**A Gentle Introduction to Pooling Layers for Convolutional Neural Networks**

<https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>

[Convolutional layers](https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/) in a convolutional neural network summarize the presence of features in an input image.

A problem with the output feature maps is that they are sensitive to the location of the features in the input. One approach to address this sensitivity is to down sample the feature maps. This has the effect of making the resulting down sampled feature maps more robust to changes in the position of the feature in the image, referred to by the technical phrase “*local translation invariance*.”

Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.

In this tutorial, you will discover how the pooling operation works and how to implement it in convolutional neural networks.

After completing this tutorial, you will know:

* Pooling is required to down sample the detection of features in feature maps.
* How to calculate and implement average and maximum pooling in a convolutional neural network.
* How to use global pooling in a convolutional neural network.

This tutorial is divided into five parts; they are:

1. Pooling
2. Detecting Vertical Lines
3. Average Pooling Layers
4. Max Pooling Layers
5. Global Pooling Layers

## POOLING LAYERS

Convolutional layers in a convolutional neural network systematically apply learned filters to input images in order to create feature maps that summarize the presence of those features in the input.

Convolutional layers prove(d to be) very effective, and stacking convolutional layers in deep models allows layers close to the input to learn low-level features (e.g. lines) and layers deeper in the model to learn high-order or more abstract features, like shapes or specific objects.

A limitation of the feature map output of convolutional layers is that they record the precise position of features in the input. This means that small movements in the position of the feature in the input image will result in a different feature map. This can happen with re-cropping, rotation, shifting, and other minor changes to the input image.

A common approach to addressing this problem from signal processing is called 'down sampling'. This is where a lower resolution version of an input signal is created that still contains the large or important structural elements, without the fine detail that may not be as useful to the task.

Down sampling can be achieved with convolutional layers by changing the [stride of the convolution across the image](https://machinelearningmastery.com/padding-and-stride-for-convolutional-neural-networks/). A more robust and common approach is to use a pooling layer.

A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer; for example the layers in a model may look as follows:

1. Input Image
2. Convolutional Layer
3. Nonlinearity
4. Pooling Layer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model.

The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps.

Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2×2 pixels applied with a stride of 2 pixels.

This means that the pooling layer will always reduce the size of each feature map by a factor of 2, e.g. each dimension is halved, reducing the number of pixels or values in each feature map to one quarter the size. For example, a pooling layer applied to a feature map of 6×6 (36 pixels) will result in an output pooled feature map of 3×3 (9 pixels).

The pooling operation is specified, rather than learned. Two common functions used in the pooling operation are:

* **Average Pooling**: Calculate the average value for each patch on the feature map.
* **Maximum Pooling (or Max Pooling)**: Calculate the maximum value for each patch of the feature map.

The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model’s invariance to local translation.

*In all cases, pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change.*

— Page 342, [Deep Learning](https://amzn.to/2Dl124s), 2016.

**SPECIFIC EXAMPLES**

## DETECTING VERTICAL LINES

Ovo poglavlje samo pročitat.

## AVERAGE POOLING LAYER

On two-dimensional feature maps, pooling is typically applied in 2×2 patches of the feature map with a stride of (2,2).

Average pooling involves calculating the average for each patch of the feature map. This means that each 2×2 square of the feature map is down sampled to the average value in the square.

(deeplizard)

Given the (2,2) stride, the operation would (when it reaches the end of the first line) then be moved down two rows and back to the first column and the process continued.

Because the downsampling operation halves each dimension, we will expect the output of pooling applied to the 6×6 feature map to be a new 3×3 feature map. Given the horizontal symmetry of the feature map input, we would expect each row to have the same average pooling values.

*This can be achieved in Keras by using the AveragePooling2D layer. The default pool\_size (e.g. like the kernel size or filter size) of the layer is (2,2) and the default strides is None, which in this case means using the pool\_size as the strides, which will be (2,2).*

Average pooling works well, although it is more common to use max pooling.

## MAX POOLING LAYER

Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map.

The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature in the case of average pooling. This has been found to work better in practice than average pooling for computer vision tasks like image classification.

*In a nutshell, the reason is that features tend to encode the spatial presence of some pattern or concept over the different tiles of the feature map (hence, the term feature map), and it’s more informative to look at the maximal presence of different features than at their average presence.*

— Page 129, [Deep Learning with Python](https://amzn.to/2Dnshvc), 2017.

## GLOBAL POOLING LAYERS

There is another type of pooling that is sometimes used called global pooling.

Instead of down sampling patches of the input feature map, global pooling down samples the entire feature map to a single value. This would be the same as setting the *pool\_size* to the size of the input feature map.

Global pooling can be used in a model to aggressively summarize the presence of a feature in an image. It is also sometimes used in models as an alternative to using a fully connected layer to transition from feature maps to an output prediction for the model.

*Both global average pooling and global max pooling are supported by Keras via the GlobalAveragePooling2D and GlobalMaxPooling2D classes respectively.*

In this tutorial, you discovered how the pooling operation works and how to implement it in convolutional neural networks.

Specifically, you learned:

* Pooling is required to down sample the detection of features in feature maps.
* How to calculate and implement average and maximum pooling in a convolutional neural network.
* How to use global pooling in a convolutional neural network.