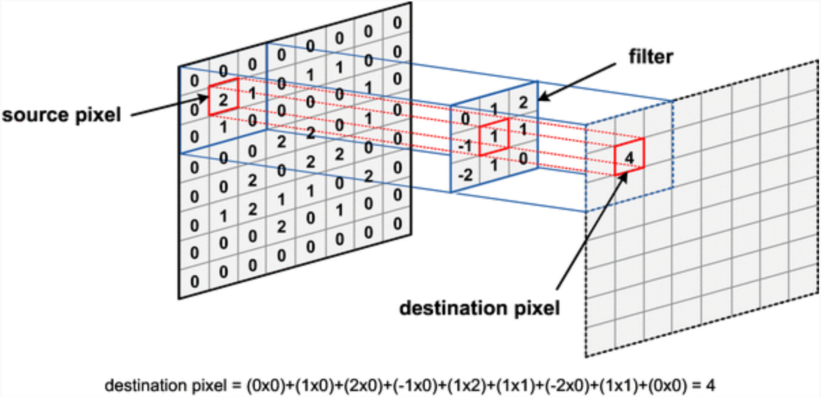
Pooling Methods in Convolutional Neural Networks

Romain Hermary

July 2021

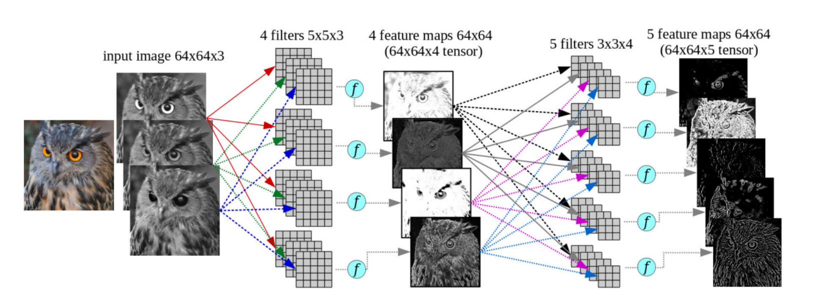
Introduction

Convolution



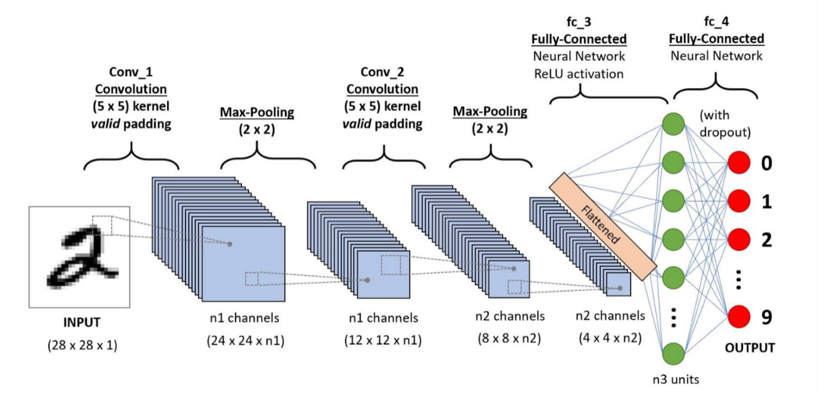
[Banharnsakun, 2019]

Convolution



[Ponti et al., 2017]

Convolutional Neural Network (CNN)



Towards Data Science

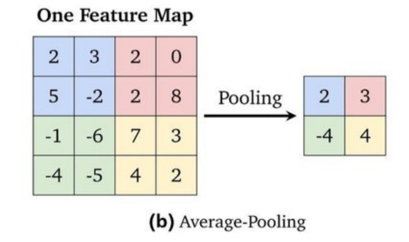
Pooling Layer

Reducing feature map resolution

- Helps at increasing distortions invariance
- Reduce computational complexity

Pooling Methods

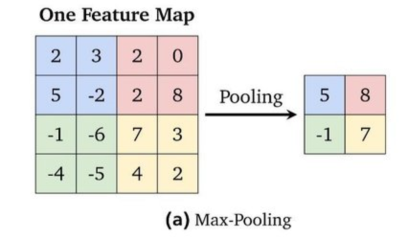
Average Pooling [LeCun et al., 1989]



[Guissous, 2019]

- Takes every neighbor values into account
- Areas of high activation are down-weighted

Max Pooling [Ranzato et al., 2007]



[Guissous, 2019]

- *Idea*: Areas of interest are of high intensity
- Globally better than average-pooling
- Does not take low intensities into consideration
- Worse at preserving localization

Stochastic Pooling [Zeiler and Fergus, 2013]

- Calculates probabilities by normalizing the activations within the region

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$$

- Multinomial distribution selects an activation value within the region

$$s_j = a_l \quad \text{where } l \sim P(p_1, \ldots, p_{|R_j|})$$

- Gives higher chances to stronger activations
- Includes the non-maximal activations
- Prohibits overfitting because of the stochastic component
- Performs better than max pooling

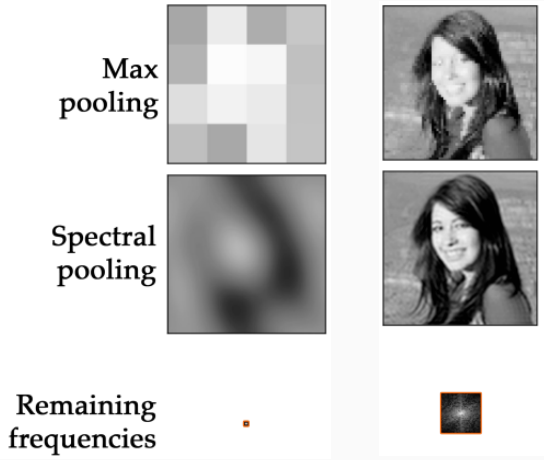
Mixed Pooling [Yu et al., 2014]

$$y_{kij} = \lambda \cdot \max_{(p,q)\in\mathcal{R}_{ij}} x_{kpq} + (1-\lambda) \cdot \frac{1}{|\mathcal{R}_{ij}|} \sum_{(p,q)\in\mathcal{R}_{ij}} x_{kpq},$$

- $\lambda \in \{0, 1\}$, picked randomly
- $\lambda = 0$, average pooling
- $\lambda = 1$, max pooling

- Stochastic procedure
- Prohibits overfitting
- Performs better than average, max, stochastic pooling

Spectral Pooling [Rippel et al., 2015]



Pooling result by cropping image dimensionality (max pooling) or frequency domain matrix (spectral pooling)

- Faster convergence
- Pooling to any desired output dimensionality while retaining significantly more information
- Incessant domain switching

Detail-Preserving Pooling (DPP) [Saeedan et al., 2018]

$$\mathcal{D}_{\alpha,\lambda}(I)[p] = \frac{1}{\sum_{q' \in \Omega_p} w_{\alpha,\lambda}[p, q']} \sum_{q \in \Omega_p} w_{\alpha,\lambda}[p, q] I[q]$$

$$w_{\alpha,\lambda}[p, q] = \alpha + \rho_\lambda \left(I[q] - \tilde{I}[p] \right)$$

$$\rho_{\text{Sym}}(x) = \left(\sqrt{x^2 + \epsilon^2} \right)^\lambda$$

Computes a spatially weighted average of the input nodes in a neighborhood, weights depending on pixel values distances

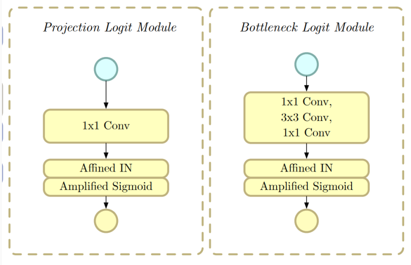
- Aims to preserve small details
- Performs at least as good than standard pooling layers
- Does not disrupt the flow of gradients of the backward pass
- Stochastic regularization techniques can be integrated
- The more detailed features might be the less discriminative ones

Local Importance-based Pooling (LIP) [Gao et al., 2019]

$$O_{x',y'} = \frac{\sum_{(\Delta x, \Delta y) \in \Omega} I_{x+\Delta x, y+\Delta y} \exp(\mathcal{G}(I))_{x+\Delta x, y+\Delta y}}{\sum_{(\Delta x, \Delta y) \in \Omega} \exp(\mathcal{G}(I))_{x+\Delta x, y+\Delta y}}$$

Computes weighted average over neighborhood

Where $\mathcal{G}$ is the one of the following module with learnable components:



- Automatically enhance discriminative features during the downsampling procedure by learning adaptive importance weights based on inputs
- Yields notable gains with different depths and different architectures on classification tasks

Remark

There are many other existing method for pooling:

- L_p pooling, a biologically inspired pooling [Hyvärinen and Köster, 2007]
- Spatial Pyramid Pooling (SPP) [He et al., 2015]
- Multi-scale Orderless Pooling (MOP) [Gong et al., 2014]
- Super-pixel Pooling [Ren and Malik, 2003]
- PCA Networks [Chan et al., 2015]
- Compact Bilinear Pooling [Lin et al., 2015]
- Edge-aware Pyramid Pooling [Xu et al., 2019]
- Lead Asymmetric Pooling (LAP) [Liu et al., 2018]
- ...

Other Way for Image Analysis

Progress

Morphological Neural Networks

- *Morphology neural networks: An introduction with applications* [Davidson and Hummer, 1993]
- *A Learning Framework for Morphological Operators Using Counter-Harmonic Mean* [Masci et al., 2013]
- *Deep morphological networks* [Franchi et al., 2020]
- *Going beyond p-convolutions to learn grayscale morphological operators* [Kirszenberg et al., 2021]

Conclusion

Conclusion

The choice of the pooling layer depends on:

- The complexity and diversity of the data
- The available implementation and learning time
- The developer skills

References

Banharnsakun, A. (2019).
Towards improving the convolutional neural networks for deep learning using the distributed artificial bee colony method.
International Journal of Machine Learning and Cybernetics, 10:1301–1311.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2015).
Pcanet: A simple deep learning baseline for image classification?
IEEE Transactions on Image Processing, 24:5017–5032.

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

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

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

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

Guissous, A. E. (2019).
Skin lesion classification using deep neural network.
ArXiv, abs/1911.07817.

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

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

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

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

Lin, T.-Y., RoyChowdhury, A., and Maji, S. (2015).
Bilinear cnn models for fine-grained visual recognition.
2015 IEEE International Conference on Computer Vision (ICCV), pages 1449–1457.

Liu, W., Zhang, M., Zhang, Y., Liao, Y., Huang, Q., Chang, S., Wang, H., and He, J. (2018).
Real-time multilead convolutional neural network for myocardial infarction detection.
IEEE Journal of Biomedical and Health Informatics, 22:1434–1444.

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

Ponti, M., Ribeiro, L. S. F., Nazaré, T. S., Bui, T., and Collomosse, J. (2017).
Everything you wanted to know about deep learning for computer vision but were afraid to ask.
2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), pages 17–41.

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

Ren, X. and Malik, J. (2003).
Learning a classification model for segmentation.
Proceedings Ninth IEEE International Conference on Computer Vision, pages 10–17 vol.1.

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

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

Xu, L., Yan, S., Chen, X., and Wang, P. (2019).
Motion recognition algorithm based on deep edge-aware pyramid pooling network in human–computer interaction.
IEEE Access, 7:163806–163813.

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

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

The End

Any Questions?

Appendix

	Train Error %	Test Error %
3-layer Conv. Net [2]	–	16.6
3-layer Conv. Net + 1 Locally Conn. layer with dropout [2]	–	15.6
Avg Pooling	1.92	19.24
Max Pooling	0.0	19.40
Stochastic Pooling	3.40	**15.13**

CIFAR-10 Classification performance for various pooling methods compared to the state-of-the-art performance with and without dropout [Zeiler and Fergus, 2013]

Method	Training error (%)	Accuracy (%)
3-layer Convnet [11]	-	83.4%
10-layer DNN [5]	-	88.79%
Stochastic pooling [20]	-	84.87%
Max pooling	3.01%	88.64%
Average pooling	4.52%	86.25%
Mixed pooling	6.25%	**89.20%**

Comparative classification performances with various pooling methods on the CIFAR-10 dataset [Yu et al., 2014]

Method	**CIFAR-10**	**CIFAR-100**
Stochastic pooling	15.13%	41.51%
Maxout	11.68%	38.57%
Network-in-network	10.41%	35.68%
Deeply supervised	9.78%	34.57%
Spectral pooling	**8.6%**	**31.6%**

Test errors on CIFAR-10/100 without data augmentation of the optimal spectral pooling architecture [Rippel et al., 2015]

Appendix

rank	team	top-5 test
1	GoogLeNet [32]	**6.66**
2	VGG [33]	7.32
3	<u>ours</u>	<u>8.06</u>
4	Howard	8.11
5	DeeperVision	9.50
6	NUS-BST	9.79
7	TTIC_ECP	10.22

The competition results of ILSVRC 2014 classification [He et al., 2015]

Appendix

	Method	VGG	NIN	ResNet
Deterministic methods	Strided conv.	8.43±0.20	10.97±0.10	6.23$^{(*)}$
	Max	7.43±0.20$^{(*)}$	9.42±0.07	6.52
	Average	7.12±0.18	8.75±0.15	6.33
	NIN	–	9.01±0.11$^{(*)}$	–
	Mixed (50/50)	7.27±0.20	8.68±0.23	6.05
	Gated	7.25±0.14	8.67±0.22	7.12
	L_2	7.15±0.18	8.65±0.12	7.29
	Lite-DPP$_{\text{Asym}}$	<u>7.10</u>±0.15	8.62±0.10	6.17
	Full-DPP$_{\text{Asym}}$	7.17±0.18	8.73±0.05	6.23
	Lite-DPP$_{\text{Sym}}$	7.19±0.10	8.58±0.11	6.05
	Full-DPP$_{\text{Sym}}$	**7.02**±0.18	8.70±0.14	5.97
Stoch.	Stochastic	7.67±0.10	8.92±0.09	5.83
	S3pool	7.21±0.14	<u>7.23</u>±0.08	<u>5.55</u>
	Lite-S3DPP$_{\text{Sym}}$	–	**7.13**±0.09	**5.42**

Comparison of different architectures and pooling layers on the CIFAR10 dataset [Saeedan et al., 2018]