D7047E exercise 1

Rasmus Högström
rashgs-5@student.ltu.se

April 2020

Theoretical assignment

We're given a image I and a kernel k. We use zero-padding to manage the margins and retain the size of the convoluted image. And I becomes

$$I_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 1 & 0 \\ 0 & 1 & -3 & -4 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The convolution $I_0 * k$ uses **stride(1,1)** which means that the kernel takes equal steps of 1 along either axis when performing the convolution. Which is a sum of element-wise product between the kernel and a subsection of the image of the same size as the kernel. I did this by hand and the result is seen below

$$I_0 * k = \begin{bmatrix} 4 & 5 & 6 & 4 \\ 5 & 3 & 3 & 6 \\ 1 & -7 & -7 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix}$$

We apply the activation function **ReLU** to this. $\mathbf{ReLU}(x)$ is defined as 0 for $x < 0$ and x for $x >= 0$ which yields

$$\mathbf{ReLU}(I_0 * k) = \begin{bmatrix} 4 & 5 & 6 & 4 \\ 5 & 3 & 3 & 6 \\ 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix}$$

Next step is *pooling* which is a way of mapping the convoluted image to a lower order. In our case the image is 4x4 and the pooling 2x2 using **stride(2,2)** meaning we'll get a 2x2 output. The function of the pooling is the **max** function taking a 2x2 matrix and returning the highest value (no need to consider magnitude of negative values since we've used **ReLU**). This gives

$$\mathbf{max\_pool}(\mathbf{ReLU}(I_0 * k)) = \begin{bmatrix} 5 & 6 \\ 4 & 4 \end{bmatrix}$$

Lets call this A. Now we want to flatten A to a vector, we simply cut and attach consecutive rows in A at the end of the first row, $A_{flat} = [\mathbf{row1}, \mathbf{row2}, ..., \mathbf{rowN}]^T$. This gives

$$A_{flat} = \begin{bmatrix} 5 \\ 6 \\ 4 \\ 4 \end{bmatrix}$$

Now we're given a weight-matrix W as

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

These are the weights for the fully connected connection between our features in A_{flat} and the output layer. To calculate this output we calculate the matrix multiplication

$$WA_{flat} = \begin{bmatrix} 45 \\ 121 \end{bmatrix}$$

Finally we apply a **softmax** function to this output to get a more well behaved output for our purposes. Each element i in v is evaluated in the **softmax** function as $\mathbf{exp}(i)/\mathbf{sum}(\mathbf{exp}(v))$ where $\mathbf{exp}(v)$ is element-wise. This gives us

$$\mathbf{softmax}\left(\begin{bmatrix} 45 \\ 121 \end{bmatrix}\right) = \begin{bmatrix} \frac{1}{1+e^{76}} \\ \\ \frac{e^{76}}{1+e^{76}} \end{bmatrix} = \begin{bmatrix} 9.85 \cdot 10^{-34} \\ 1 - 9.85 \cdot 10^{-34} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

This answer indicates the second class. At my gitpage for this exercise is my handwritten solution for this exercise. somewhat transcribed from my initial messy solution.