CNN-based Face Recognition

09015128 张敏学

Problem Description

Aaron_Eckhart

MTCNN

My Work

```
cnn = CNN(img_width=img_width, input_size=img_width ** 2, output_size=num)

x_image = cnn.format_input()

conv_output1 = cnn.conv(x_image, 1, kernel_size1, feature_map1_num, 'conv1')
conv_output2 = cnn.conv(conv_output1, feature_map1_num, kernel_size2, feature_map2_num, 'conv2')
conv_output3 = cnn.conv(conv_output2, feature_map2_num, kernel_size3, feature_map3_num, 'conv3')

conv_output3_flat, flat_size = cnn.flat(conv_output3, feature_map3_num)

fc_output1 = cnn.fc(conv_output3_flat, flat_size, fully_connect_size, 'fc1')
y_conv = cnn.output_layer(fc_output1, fully_connect_size, num, 'fc2')

cnn.train()
```

Results

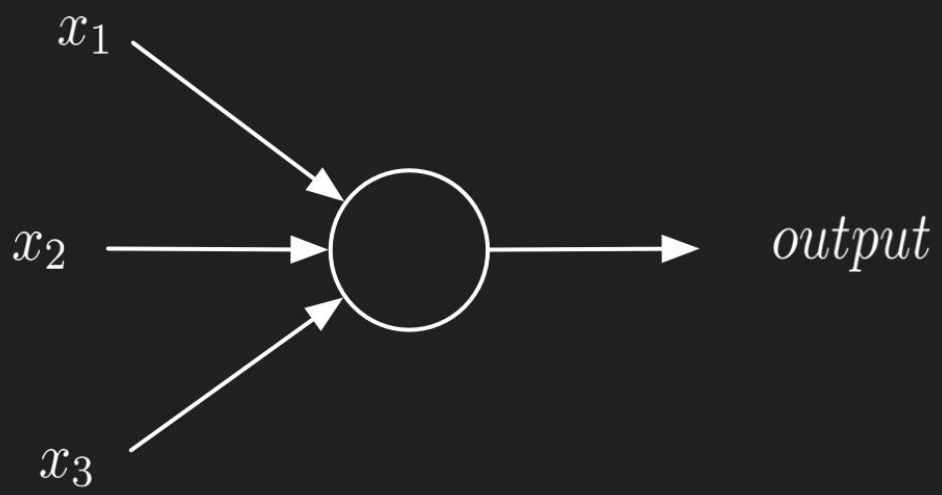
	人脸朝向	人脸表情	人脸名字	是否戴太阳镜	FaceScrub
CNN	98.2%	18.7%	98.3%	94.5%	76%~82%
ANN	92%	17%	97.5%	87%	

Procedure

1. choosing model
2. data preprocessing
3. training network
4. improve the profermance

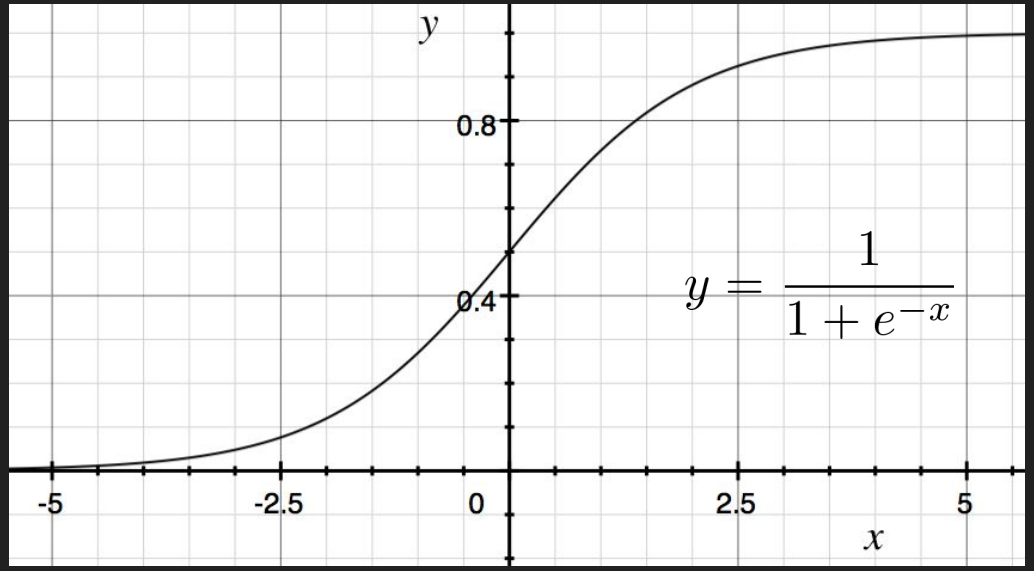
Neural Network

Perceptrons

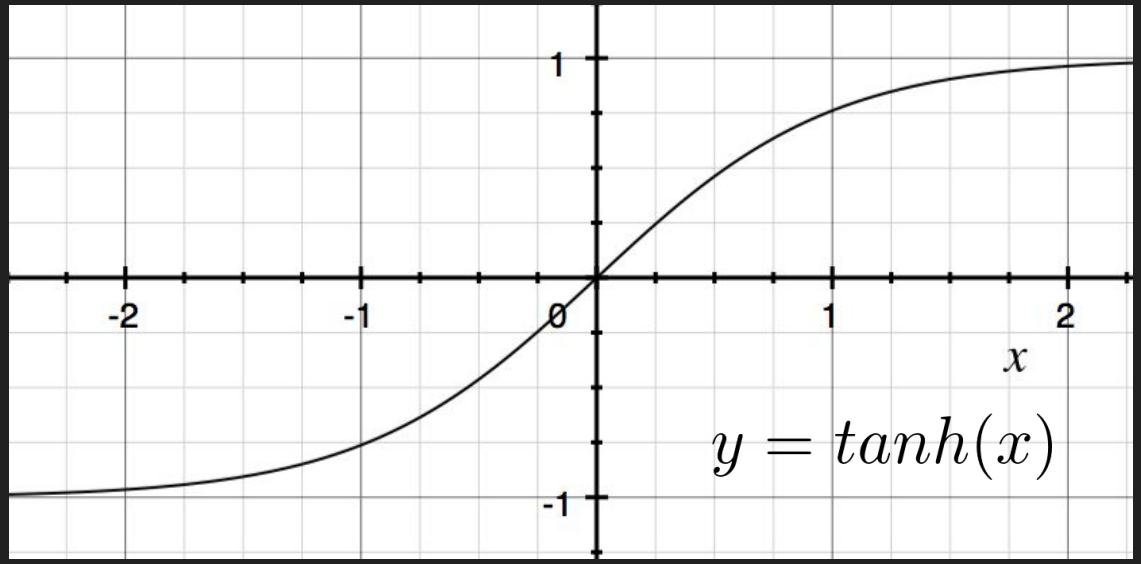


$$output = f(x_1, x_2, x_3) = f(\sum_{i=1}^{n} w_i x_i - \theta)$$

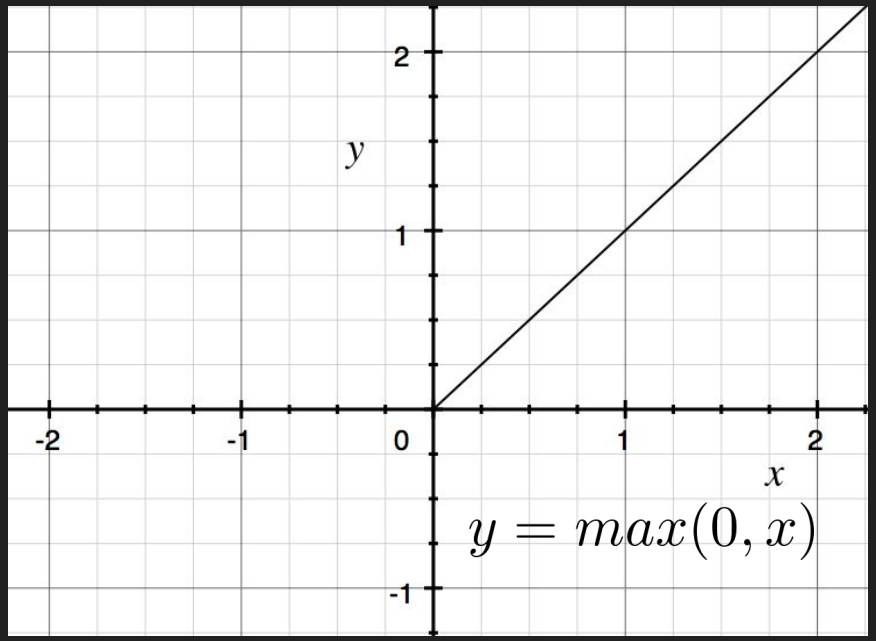
Activation Function



Activation Function

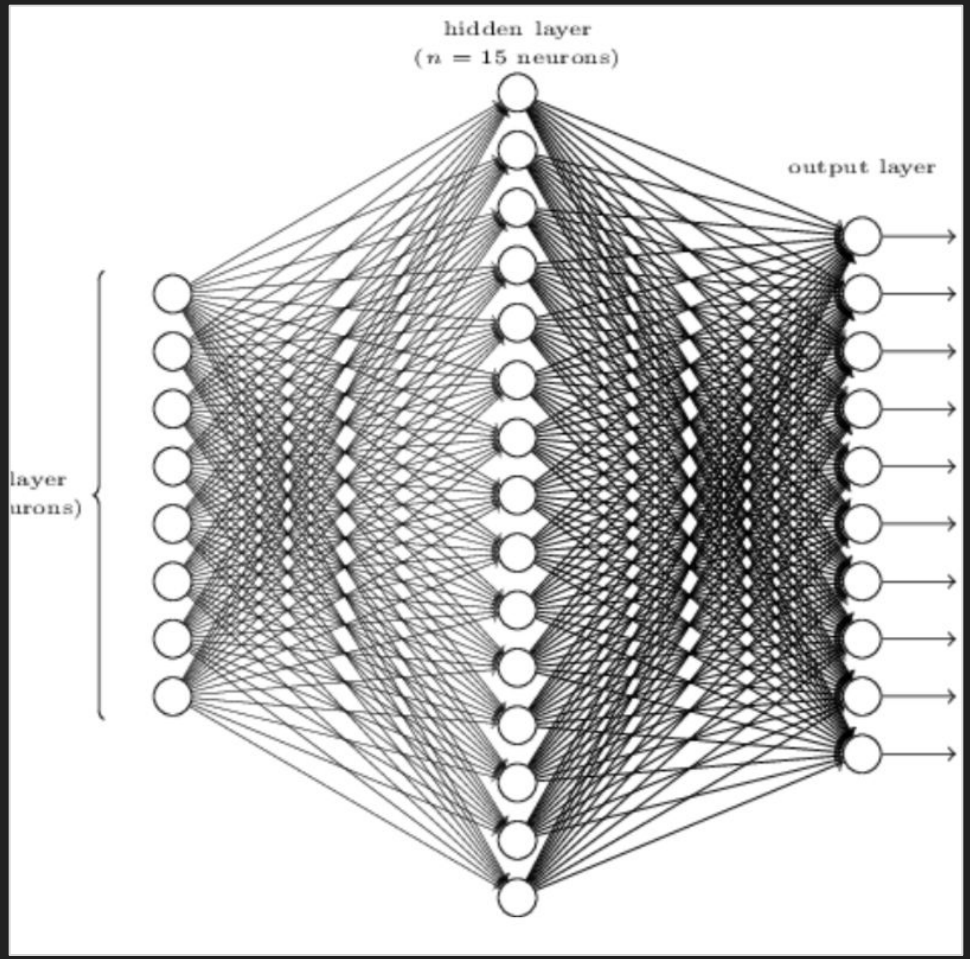


Activation Function



ANN

1. Imitating biological neuron structure
2. Fully connected

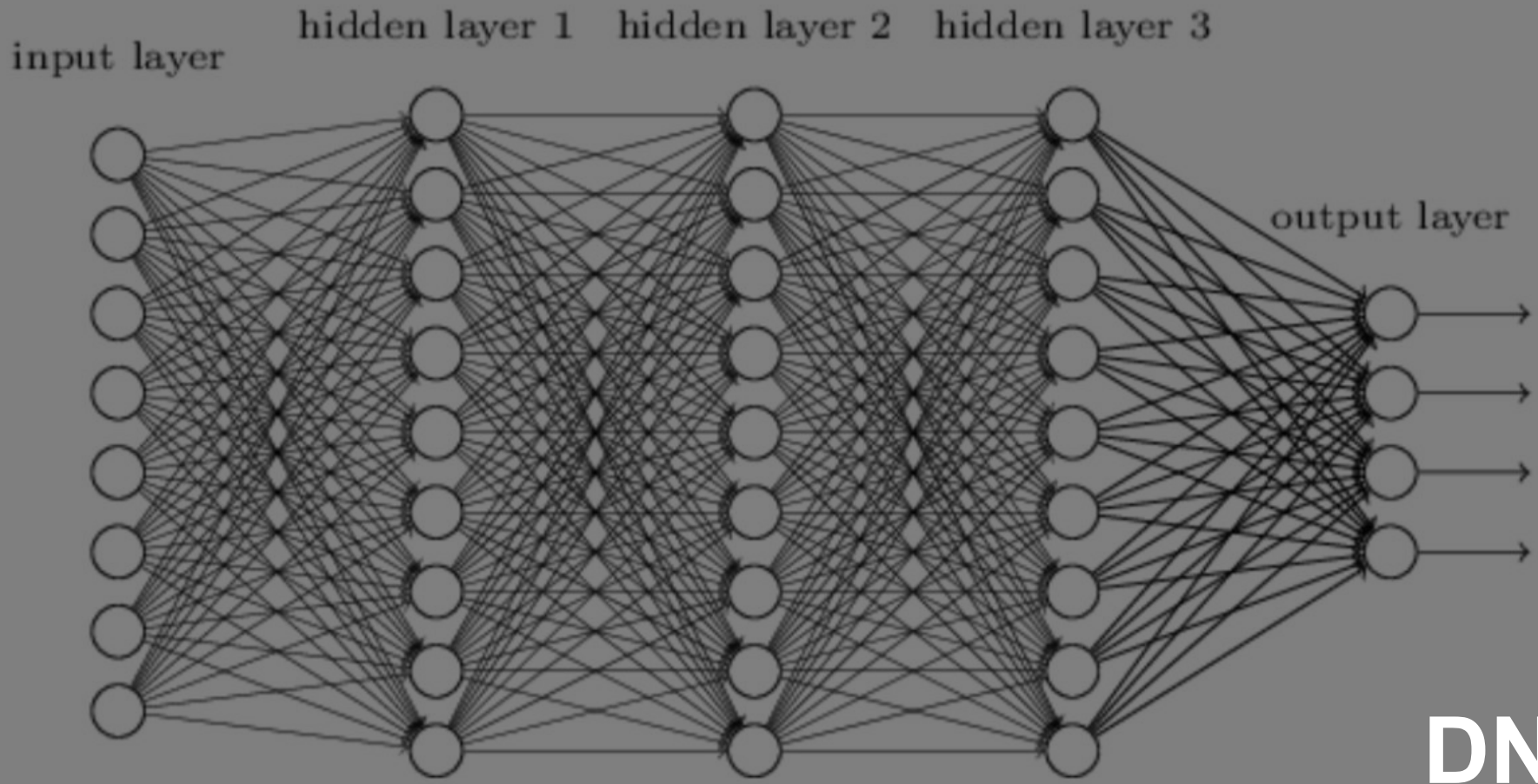


Loss Function

$$C = \frac{1}{2n} \sum_x ||y - a||^2$$

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)]$$

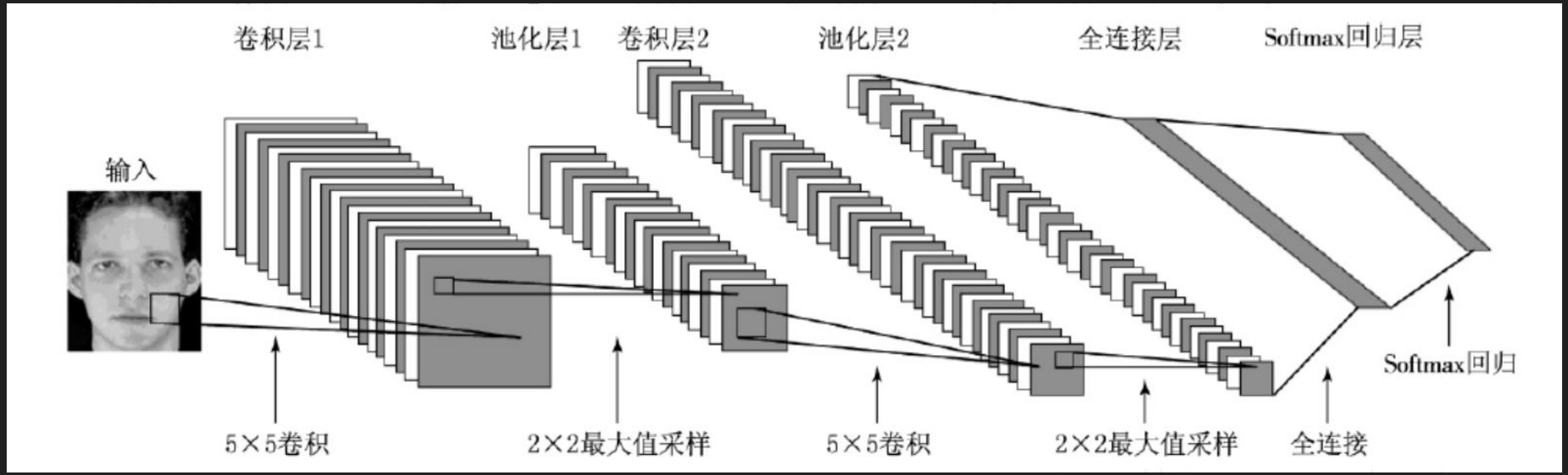
$y\ is\ the\ label\ of\ data, and\ a\ is\ the\ prediction\ of\ network$


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

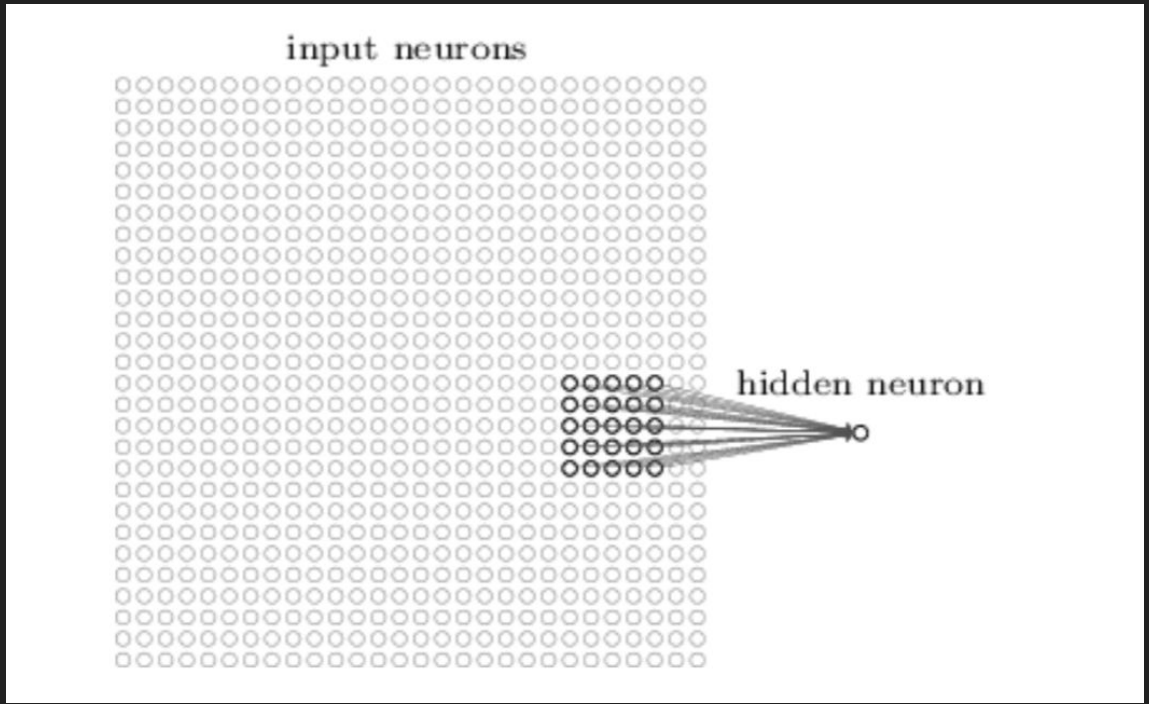
DN

Convolutional Neural Network

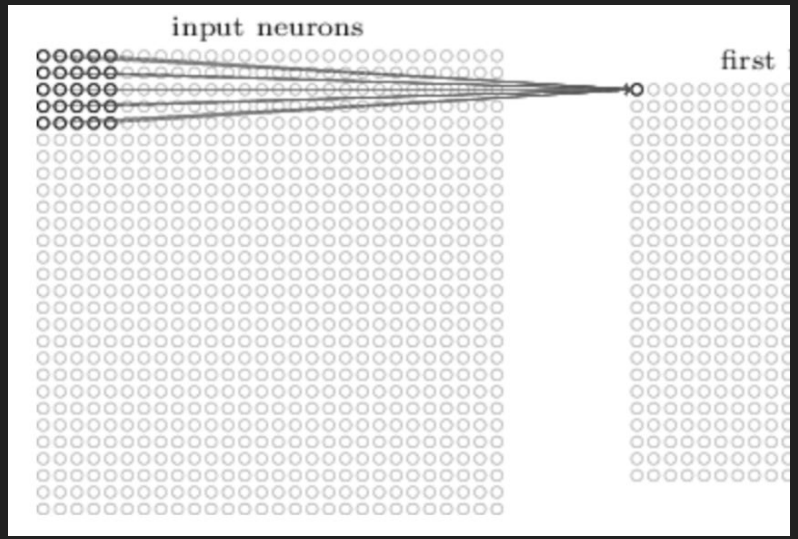
Convolutional Neural Network

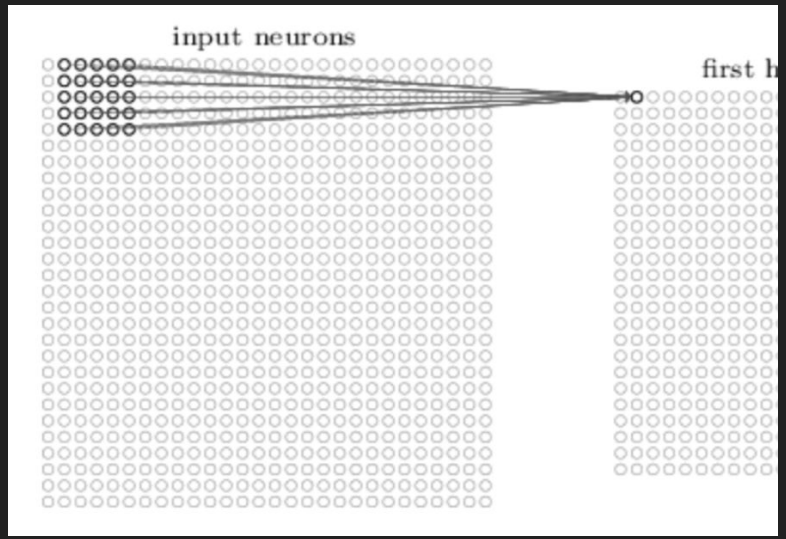


local receptive fields

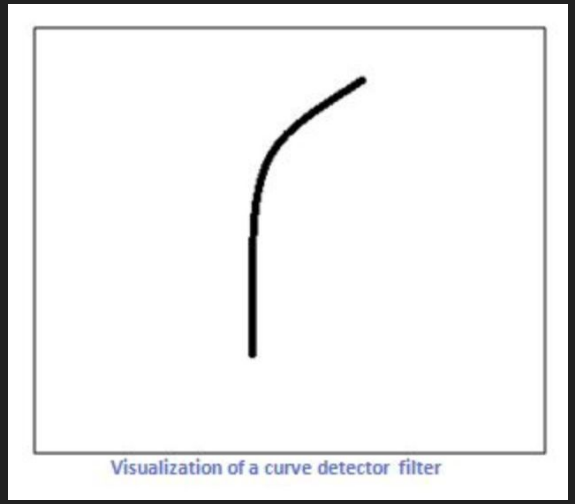


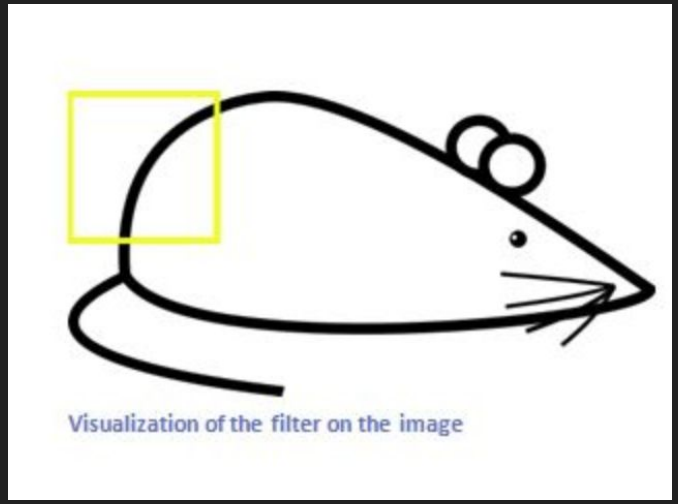
kernel



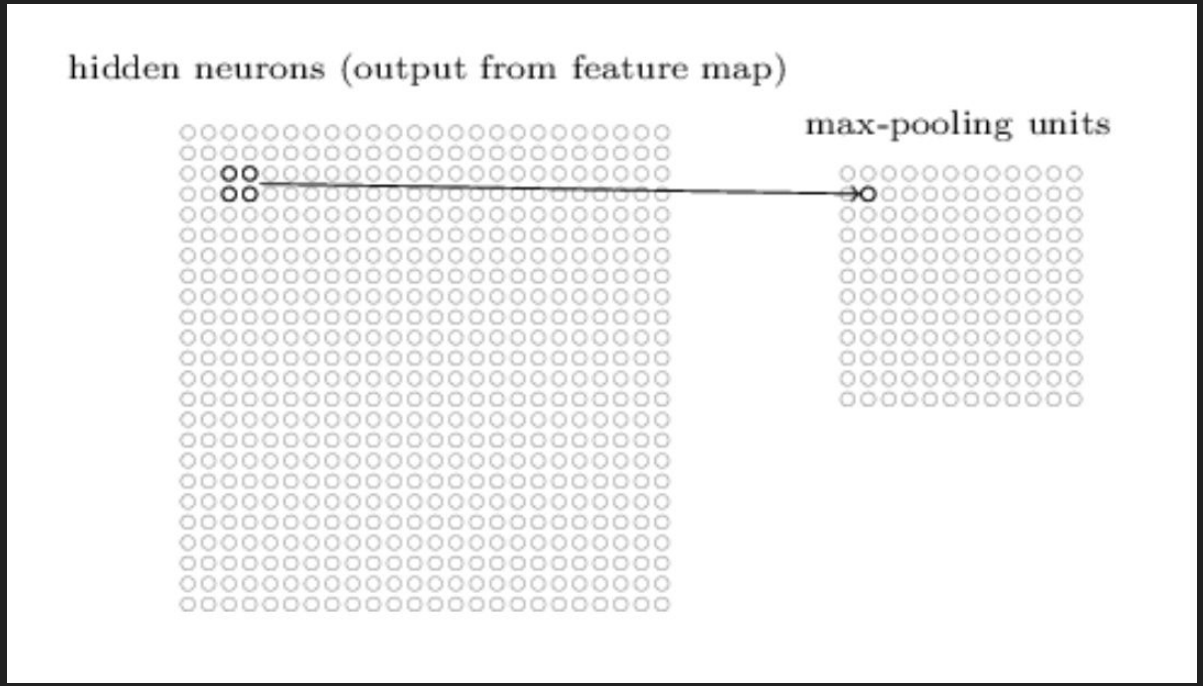


convolution

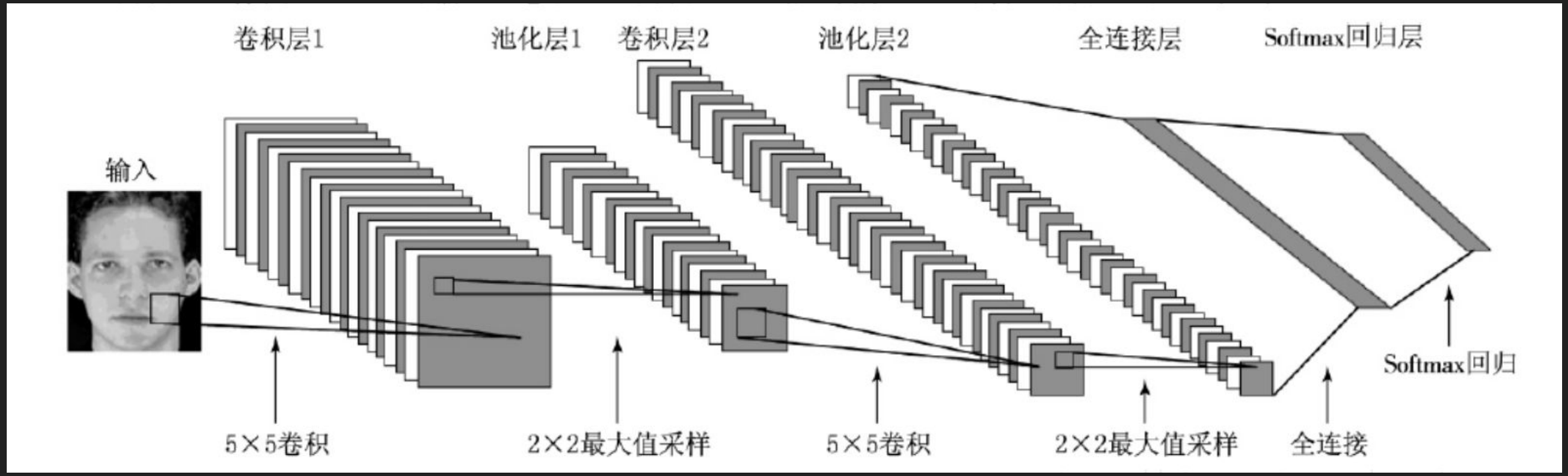


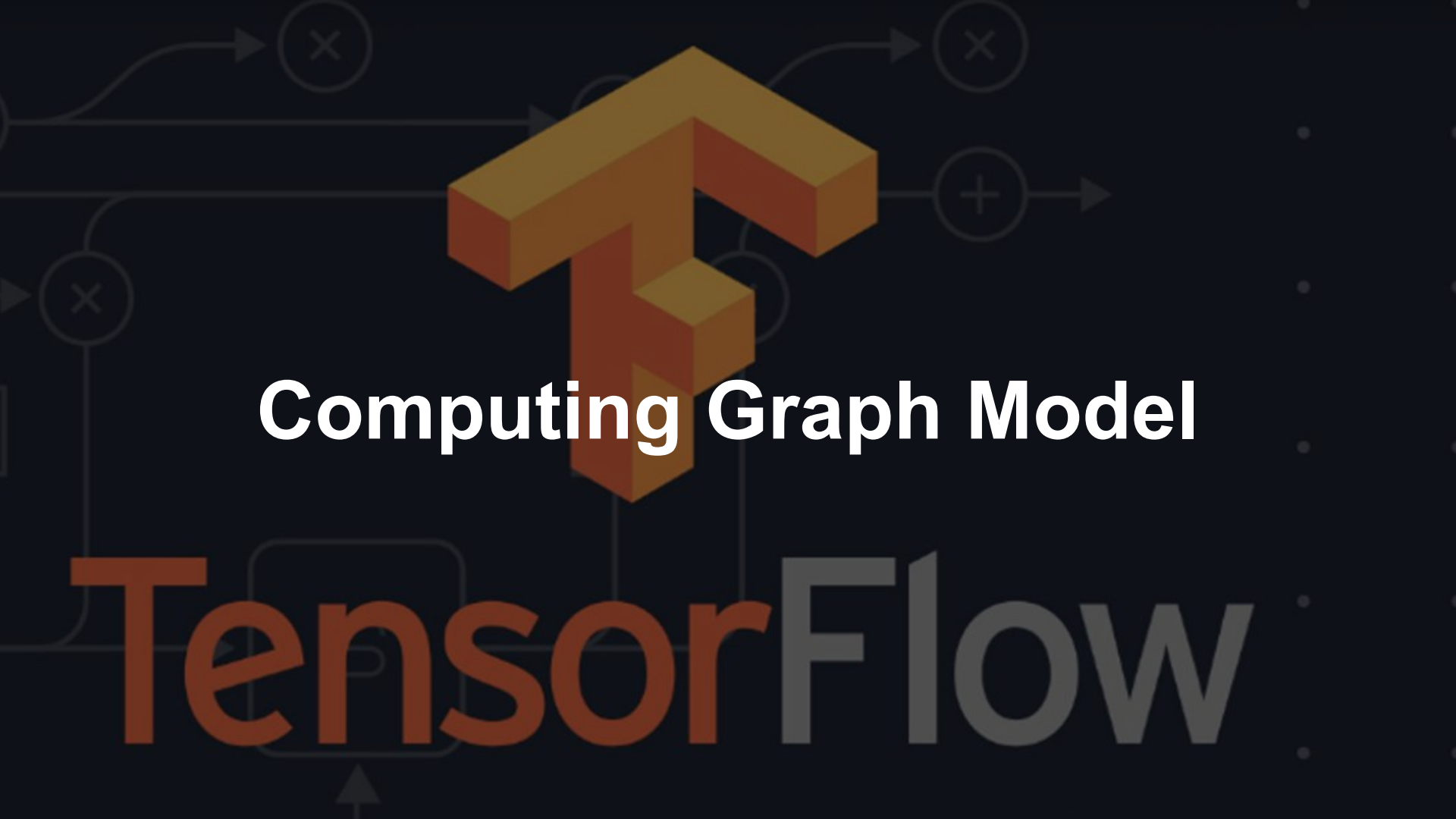


pooling



architecture




Computing Graph Model
TensorFlow

Data Preprocessing

1.

Data Preprocessing

1. Extract faces from input pictures
2. Change face images to grayscale images
3. Mapping the pixel values of the images to the [-1, 1] interval
4. Whitening(unfinished)
5. Compress the image to one dimension

Training Network

1.

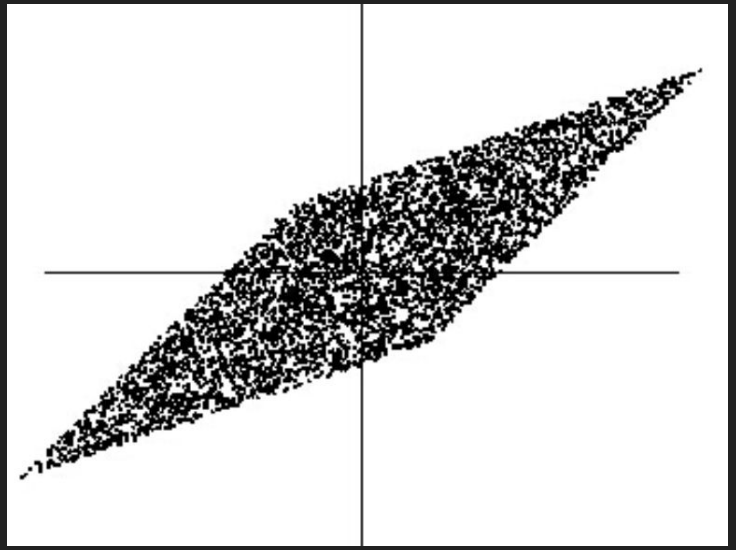
TensorBoard

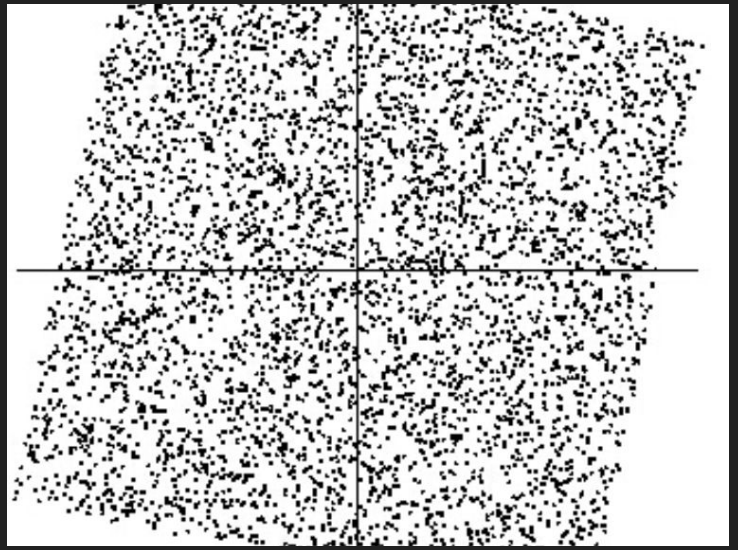
loss, accuracy

Improve the Profermance

Learning Rate

AdamOptimizer





Whitening

Batch Normalization

Whitening every layer approximately in the network

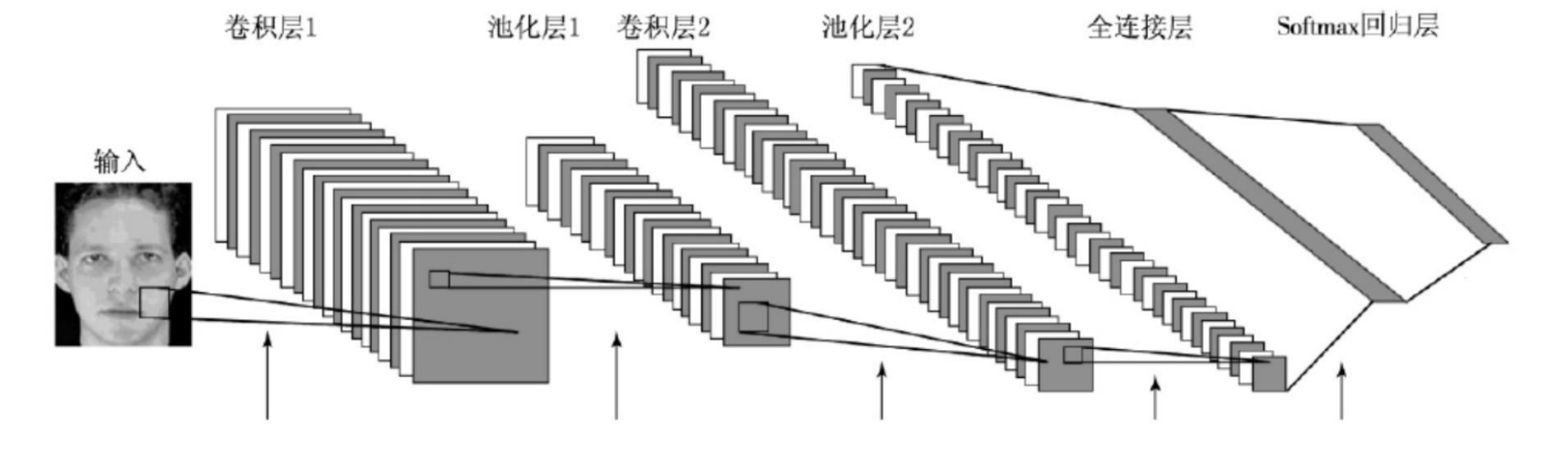
Batch Normalization

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$$



Batch Normalization

Distribution

WHY

神经网络喜欢独立同分布的数据，更喜欢白化后的数据

BN可以加快训练速度，可以防止过拟合，可以更快地收敛

Reference

Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", 2015

Thank you