Convolutional neural networks

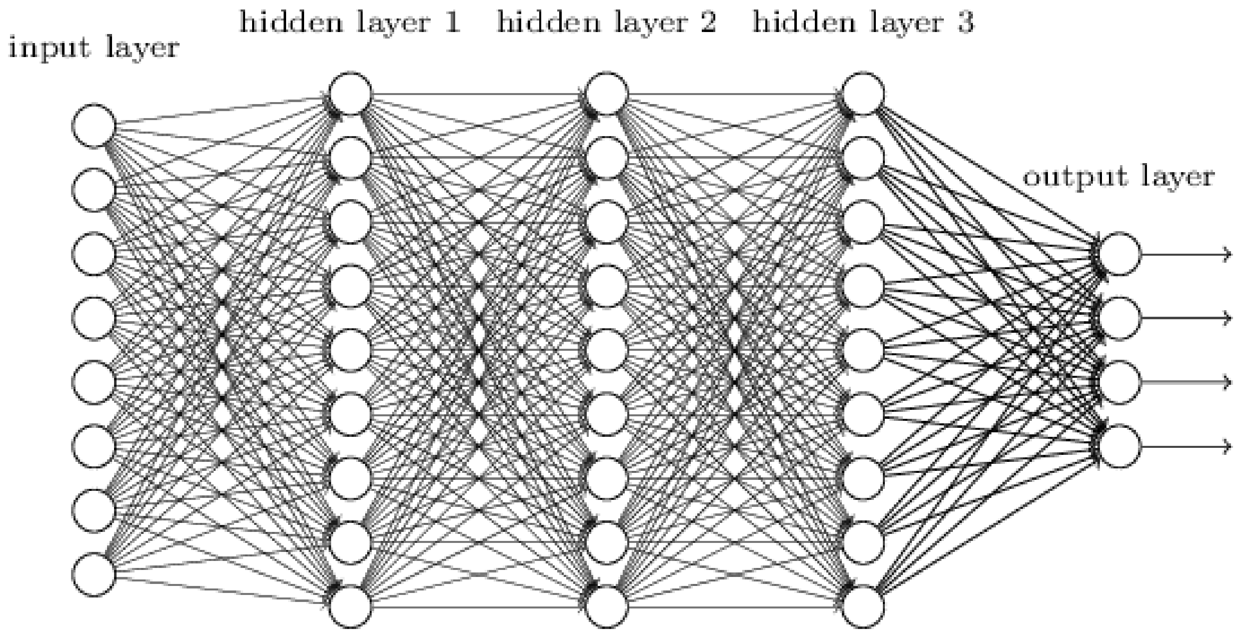
Seongok Ryu
KAIST Chemistry

Contents

- Weight sharing
- Convolution operation
- Convolutional neural networks
- Residual networks
- Beyond

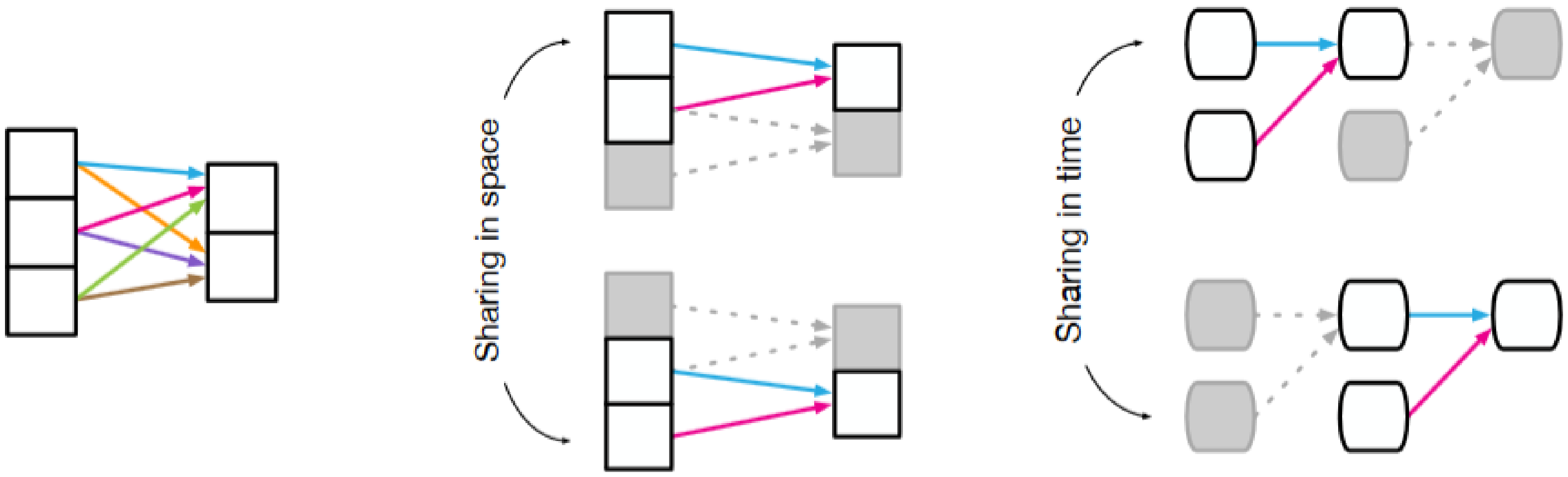
Weight sharing

- Multi-layer perceptrons (or sometimes called as fully-connected neural networks) connect all the neurons between the layers.
- This model structure use too much number of parameters, vulnerable to over-fitting problems.
- For example, as an input with images having a dimensionality $H \times W \times C$ and hidden dimension d, a MLP uses total $H \times W \times C \times d$ number of weight parameters. If $H = W = 32, C = 3$ and $\mathrm{d} = 512$, then total 1.57M number of parameters is required.



Weight sharing

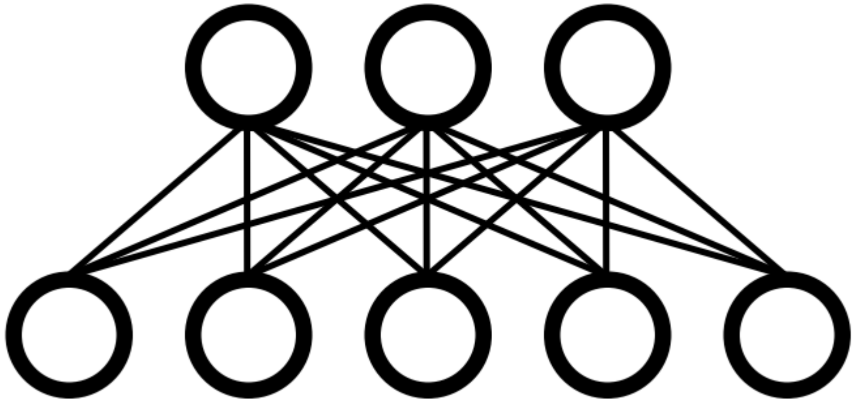
- For many domains, a same operation can be applied on local regions of input data.
- Images: same convolution operations on different pixels in images.
- Languages: same recurrent/convolution operations on different words in sentences.
- Molecules: same graph convolution operations on different atoms in molecules.
- This makes designing a model architecture much simpler and helps using less number of parameters.



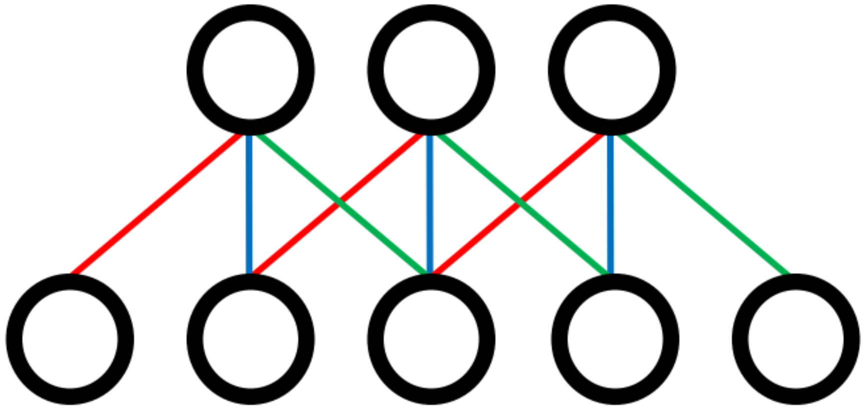
(a) Fully connected (b) Convolutional (c) Recurrent

Convolution operation

- Convolution is the most commonly used operation in modern deep neural networks.
- Using non-zero weight values on local regions.
- Sharing a same weight parameter (also referred to as a receptive field) for different regions.



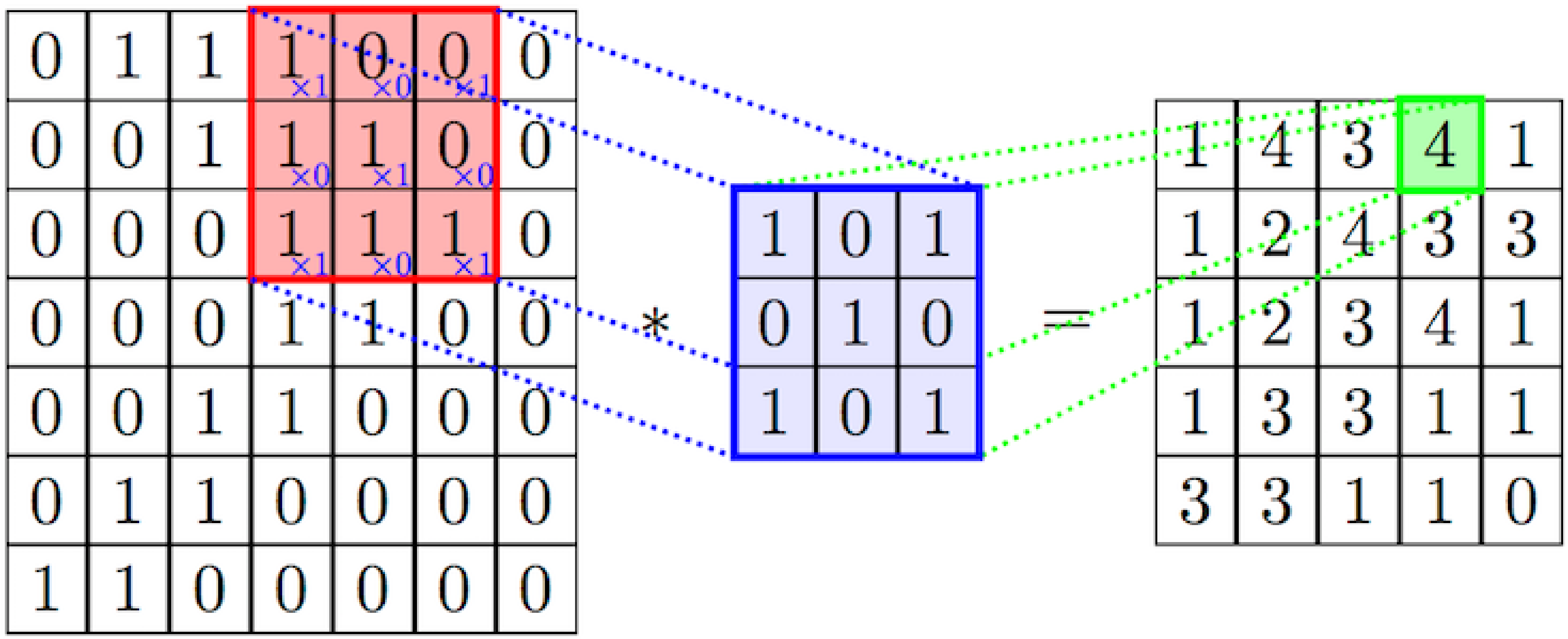
Fully-connected NN



Convolutional NN

Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

Convolution operation



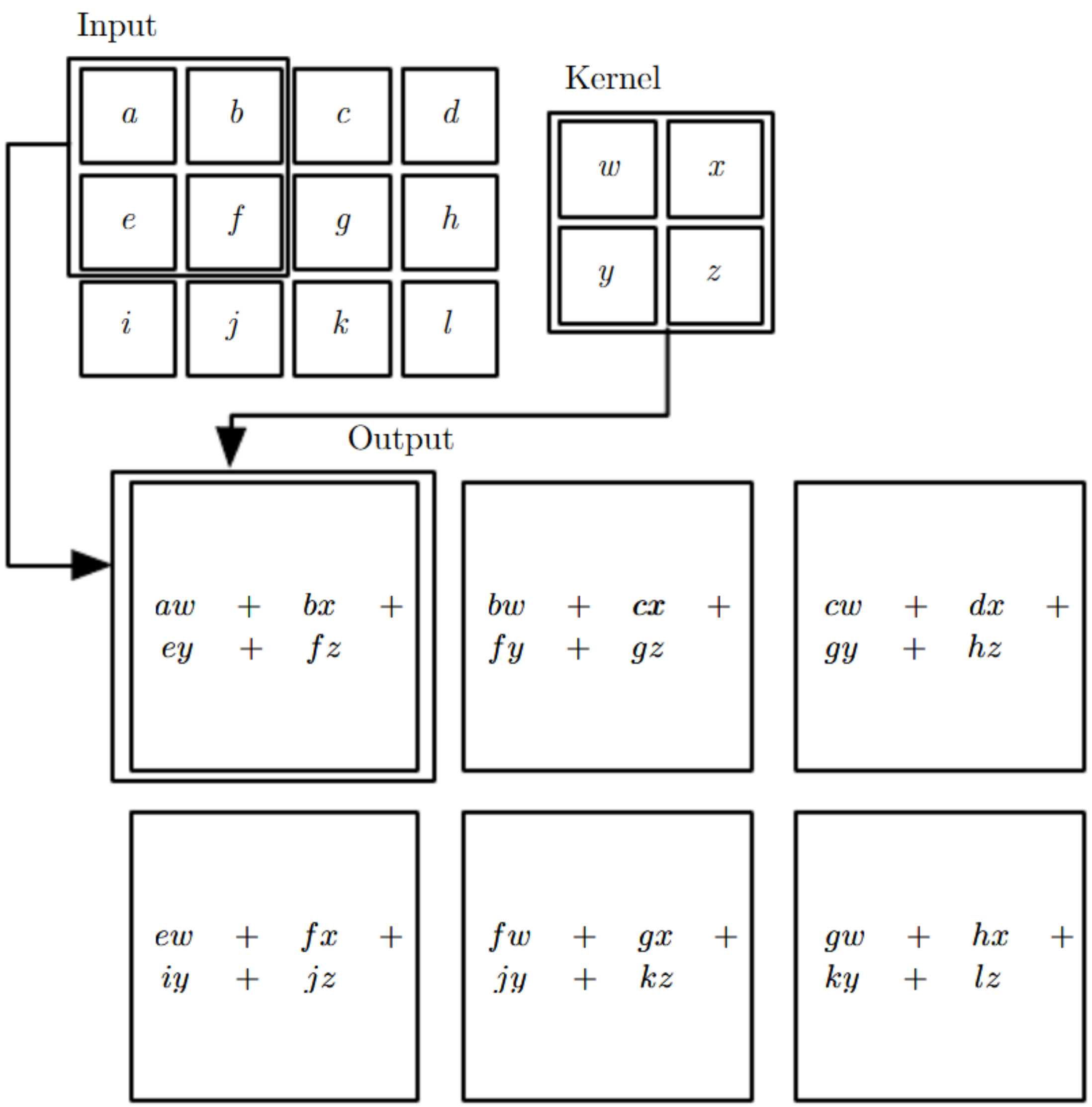
$k = 1$ for 3×3 convolution

$$\boldsymbol{X_i^{(l+1)}} = \boldsymbol{\sigma}\left(\sum_{j \in [i-k, i+k]} \boldsymbol{W_j^{(l)}} \boldsymbol{X_j^{(l)}} + \boldsymbol{b^{(l)}}\right)$$

learnable parameters are shared

Convolution operation

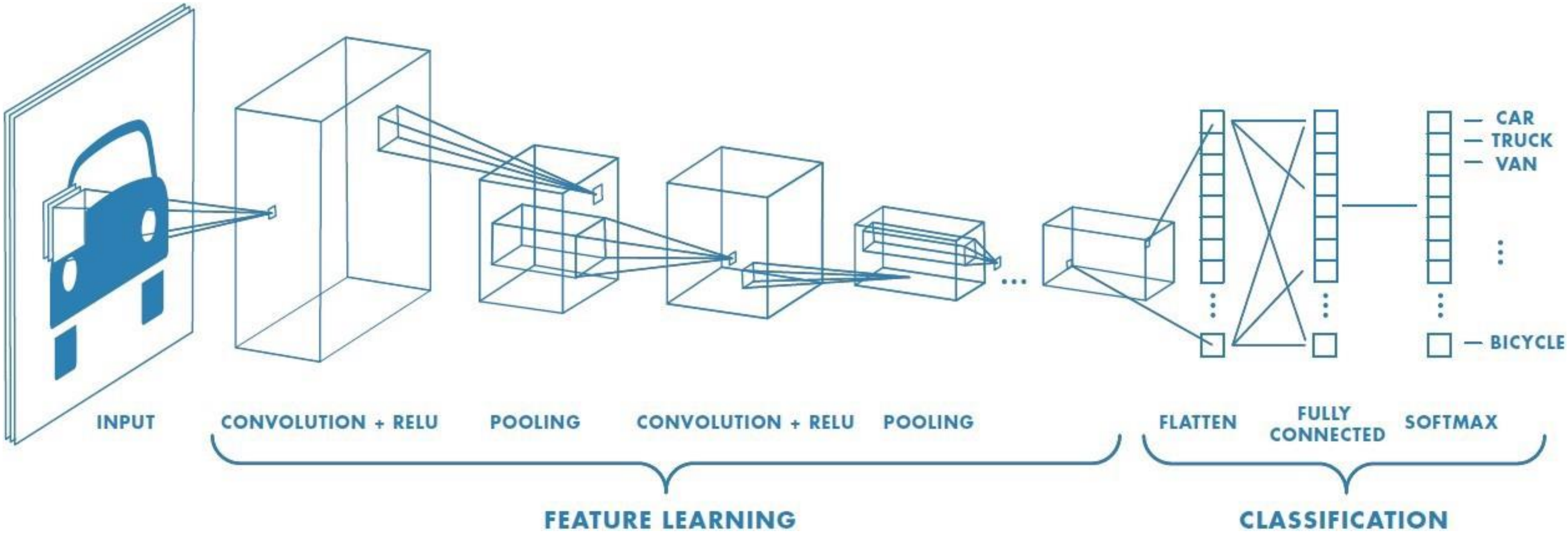
- An example of 2-D convolution



Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

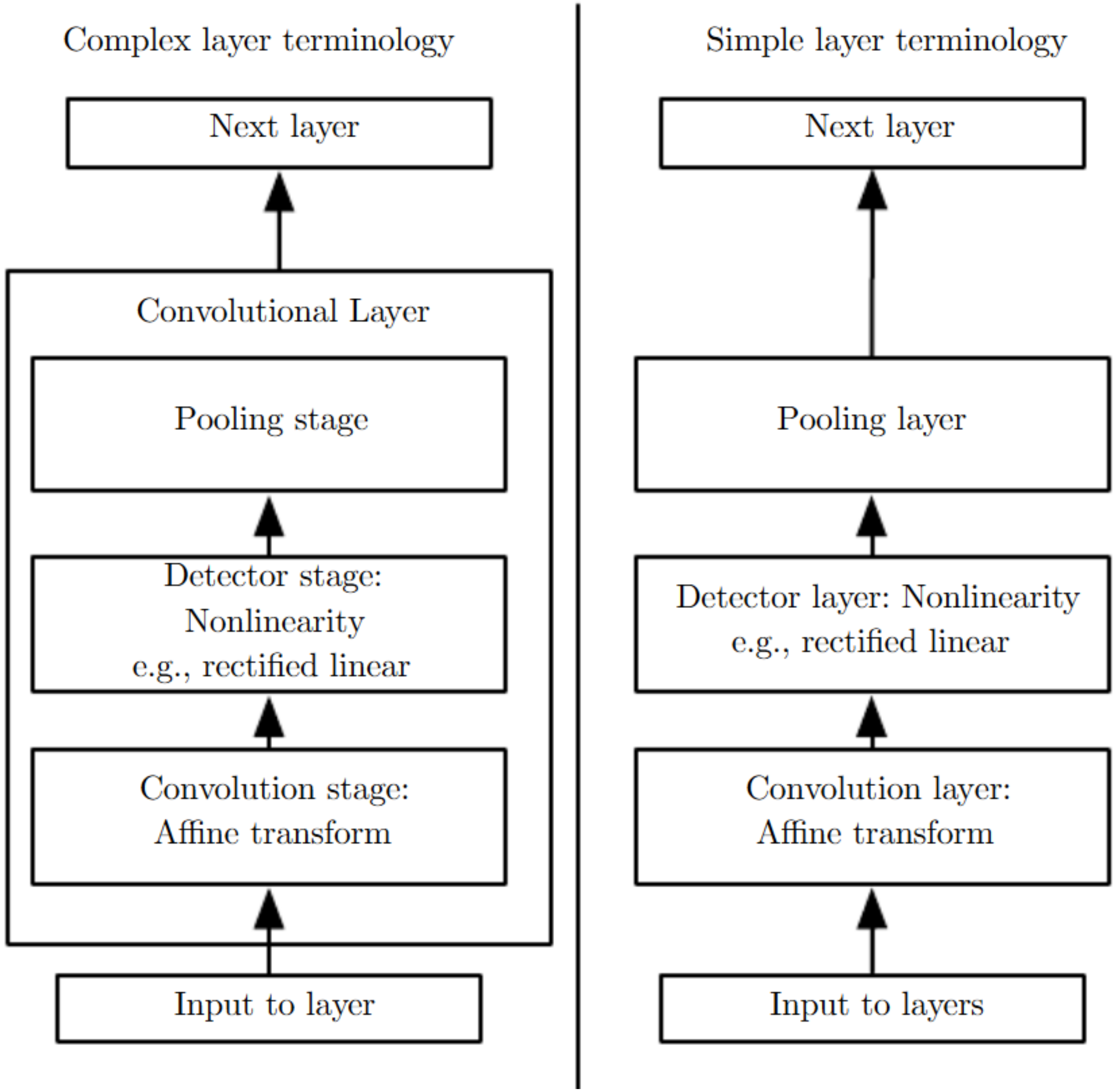
Convolutional neural networks

- “CNNs are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.” – Deep learning book

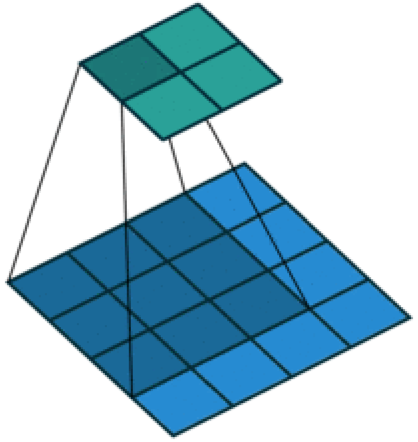


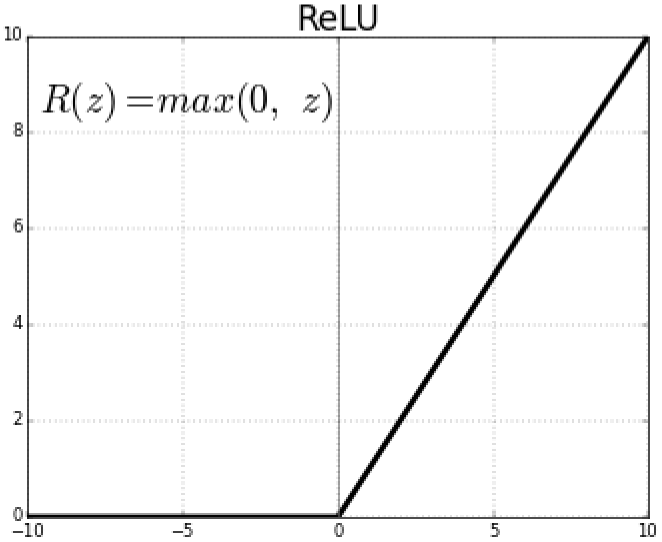
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Convolutional neural networks



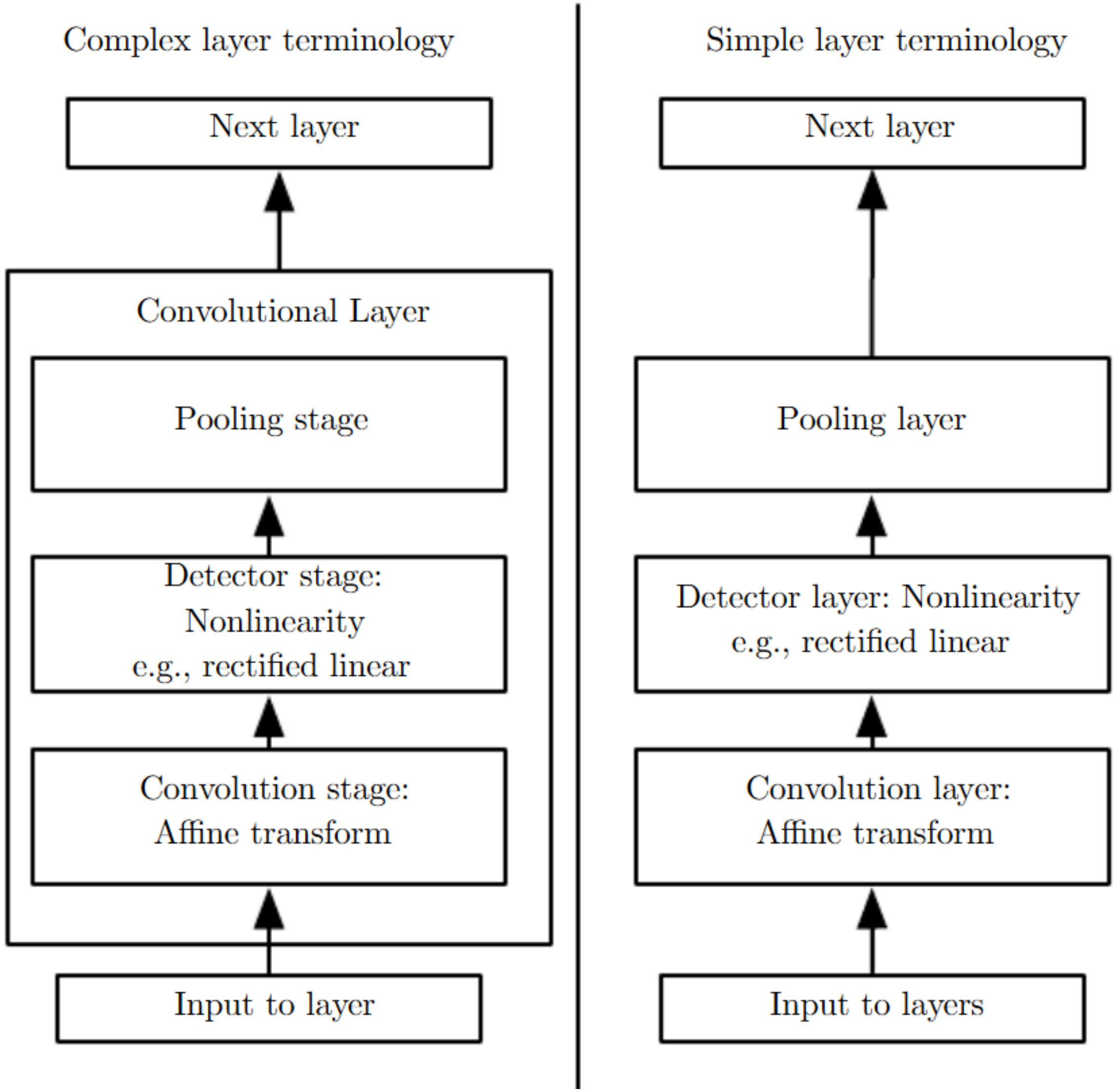
- Convolution operations compute the linear combination of pixel values.
- Then, non-linear activations computes output responses and detects large ones.



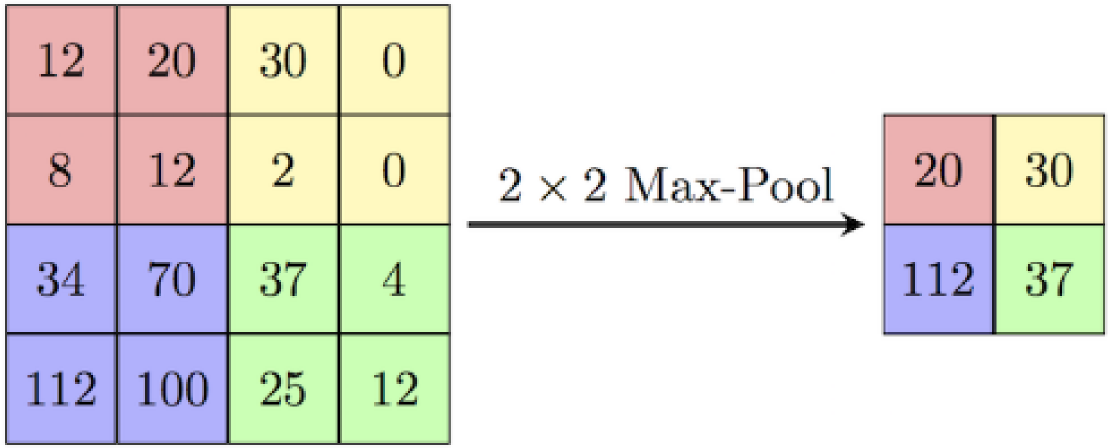


Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Convolutional neural networks



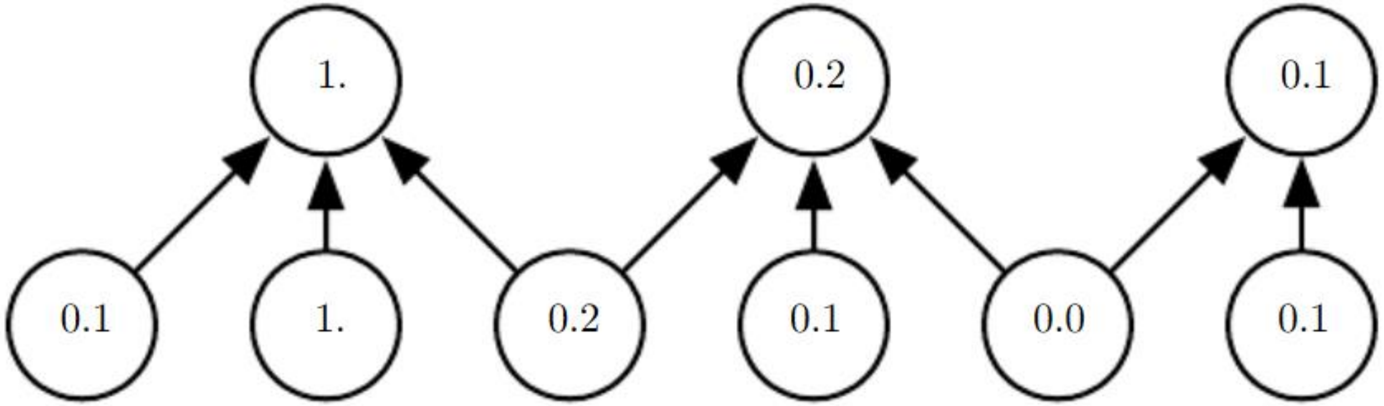
- Then, the pooling operation is applied to summarize the statistics of features.
- Max-pooling, L2-norm pooling, average pooling, …



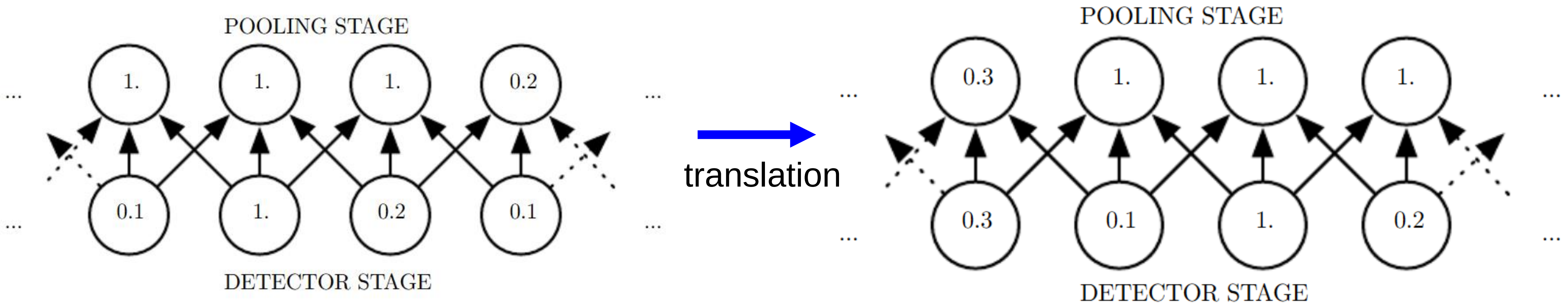
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Convolutional neural networks

- Pooling operation with strides larger than 2 can provide down-sampling of images (feature-maps).



- Pooling operation can approximate invariance to local translation

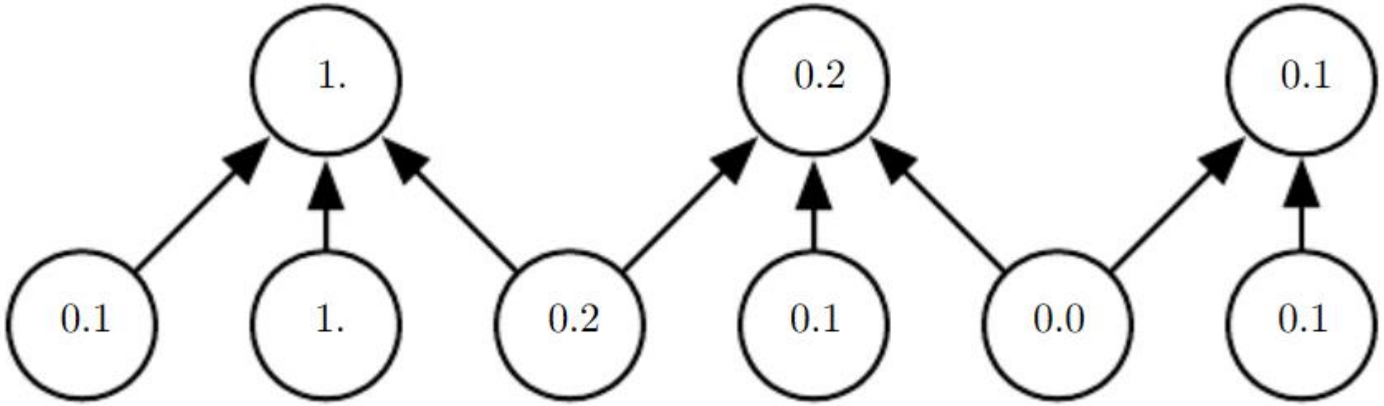


- Every value in the botton row has changed, but only half of the values in the top row have changed.

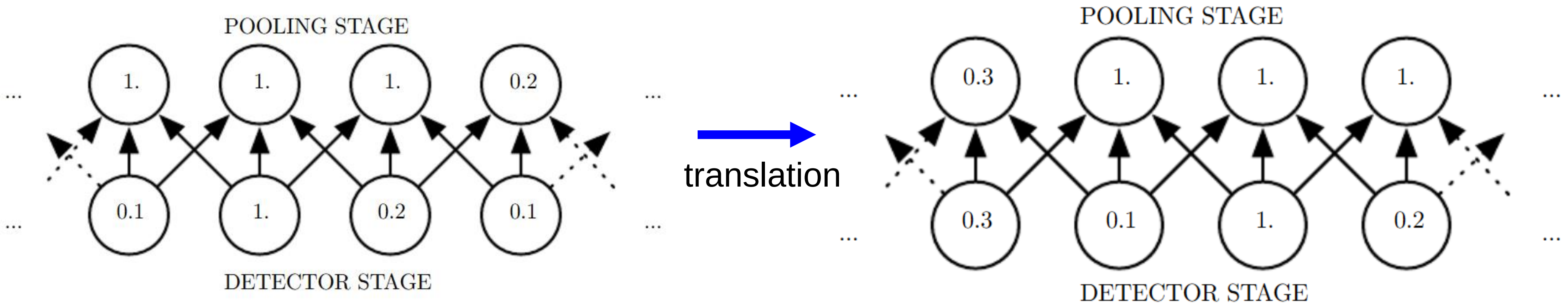
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Convolutional neural networks

- Pooling operation with strides larger than 2 can provide down-sampling of images (feature-maps).



- Pooling operation can approximate invariance to local translation

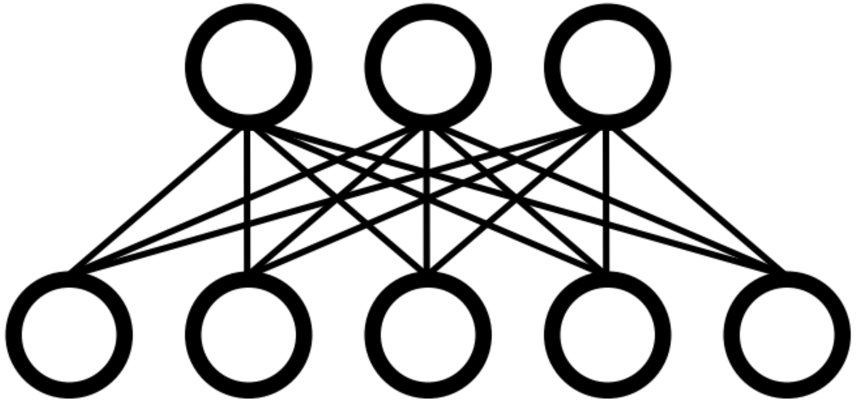


- Every value in the botton row has changed, but only half of the values in the top row have changed.

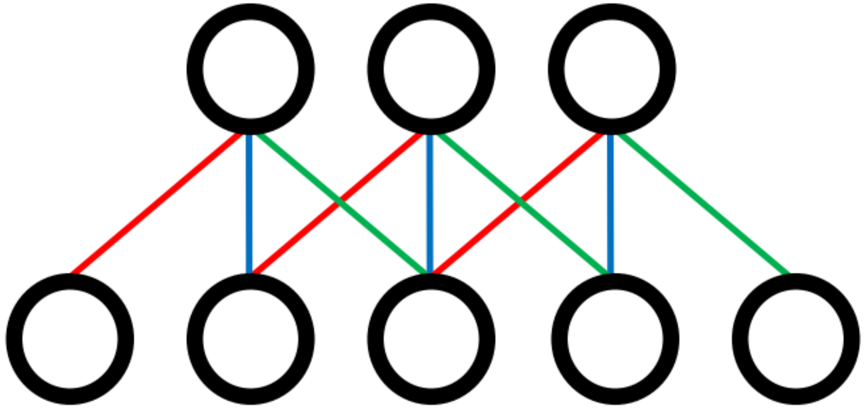
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Convolutional neural networks

- We can understand convolution and pooling as *an infinitely strong prior*.
- *An infinitely strong prior* places zero probability on some parameters and says that theses parameter values are completely forbidden, regardless of how much support the data give to those values.
- Using those operations endue strong *inductive biases* on a hypothesis.



Fully-connected NN

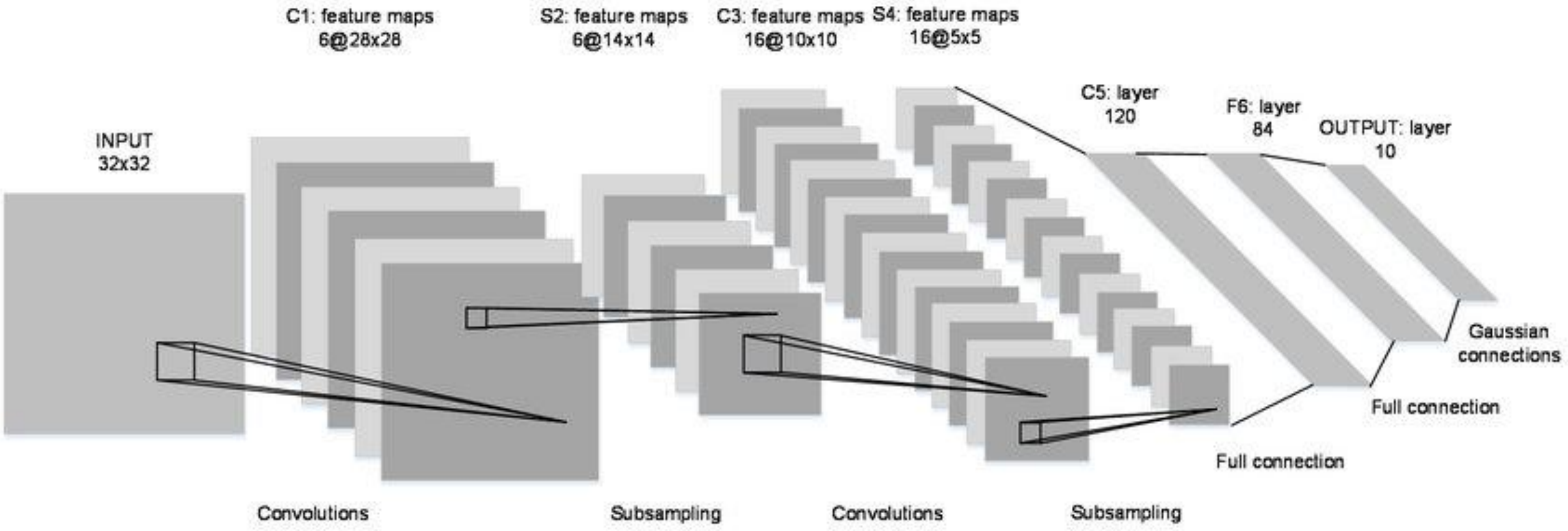


Convolutional NN

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

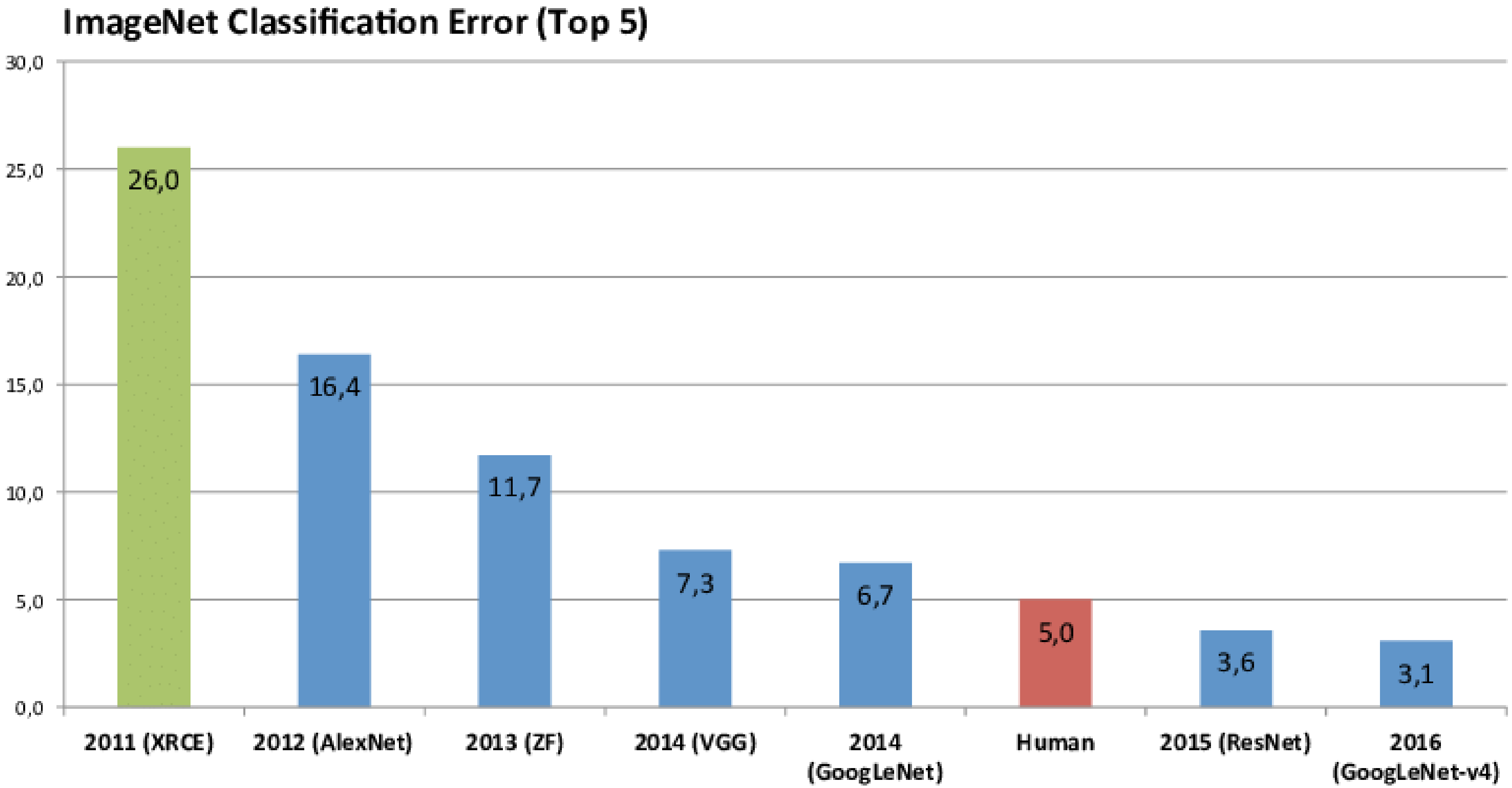
Convolutional neural networks

- In early 1990s, LeNet was invented by Yann Lecun, who is one of the pioneers of deep learning researches.
- Using simple convolution and pooling operations for featurizations and fully-connected layers for a classifier.
- CNN demo from 1993, https://www.youtube.com/watch?v=FwFduRA_L6Q



Convolutional neural networks

- Representative CNN models wons ImageNet Challenge (ILSVRC)

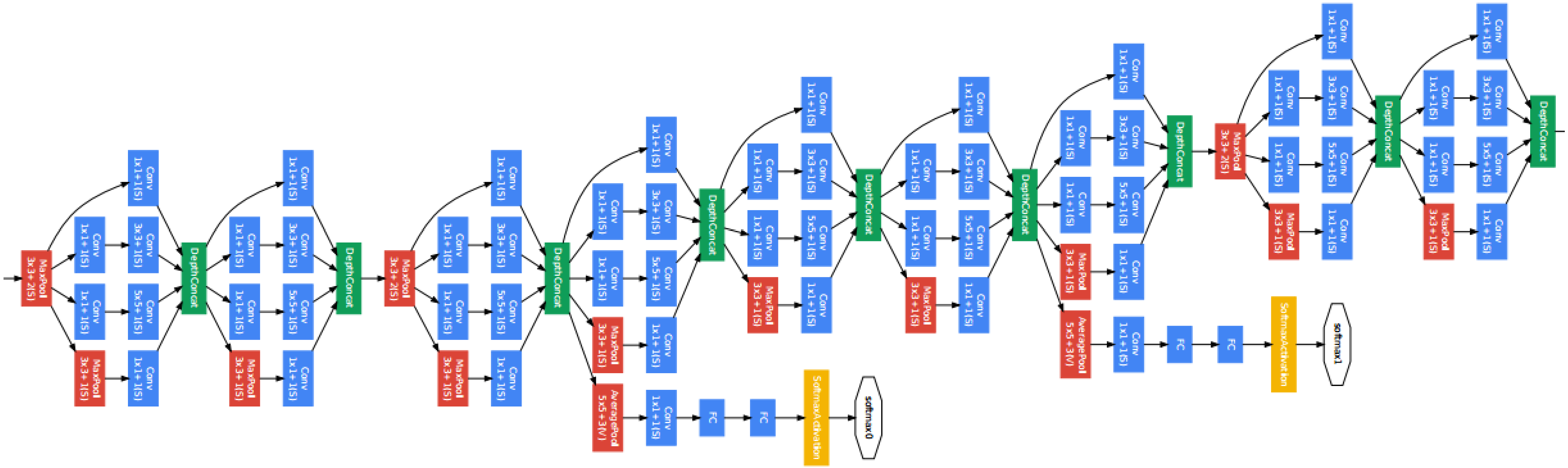


Convolutional neural networks

- VGG-Net
- Total 19 layers with a simple combination of convolution and pooling operations

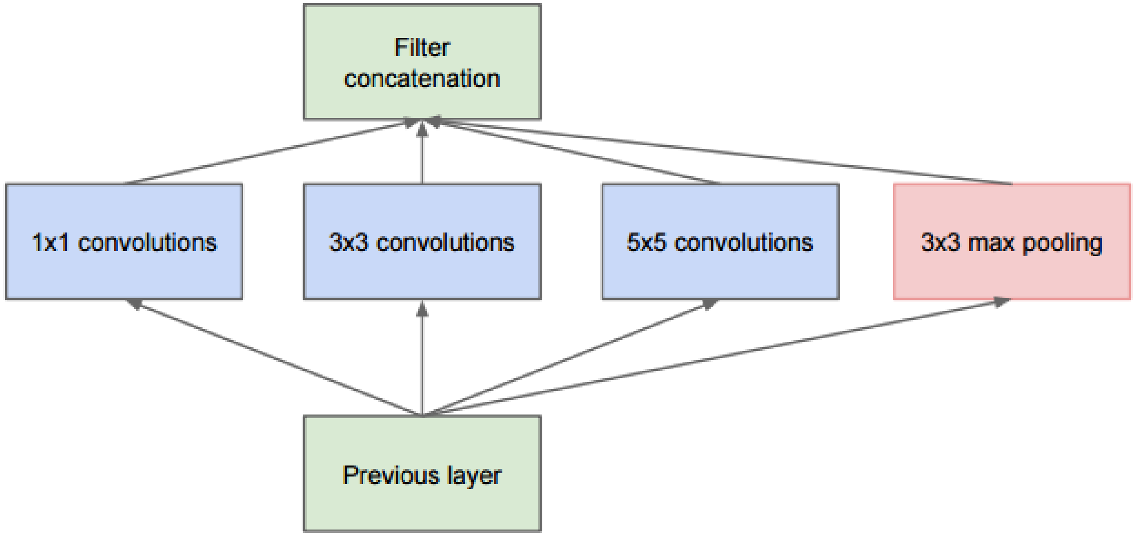
Convolutional neural networks

- GoogLeNet (Inception network)
- Inception module is combined with different receptive fields and max-pooling operations.

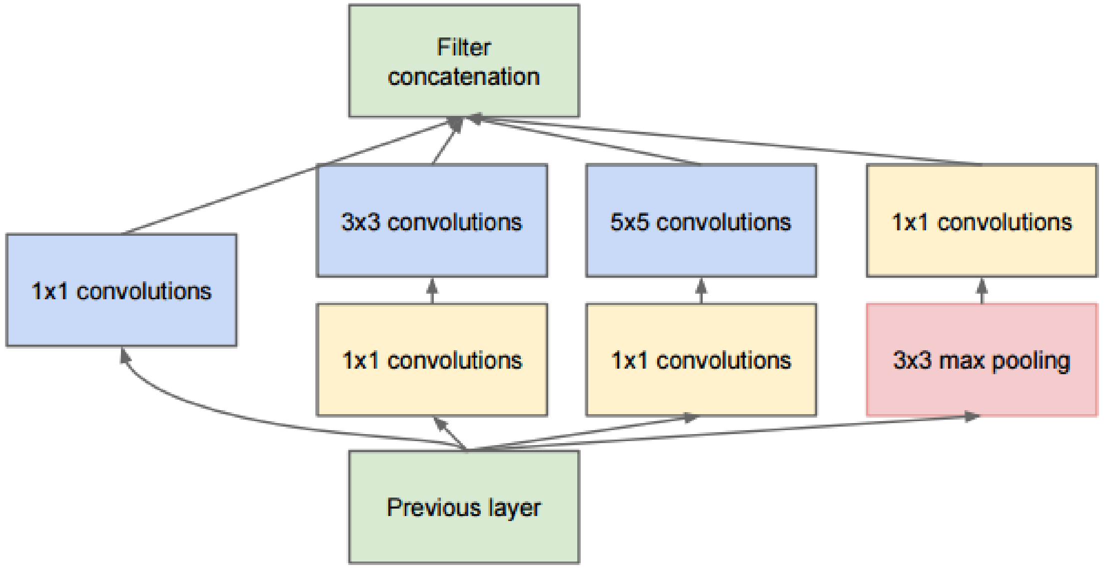


Convolutional neural networks

- GoogLeNet (Inception network)
- Inception module is combined with different receptive fields and max-pooling operations.



(a) Inception module, naïve version



(b) Inception module with dimension reductions

- Make CNN models wider as well as deeper.
- Efficient use of parameters: showing better performance with using less number of parameters than VGGNet.

Residual Networks

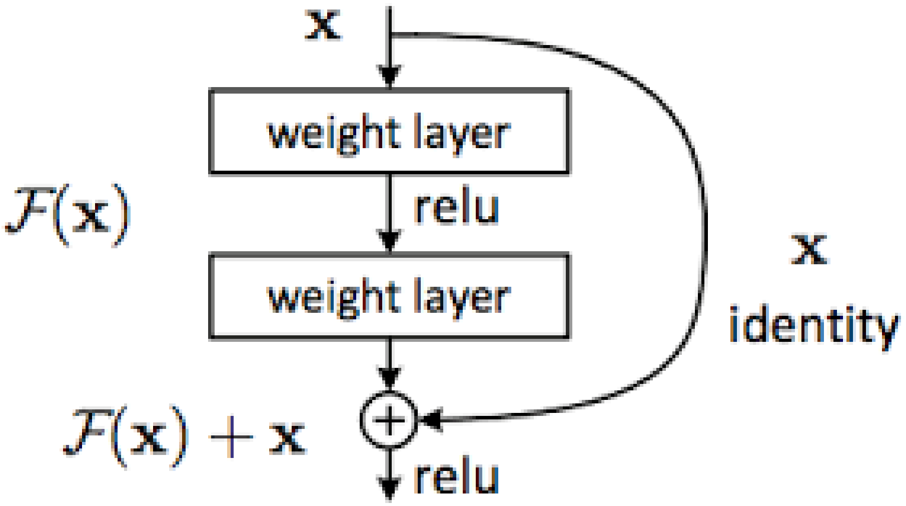
- Using residual blocks or skip-connections allows model to be much deeper.
- The residual block updates the feature by the amount of $F(x)$:

$$x^{l+1} - x^{l} = F(x^{l})$$

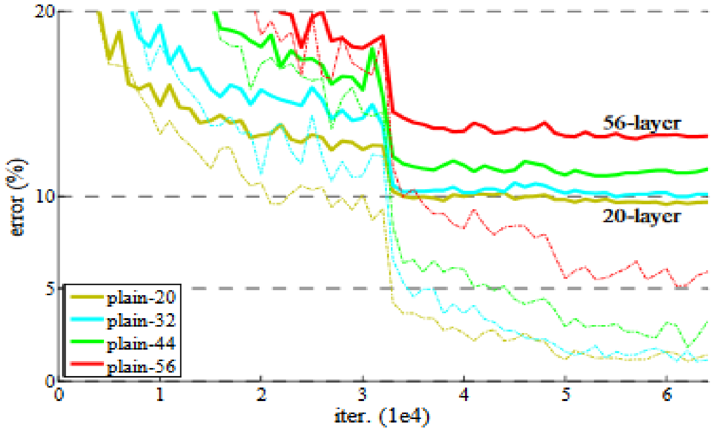
- Residual learning can mitigate vanishing gradient problems.
- Let $y = x + F(x)$, then the gradient of loss is given by:

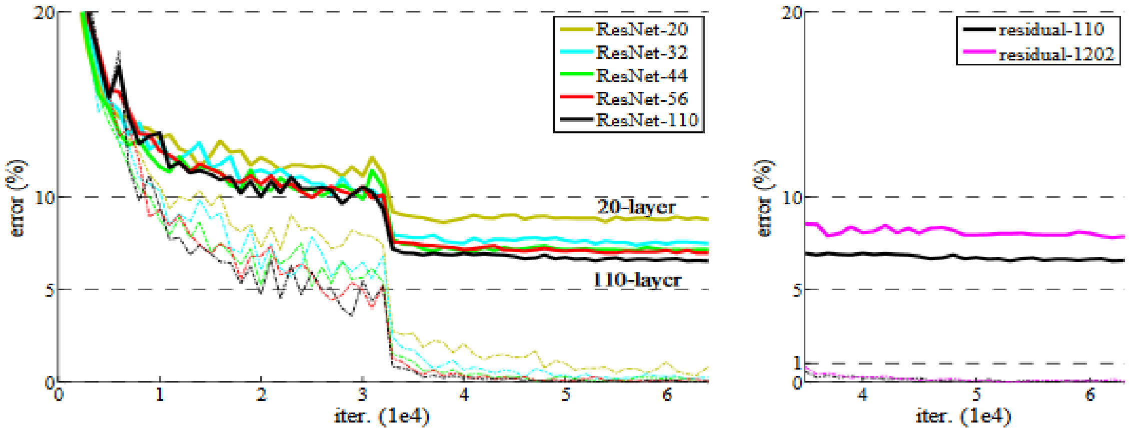
$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x} = \frac{\partial L}{\partial y}\left(1 + \frac{\partial F(x)}{\partial x}\right)$$

- In contrast to the gradient of plain networks is given by $\frac{\partial L}{\partial y} \cdot \frac{\partial F(x)}{\partial x}$.



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.





Residual Networks

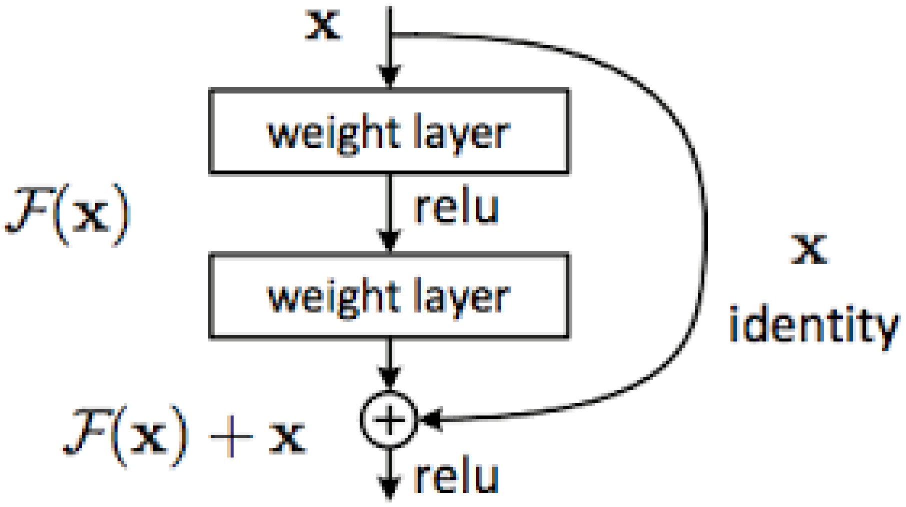
- Using residual blocks or skip-connections allows model to be much deeper.
- The residual block updates the feature by the amount of $F(x)$:

$$x^{l+1} - x^{l} = F(x^{l})$$

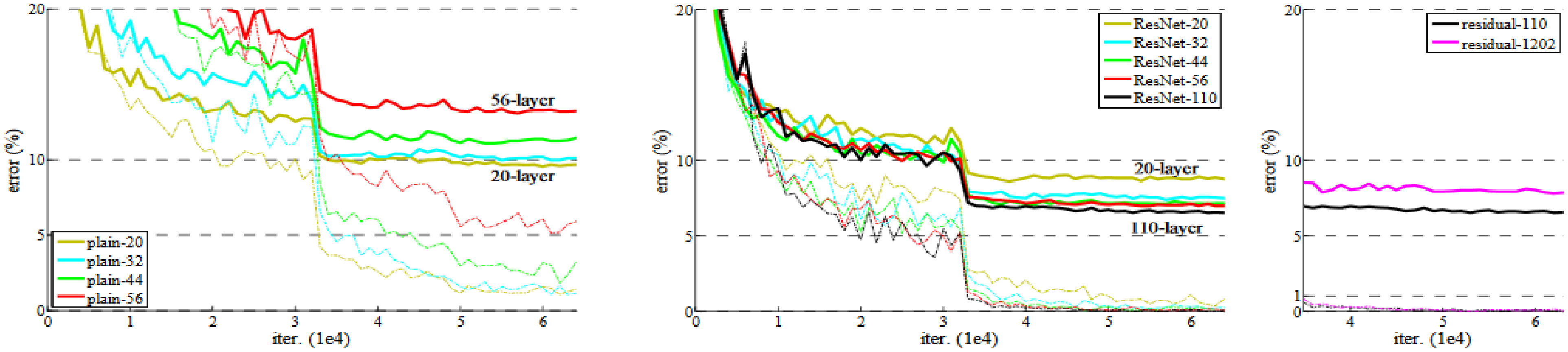
- Residual learning can mitigate vanishing gradient problems.
- Let $y = x + F(x)$, then the gradient of loss is given by:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x} = \frac{\partial L}{\partial y}\left(1 + \frac{\partial F(x)}{\partial x}\right)$$

- In contrast to the gradient of plain networks is given by $\frac{\partial L}{\partial y} \cdot \frac{\partial F(x)}{\partial x}$.



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.



Beyond ResNet (~2015)

- Neural architecture search (NAS): searching model architectures by neural networks

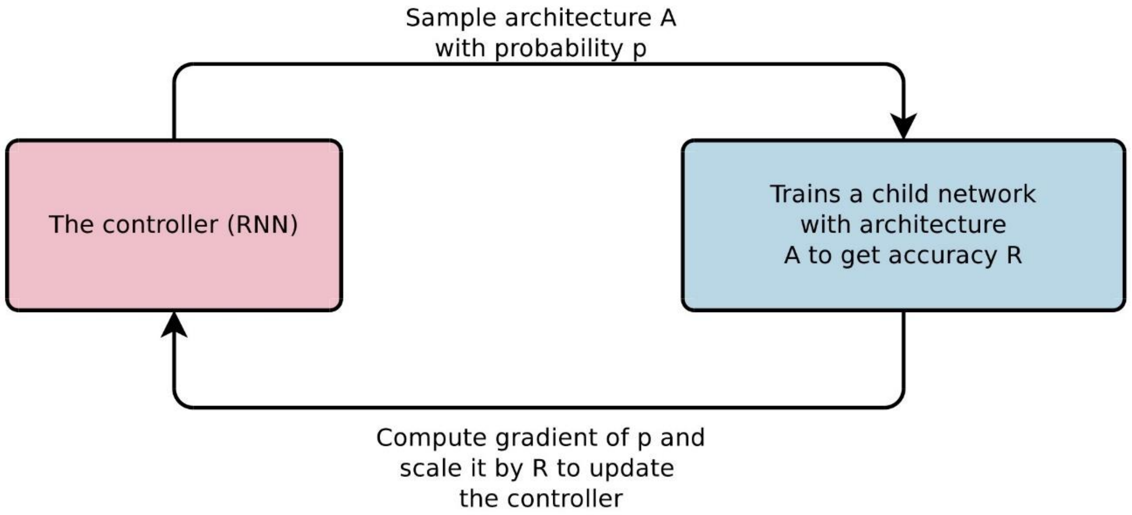
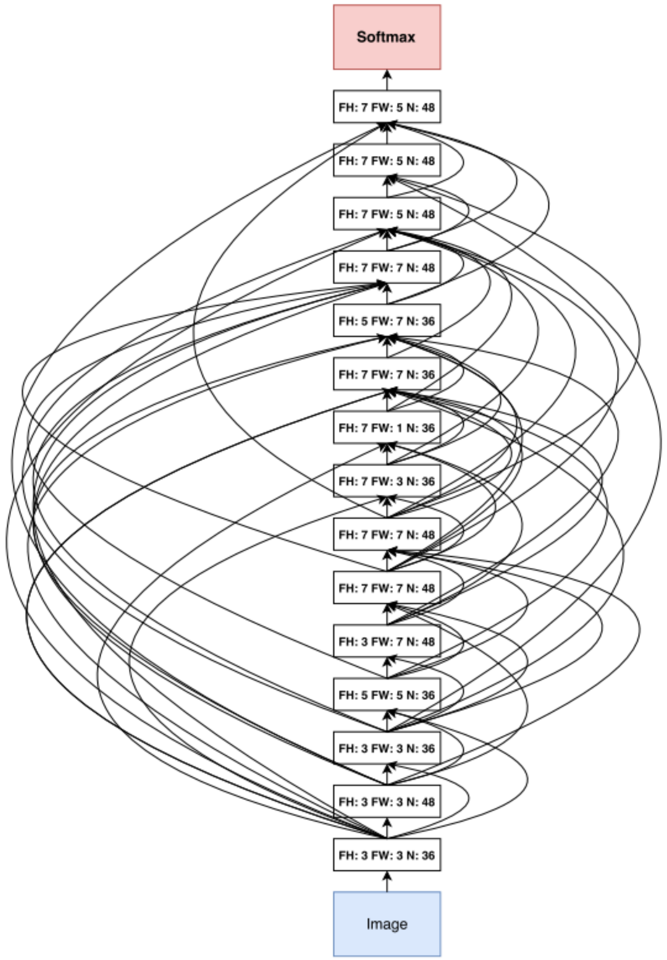
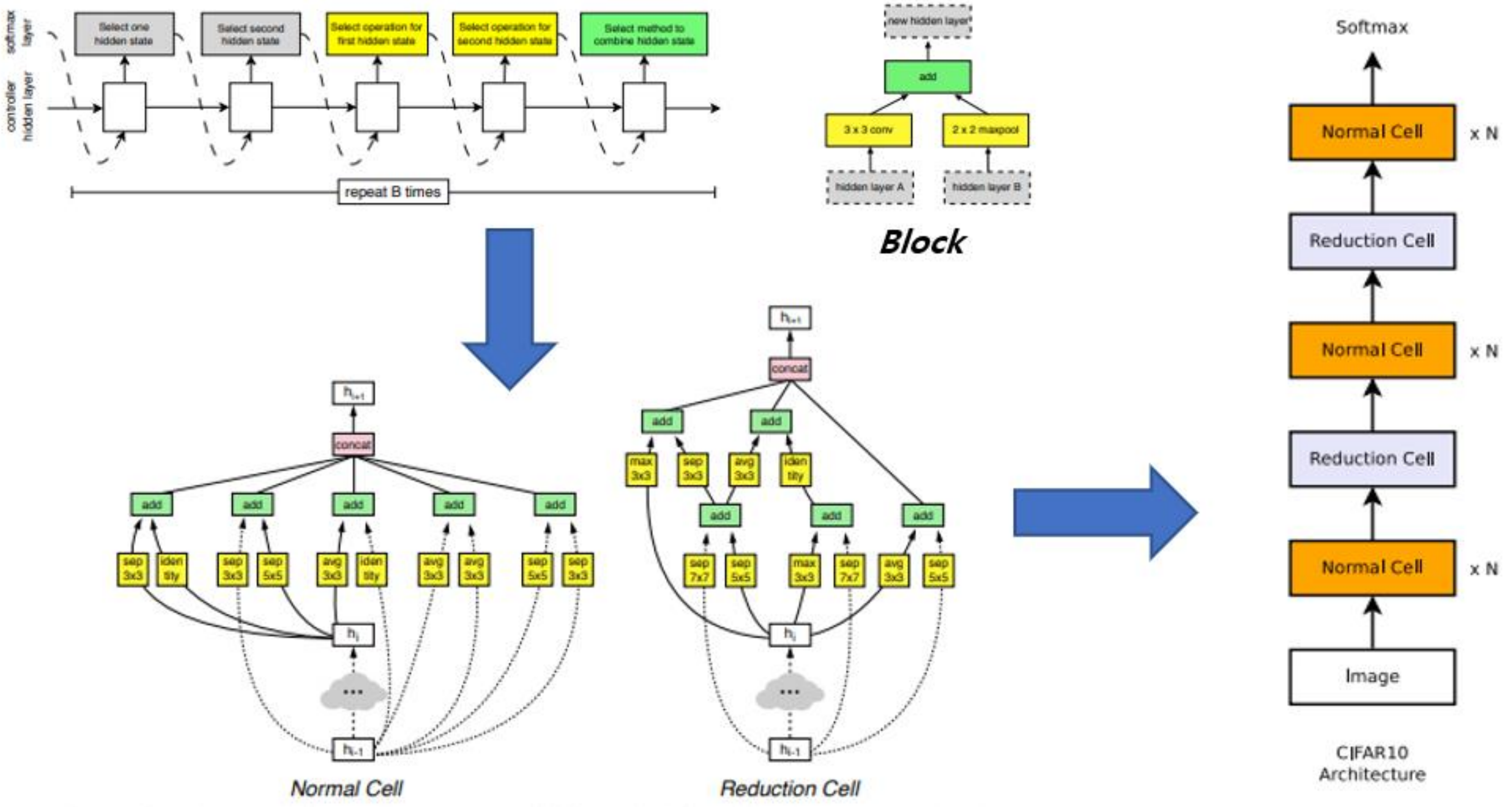


Figure 1: An overview of Neural Architecture Search.



Beyond ResNet (~2015)

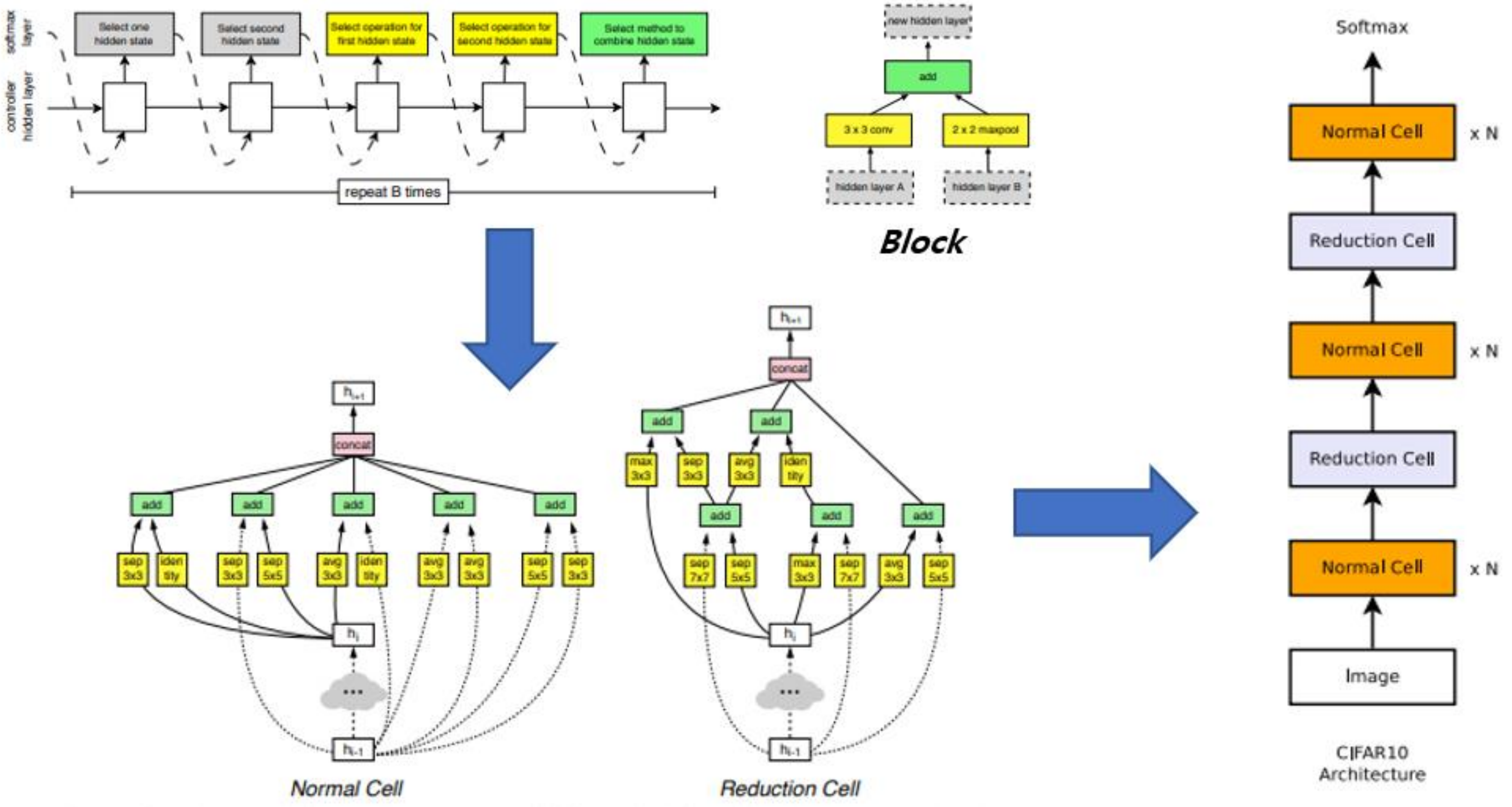
- Neural architecture search (NAS): searching model architectures by neural networks



Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." *arXiv preprint arXiv:1611.01578* (2016).
Pham, Hieu, et al. "Efficient neural architecture search via parameter sharing." *arXiv preprint arXiv:1802.03268* (2018).

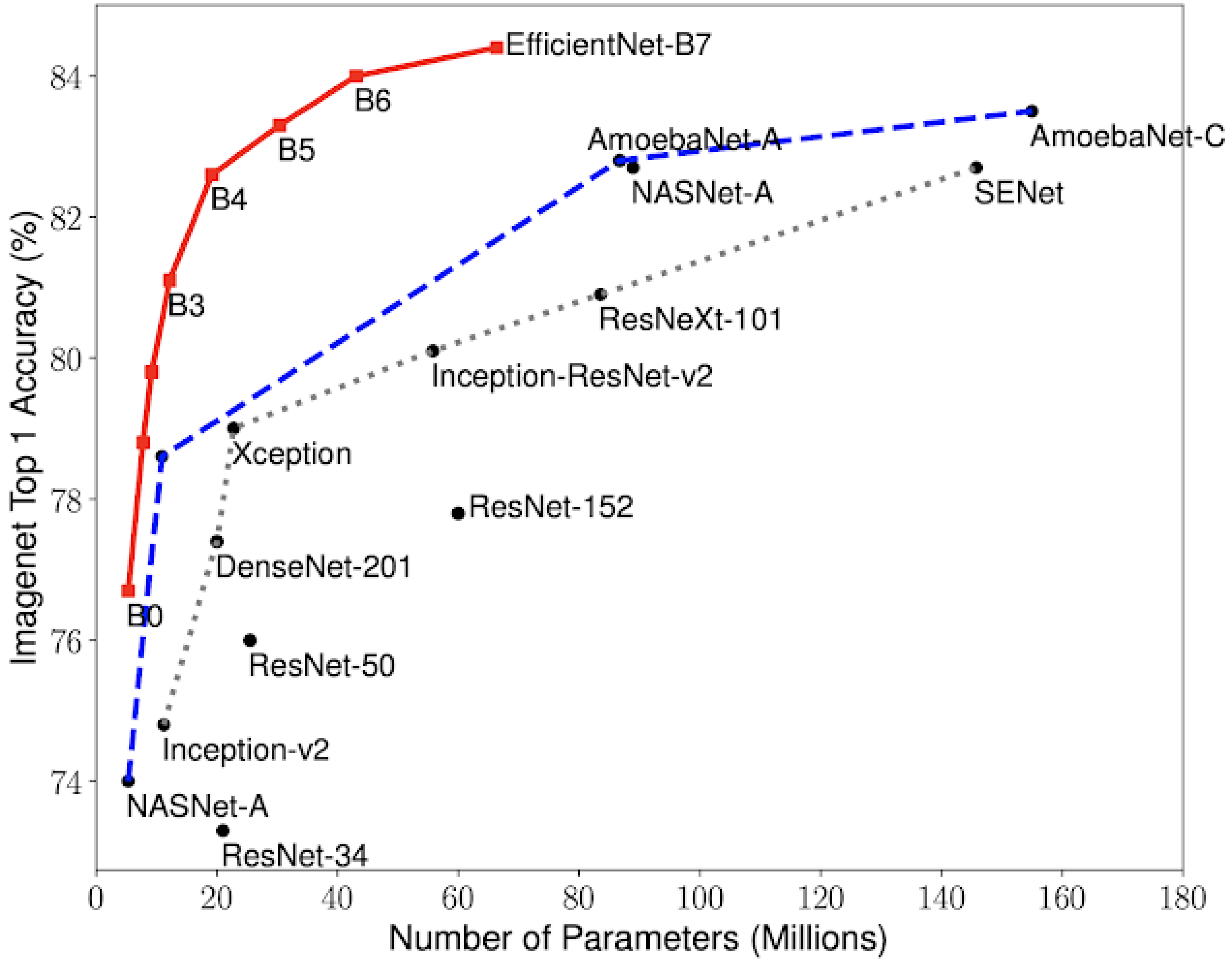
Beyond ResNet (~2015)

- Neural architecture search (NAS): searching model architectures by neural networks



Beyond ResNet (~2015)

- Improve parameter efficiency - very important for practical uses (time costs, electricity consumption, …)



https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html