

Deep Learning

Convolutional Neural Networks

- Image classification
- Object detection
- Image segmentation, etc.

Images

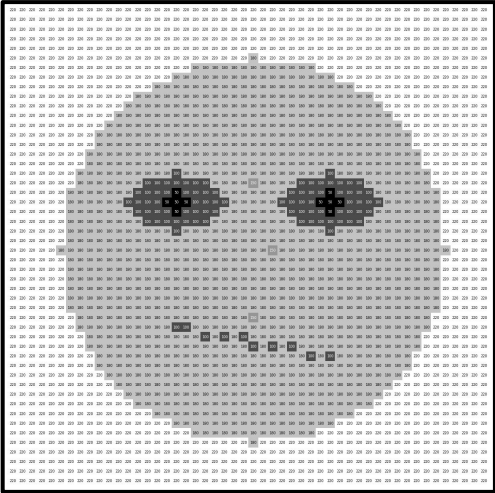
[illegible]

Matrix

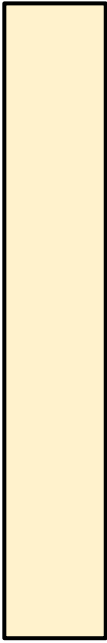
Deep Learning

Images

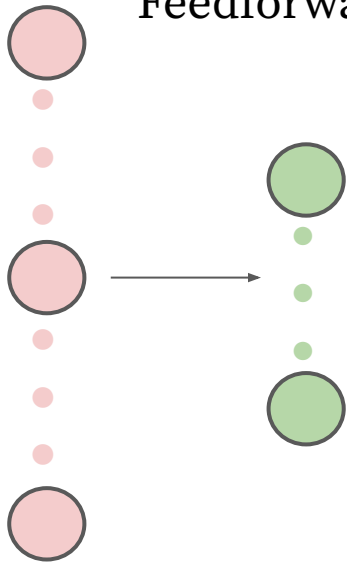
50x50



2500 x 1



Feedforward neural network

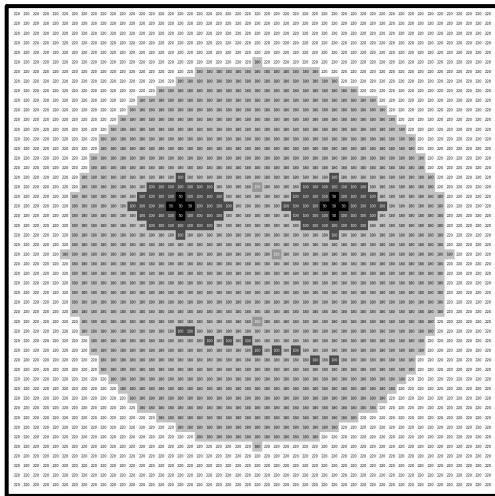


Face/not?

Deep Learning

Images

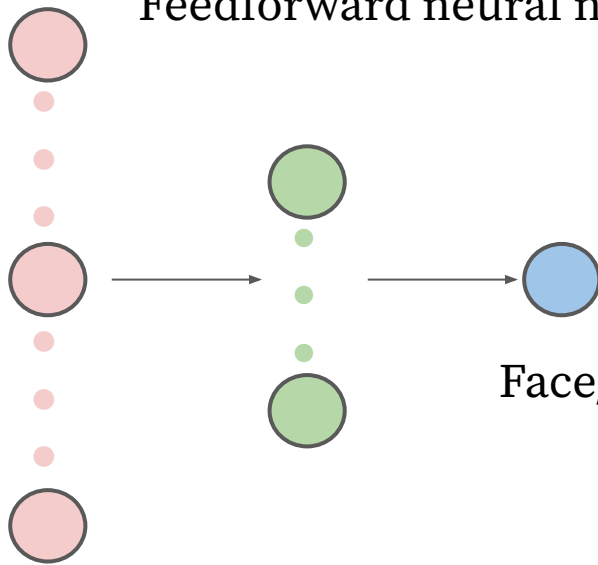
50x50



2500 x 1



Feedforward neural network



Face/not?

Issues?

- Number of parameters
- Loss of spatial information

Deep Learning

Convolutional Neural networks

- Preserve spatial relationship
- Weight sharing
- Computationally efficient

Main components of CNN

Convolutions

| | | |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

filter/kernel

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

Feature map

Filters move over the image, perform element-wise multiplication and summing results to create a feature map.

Convolutions

Stride refers to how much the kernel is moved during the convolution operation.

$$\text{Output Size} = \left(\frac{\text{Input Size} - \text{Filter Size}}{\text{Stride}} + 1 \right)$$

Padding helps preserve spatial dimension ('valid', 'same')

$$\text{Padding size} = \left\lfloor \frac{\text{Filter size} - 1}{2} \right\rfloor$$

Activation function

Pooling reduces spatial dimension (average, max pooling)

Overfitting

Model performs well on training data but poorly on test data

- Regularization
- Early stopping
- Drop out, etc.

Underfitting

Tensorflow example - CNN

Google colab —→ Classification - Wisconsin breast cancer dataset, Cifar dataset