}} \sum_{e'=1}^{N} TT^{(4)k_4}_{e'|u_4,v_4} \\
&- \frac{1}{N} \frac{\alpha^{(4)k_4}_{u_4,v_4} \left(Z^{(41)k_4}_{e|u_4,v_4} - \mu^{(41)k_4}_{u_4,v_4} \right)}{\left(V^{(41)k_4}_{u_4,v_4} + \epsilon \right)^{3/2}} \left(\sum_{e'=1}^{N} TT^{(4)k_4}_{e'|u_4,v_4} Z^{(41)k_4}_{e'|u_4,v_4} - \mu^{(41)k_4}_{u_4,v_4} \sum_{e'=1}^{N} TT^{(4)k_4}_{e'|u_4,v_4} \right).
\end{aligned}
$$

Layer 41 $\Rightarrow$ Layer 3

$$
\begin{aligned}
&\Gamma^{(4)} = zeros(N, 24, 24, 16); \\
&\Gamma^{(4)}(:, 5:20, 5:20, :) = T^{(4)}(:, :, :, :); \\
&\Gamma^{(3)k_3}_{e|u3,v3} = \sum_{k_4=1}^{16} \sum_{o_3=1}^{5} \sum_{e_3=1}^{5} \Gamma^{(4)k_4}_{e|u_3-o_3+5,v_3-e_3+5} w^{(3)k_4,k_3}_{o_3,e_3}; \\
&T^{(3)k_3}_{e|u_3,v_3} = \Gamma^{(3)k_3}_{e|u3+2,v3+2}; \\
&\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_3=1}^{6} \sum_{u_3=1}^{16} \sum_{v_3=1}^{16} T^{(3)k_3}_{e|u3,v3} \frac{\partial}{\partial x} A^{(3)k_3}_{e|u_3,v_3};
\end{aligned}
$$

Layer 3 $\Rightarrow$ Layer 23

$$
\begin{aligned}
&TTTT^{(2)k_2}_{e|u_2,v_2} = T^{(3)k_2}_{e|ceil(u_2/2),ceil(v_2/2)}; \\
&P2^{k_2}_{u_2,v_2} = \begin{cases} 1 & \text{if this pair } u_2 \text{ and } v_2 \text{ corresponds to the pooling maximum;} \\ 0 & \text{otherwise;} \end{cases} \\
&TTT^{(2)} = TTTT^{(2)} .* P2; \\
&\frac{\partial}{\partial x} J = \sum_{e=1}^{N} \sum_{k_2=1}^{6} \sum_{u_2=1}^{32} \sum_{v_2=1}^{32} TTT^{(2)k_2}_{e|u_2,v_2} \frac{\partial}{\partial x} Z^{(23)k_2}_{e|u_2,v_2};
\end{aligned}
$$

And here is pseudo code to calculate $TTT^{(2)}$:

$$
\begin{aligned}
&TTTT2 = zeros(N, 32, 32, 6); \\
&TTT2 = zeros(N, 32, 32, 6); \\
&P2 = zeros(N, 32, 32, 6); \\
&\\
&P3aux = zeros(N, 6); \\
&P2aux1 = zeros(N, 6); \\
&P2aux2 = zeros(N, 6); \\
&P2aux3 = zeros(N, 6); \\
&P2aux4 = zeros(N, 6); \\
&\\
&\quad for\ i = 1:16 \\
&\qquad for\ j = 1:16 \\
&\\
&\qquad TTTT2(:, 2*i-1, 2*j-1, :) = T3(:, i, j, :); \\
&\qquad TTTT2(:, 2*i-1, 2*j, :) = T3(:, i, j, :); \\
&\qquad TTTT2(:, 2*i, 2*j-1, :) = T3(:, i, j, :); \\
&\qquad TTTT2(:, 2*i, 2*j, :) = T3(:, i, j, :);
\end{aligned}
$$

$P3aux(:,:) = uint8(P3(:,i,j,:));$

$P2aux1(:,:) = P3aux(:,:);$
$P2aux2(:,:) = P3aux(:,:)/2;$
$P2aux3(:,:) = P3aux(:,:)/3;$
$P2aux4(:,:) = P3aux(:,:)/4;$

$P2aux1(P2aux1 \sim= 1) = 0;$
$P2aux2(P2aux2 \sim= 1) = 0;$
$P2aux3(P2aux3 \sim= 1) = 0;$
$P2aux4(P2aux4 \sim= 1) = 0;$

$P2(:, 2*i-1, 2*j-1, :) = P2aux1(:,:);$
$P2(:, 2*i-1, 2*j, :) = P2aux2(:,:);$
$P2(:, 2*i, 2*j-1, :) = P2aux3(:,:);$
$P2(:, 2*i, 2*j, :) = P2aux4(:,:);$

end

end

$TTT2 = TTTT2. * P2;$

Layer 23 $\Rightarrow$ Layer 22

$TT^{(2)} = TTT^{(2)}. * ELUD(Y^{(22)});$

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial x}Y^{(22)k_2}_{e|u_2,v_2},$$

Layer 22 $\Rightarrow$ Layer 21

$$\frac{\partial}{\partial x}J = \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} T^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial x}Z^{(21)k_2}_{e|u_2,v_2},$$ where $T^{(2)}$ is defined as:

$$T^{(2)k_2}_{e|u_2,v_2} = TT^{(2)k_2}_{e|u_2,v_2}\frac{\alpha^{(2)k_2}_{u_2,v_2}}{\sqrt{V^{(21)k_2}_{u_2,v_2}+\epsilon}} - \frac{1}{N}\frac{\alpha^{(2)k_2}_{u_2,v_2}}{\sqrt{V^{(21)k_2}_{u_2,v_2}+\epsilon}}\sum_{e'=1}^{N} TT^{(2)k_2}_{e'|u_2,v_2}$$
$$- \frac{1}{N}\frac{\alpha^{(2)k_2}_{u_2,v_2}\left(Z^{(21)k_2}_{e|u_2,v_2}-\mu^{(21)k_2}_{u_2,v_2}\right)}{\left(V^{(21)k_2}_{u_2,v_2}+\epsilon\right)^{3/2}}\left(\sum_{e'=1}^{N} TT^{(2)k_2}_{e'|u_2,v_2}Z^{(21)k_2}_{e'|u_2,v_2} - \mu^{(21)k_2}_{u_2,v_2}\sum_{e'=1}^{N} TT^{(2)k_2}_{e'|u_2,v_2}\right).$$

20 Final formulae for derivatives with batch normalization

When using batch norm, the table of learnable parameters needs to be expanded to include additional learnable parameters α and β:

Learnable parameter	**Dimensions**	**Size**
$b^{(1)k_2}$	1	6 $(k_2 = 1:6)$
$w^{(1)k_2}_{o_1,e_1}$	3	5x5x6 $(o_1 = 1:5, e_1 = 1:5, k_2 = 1:6)$
$b^{(3)k_4}$	1	16 $(k_4 = 1:16)$
$w^{(3)k_4,k_3}_{o_3,e_3}$	4	5x5x16x6 $(o_3 = 1:5, e_3 = 1:5, k_4 = 1:16, k_3 = 1:6)$
$b^{(6)}_{s_7}$	1	65 $(s_7 = 1:65)$
$\Theta^{(6)}_{s_7,s_6}$	2	65x1024 $(s_7 = 1:65, s_6 = 1:1024)$
$b^{(7)}_{s_8}$	1	12 $(s_8 = 1:12)$
$\Theta^{(7)}_{s_8,s_7}$	2	12x65 $(s_8 = 1:12, s_7 = 1:65)$
$\alpha^{(2)k_2}_{u_2,v_2}$	3	32x32x6 $(u_2 = 1:32, v_2 = 1:32, k_2 = 1:6)$
$\beta^{(2)k_2}_{u_2,v_2}$	3	32x32x6 $(u_2 = 1:32, v_2 = 1:32, k_2 = 1:6)$
$\alpha^{(4)k_4}_{u_4,v_4}$	3	16x16x16 $(u_4 = 1:16, v_4 = 1:16, k_4 = 1:16)$
$\beta^{(4)k_4}_{u_4,v_4}$	3	16x16x16 $(u_4 = 1:16, v_4 = 1:16, k_4 = 1:16)$
$\alpha^{(7)}_{s_7}$	1	65 $(s_7 = 1:65)$
$\beta^{(7)}_{s_7}$	1	65 $(s_7 = 1:65)$

Formulae for partial derivatives with respect to "old" parameters" (i.e. parameters which we had without batch norm - $b^{(1)}, w^{(1)}, b^{(3)}, w^{(3)}, b^{(6)}, \Theta^{(6)}, b^{(7)}, \Theta^{(7)}$) look exactly the same as before (with the only difference that $A^{(7)}$ is now called $A^{(73)}$). All effects of batch norm are hidden within

T-matrices. For readers convenience below we provide those formulae again:

$$\frac{\partial}{\partial b^{(1)k}}J = \sum_{e=1}^{N}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} T^{(2)k}_{e|u_2,v_2};$$

$$\frac{\partial}{\partial w^{(1)k}_{s,s'}}J = \sum_{e=1}^{N}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} T^{(2)k}_{e|u_2,v_2} AA^{(1)}_{e|u_2+s-1,v_2+s'-1};$$

$$\frac{\partial}{\partial b^{(3)k}}J = \sum_{e=1}^{N}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} T^{(4)k}_{e|u_4,v_4};$$

$$\frac{\partial}{\partial w^{(3)k,k'}_{s,s'}}J = \sum_{e=1}^{N}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} T^{(4)k}_{e|u_4,v_4} AA^{(3)k'}_{e|u_4+s-1,v_4+s'-1};$$

$$\frac{\partial}{\partial b^{(6)}_{i}}J = \sum_{e=1}^{N} T^{(7)}_{e|i};$$

$$\frac{\partial}{\partial \Theta^{(6)}_{i,j}}J = \sum_{e=1}^{N} T^{(7)}_{e|i} A^{(6)}_{e|j} = \left(T^{(7)^T} * A^{(6)}\right)_{i,j};$$

$$\frac{\partial}{\partial b^{(7)}_{i}}J = \sum_{e=1}^{N} T^{(8)}_{e|i};$$

$$\frac{\partial}{\partial \Theta^{(7)}_{i,j}}J = \sum_{e=1}^{N} T^{(8)}_{e|i} A^{(7)}_{e|j} = \left(T^{(8)^T} * A^{(73)}\right)_{i,j}.$$

Formulae for partial derivatives with respect to "new parameters" α and β need to be calculated. For derivatives with respect to $\alpha^{(2)}$ and $\beta^{(2)}$ we will need the 22^{nd} layer formula:

$$\begin{aligned}
\frac{\partial}{\partial \alpha^{(2)k}_{u,v}}J &= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial \alpha^{(2)k}_{u,v}} Y^{(22)k_2}_{e|u_2,v_2} \\
&= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial \alpha^{(2)k}_{u,v}}\left(\alpha^{(2)k_2}_{u_2,v_2} Z^{(22)k_2}_{e|u_2,v_2} + \beta^{(2)k_2}_{u_2,v_2}\right) \\
&= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2} Z^{(22)k_2}_{e|u_2,v_2}\delta_{kk_2}\delta_{uu_2}\delta_{vv_2} \\
&= \sum_{e=1}^{N} TT^{(2)k}_{e|u,v} Z^{(22)k}_{e|u,v};
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \beta^{(2)k}_{u,v}}J &= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial \beta^{(2)k}_{u,v}} Y^{(22)k_2}_{e|u_2,v_2} \\
&= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\frac{\partial}{\partial \beta^{(2)k}_{u,v}}\left(\alpha^{(2)k_2}_{u_2,v_2} Z^{(22)k_2}_{e|u_2,v_2} + \beta^{(2)k_2}_{u_2,v_2}\right) \\
&= \sum_{e=1}^{N}\sum_{k_2=1}^{6}\sum_{u_2=1}^{32}\sum_{v_2=1}^{32} TT^{(2)k_2}_{e|u_2,v_2}\delta_{kk_2}\delta_{uu_2}\delta_{vv_2} \\
&= \sum_{e=1}^{N} TT^{(2)k}_{e|u,v};
\end{aligned}$$

For derivatives with respect to $\alpha^{(4)}$ and $\beta^{(4)}$ we will need the 42^{nd} layer formula:

$$\begin{aligned}
\frac{\partial}{\partial \alpha^{(4)k}_{u,v}}J &= \sum_{e=1}^{N}\sum_{k_4=1}^{16}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} TT^{(4)k_4}_{e|u_4,v_4}\frac{\partial}{\partial \alpha^{(4)k}_{u,v}} Y^{(42)k_4}_{e|u_4,v_4} \\
&= \sum_{e=1}^{N}\sum_{k_4=1}^{16}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} TT^{(4)k_4}_{e|u_4,v_4}\frac{\partial}{\partial \alpha^{(4)k}_{u,v}}\left(\alpha^{(4)k_4}_{u_4,v_4} Z^{(42)k_4}_{e|u_4,v_4} + \beta^{(4)k_4}_{u_4,v_4}\right) \\
&= \sum_{e=1}^{N}\sum_{k_4=1}^{16}\sum_{u_4=1}^{16}\sum_{v_4=1}^{16} TT^{(4)k_4}_{e|u_4,v_4} Z^{(42)k_4}_{e|u_4,v_4}\delta_{kk_4}\delta_{uu_4}\delta_{vv_4}
\end{aligned}$$

$$= \sum_{e=1}^{N} TT_{e|u,v}^{(4)k} Z_{e|u,v}^{(42)k};$$

$$\begin{aligned}
\frac{\partial}{\partial \beta_{u,v}^{(4)k}} J &= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e|u_4,v_4}^{(4)k_4} \frac{\partial}{\partial \beta_{u,v}^{(4)k}} Y_{e|u_4,v_4}^{(42)k_4} \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e|u_4,v_4}^{(4)k_4} \frac{\partial}{\partial \beta_{u,v}^{(4)k}} \left(\alpha_{u_4,v_4}^{(4)k_4} Z_{e|u_4,v_4}^{(42)k_4} + \beta_{u_4,v_4}^{(4)k_4} \right) \\
&= \sum_{e=1}^{N} \sum_{k_4=1}^{16} \sum_{u_4=1}^{16} \sum_{v_4=1}^{16} TT_{e|u_4,v_4}^{(4)k_4} \delta_{kk_4} \delta_{uu_4} \delta_{vv_4} \\
&= \sum_{e=1}^{N} TT_{e|u,v}^{(4)k};
\end{aligned}$$

For derivatives with respect to $\alpha^{(7)}$ and $\beta^{(7)}$ we will need the 72^{nd} layer formula:

$$\begin{aligned}
\frac{\partial}{\partial \alpha_i^{(7)}} J &= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} \frac{\partial}{\partial \alpha_i^{(7)}} Y_{e|s_7}^{(72)} \\
&= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} \frac{\partial}{\partial \alpha_i^{(7)}} \left(\alpha_{s_7}^{(7)} Z_{e|s_7}^{(72)} + \beta_{s_7}^{(7)} \right) \\
&= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} Z_{e|s_7}^{(72)} \delta_{is_7} \\
&= \sum_{e=1}^{N} H_{e|i}^{(7)} Z_{e|i}^{(72)};
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \beta_i^{(7)}} J &= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} \frac{\partial}{\partial \beta_i^{(7)}} Y_{e|s_7}^{(72)} \\
&= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} \frac{\partial}{\partial \beta_i^{(7)}} \left(\alpha_{s_7}^{(7)} Z_{e|s_7}^{(72)} + \beta_{s_7}^{(7)} \right) \\
&= \sum_{e=1}^{N} \sum_{s_7=1}^{65} H_{e|s_7}^{(7)} \delta_{is_7} \\
&= \sum_{e=1}^{N} H_{e|i}^{(7)}.
\end{aligned}$$