Example of convolution operation:

The convolution is between a kernel (Filter) and the image. The result is called feature map.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 31 | 19 | 30 | 21 |
| 31 | 25 | 30 | 25 |
| 31 | 28 | 30 | 28 |
| 31 | 22 | 32 | 22 |

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Input (6x6) Kernel (3x3) Output(4x4)

In the above example we have an input image of size 6x6 convolved with kernel filter of size 3x3 with strides = 1, padding = 0. The result is an output feature map of size 4x4.

C = ((n-f+2p)/s) +1 🡪C = size of the output. n = size of input matrix, f = size of kernel, p = padding amount, s = stride applied.

* There is an undesirable side effect in filtering with a kernel (border effect problem). The size of the filtered image is smaller than that of the original. This problem worsens as the kernel size and/or the stride get bigger. As a measure to prevent the size reduction problem, we can add some “padding” to the outer boundary of the image. Padding consists in attaching cells filled with value 0. The required padding size obviously depends on the kernel size.
* Increasing strides makes convolution faster with less operations to fulfill and downsizes the feature map but might cause losing important features from the image.
* Adding pooling layers also down samples the feature image but without losing as much information as increasing the stride in convolutional layers.
* Pooling layers: low version of feature maps where detecting the presence of certain features is more important than determining their exact position.
* Dropout regularization: At every iteration, it randomly removes specific neurons from a dropout layer.

Let us apply a max pooling layer to our previous feature map output. the max pooling applied has a filter of 2x2 and a stride of 2x2.

|  |  |  |  |
| --- | --- | --- | --- |
| 31 | 19 | 30 | 21 |
| 31 | 25 | 30 | 25 |
| 31 | 28 | 30 | 28 |
| 31 | 22 | 32 | 22 |

|  |  |
| --- | --- |
| 31 | 30 |
| 31 | 32 |

🡪

**Suppose you have the following dataset:**

|  |  |  |
| --- | --- | --- |
| Blood pressure | Height | Weight |
| 130 | 1.64 | 66.5 |
| 140 | 1.70 | 80 |
| 125 | 1.66 | 62 |
| 110 | 1.80 | 77 |
| 127 | 1.90 | 85 |

**Assuming that the weight vector of the perceptron is and the bias is 0, write the equation of the perceptron and what can you tell about feature scale?**

**and** is the step activation function used with the perceptron. There is a feature scale problem.

**Write the training set that results after the application of the preprocessing formula below.**

**array([[0.37, -1, -0.88 ],**

**[ 1.4, -0.4, 0.68],**

**[-0.14, -0.8, -1.4],**

**[ -1.69, 0.6, 0.33],**

**[0.06, 1.6, 1.26]])**

What is next?

1. **After scaling, we need to retrain the perceptron on the new scaled data to get new weights.**
2. **Training on a scaled data leads to a faster convergence.**