STOR566: Introduction to Deep Learning

Lecture 7: Convolutional Neural Networks

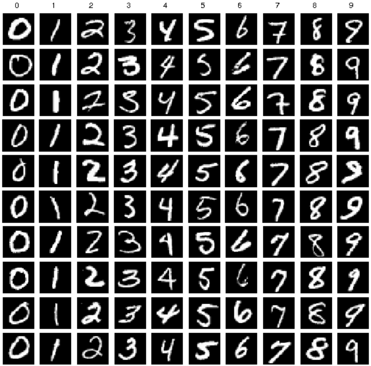
Yao Li
UNC Chapel Hill

Sep 12, 2024

Materials are from *Learning from data (Caltech)* and *Deep Learning (UCLA)*

MNIST

- Hand-written digits (0 to 9)
- Total $60,000$ samples, 10-class classification.



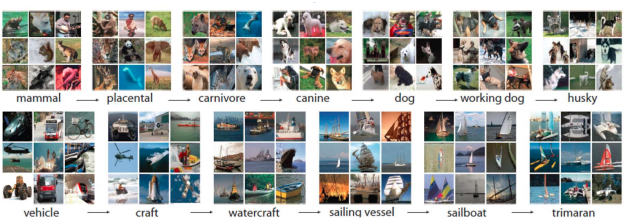
MNIST Classification Accuracy

- See the website by Yann LeCun:

  `http://yann.lecun.com/exdb/mnist/`

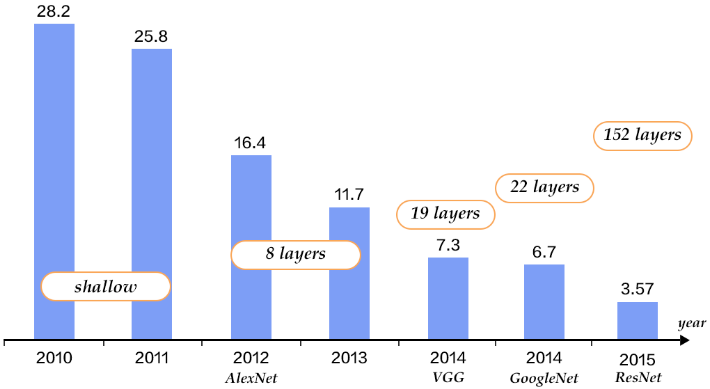
Classifier	Test Error
Linear classifier	12.0 %
SVM, Gaussian kernel	1.4%
SVM, degree 4 polynomial	1.1%
Best SVM result	0.56%
2-layer NN	$\sim$ 3.0%
3-layer NN	$\sim$ 2.5%
CNN, LeNet-5 (1998)	0.85%
Larger CNN (2011, 2012)	$\sim$ 0.3%

ImageNet Data



- ILSVRC competition: 1000 classes and about 1.2 million images
- Full imagenet: $> 20,000$ categories, each with about a thousand images.

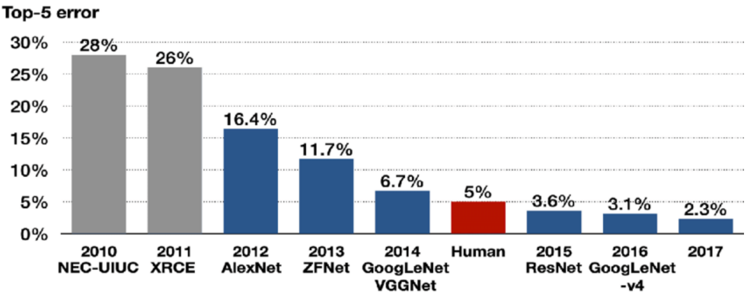
ImageNet Results



Top-5 error rates on ILSVRC image classification

picture from `http://www.paddlepaddle.org/documentation/book/en/0.14.0/03.image_classification/index.html`

ImageNet Results



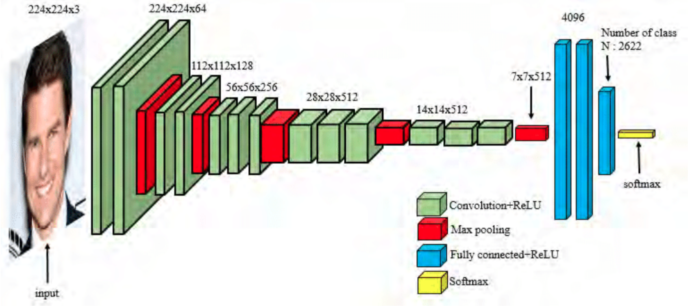
Top-5 error rates on ILSVRC image classification

Picture from `https://www.researchgate.net/figure/Algorithms-that-won-the-ImageNet-Large-Scale-Visual-Recognition-Challenge-I`
`fig2_346091812`

Convolutional Neural Network

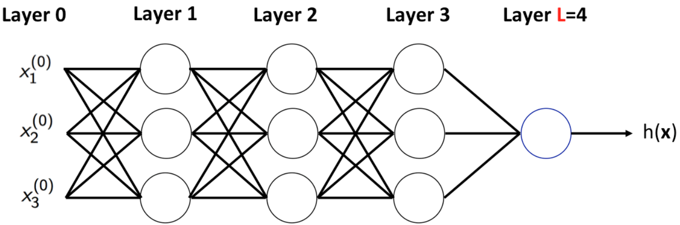
The structure of CNN

- Structure of VGG



- Two important layers:
  - Convolution
  - Pooling

Neural Networks



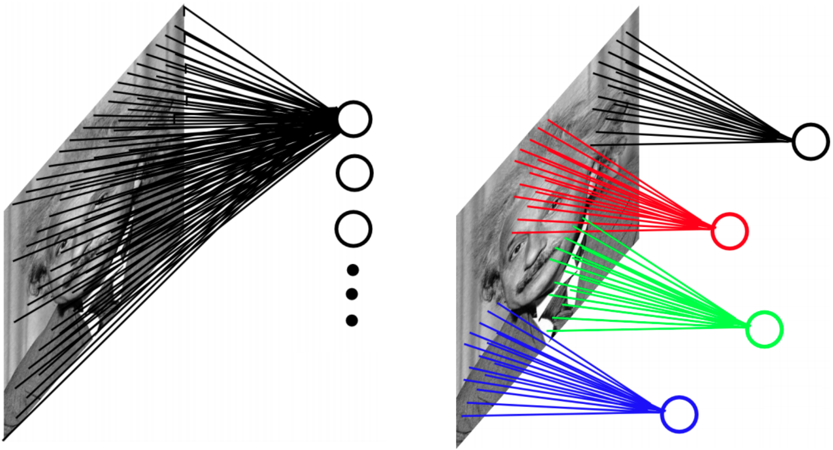
Number of parameters in the network?

Convolution Layer

- Fully connected layers have too many parameters

  $\Rightarrow$ poor performance
- Example: VGG first layer
  - Input: $224 \times 224 \times 3$
  - Output: $224 \times 224 \times 64$
  - Number of parameters if we use fully connected net: $(224 \times 224 \times 3) \times (224 \times 224 \times 64) =$ 483 billion
- Convolution layer leads to:
  - Local connectivity
  - Parameter sharing

Local connectivity



(Figure from Salakhutdinov 2017)

Parameter Sharing

- Making a reasonable assumption:

If one feature is useful to compute at some spatial position (x, y), then it should also be useful to compute at a different position (x_2, y_2)

- Using the convolution operator

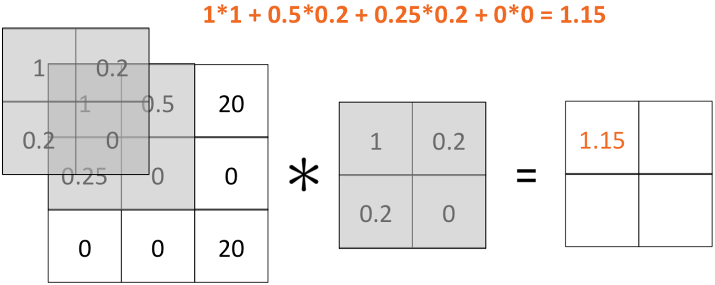
Convolution

- The convolution of an image x with a kernel k is computed as

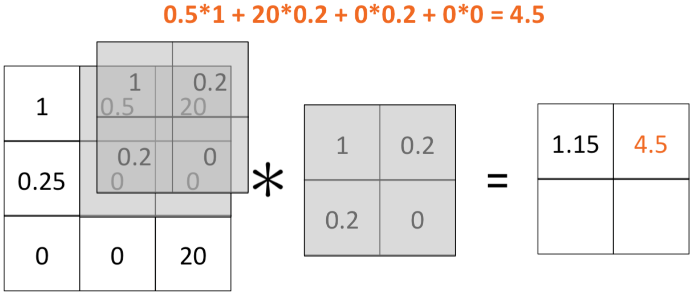
$$(x * k)_{ij} = \sum_{pq} x_{i+p,j+q} k_{p,q}$$

$$\begin{array}{|c|c|c|}\hline 1 & 0.5 & 20 \\ \hline 0.25 & 0 & 0 \\ \hline 0 & 0 & 20 \\ \hline \end{array} * \begin{array}{|c|c|}\hline 1 & 0.5 \\ \hline 0.25 & 0 \\ \hline \end{array} = \begin{array}{|c|c|}\hline \quad & \quad \\ \hline \quad & \quad \\ \hline \end{array}$$

Convolution

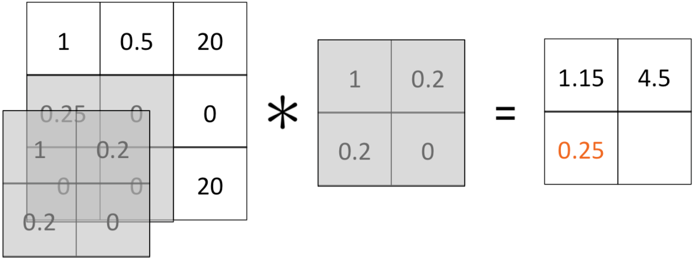


Convolution



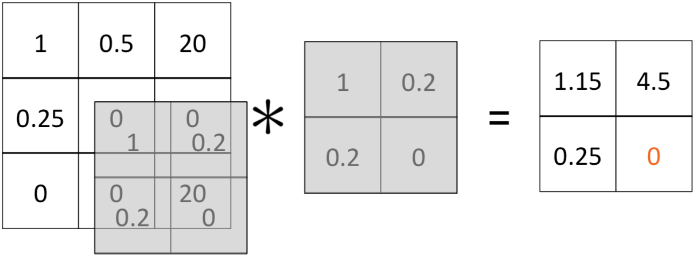
Convolution

0.25*1 + 0*0.2 + 0*0.2 + 0*0 = 0.25



Convolution

0*1 + 0*0.2 + 0*0.2 + 20*0 = 0



Multiple Channels

- Multiple input channels:

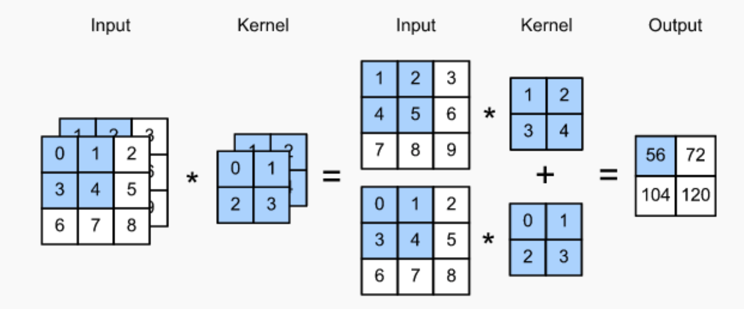
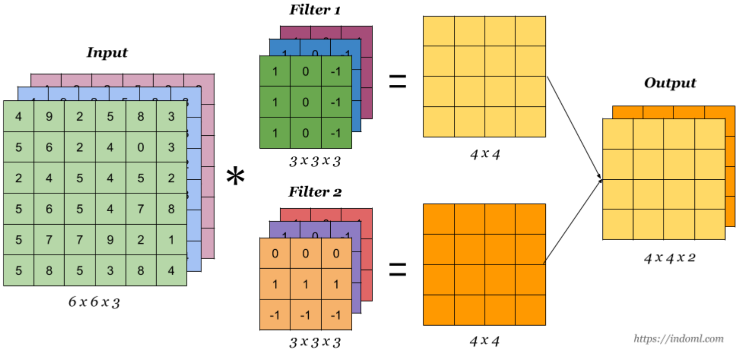


Image from Dive into Deep Learning

- $(1 \times 1 + 2 \times 2 + 4 \times 3 + 5 \times 4) + (0 \times 0 + 1 \times 1 + 3 \times 2 + 4 \times 3) = 56$

Multiple Channels

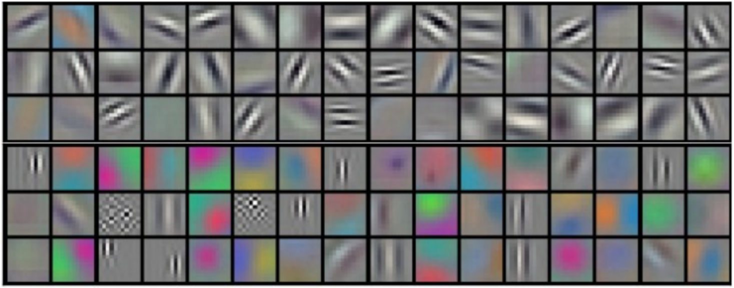
- Multiple input channels and output channels:



- Number of parameters: $k_1 \times k_2 \times d_{in} \times d_{out} + d_{out}$

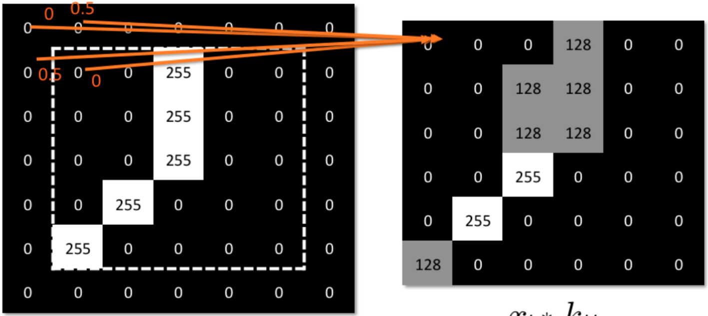
Learned Kernels

- Example kernels learned by AlexNet



Padding

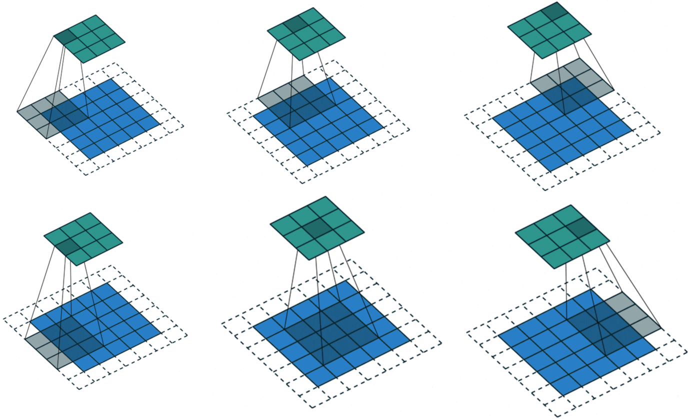
- Use zero padding to allow going over the boundary
    - Easier to control the size of output layer



x_i $\quad$ $x_i * k_{ij}$

Strides

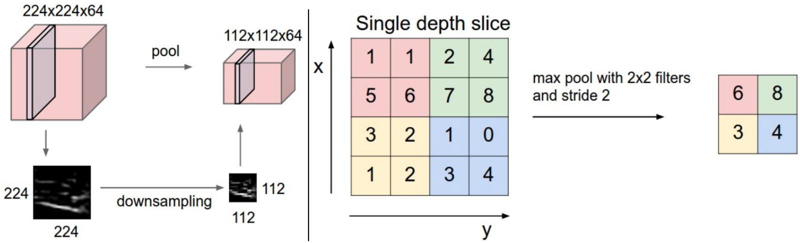
- Stride: The amount of movement between applications of the filter to the input image
- Strude $= (1, 1)$: no stride



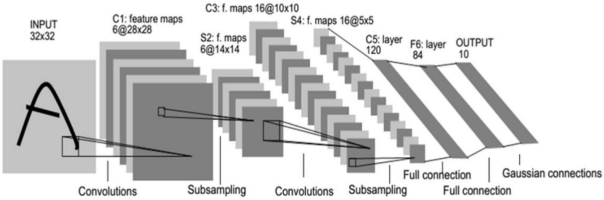
stride = (2,2)

Pooling

- It's common to insert a pooling layer in-between successive convolutional layers
- Reduce the size of representation, down-sampling
- Example: Max Pooling



Example: LeNet5

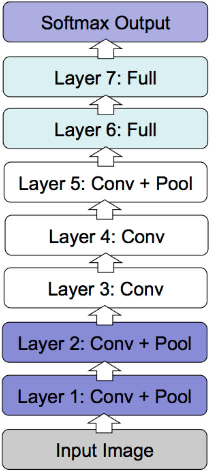


- Input: 32×32 images (MNIST)
- Convolution 1: 6 5×5 filters, stride 1
  - Output: 6 28×28 maps
- Pooling 1: 2×2 max pooling, stride 2
  - Output: 6 14×14 maps
- Convolution 2: 16 5×5 filters, stride 1
  - Output: 16 10×10 maps
- Pooling 2: 2×2 max pooling with stride 2
  - Output: 16 5×5 maps (total 400 values)
- 3 fully connected layers: $120 \Rightarrow 84 \Rightarrow 10$ neurons

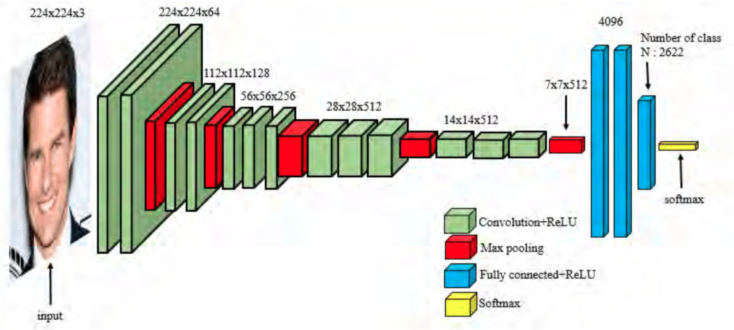
AlexNet

- 8 layers in total, about 60 million parameters and 650,000 neurons.
- Trained on ImageNet dataset

  "ImageNet Classification with Deep Convolutional Neural Networks", by Krizhevsky, Sustskever and Hinton, NIPS 2012.

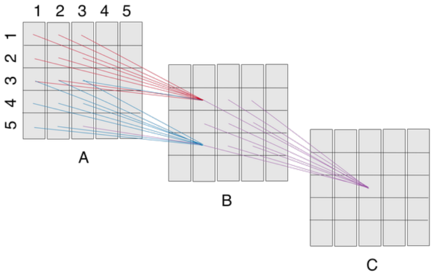


Example: VGG Network



What do the kernels learn?

- The receptive field of a neuron is the input region that can affect the neuron's output
- The receptive field for a first layer neuron is its neighbors (depending on kernel size) $\Rightarrow$ capturing very local patterns
- For higher layer neurons, the receptive field can be much larger $\Rightarrow$ capturing global patterns



Conclusions

- Convolution
- Pooling

Questions?