CS489/698
Lecture 17: March 8, 2017

Convolutional Neural Networks

[GBC] Chap. 9

Large networks

- What kind of neural networks can be used for large or variable length input vectors (e.g., time series)?

- Common networks:
  - Convolutional networks
  - Recursive networks
  - Recurrent networks

Convolution

- Convolution: mathematical operation on two functions $x()$ and $w()$ that produces a third function $y()$ that can be viewed as a modified version of one of the original functions $x()$

$$y(i) = \int_t x(t)w(i-t)dt$$
$$y(i) = (x * w)(t)$$

Where $*$ is an operator denoting a convolution

Example Smoothing

Discrete convolution

- Discrete convolution

$$y(i) = \sum_{t=-\infty}^{\infty} x(t)w(i-t)$$

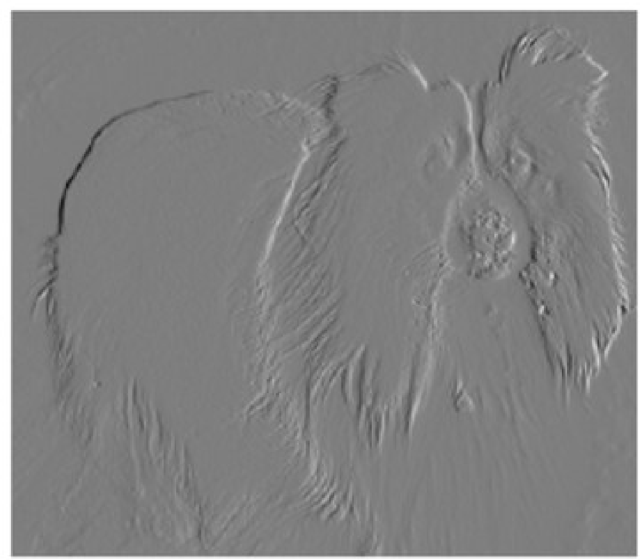
- Multidimensional convolution

$$y(i,j) = \sum_{t_1=-\infty}^{\infty} \sum_{t_2=-\infty}^{\infty} x(t_1,t_2)w(i-t_1,j-t_2)$$

Example: Edge Detection

- Consider a grey scale image
- Detect vertical edges: $y(i,j) = x(i,j) - x(i-1,j)$

$$\text{hence } w(i-t_1, j-t_2) = \begin{cases} 1 & t_1 = i, t_2 = j \\ -1 & t_1 = i+1, t_2 = j \\ 0 & \text{otherwise} \end{cases}$$



Convolutions for feature extraction

- In neural networks
  - A **convolution** denotes the linear combination of a **subset of units** based on a **specific pattern of weights.**

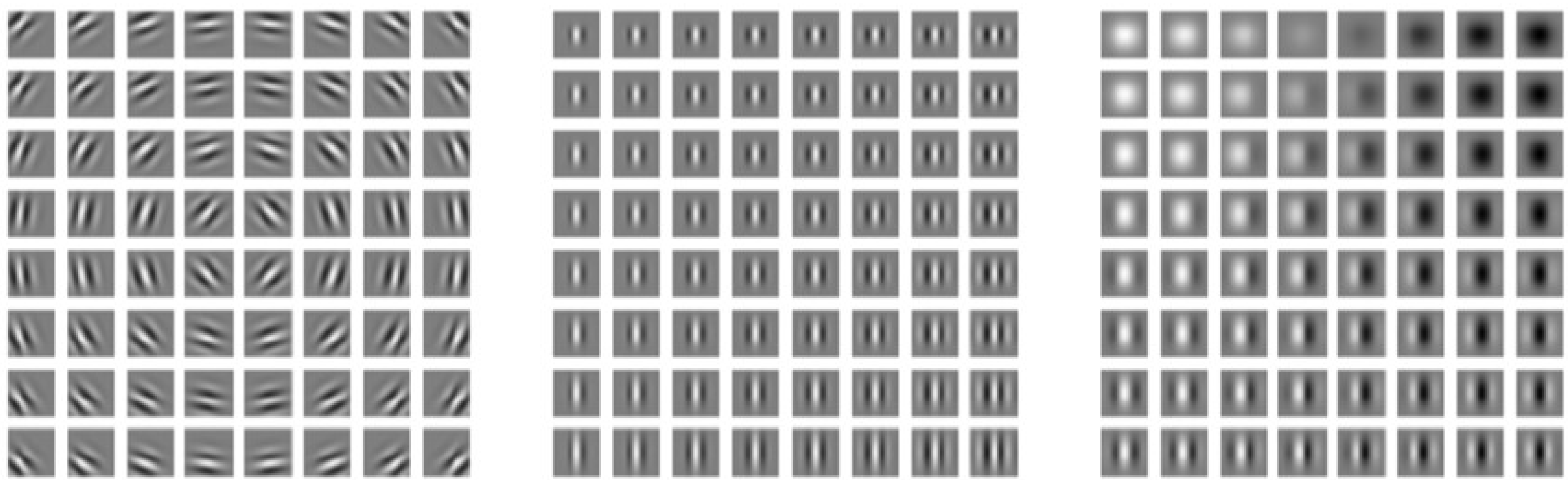
$$a_j = \sum_i w_{ji} z_i$$

  - Convolutions are often combined with an activation function to produce a feature

$$z_j = h(a_j) = h\left(\sum_i w_{ji} z_i\right)$$

Gabor filters

- Gabor filters: common feature maps inspired by the human vision system



- Weights:

  Grey: zero White: positive Black: negative

Convolution Neural Network

- A **convolutional neural network** refers to any network that includes an **alternation of convolution and pooling layers**, where **some of the convolution weights are shared.**
- Architecture:

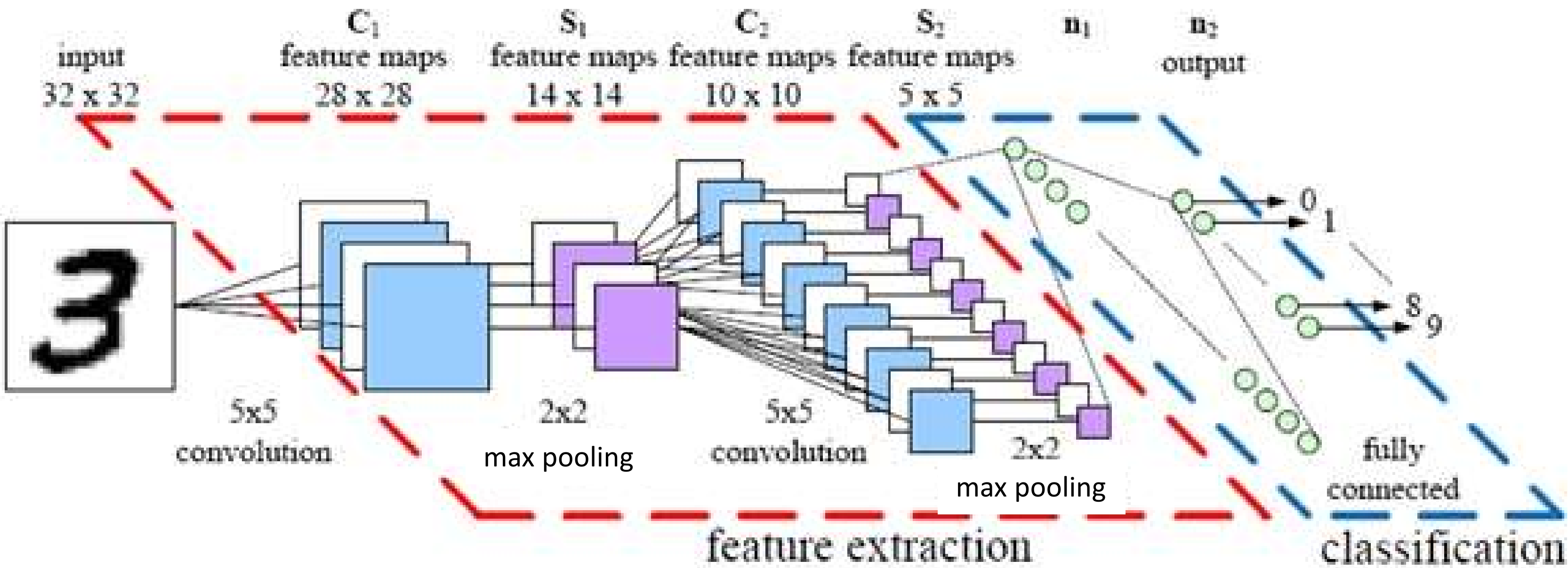
Pooling

- Pooling: **commutative** mathematical operation that combines several units
- Examples:
  - Max, sum, product, Euclidean norm, etc.

- Commutative property (order does not matter):

$$\max(a, b) = \max(b, a)$$

Example: Digit Recognition



Benefits

- Sparse interactions
  - Fewer connections
- Parameter sharing
  - Fewer weights
- Locally equivariant representation
  - Locally invariant to translations
  - Handle inputs of varying length

Training

- Convolutional neural networks are trained in the same way as other neural networks
  - E.g., backpropagation

- Weight sharing:
  - Combine gradients of shared weights into a single gradient

Applications

- Image processing
- Data with sequential, spatial, or tensor patterns