THE CHINESE UNIVERSITY
OF HONG KONG (SHENZHEN)
School of Science and Engineering

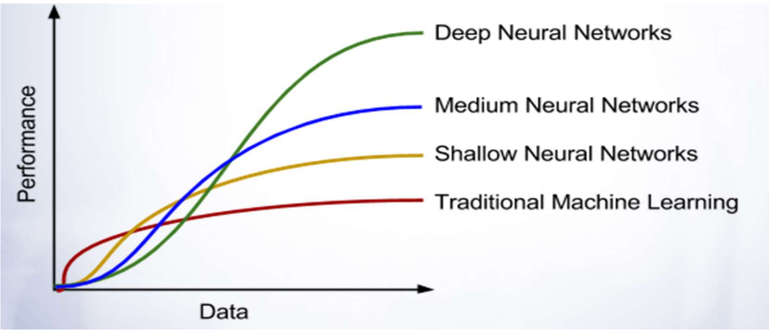
Tutorial - 12

Fundamental of Artificial Intelligence **CSC3180**

1. **Answer the following questions:**

(a) Why deep learning (DL)? What difference between ANN and DL? (L13: P4-6)

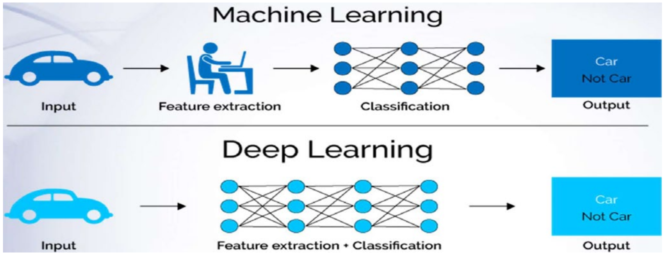
- Deep Architectures can be representationally efficient
- Deep Representations might allow for a hierarchy or representation

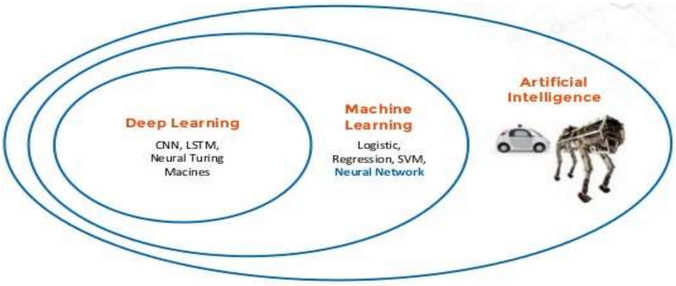


- Deep Learning = Deep Feature/Representation Learning; Learning good features automatically from raw data.
- Learning hierarchical representations with multiple levels of abstraction by using deep neural network.
- Deep Learning is a neural network with multiple hidden layers between input and output, which will allow us to compute much more complex features of the input(vs. shallow nets with just a couple of layers).

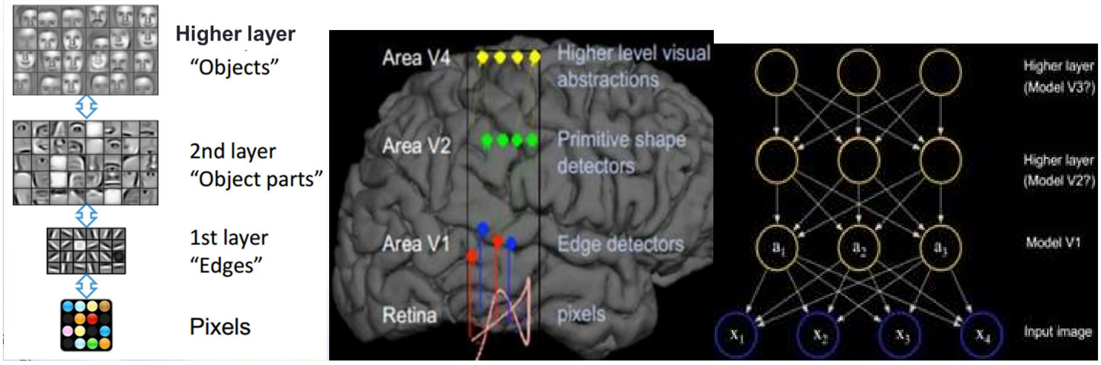
(b) Compared with Machine Learning (ML), what is the main characteristics? (L13: P7-8)

- Hand-crafted Feature/Learned Feature




Deep Learning
CNN, LSTM, Neural Turing Macines
Machine Learning
Logistic, Regression, SVM, Neural Network
Artificial Intelligence

(c) Please understand DL process and its feature analysis. (L13: P9-10)



(d) What's the key points of Convolution Networks? Please list them out and give one or two techniques for each key point. (L13: P21-23)

(1) Convolution Layer： zero padding

(2) Activation Layer： ReLU, Sigmoid, Tanh

(4)Pooling Layer： MaxPooling, AveragePooling

(5) Fully-connected Layer

(6) Regularizations: dropout

(7) Data augment: crop, mirror, rotation

(8) Data normalization:

(9) Loss Function: Softmax

(10) Training algorithm: SGD, mini batch SGD.

(e) Why the locally connected NNs are better than fully connected NNs? (L13: P24-25)

Problems of Fully Connected NNs：

- Every output unit interacts with every input unit
- The number of weights grows largely with the size of the input image. Also pixels in distance are less correlated.

Locally Connected NNs：

- Sparse connectivity: a hidden unit is only connected to a local patch (weights connected to the patch are called filter or kernel)
- It is inspired by biological systems, where a cell is sensitive to a small sub-region of the input space, called a receptive field.
- The design of such sparse connectivity is based on domain knowledge.

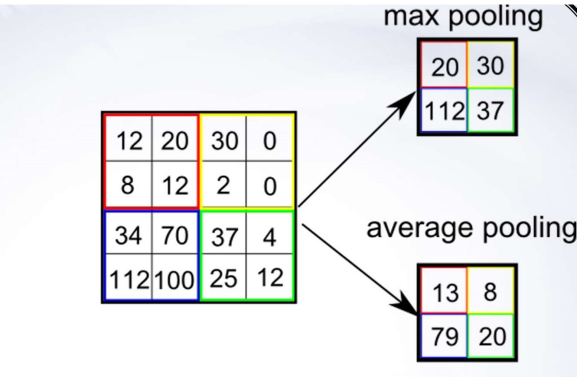
(f) Why the shared weights method is useful in ANN design? (L13: P26)

- Translation invariance: capture statistics in local patches and they are independent of locations
    - Similar edges may appear at different locations
- Hidden nodes at different locations share the same weights. It greatly reduces the number of parameters to learn

- In some applications (especially images with regular structures), we may only locally share weights or not share weights at top layers

(g) Why pooling layer? Please explain two main approaches, Max pooling and Average pooling. (L13: P30-31)

Make the representations smaller and more manageable



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

13	8
79	20

(h) What is Receptive Field? What difference between receptive field of pooling and convolution? (L13: P32-35)

Receptive Field：

- The receptive filed(感受野) is defined as the region in the input space that a particular CNN’ s feature is looking at.
- It’s the same as elements in the images used to calculate one feature in certain CNN layer.
- The feature in the last layer is better to have a large receptive filed to cover the object to be recognized.

The receptive filed of pooling layer is $S\times S=(T+(P-1)\times D)^2$. The receptive filed of the convolution layer is $(T+(K-1)\times D)^2$. P is the pooling size. T is the convolution size. T is the receptive filed of each ceil of before the pooling layer. D is the product of the downsample multiples before the pooling layer.

(i) There are three activation functions, Sigmoid, Tanh and ReLU. Why ReLU is widely adapted in DL? (L13: P37)

- The traditional activation functions suffers gradient vanishing problem.
- The gradient will reduce in backpropogation while pass through the activation layer such as Sigmoid and Tanh.
- ReLU does not suffer from the gradient vanishing problem. Widely adapted in deep networks.

(j) Please understand Loss layer: Softmax and its three optimization approaches. (L13: P41-46)

Softmax: transforms logits into probabilities.

Optimization Approaches:

- Gradient decent (GD)
- Stochastic gradient decent (SGD)
- Mini batch gradient decent

Now, if given a dataset with two classes of image. We want to take use of deep learning to solve the classification problem. Please answer the following Question 2 to 5.

2. **Currently, we have only 2,000 images for training. It's not enough to train a convolution network. Thus, enlarging the dataset is needed. Let's set the image size is 64×64. In order to keep most part of the image, the minimum input image size of the network is 56×56. Please use crop and mirror to do data argument.**

   (1) Describe the crop and mirror method.

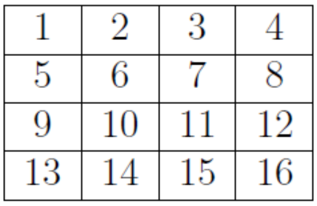
   Crop: Randomly extract a sub-image from the whole image and use it as a training sample.

   Mirror: Flip the image in horizontal direction.

   (2) Calculate the total number of training samples after argument.

   For cropping, we can extract (64-56+1)×(64-56+1)=9×9 =81 patches from one single image. For each patch, flip the image in horizontal direction can double the patches. Thus, there are 81×2=162 patches for one single images. After augment, there with be 162×2,000=324,000 training samples.

   (3) Given a small image with the size of 4×4, please first crop the image to 3×3 patches and then mirror the cropped patches.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3
5	6	7
9	10	11

2	3	4
6	7	8
10	11	12

5	6	7
9	10	11
13	14	15

6	7	8
10	11	12
14	15	16

3	2	1
7	6	5
11	10	9

4	3	2
8	7	6
12	11	10

7	6	5
11	10	9
15	14	13

8	7	6
12	11	10
16	15	14

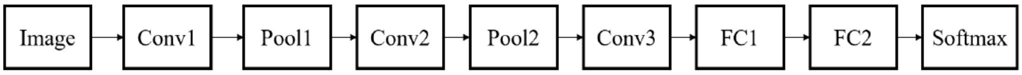


Figure 1: Structure of the Convolution Network.

The input image size for classification is 56×56 after enlargement. Now, we have designed a convolution network for this classification problem in Figure 1. Conv1 do 3×3 convolution with 64 filters without padding. Pool1 do 3×3 max pooling. Conv2 do 7×7 convolution with 128 filters without padding. Pool2 do 2×2 max pooling. Conv3 do 5×5 convolution with 128 filters and zero-padding, pad 1. FC1 has 512 neurons and FC2 has 2 neurons. Conv1, Conv2, Conv3 and FC1 is followed by a ReLU activation.

3. Now calculate the following problem.

(1) Calculate the size of the output of each layer (height×width×channel).

Conv1: 54×54×64 (hint 56-3+1=54)

Pool1: 18×18×64 (hint 54/3=18)

Conv2: 12×12×128 (hint 18-7+1=12)

Pool2: 6×6×128

Conv3: 4×4×128 (hint 6+2-5+1=4)

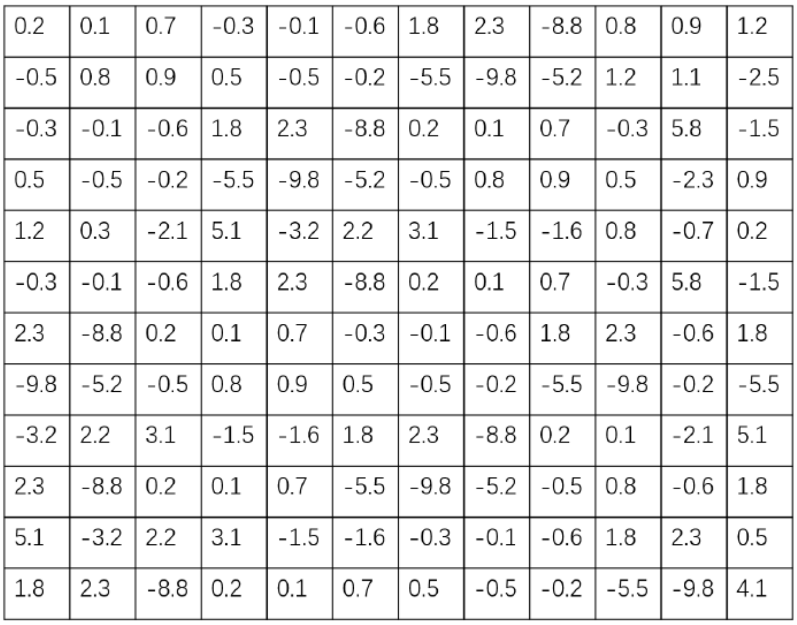
FC1: 512

FC2: 2

(2) Calculate the number of parameters of Conv2.

128*64*7*7+128=401536

(3) Figure 2 gives the output of Conv2 before ReLU. Please calculate the results after ReLU and Pool2.



0.2	0.1	0.7	-0.3	-0.1	-0.6	1.8	2.3	-8.8	0.8	0.9	1.2
-0.5	0.8	0.9	0.5	-0.5	-0.2	-5.5	-9.8	-5.2	1.2	1.1	-2.5
-0.3	-0.1	-0.6	1.8	2.3	-8.8	0.2	0.1	0.7	-0.3	5.8	-1.5
0.5	-0.5	-0.2	-5.5	-9.8	-5.2	-0.5	0.8	0.9	0.5	-2.3	0.9
1.2	0.3	-2.1	5.1	-3.2	2.2	3.1	-1.5	-1.6	0.8	-0.7	0.2
-0.3	-0.1	-0.6	1.8	2.3	-8.8	0.2	0.1	0.7	-0.3	5.8	-1.5
2.3	-8.8	0.2	0.1	0.7	-0.3	-0.1	-0.6	1.8	2.3	-0.6	1.8
-9.8	-5.2	-0.5	0.8	0.9	0.5	-0.5	-0.2	-5.5	-9.8	-0.2	-5.5
-3.2	2.2	3.1	-1.5	-1.6	1.8	2.3	-8.8	0.2	0.1	-2.1	5.1
2.3	-8.8	0.2	0.1	0.7	-5.5	-9.8	-5.2	-0.5	0.8	-0.6	1.8
5.1	-3.2	2.2	3.1	-1.5	-1.6	-0.3	-0.1	-0.6	1.8	2.3	0.5
1.8	2.3	-8.8	0.2	0.1	0.7	0.5	-0.5	-0.2	-5.5	-9.8	4.1

Figure 2: Figure for problem 3.3.

Result after ReLU:

0.2	0.1	0.7	0	0	0	1.8	2.3	0	0.8	0.9	1.2
0	0.8	0.9	0.5	0	0	0	0	0	1.2	1.1	0
0	0	0	1.8	2.3	0	0.2	0.1	0.7	0	5.8	0
0.5	0	0	0	0	0	0	0.8	0.9	0.5	0	0.9
1.2	0.3	0	5.1	0	2.2	0	0	0	0.8	0	0.2
0	0	0	1.8	2.3	0	0.2	0.1	0.7	0	5.8	0
2.3	0	0.2	0.1	0.7	0	0	0	1.8	2.3	0	1.8
0	0	0	0.8	0.9	0.5	0	0	0	0	0	0
0	2.2	3.1	0	0	1.8	2.3	0	0.2	0.1	0	5.1
2.3	0	0.2	0.1	0.7	0	0	0	0	0.8	0	1.8
5.1	0	2.2	3.1	0	0	0	0	0	1.8	2.3	0.5
1.8	2.3	0	0.2	0.1	0.7	0.5	0	0	0	0	4.1

Results after Pool2:

0.8	0.9	0	2.3	1.2	1.2
0.5	1.8	2.3	0.8	0.9	5.8
1.2	5.1	2.3	0.2	0.8	5.8
2.3	0.8	0.9	0	2.3	1.8
2.3	3.1	1.8	2.3	0.8	5.1
5.1	3.1	0.7	0.5	1.8	4.1

4. **Calculate the receptive filed of Conv1, Pool1, Conv2, Pool2 and Conv3.**

   Conv1: 3×3

   Pool1: (3+3-1)×(3+3-1)=5×5

   Conv2: (5+(7-1)*3)×(5+(7-1)*3)=23×23

   Pool2: (23+(2-1)*3)×(23+(2-1)*3)=26×26

   Conv3: (26+(5-1)*6)×(26+(5-1)*6)=50×50

5. **Describe the steps of designing a Convolution Network to solve a classification problem.**

   Hand Written Example： L13: Page49-53.