Convolutional Neural Nets (CNNs)

Nan Ye

School of Mathematics and Physics
The University of Queensland

Applications

- CNNs are inspired by how biological vision works.
- CNNs are useful for dealing with array inputs in which nearby values are correlated.
- Examples: images, video, sound

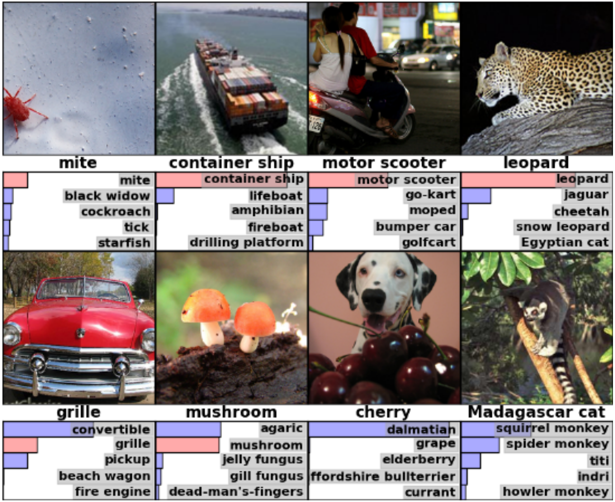
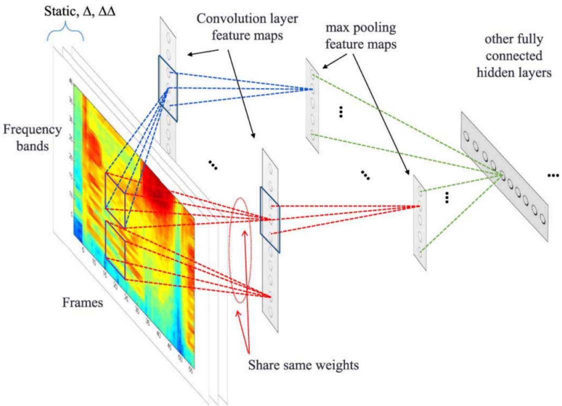


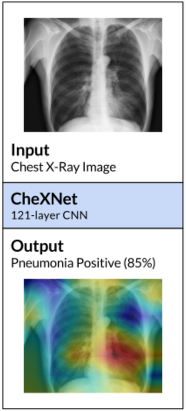
Image classification

Krizhevsky, Sutskever, and Hinton, Imagenet classification with deep convolutional neural networks, 2012



Speech recognition

Abdel-Hamid et al., Convolutional neural networks for speech recognition, 2014



Pneumonia detection from chest X-rays

Rajpurkar et al., CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 2017

Biological Vision

- Hubel & Wiesel (1950s and 1960s) showed that cat and monkey visual cortices contain neurons that individually respond to small regions of the visual field.
- The firing of a single neuron is affected by a certain region of the visual space, known as the receptive field of the neuron.
- Neighboring cells have similar and overlapping receptive fields.
- Some cells can detect edges irrespective of where they occur.

Convolutional Neural Nets (CNNs)

- CNNs are multilayer feedforward neural networks
  - they are MLPs where the weights have been constrained to mimic how biological vision works
- Three architectural ideas
  - Local receptive fields
  - Shared weights
  - Spatial or temporal sub-sampling

  These ensure some degree of shift, scale, and distortion invariance.

- There are two key building blocks
  - The convolutional layer, which consists of a number of filters
    - filters are also called kernels, feature detectors
    - each filter scans small patches in the input to detect features
  - The downsampling layer, which reduces the resolution of the image for learning higher-level features.

Convolution

- Convolution in CNNs is not convolution in maths.
- Convolution in CNNs is known as cross-correlation, or sliding inner product in maths.

2D Convolution (in CNN)

- Given an $N \times N$ input, the convolution operation slides one filter through the input to extract features
  - An $F \times F$ filter is simply an $F \times F$ weight matrix.
  - We slide the filter over all $F \times F$ subarrays.
  - For each subarray, we compute the weighted sum of its elements (i.e., the dot product between the filter and the sub-array).
  - This gives us an $(N - F + 1) \times (N - F + 1)$ feature/activation map.

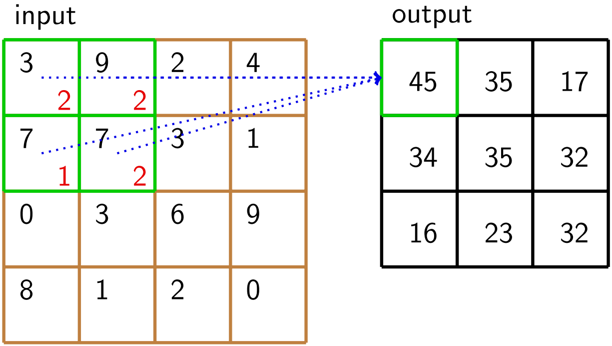
Example. 2x2 filter applied to 4x4 input

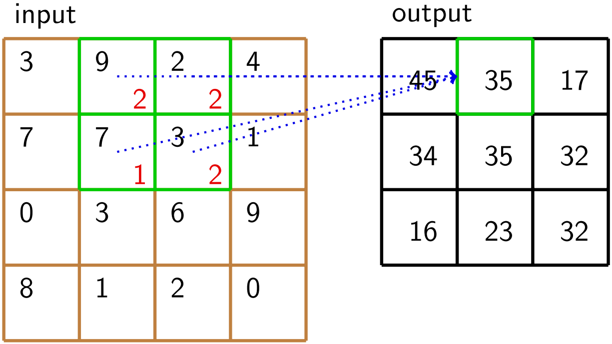
input

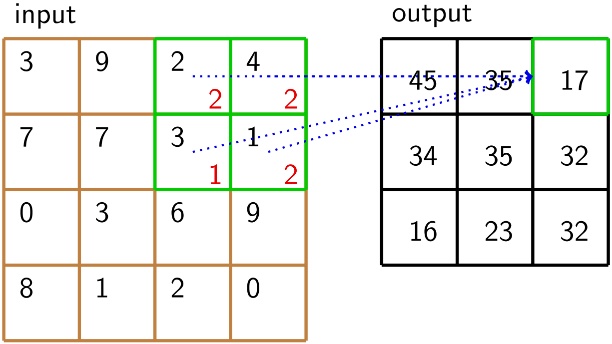
3	9	2	4
7	7	3	1
0	3	6	9
8	1	2	0

filter

2	2
1	2


input
3 9 2 4
2 2
7 7 3 1
1 2
0 3 6 9
8 1 2 0
output
45 35 17
34 35 32
16 23 32


input
3 9 2 4
7 7 3 1
0 3 6 9
8 1 2 0
2 2
1 2
output
45 35 17
34 35 32
16 23 32


input
3 9 2 4
7 7 3 1
0 3 6 9
8 1 2 0
2 2
1 2
output
45 35 17
34 35 32
16 23 32

In the language of neural nets...

- 4x4 input matrix = outputs of 4x4 input neurons
- 3x3 output matrix = outputs of 3x3 neurons in the conv. layer
- Each output neuron is connected to 4 of the 4x4 input neurons.
- The 4 weights are shared for all the output neurons.

input

3	9	2	4
7	7	3	1
0	3	6	9
8	1	2	0

output

45	35	17
34	35	32
16	23	32

Example. 2x2 filter applied to 5x5 input with stride 2

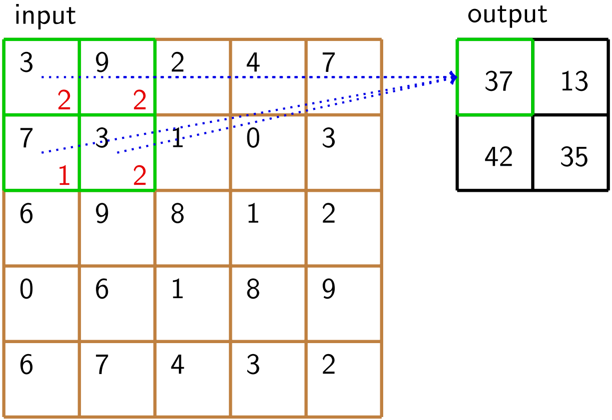
input

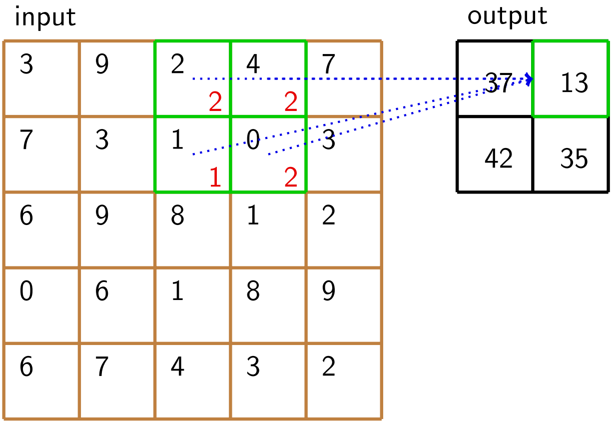
3	9	2	4	7
7	3	1	0	3
6	9	8	1	2
0	6	1	8	9
6	7	4	3	2

filter

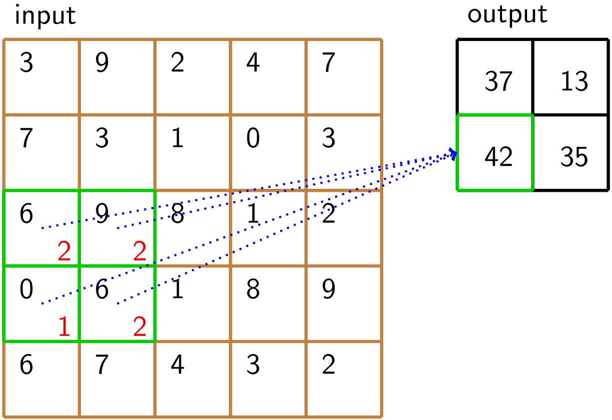
2	2
1	2

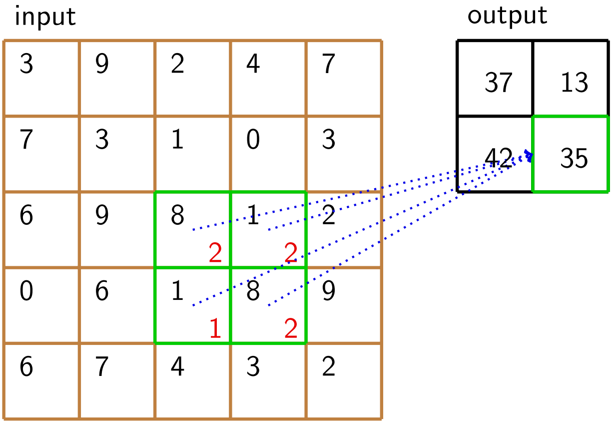
With stride=2, we skip 2 cells each time.


input
3 9 2 4 7
7 3 1 0 3
6 9 8 1 2
0 6 1 8 9
6 7 4 3 2
2 2
1 2
output
37 13
42 35



input
3
9
2
2
4
2
7
7
3
1
1
0
2
3
6
9
8
1
2
0
6
1
8
9
6
7
4
3
2
output
37
13
42
35


input
output
3 9 2 4 7
7 3 1 0 3
6 9 8 1 2
0 6 1 8 9
6 7 4 3 2
2 2
1 2
37 13
42 35



$N \times N$ input, $F \times F$ filter with stride S $\Rightarrow$ output size $\lfloor \frac{N-F}{S} \rfloor + 1$

Zero-padding, dilation and bias

- We often pad each side of the input with P zeros (or other constants)
    - this allows the filters to scan elements near the borders
- Sometimes, in a filter with dilation D, its cells are D cells apart ($D = 1$ in previous examples).
- $N \times N$ input, $F \times F$ filter, pad P zeros on each side, dilation D, stride $S \Rightarrow$ output size $\lfloor \frac{N+2P-D(F-1)-1}{S} \rfloor + 1$
- In general, each filter has a bias term as well.

Convolution beyond 2D

- In general, the input is not necessarily a 2D matrix, but can be a general N-dimensional array (1D, 2D, 3D,...)
- Similarly, a filter can be a general M-dimensional array (you can slide it through the input array as long as $M \leq N$).

Convolutional layer

- A convolutional layer often has several filters.
- Each filter produces a separate activation map.
- Filter weights are typically learned from data.

Your Turn

Which of the following statement is correct? (Multiple choice)

(a) A convolutional layer is a special kind of fully connected layer.

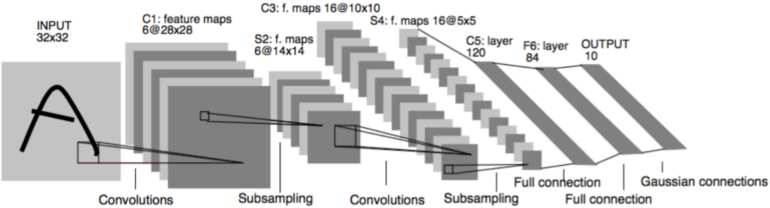
(b) Each neuron in a convolutional layer has to be connected to all input neurons.

(c) Convolutional layer is designed to extract features from array data.

Sub-sampling

- Sub-sampling (or pooling) is very similar to convolution.
- In average pooling, when we slide the filter through the input, we simply take the average of the input elements being scanned as the output.
- In max pooling, we replace average by max.
- The default stride is equal to the filter size (i.e. we do not pool the same element twice).

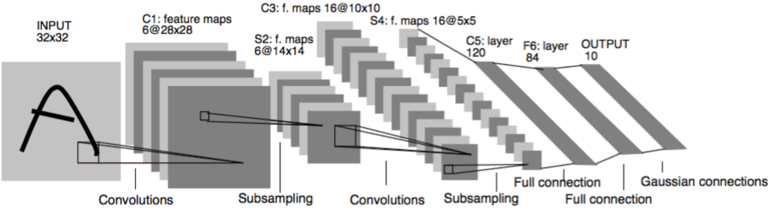
LeNet-5 (1998)



- 7 layers (excluding input layer)
- Layer 1,3,5 are convolution layers (C1, C3, C5)
- Layer 2,4 are sub-sampling layers (S2, S4)
- Layer 6 is fully-connected (F6)
- Layer 7 is the output layer

LeCun et al., Gradient-based learning applied to document recognition, 1998

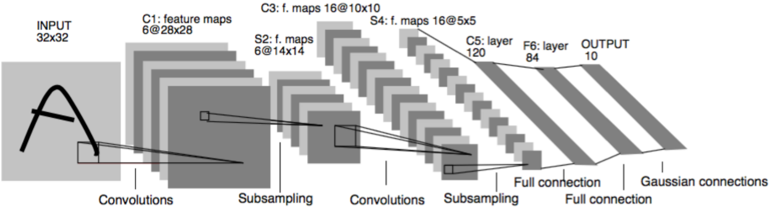
LeNet-5 (1998)



- Activation function is hyperbolic tangent up to F6.
- Output layer uses the Euclidean Radial Basis Function (RBF) units (each computes the squared distance between the input vector and the weight vector of the unit).

LeCun et al., Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)

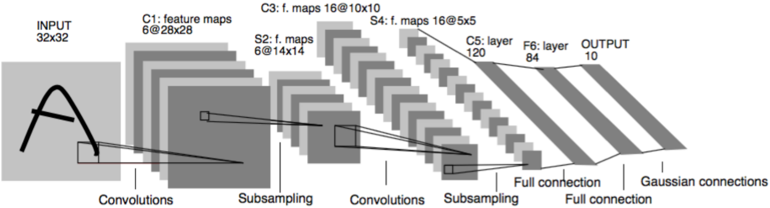


Convolutional layers

- Each convolutional layer has units organized as several 2D arrays.
- C1: 6 filters of size 5x5
- C3: 16 filters of size 5x5

LeCun et al., Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)

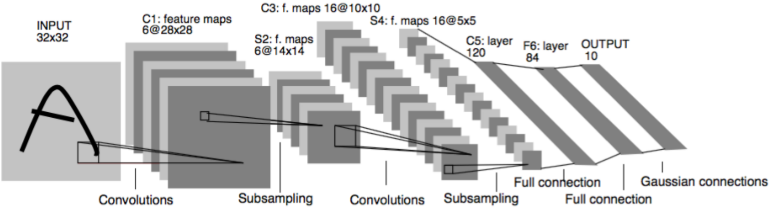


Sub-sampling/pooling layers

- Each sub-sampling layer has units organized as the same number of 2D arrays as previous convolutional layer.
- Reduces each 2D array in the previous convolutional layer to a lower resolution, by taking the average of each non-overlapping 2x2 neighborhood and adding a bias to it.

LeCun et al., Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)



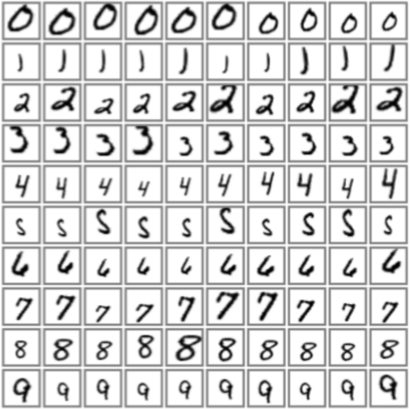
Trainable using backprop.

LeCun et al., Gradient-based learning applied to document recognition, 1998

Performance

- MNIST dataset: 60,000 training examples, 10,000 test examples, resized to 32x32.
- 0.95% error.

Adding distorted training data helps



- Additional 540,000 distorted training examples.
- Error improved to 0.8%.

Errors made by LeNet5

Variants

- Max-pooling is found to work better than average-pooling.
- Overlapping pooling is sometimes used.
- Rectified linear unit (ReLU, $\max(0, x)$) is now often used instead of sigmoid units ($\tanh(x)$ or $\sigma(x)$).

Modern CNNs

- Modern CNNs are generally much deeper and are more expressive.
- They also make use of various other ideas, such as shortcut connections, batch normalization, dropout.
- Examples: AlexNet, GoogLeNet, ResNet

More in STAT3007 Deep Learning

What You Need to Know

Convolutional neural nets

- They are special types of MLPs with sparse connections between layers.
- Three key architectural ideas: local receptive fields, weight sharing, sub-sampling.
- Two special types of layers
  - Convolutional layers
  - Sub-sampling layers