Survey on Convolutional Neural Networks and Pooling Strategies

Romain Hermary

July 2021

Abstract

When talking about object detection, image analysis, image classification and essentially every algorithm that tries to apply the conveniences of deep learning and back-propagation in signal related fields, Convolutional Neural Networks (CNN) appear to be the best contenders. Throughout the years, many variants were proposed with architectures and layer arrangements very peculiar and different, but also diverse pooling/sub-sampling methods, which CNNs heavily rely on to increase invariance to distortions and have a broader feature detection. In this paper, we overview the structure of Neural Networks and Convolutional Neural Networks, review the existing approaches of pooling layers and present possible future directions.

1 Introduction

CNNs started with the idea of mimicking the natural visual perception mechanism of the living creatures (Fukushima & Miyake, 1982; Hubel & Wiesel, 1968), and was first introduced in the way we know them today by LeCun et al. (1989), with the LeNet network model. To be more specific, CNNs try to detect images' distinct features that are key-points to describe the picture and derive conclusions – i.e, in recurring application, what is on it.

From this point and thanks to the ever growing datasets and improving computing capabilities, CNNs are now a big part in artificial intelligence, and researches on the subject to improve the used methods, algorithms and architectures allow a constant evolution of networks models; to cite a few major breakthroughs or references, AlexNet (Russakovsky et al., 2015), VGGNet (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), ZSFNet (Zeiler & Fergus, 2014) and ResNet (He, Zhang, Ren, & Sun, 2016).

In this survey, we will cover the base structure of CNNs with greater focus on pooling layers and their usefulness, with an overview of existing pooling methods, finishing with the study of a potential CNN alternative.

2 Neural Networks: Underlying Secrets

The idea of a neural network is to evaluate an input and give an opinion on what was given to it; of course, its answer depends drastically on what data it was given

to train on and the output the developer expected for each entry. One could see a neural network as a bunch of parameters which are just numbers called *weights*, and each of them acts as a feature detector, i.e, when different entries are given, the combination of the weights' and entries' values will give different results, thus making it possible to make decisions.

For the network to learn and improve the value of those weights, it compares its prediction to the expected output and changes the weights' values according to the error. In fact, this error is calculated with a function and the net's goal is to find the weights' values that minimize the error, that is, finding the minimum of this multidimensional function.

3 Less Weights, More Convolutions

Analyzing a signal like an image would require a crazy amount of weights with a classic Neural Network as of the great number of pixels in a picture and the complexity of the perception process, causing a difficult training. Fortunately, CNNs are well suited for this task and overcome some of the problems one could face with classic perceptrons.

CNNs have limitless architectures possibilities, but the core idea is to chain convolution layers and pooling layers – with an activation function in between, and ending in fully connected layers.

3.1 Convolution Layers

Firstly, the convolution layer. Its biggest asset is to introduce shared weights between every pixel of the image; indeed, a convolution layer is composed of multiple *kernels*, small matrices, each one takes the role of a feature detector and its output (feature map) serves for the decision making.

The convolution process to get the feature map is very straightforward, as it only consists in sliding the kernel and centering it on each pixel, multiplying the overlapped values together and summing the results to get the corresponding feature map's pixel value.

This structure of kernels shared by each pixel have the advantage of making it easier to train and find a global solution by reducing the model complexity, but also helps with spacial discrepancy. There exists multiple types and improvements of convolutions such as tiled convolution (Le et al., 2010), transposed convolution (Zeiler & Fergus, 2014), dilated convolution (F. Yu & Koltun, 2016), and other improvements over the layer architecture, each one aiming at improving the spatial invariance character of the layer and even more (Gu et al., 2018).

3.2 Pooling layers

"Reducing the precision is actually advantageous, since a slight distortion or translation of the input will have reduced effect on the representation" (LeCun et al., 1989), that's why pooling layers always go hand in hand with a convolution layer. The final objective is always to reduce the feature map size and resolution to introduce a certain level of invariance to distortions and translations, increase the receptive field size, but different techniques exist to help carrying the more information possible

to the depth of the network, thus making it possible to stack layers robustly.

Starting with the ***average pooling***, introduced in LeCun et al. (1989), this layer performs down sampling by dividing the input into rectangular pooling regions and computing the average values of each region. This has the tendency of smoothing the feature map.

Also widely use and with a similar process, there is the ***max pooling*** (Ranzato, Boureau, & LeCun, 2007), which instead of taking the average of the region value, takes the maximum.

Combining the too approaches, the ***mixed pooling*** (D. Yu, Wang, Chen, & Wei, 2014) introduces a stochastic procedure by randomly using the conventional max pooling and average pooling methods.

L_p ***pooling***, a biologically inspired pooling (Hyvärinen & Köster, 2007), is a pooling using a weighted average of the feature map's regions. The parameter p can be tuned for the layer to mimic max pooling or average pooling, but its generalization ability is claimed to be better than max pooling (Bruna, Szlam, & LeCun, 2014).

Instead of picking the maximum value within each pooling region as max pooling does, ***stochastic pooling*** (Zeiler & Fergus, 2013) randomly picks the activations according to a multinomial distribution, which ensures that the non-maximal activations of feature maps are also possible to be utilized. Compared with max pooling, stochastic pooling can avoid over-fitting due to the stochastic component.

Spectral pooling (Rippel, Snoek, & Adams, 2015) performs dimensionality reduction by cropping the representation of input in frequency domain. Spectral pooling first computes the *discrete Fourier transform* (DFT) of the input feature map, then crops the frequency representation by maintaining only the central submatrix of the frequencies, and finally uses inverse DFT to map the approximation back into spatial domain. Compared with max pooling, the linear low-pass filtering operation of spectral pooling can preserve more information for the same output dimensionality.

Spatial Pyramid Pooling (He, Zhang, Ren, & Sun, 2015) (***SPP***) pools the input's feature map in local spatial bins with sizes proportional to the image size, resulting in a fixed number of bins, thus the key advantage of SPP is that it can generate a fixed-length representation regardless of the input sizes.

Multi-scale Orderless Pooling (***MOP***) (Gong, Wang, Guo, & Lazebnik, 2014) improves the invariance of CNNs without degrading their discriminative power. It extracts deep activation features for both the whole image and local patches of several scales.The activations of local patches are aggregated by VLAD encoding (Jégou et al., 2012), which aims to capture more local, fine-grained details of the image as well as enhancing invariance.

Detail-Preserving Pooling (***DDP***) (Saeedan, Weber, Goesele, & Roth, 2018) appeared inspired by the human visual system, which focuses on local spatial changes, to propose an adaptive pooling method that magnifies spatial changes and preserves important structural detail.

Local Importance-based Pooling (***LIP***) (Gao, Wang, & Wu, 2019) is a layer able to adaptively determine which features are more important to be kept through downsampling. For instance, LIP enables the network to preserve features of tiny targets while discarding false activations of the background clutter when recognizing

or detecting small objects. Moreover, LIP is a more generic pooling method than the existing methods, in sense that it is capable of mimicking the behavior of average pooling, max pooling and detail-preserving pooling.

4 Area of Improvement: Morphological Neural Networks

Mathematical morphology operations are techniques part of computer vision and image algebra (Ritter & Wilson, 1996). The combination of those operations and neural networks arose theoretically with Davidson and Ritter (1990) and concretely latter on with Davidson and Hummer (1993). Those premises of a promising field of study do not fails to generate enthusiasm in the mathematical morphology community, and we start to see networks implementations with the concern of finding the best way to mix those operations and deep learning (Masci, Angulo, & Schmidhuber, 2013).

Studies start to arise comparing the efficiency of CNNs compared to that of MNNs, (Franchi, Fehri, & Yao, 2020), and regarding the de-noising capability of those latter, MNNs are better; we could see other applications as of image analysis and object detection appear on further works, like the inspiring layers of Kirszenberg, Tochon, Puybareau, and Angulo (2021) which could in a short term be improved or included, mixed with other networks and layers, to outperform today standards.

5 Conclusion

We have seen in this survey the basics of Neural Networks and Convolutional Neural Networks, focusing on the main existing pooling layers, and highlighting a promising field of study that are Morphological Neural Networks. CNNs are still evolving throughout the time and very quickly, researchers are perpetually trying to push this technology further, and it already made a big breakthrough in image processing.

References

Bruna, J., Szlam, A. D., & LeCun, Y. (2014). Signal recovery from pooling representations. In *Icml.*

Davidson, J., & Hummer, F. (1993). Morphology neural networks: An introduction with applications. *Circuits, Systems and Signal Processing*, *12*, 177-210.

Davidson, J., & Ritter, G. (1990). Theory of morphological neural networks. In *Photonics west - lasers and applications in science and engineering.*

Franchi, G., Fehri, A., & Yao, A. (2020). Deep morphological networks. *Pattern Recognit.*, *102*, 107246.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition..

Gao, Z., Wang, L., & Wu, G. (2019). Lip: Local importance-based pooling. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3354-3363.

Gong, Y., Wang, L., Guo, R., & Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *Eccv.*

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognit.*, *77*, 354-377.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*, 1904-1916.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

Hubel, D., & Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*.

Hyvärinen, A., & Köster, U. (2007). Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems*, *18*, 100 - 81.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., & Schmid, C. (2012). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 1704-1716.

Kirszenberg, A., Tochon, G., Puybareau, É., & Angulo, J. (2021). Going beyond p-convolutions to learn grayscale morphological operators. In *Dgmm.*

Le, Q. V., Ngiam, J., Chen, Z., Chia, D. J., Koh, P. W., & Ng, A. (2010). Tiled convolutional neural networks. In *Nips.*

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. In *Nips.*

Masci, J., Angulo, J., & Schmidhuber, J. (2013). A learning framework for morphological operators using counter-harmonic mean. In *Ismm.*

Ranzato, M., Boureau, Y.-L., & LeCun, Y. (2007). Sparse feature learning for deep belief networks. In *Nips.*

Rippel, O., Snoek, J., & Adams, R. P. (2015). Spectral representations for convolutional neural networks. In *Nips.*

Ritter, G., & Wilson, J. N. (1996). Handbook of computer vision algorithms in image algebra..

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211-252.

Saeedan, F., Weber, N., Goesele, M., & Roth, S. (2018). Detail-preserving pooling in deep networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9108-9116.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *CoRR*, *abs/1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

Yu, D., Wang, H., Chen, P., & Wei, Z. (2014). Mixed pooling for convolutional neural networks. In *Rskt.*

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. *CoRR*, *abs/1511.07122*.

Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *CoRR*, *abs/1301.3557*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Eccv*.