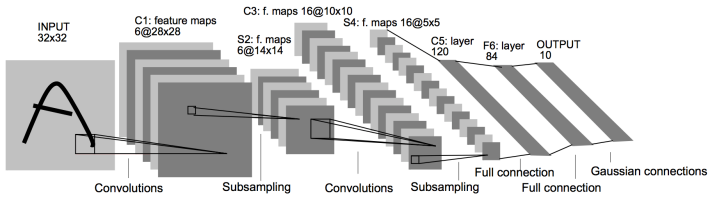
Deep Learning

Modern Architectures - I



Learning goals

- LeNet
- AlexNet
- VGG
- Network in Network

LeNet

LENET ARCHITECTURE

- Pioneering work on CNNs by Yann Lecun in 1998.
- Applied on the MNIST dataset for automated handwritten digit recognition.
- Consists of convolutional, "subsampling" and dense layers.
- Complexity and depth of the net was mainly restricted by limited computational power back in the days.

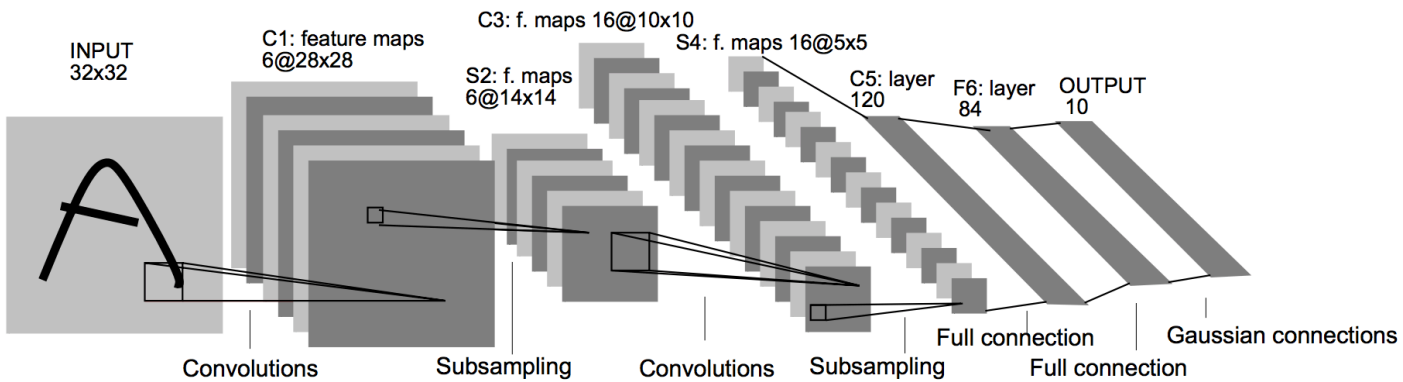


Figure: LeNet architecture: two conv layers with subsampling, followed by dense layers and a 'Gaussian connections' layer.

LENET ARCHITECTURE

- A neuron in a subsampling layer looks at a 2×2 region of a feature map, sums the four values, multiplies it by a trainable coefficient, adds a trainable bias and then applies a sigmoid activation.
- A stride of 2 ensures that the size of the feature map reduces by about a half.
- The 'Gaussian connections' layer has a neuron for each possible class.
- The output of each neuron in this layer is the (squared) Euclidean distance between the activations from the previous layer and the weights of the neuron.

AlexNet

ALEXNET

- AlexNet, which employed an 8-layer CNN, won the ImageNet Large Scale Visual Recognition (LSVR) Challenge 2012 by a phenomenally large margin.
- The network trained in parallel on two small GPUs, using two streams of convolutions which are partly interconnected.
- The architectures of AlexNet and LeNet are very similar, but there are also significant differences:
  - First, AlexNet is deeper than the comparatively small LeNet5. AlexNet consists of eight layers: five convolutional layers, two fully-connected hidden layers, and one fully-connected output layer.
  - Second, AlexNet used the ReLU instead of the sigmoid as its activation function.

ALEXNET

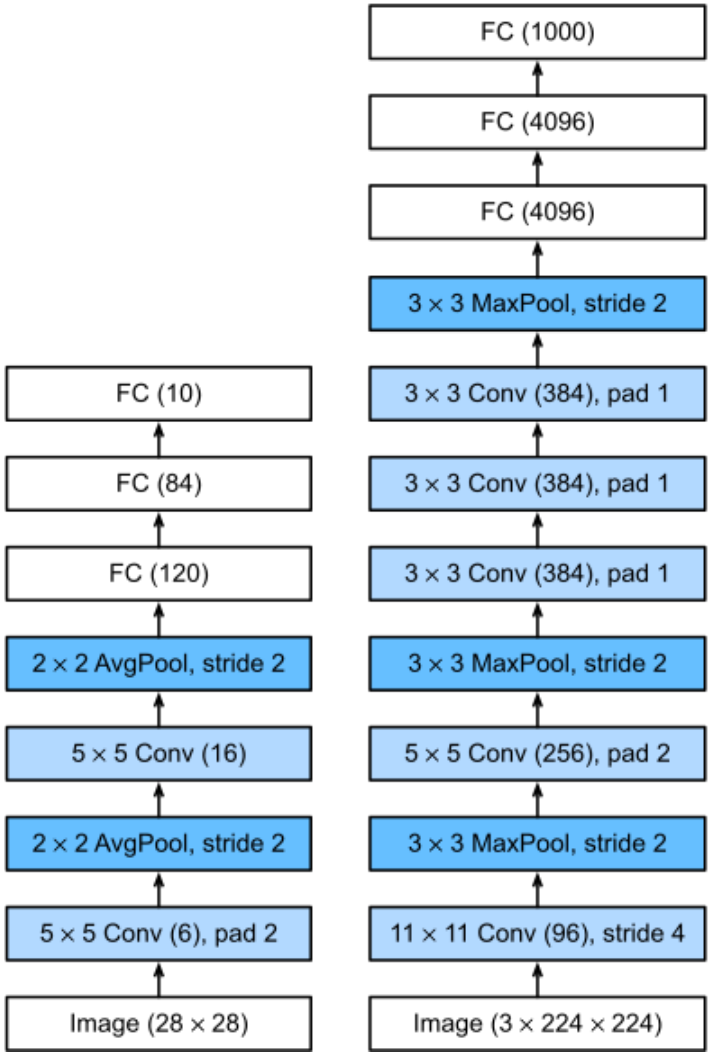
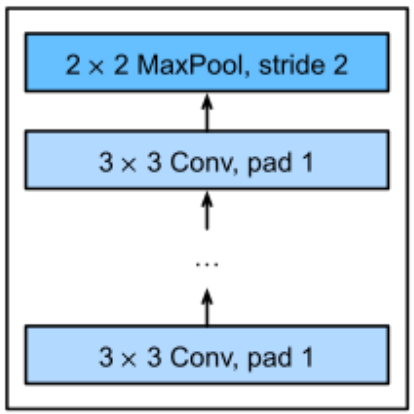


Figure: From LeNet (left) to AlexNet (right).

VGG

VGG BLOCKS

- The block composed of convolutions with 3×3 kernels with padding of 1 (keeping height and width) and 2×2 max pooling with stride of 2 (halving the resolution after each block).
- The use of blocks leads to very compact representations of the network definition.
- It allows for efficient design of complex networks.



credit : D2DL

Figure: VGG block.

VGG NETWORK

- Architecture introduced by Simonyan and Zisserman, 2014 as "Very Deep Convolutional Network".
- A deeper variant of the AlexNet.
- Basic idea is to have small filters and Deeper networks
- Mainly uses many cnn layers with a small kernel size 3×3.
- Stack of three 3×3 cnn (stride 1) layers has same effective receptive field as one 7×7 conv layer.
- Performed very well in the ImageNet Challenge in 2014.
- Exists in a small version (VGG16) with a total of 16 layers (13 cnn layers and 3 fc layers) and 5 VGG blocks while a larger version (VGG19) with 19 layers (16 cnn layers and 3 fc layers) and 6 VGG blocks.

VGG NETWORK

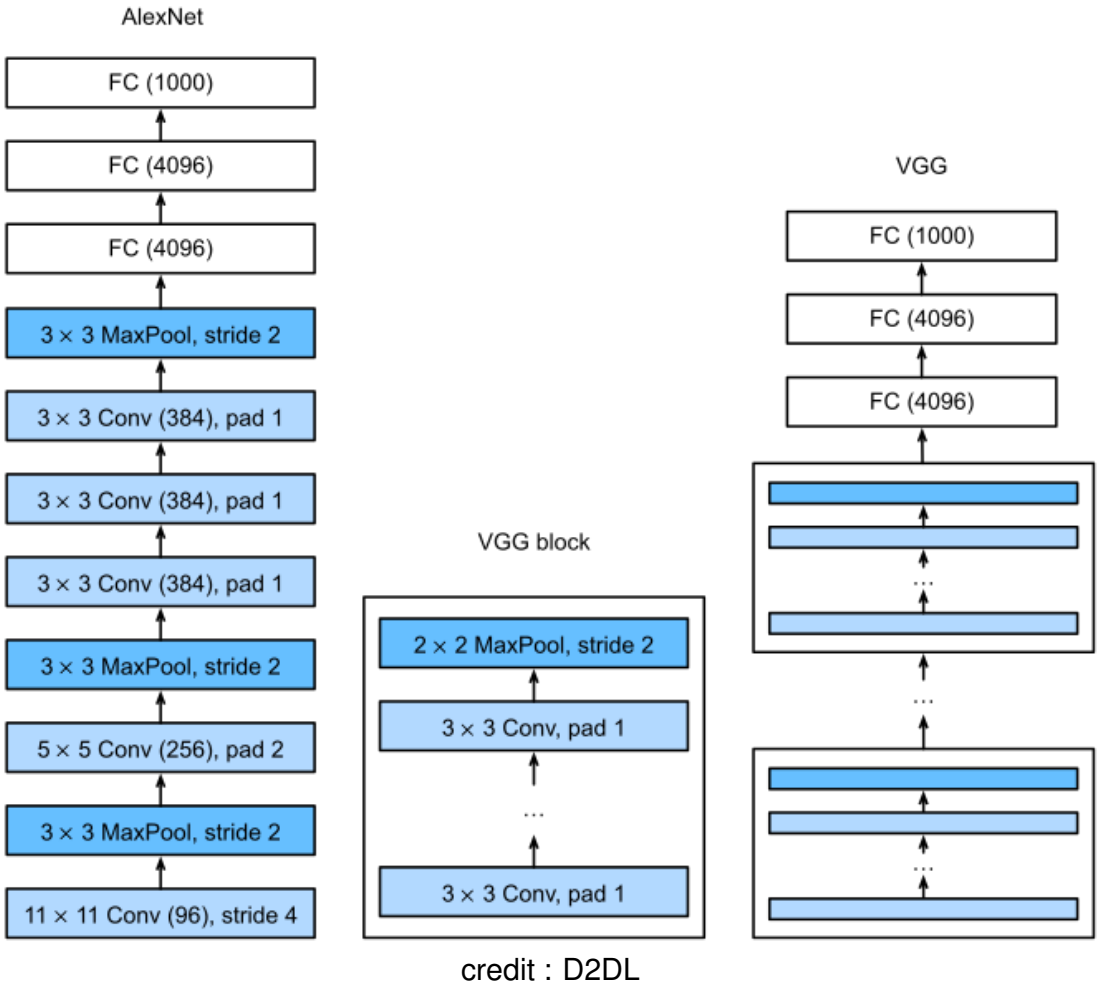


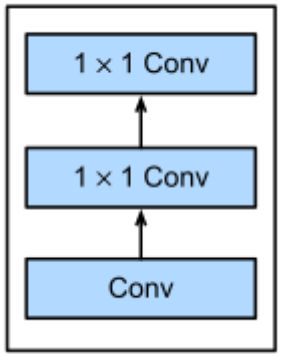
Figure: From AlexNet to VGG that is designed from building blocks.

Network in Network (NiN)

NIN BLOCKS

- The idea behind NiN is to apply a fully-connected layer at each pixel location (for each height and width). If we tie the weights across each spatial location, we could think of this as a 1×1 convolutional layer.
- The NiN block consists of one convolutional layer followed by two 1×1 convolutional layers that act as per-pixel fully-connected layers with ReLU activations.
- The convolution window shape of the first layer is typically set by the user. The subsequent window shapes are fixed to 1×1.

NIN BLOCKS



credit : D2DL

Figure: NiN block.

GLOBAL AVERAGE POOLING

- Problem setting: tackle overfitting in the final fully connected layer.
  - Classic pooling removes spatial information and is mainly used for dimension and parameter reduction.
  - The elements of the final feature maps are connected to the output layer via a dense layer. This could require a huge number of weights increasing the danger of overfitting.
  - Example: 256 feature maps of dim 100x100 connected to 10 output neurons lead to 256×10^5 weights for the final dense layer.

GLOBAL AVERAGE POOLING

- Solution:
  - Average each final feature map to the element of one global average pooling (GAP) vector.
  - Example: 256 feature maps are now reduced to GAP-vector of length 256 yielding a final dense layer with 2560 weights.

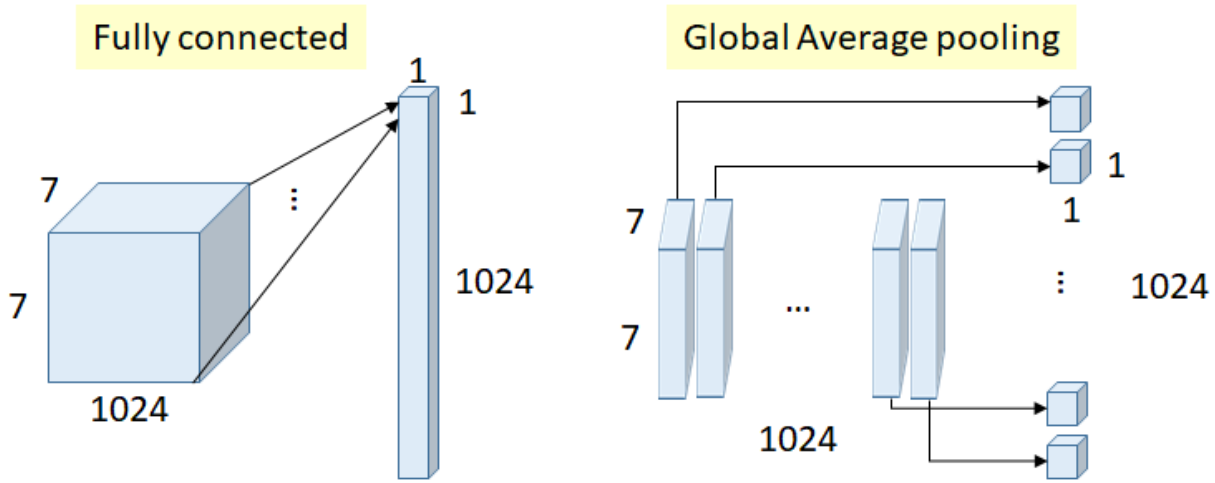


Figure: An Example of Fully Connected Layer VS Global Average Pooling Layer

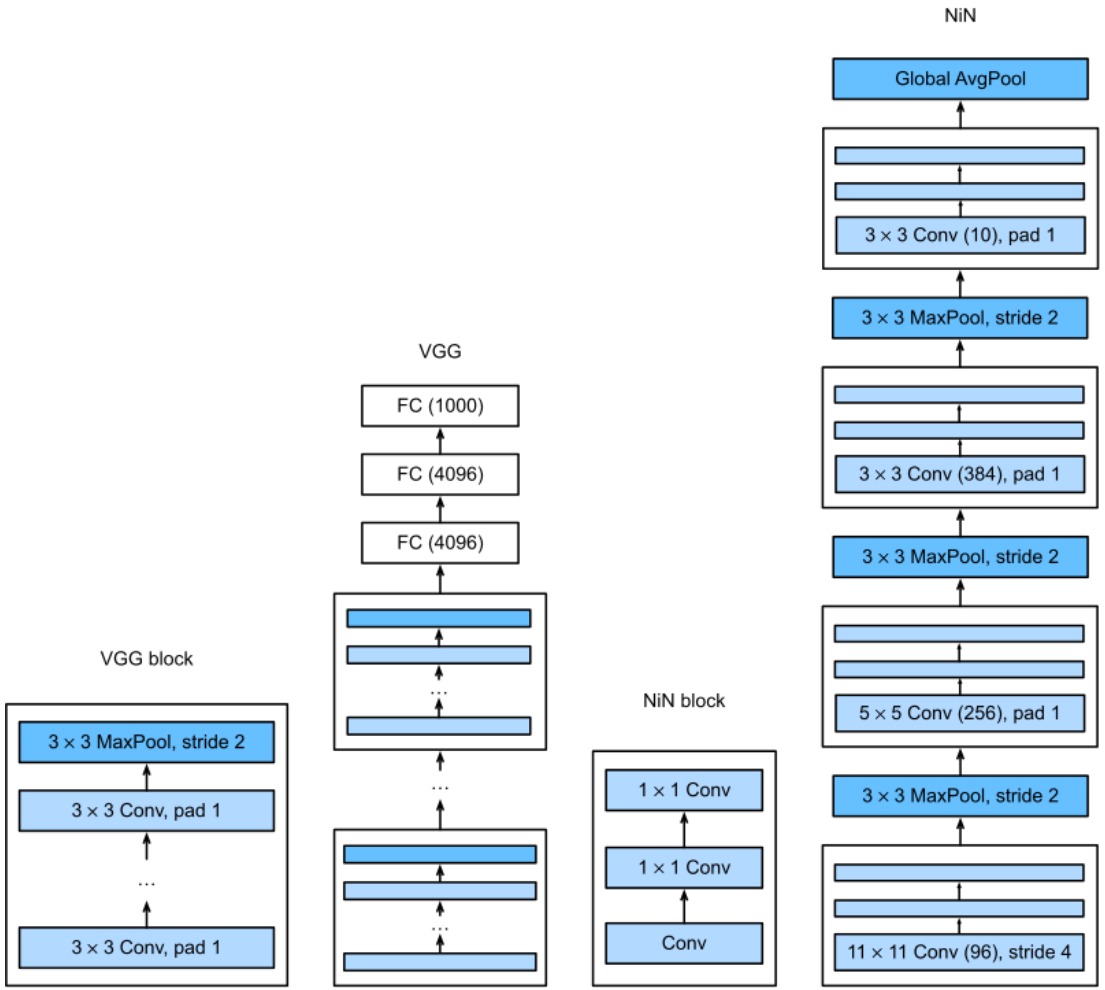
GLOBAL AVERAGE POOLING

- GAP preserves whole information from the single feature maps whilst decreasing the dimension.
- Mitigates the possibly **destructive** effect of pooling.
- Each element of the GAP output represents the activation of a certain feature on the input data.
- Acts as an additional regularizer on the final fully connected layer.

NETWORK IN NETWORK (NIN)

- NiN uses blocks consisting of a convolutional layer and multiple 1×1 convolutional layers. This can be used within the convolutional stack to allow for more per-pixel nonlinearity.
- NiN removes the fully-connected layers and replaces them with global average pooling (i.e., summing over all locations) after reducing the number of channels to the desired number of outputs (e.g., 10 for Fashion-MNIST).
- Removing the fully-connected layers reduces overfitting. NiN has dramatically fewer parameters.
- The NiN design influenced many subsequent CNN designs.

NETWORK IN NETWORK (NIN)



credit : D2DL

Figure: Comparing architectures of VGG and NiN, and their blocks.

REFERENCES

B. Zhou, Khosla, A., Labedriza, A., Oliva, A. and A. Torralba (2016)

Deconvolution and Checkerboard Artifacts

http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich (2014)

Going deeper with convolutions

https://arxiv.org/abs/1409.4842

Kaiming He, Zhang, Xiangyu, Ren, Shaoqing, and Jian Sun (2015)

Deep Residual Learning for Image Recognition

https://arxiv.org/abs/1512.03385

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva and Antonio Torralba (2016)

Learning Deep Features for Discriminative Localization

http://cnnlocalization.csail.mit.edu/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf