Deep Learning

Lecture 3

Recap

- Gradient descent for neural networks
- Weight decay
- Dropout

Weight Initialization

Zero initialization or constant initialization

Random initialization

Too big and too low values

More smart approaches

Idea

$$a^{l-1} = g^{l-1}(z^{l-1})$$
$$z^l = W^l a^{l-1} + b^l$$
$$a^l = g^l(z^l)$$

$$\mathbb{E}(a^{l-1}) = \mathbb{E}(a^l)$$
$$Var(a^{l-1}) = Var(a^l)$$

Xavier and He initialization

$$W^l \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{1}{n^{l-1}}) \quad b^l = 0$$

$$W^l \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{2}{n^{l-1}}) \quad b^l = 0$$

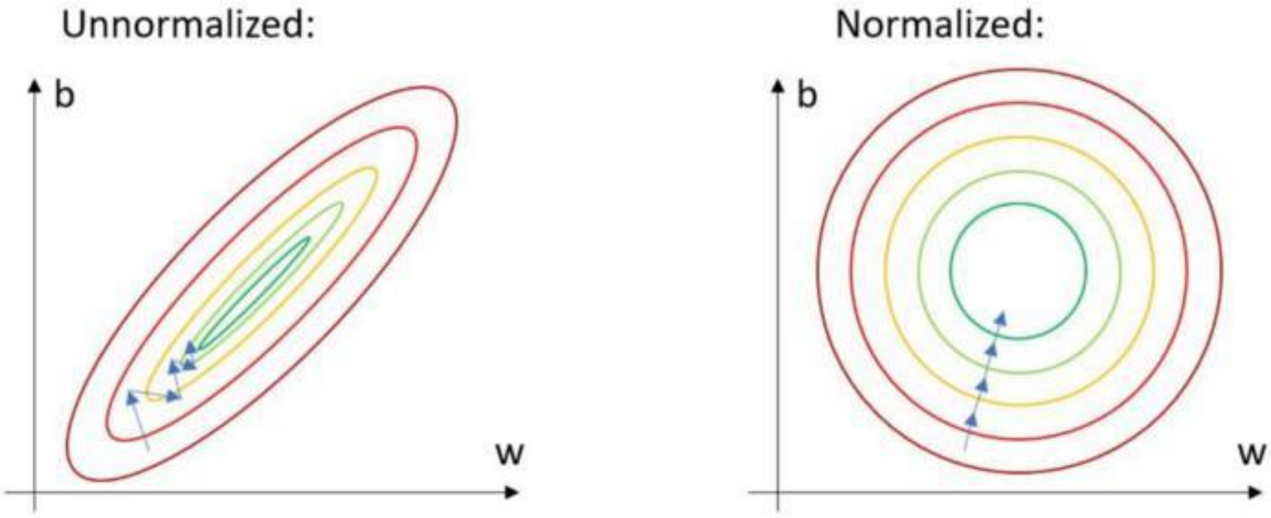
https://www.deeplearning.ai/ai-notes/initialization/index.html

Batch Normalization

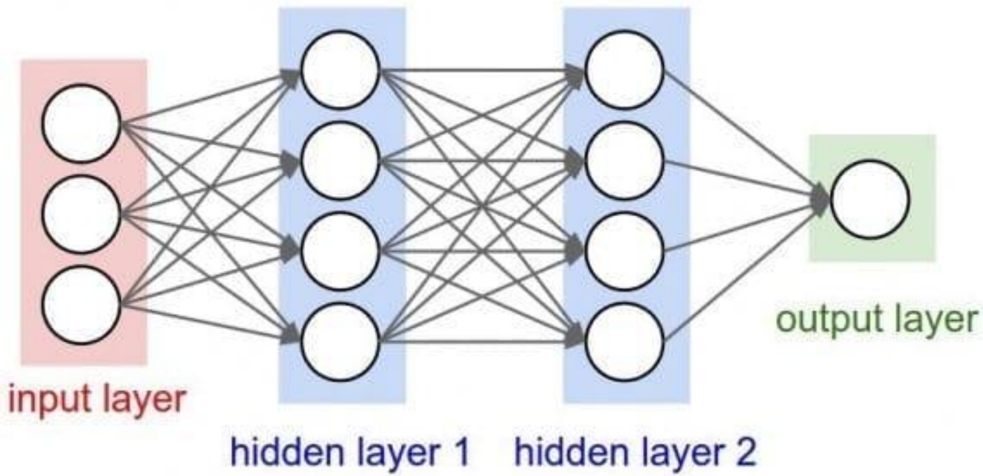
- Why we need this?

Batch Normalization

- More stable computations
- Better for optimization methods



What about neural networks?



- Can’t use whole dataset
- Idea: use minibatch

$$\{x_{ij}\}_{i=1,j=1}^{N_{batch},d}$$

$$\mu_j = \frac{\sum_{i=1}^{N_{batch}} x_{ij}}{N_{batch}}$$

$$\sigma_j^2 = \frac{\sum_{i=1}(x_{ij} - \mu_j)^2}{N_{batch}}$$

Batch normalisation

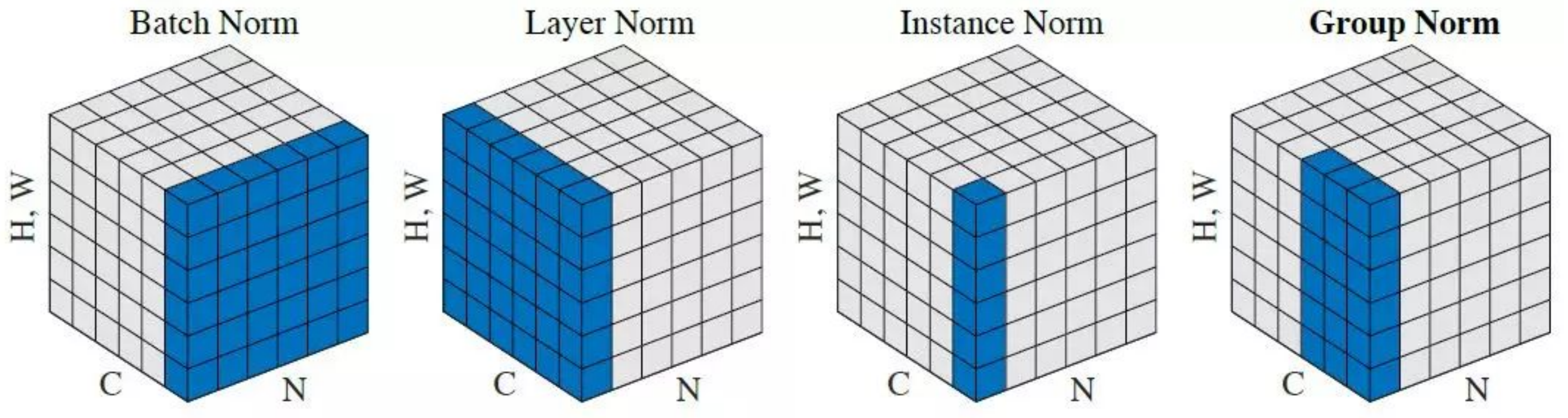
$$\mu_j = \frac{\sum_{i=1}^{N_{batch}} x_{ij}}{N_{batch}}$$

$$\sigma_j^2 = \frac{\sum_{i=1}(x_{ij} - \mu_j)^2}{N_{batch}}$$

$$\hat{y}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_{ij}^2 + \varepsilon}}$$

$$y_{ij} = \gamma_j \hat{y}_{ij} + \delta_j$$

Normalization

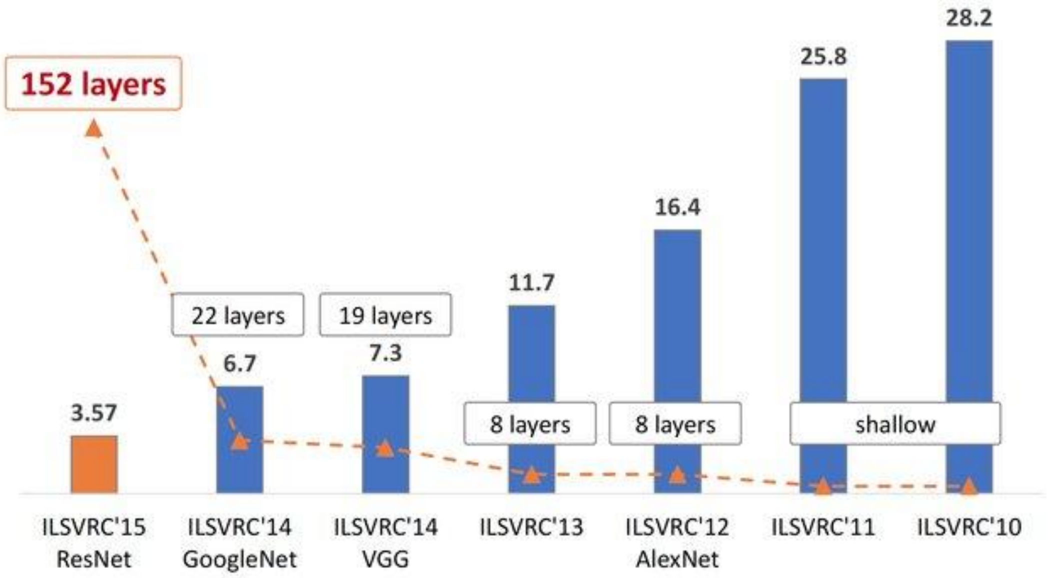


Convolution NN

ImageNet

- The ImageNet dataset contains 14,197,122 annotated images.
- Total number of non-empty WordNet synsets: 21841.
- Since 2010 the dataset is used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark in image classification and object detection.

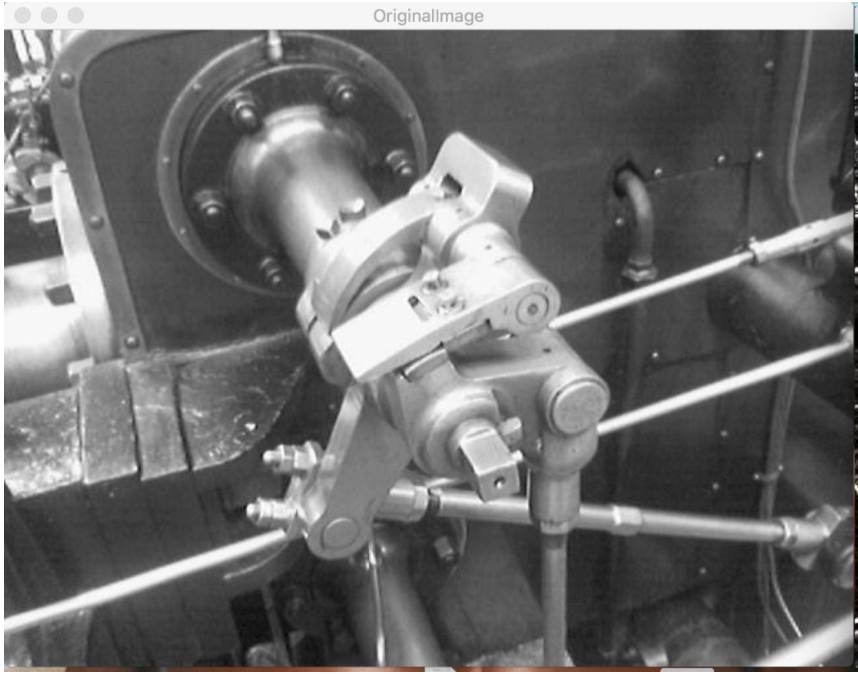
ImageNet Results



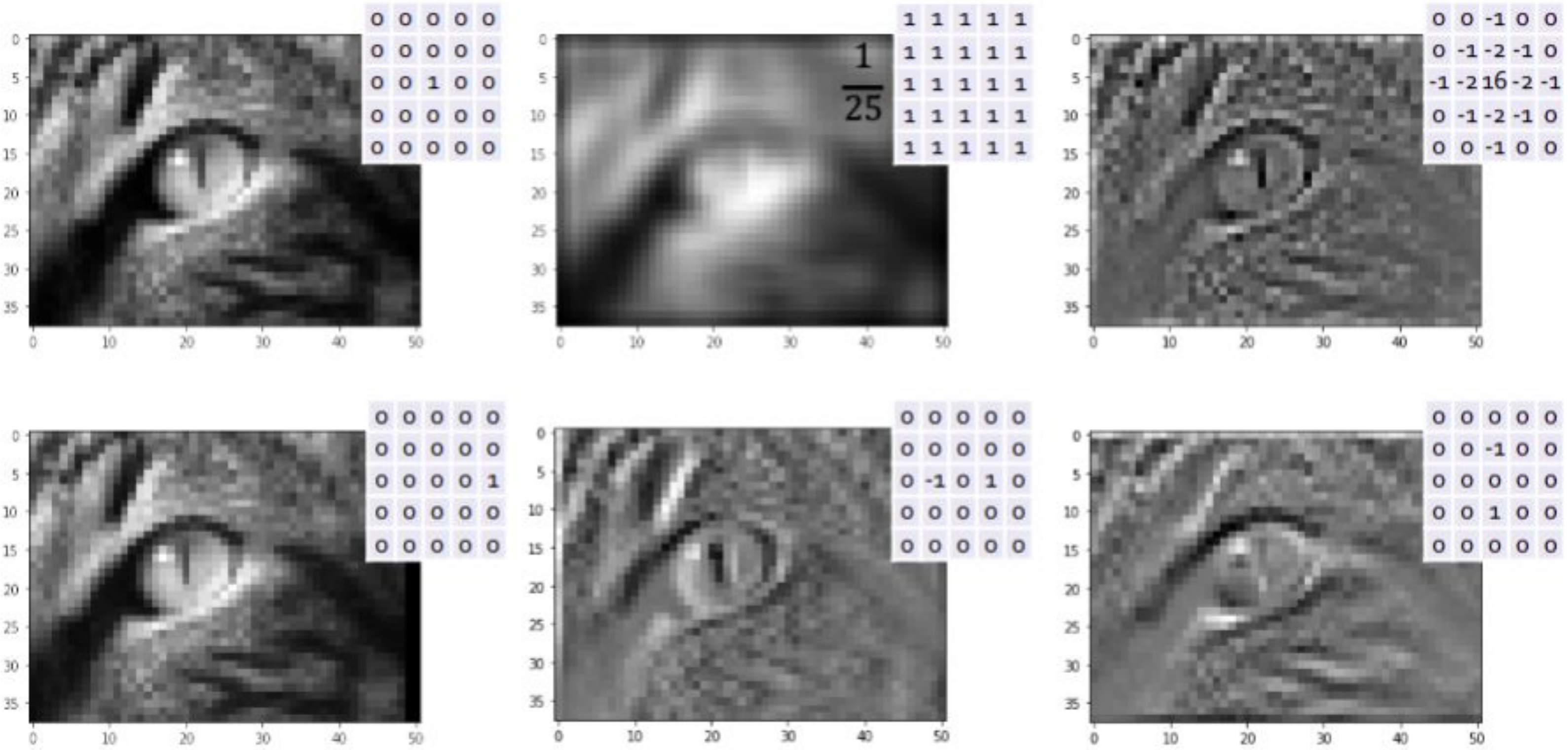
Why we can't use MLP for images?

- Too many parameters
- Fixed dimension of images
- Features will be too correlated

Sobel and other filters

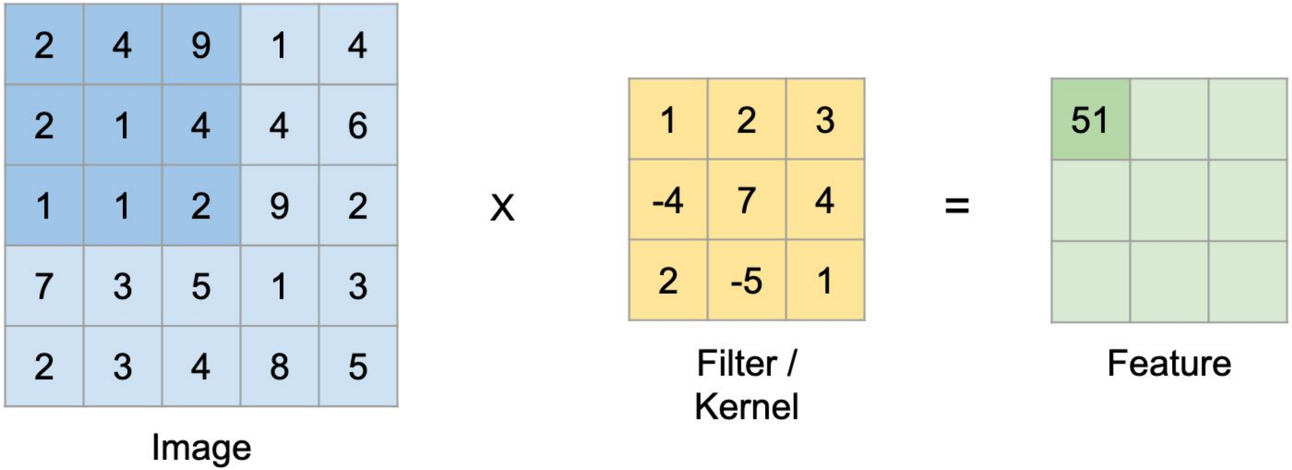


Sobel and other filters

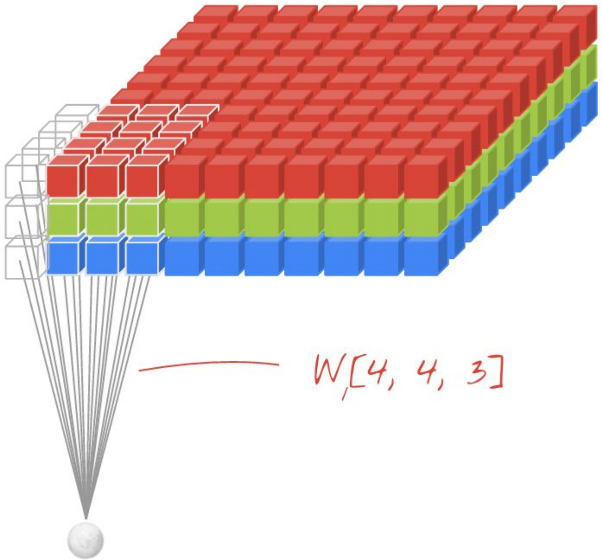


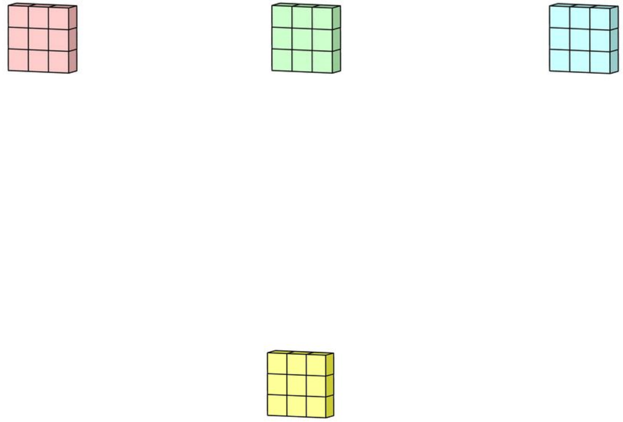
Convolution

$$Y(i,j) = \sum_{u,v} X(i+u, j+v) K(u,v)$$



Convolution for 3D tensor





Backpropagation for Conv

$$\begin{pmatrix} y_1 & y_2 \\ y_3 & y_4 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix} \times \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix}$$

$$y = Kx$$

Backpropagation for Conv

$$y = Kx$$

$$\nabla_y L \to \nabla_x L, \nabla_K L$$

$$dy = dKx + Kdx$$

$$dL = \nabla_y L^T dy = \nabla_y L^T (dKx + Kdx) = tr(\nabla_K L^T dK) + \nabla_x L^T dx$$

$$\nabla_x L = K^T \nabla_y L \qquad \nabla_K L = \nabla_y L x^T$$

Backpropagation for Conv

$$\nabla_x L = K^T \nabla_y L \quad \nabla_x L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & (\nabla_y L)_1 & (\nabla_y L)_2 & 0 \\ 0 & (\nabla_y L)_3 & (\nabla_y L)_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} k_4 & k_3 \\ k_2 & k_1 \end{pmatrix}$$

Backpropagation for Conv

$$\nabla_K L = \nabla_y L x^T \qquad dK = \begin{pmatrix} dk_1 & dk_2 & 0 & dk_3 & dk_4 & 0 & 0 & 0 & 0 \\ 0 & dk_1 & dk_2 & 0 & dk_3 & dk_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & dk_1 & dk_2 & 0 & dk_3 & dk_4 & 0 \\ 0 & 0 & 0 & 0 & dk_1 & dk_2 & 0 & dk_3 & dk_4 \end{pmatrix}$$

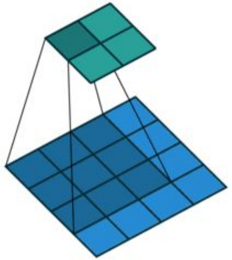
$$dL = tr(\nabla_K L^T dK) = \nabla_k L^T dk \qquad dk = \begin{pmatrix} dk_1 \\ dk_2 \\ dk_3 \\ dk_4 \end{pmatrix}$$

$$(\nabla_k L)_1 = (\nabla_K L)_{11} + (\nabla_K L)_{22} + (\nabla_K L)_{34} + (\nabla_K L)_{45}$$

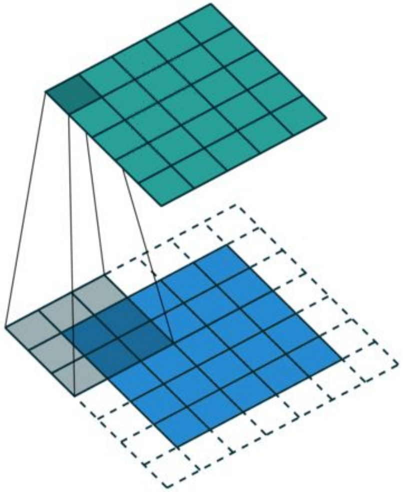
$$(\nabla_k L)_1 = (\nabla_y L)_1 x_1 + (\nabla_y L)_2 x_2 + (\nabla_y L)_3 x_4 + (\nabla_y L)_4 x_5$$

$$\nabla_k L = X \times \nabla_y L$$

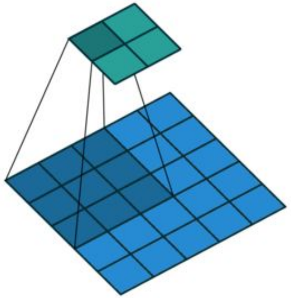
Different types of convolution



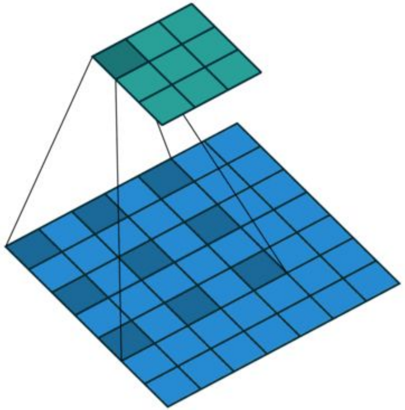
No padding,
no strides



Padding 1,
no strides

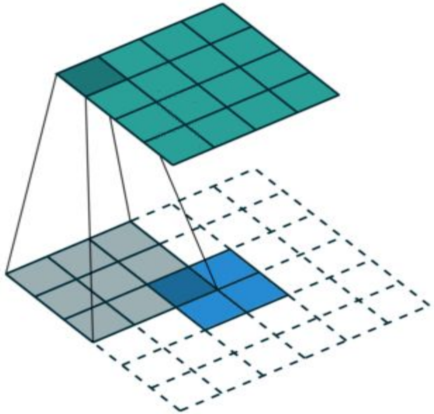


No padding,
stride = 2

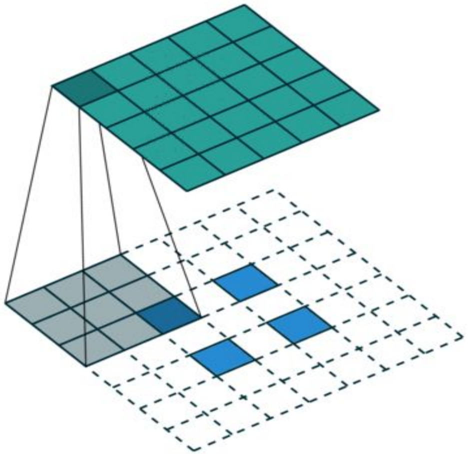


No padding,
no stride,
dilation = 2

Different types of convolution

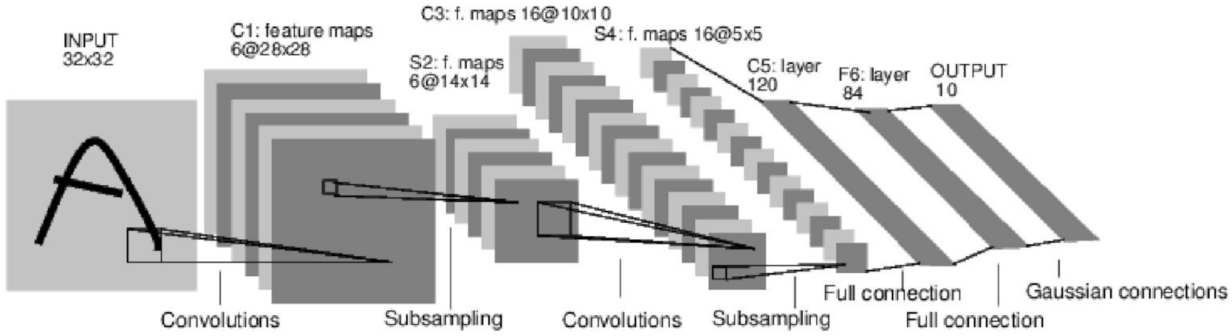


Transposed convolution



Transposed convolution
stride = 2

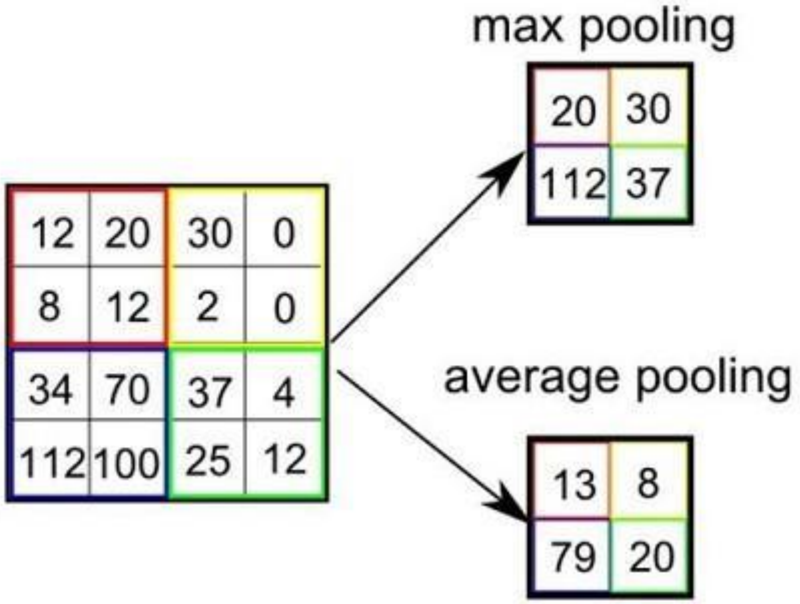
LeNet [LeCun et al., 1998]



Operation types:

- Convolutions
- Nonlinearities
- Pooling
- Full connection layers

What is pooling?



- Usual motivation: adding invariance to small shifts
- Several max-poolings can accumulate invariance to stronger shifts
- No invariance/covariance to scaling, rotation, color changes!

Recap

- Weight initialization
- Batch Normalization
- CNN