CSC520 - Artificial Intelligence

Lecture 25

Dr. Scott N. Gerard

North Carolina State University

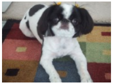
Apr 17, 2025

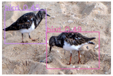
Agenda

- Computer vision tasks
- Convolution operation
- Padding and stride
- Convolution layer
- Pooling layer
- LeNet-5 Model

Computer Vision

- Computer vision's goal is to enable computers to interpret and understand images

 $\Rightarrow$ Dog

 $\Rightarrow$ 

 $\Rightarrow$ 

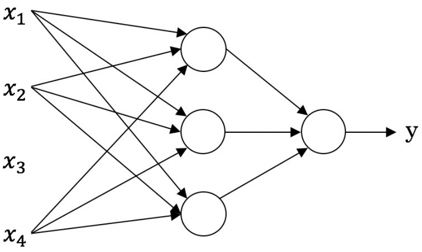
Image captioning
Face recognition
Object tracking
Human pose recognition
...

Fully-connected NN for Computer Vision Tasks

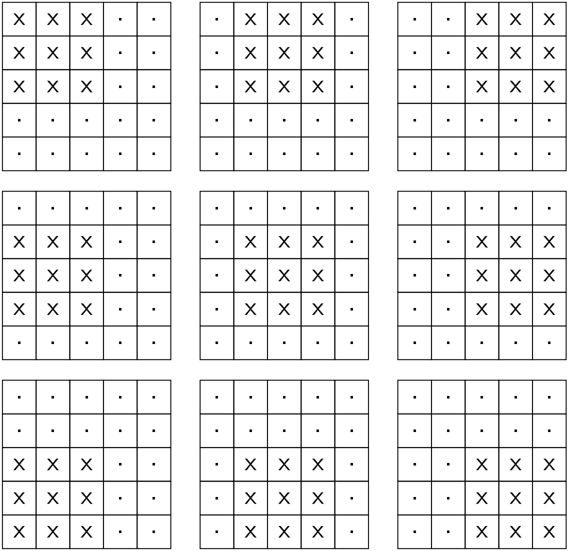
- Colored image is a 3-D (width x height x 3) grid of pixels

8	9	2	4	3
6	5	3	7	9
1	0	8	9	3
4	2	6	3	2
8	4	2	0	1
2	1	8	9	0

- For a 1000x1000 image, the number of features: 1000x1000x3 = 3M



Convolution Operation



Kernel

1	0	-1
1	0	-1
1	0	-1

Output

$\sum$	$\sum$	$\sum$
$\sum$	$\sum$	$\sum$
$\sum$	$\sum$	$\sum$

Convolution Operation

8	9	2	4	3	2
6	5	3	7	9	8
1	0	8	9	3	1
4	2	6	3	2	0
8	4	2	0	1	2
2	1	8	9	0	1

1	0	-1
1	0	-1
1	0	-1

		-2	9
-6	-12	3	10
-3	-6	10	9
-2	-5	13	9

Edge Detection using Convolution

15	15	15	0	0	0
15	15	15	0	0	0
15	15	15	0	0	0
15	15	15	15	15	15
15	15	15	15	15	15
15	15	15	15	15	15

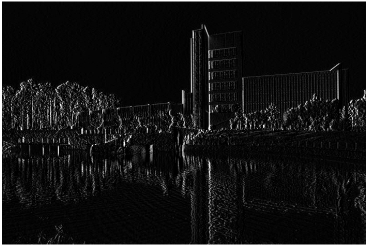
1	0	-1
1	0	-1
1	0	-1

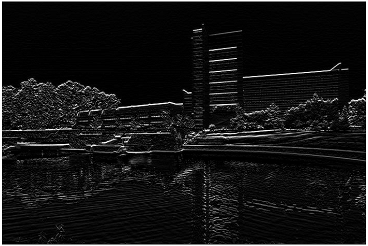
0	45	45	0
0	30	30	0
0	15	15	0
0	0	0	0

1	1	1
0	0	0
-1	-1	-1

0	0	0	0
0	-15	-30	-45
0	-15	-30	-45
0	0	0	0

Edge Detection Example





Padding

- Convolving an image with a filter may reduce the size of the output
  - Causes loss of information from the image borders
- Image is padded with a border to address this issue
  - Pixels in the padded region are typically set to 0

0	0	0	0	0	0	0	0
0	8	9	2	4	3	2	0
0	6	5	3	7	9	8	0
0	1	0	8	9	3	1	0
0	4	2	6	3	2	0	0
0	8	4	2	0	1	2	0
0	2	1	8	9	0	1	0
0	0	0	0	0	0	0	0

1	0	-1
1	0	-1
1	0	-1

-14	9	3	-7	1	12
-14	2	-6	-2	9	15
-7	-6	-12	3	10	14
-6	-3	-6	10	9	6
-7	-2	-5	13	9	3
-5	0	-4	9	6	1

Padding

- If image size is h x w, filter size is f x f, and padding size is p, then the output size is: $(h + 2p - f + 1)$ x $(w + 2p - f + 1)$
- *Valid convolution* means no padding is added
- *Same convolution* means image is padded such that output size equals image size

Stride

- Filter is moved over the image in steps equal to stride value
- Suppose stride = 2

1	·	0	·	-1	·	·
·	·	·	·	·	·	·
1	·	0	·	-1	·	·
·	·	·	·	·	·	·
1	·	0	·	-1	·	·
·	·	·	·	·	·	·
·	·	·	·	·	·	·

1	0	-1
1	0	-1
1	0	-1

Stride

- Filter is moved over the image in steps equal to stride value
- Suppose stride $= 2$

8	9	2	4	3	2	1
6	5	3	7	9	8	0
1	0	8	9	3	1	3
4	2	6	3	2	0	4
8	4	2	0	1	2	2
2	1	8	9	0	1	1
3	2	1	4	1	2	0

1	0	-1
1	0	-1
1	0	-1

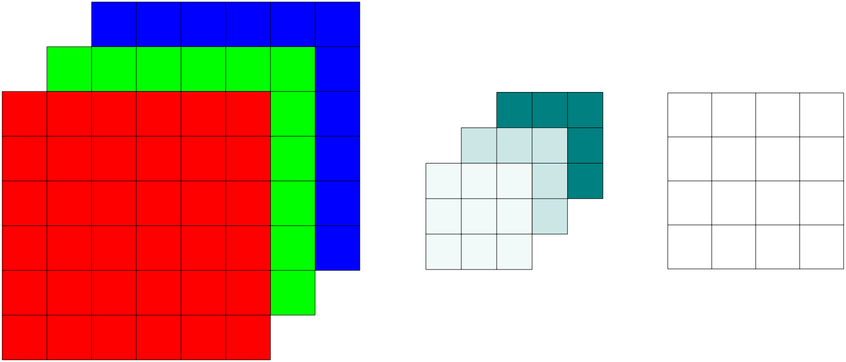
		11
-3	10	-3
2	9	-1

Output Size Calculation

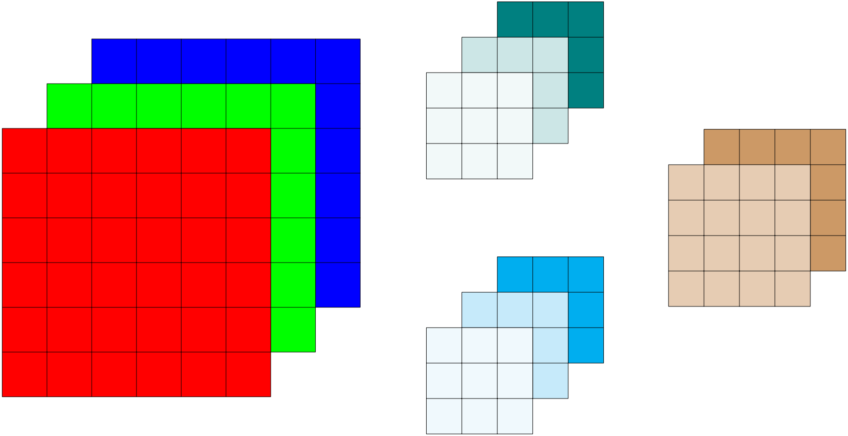
- Input size $= h \times w$
- Filter size $= f$
- Padding $= p$
- Stride $= s$
- Output size can be calculated using this formula:

$$\left\lfloor \frac{h + 2p - f}{s} + 1 \right\rfloor \text{ x } \left\lfloor \frac{w + 2p - f}{s} + 1 \right\rfloor$$

3D Convolution



3D Convolution



Convolution Layer

- Input dimensions: $h_{\ell-1} \times w_{\ell-1} \times c_{\ell-1}$
- Filter size: f_ℓ, number of filters: c_ℓ, padding: p_ℓ, stride: s_ℓ
- Output dimensions: $h_\ell \times w_\ell \times c_\ell$

$$h_\ell = \left\lfloor \frac{h_{\ell-1} + 2p_\ell - f_\ell}{s_\ell} + 1 \right\rfloor$$
$$w_\ell = \left\lfloor \frac{w_{\ell-1} + 2p_\ell - f_\ell}{s_\ell} + 1 \right\rfloor$$

- Number of parameters in one filter $= (f_\ell \times f_\ell \times c_{\ell-1}) + 1$
- Total number of parameters $= [(f_\ell \times f_\ell \times c_{\ell-1}) + 1] \times c_\ell$

Pooling Layer In Convolutional NN

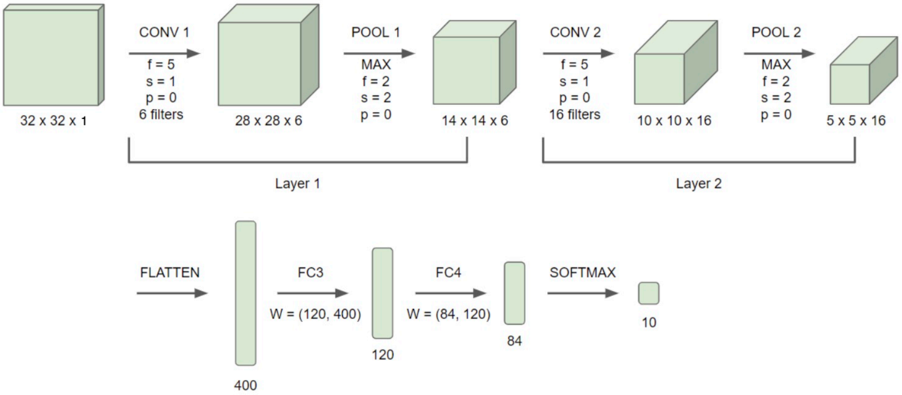
- Hyperparameters are: filter size, padding, stride
- No parameters to learn
- Two variants: max pooling and average pooling
- Example of max pooling where $f = 2$ and $s = 2$

2	4	3	5	3	2
3	5	3	7	2	1
1	0	8	9	9	1
4	2	4	8	2	0
3	4	2	0	1	2
2	1	1	2	0	1

		3
4	9	9
4	2	2

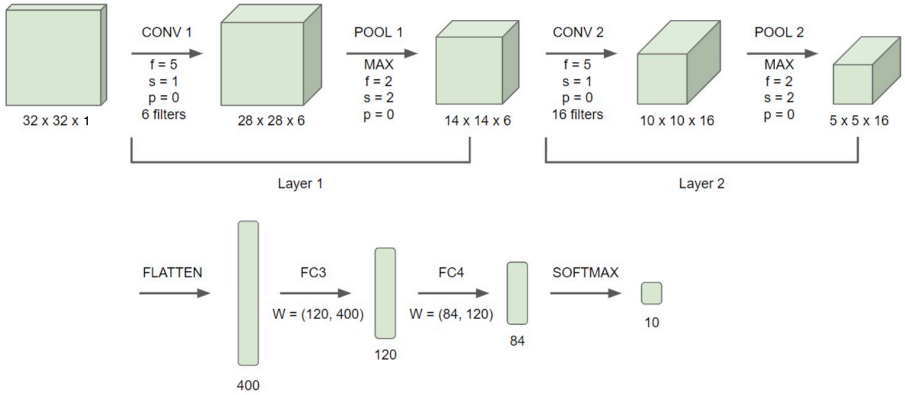
- Same formula as earlier can be used to calculate the output size

LeNet-5 CNN



CONV 1
f = 5
s = 1
p = 0
6 filters
32 x 32 x 1
28 x 28 x 6
POOL 1
MAX
f = 2
s = 2
p = 0
14 x 14 x 6
CONV 2
f = 5
s = 1
p = 0
16 filters
10 x 10 x 16
POOL 2
MAX
f = 2
s = 2
p = 0
5 x 5 x 16
Layer 1
Layer 2
FLATTEN
400
FC3
W = (120, 400)
120
FC4
W = (84, 120)
84
SOFTMAX
10

LeNet-5 CNN



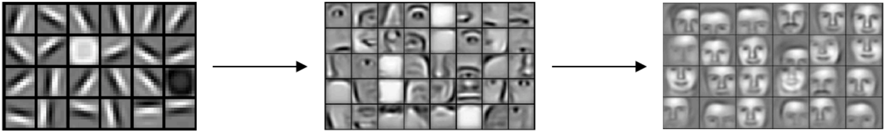


Image credit: Andrew Ng

LeNet-5 CNN Parameters

Layer	Shape	Parameters
Input	32 x 32 x 1	0
CONV1	28 x 28 x 6	$(5 * 5 * 1 + 1) * 6 = 156$
POOL1	14 x 14 x 6	0
CONV2	10 x 10 x 16	$(5 * 5 * 6 + 1) * 16 = 2416$
POOL2	5 x 5 x 16	0
FC3	120	$(400 * 120) + 120 = 48120$
FC4	84	$(120 * 84) + 84 = 10164$
Softmax	10	$(84 * 10) + 10 = 850$

Training CNN

- Can be trained using gradient descent algorithm
  - Initialize weights and baises
  - Compute activations in the forward pass
  - Compute gradient in the backward pass
  - Update weights and baises to minimize the loss

- Same loss functions we discussed earlier are used
  - Mean squared error for regression tasks
    - ★ $MSE = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$
  - Cross-entropy loss for classification tasks
    - ★ $Logloss = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} y_{ij} log(\hat{y}_{ij})$

Class Exercise

- Calculate the result of following convolution operation. Assume p = 0 and s = 1.

8	9	2	4	3	2
6	5	3	7	9	8
1	0	8	9	3	1
4	2	6	3	2	0
8	4	2	0	1	2
2	1	8	9	0	1

1	1	1
0	0	0
-1	-1	-1