Lecture 11-Dimension reduction

Outline of lecture

1. Dimension reduction
2. Neural networks

---

1. Dimension reduction

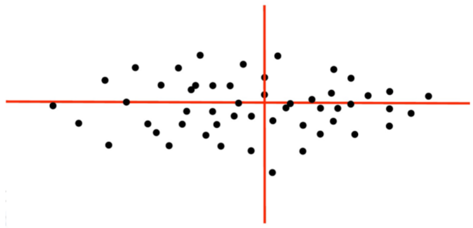
1.1 Feature selection

- Choose the **best subset** genes from all the genes
- Feature ranking
- Feature subset selection: Filter and Wrapper

1.2 Feature extraction

- Extract new features by linear or non-linear combination of the original features
    - New feature = Gene 1 + Gene 2
- New features may not have physical interpretation/meaning (usually for non-linear)
- PCA, SVD, Isomap, LLE, CCA, et. al.

1.3 Principal components analysis (PCA)



1st dimension captures max variance
2nd dimension captures the max amount of residual variance, at right angles (orthogonal) to the first

1.3.1 PCA steps

1. We first normalize each feature to make the average of each feature 0. Then, we get X'

X

X1	1	1	1
X2	2	2	2
X3	3	3	3
Average	2	2	2

X1	-1 (1−2)	-1	-1
X2	0 (2−2)	0	0
X3	1 (3−2)	1	1

X'

2. calculate the covariance matrix of X'

$$\Sigma = \frac{1}{n-1} X'^T X', \Sigma\text{: a } d \text{ by } d \text{ matrix}$$

$$\Sigma = \frac{1}{n-1} X'^T X' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

($n = 3$)

$$\frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$(-1)(-1)+0+1=2.$

3. Find the eigenvectors and eigenvalues of Σ

$$\Sigma * V = \lambda * V \qquad \lambda_1 = 3 \text{ or } \lambda_2 = 0$$

$$(\Sigma - \lambda I) * V = 0$$

$$\begin{bmatrix} 1-\lambda & 1 & 1 \\ 1 & 1-\lambda & 1 \\ 1 & 1 & 1-\lambda \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0$$

$$\begin{cases} -2V_1 + V_2 + V_3 = 0 \\ V_1 - 2V_2 + V_3 = 0 \\ V_1 + V_2 - 2V_3 = 0 \end{cases} \Rightarrow V_1 = V_2 = V_3$$

$|V_1| = 1$

$$\lambda_1 = 3 \qquad V_1 = \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}$$

Yu Li

4. M eigenvectors with the M largest eigenvalues

$$\lambda_1 = 3 \qquad V_1 = \begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix} \qquad \lambda_{2,3} = 0 \qquad V_{2,3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

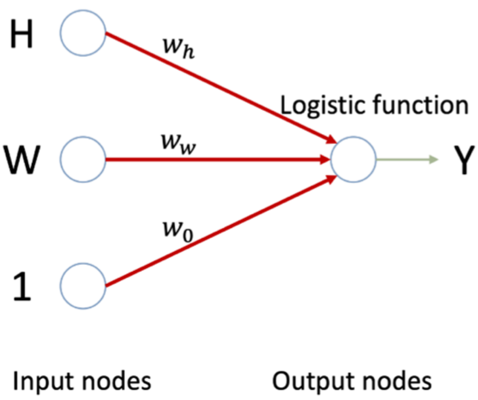
2D $$P = \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \end{bmatrix}$$

5. Project the data to the M eigenvectors' direction

$$\hat{X} = X'P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \\ \frac{\sqrt{3}}{3} & 0 \end{bmatrix} = \begin{bmatrix} -\sqrt{3} & 0 \\ 0 & 0 \\ \sqrt{3} & 0 \end{bmatrix}$$

2. Neural networks

2.1 Logistic regression



$$Y^{output} = \frac{1}{1 + e^{-(w_h H + w_w W + w_0)}}$$

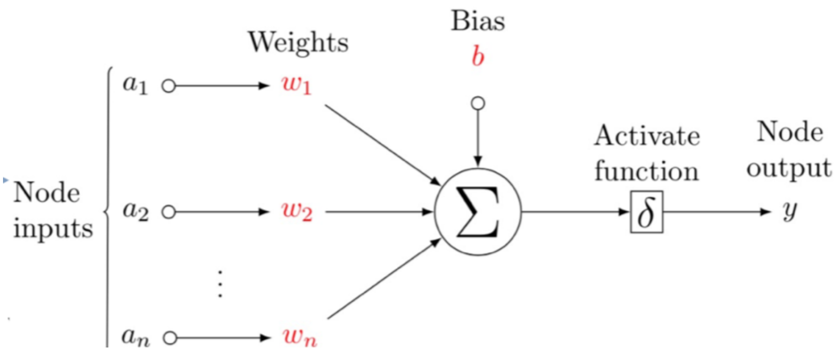
The problem of logistic regression
1. The relation between the output and input may be nonlinear
2. The relation between the output and input can be very complex

2.2 From LR to deep neural networks

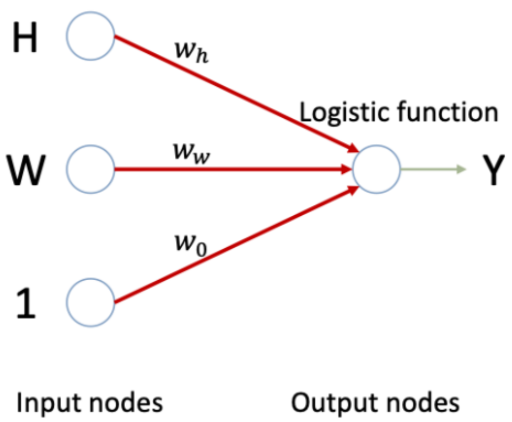
1. To resolve complicated problems
    - Increase the number of nodes
    - Increase the number of layers
    - Add non-linear function

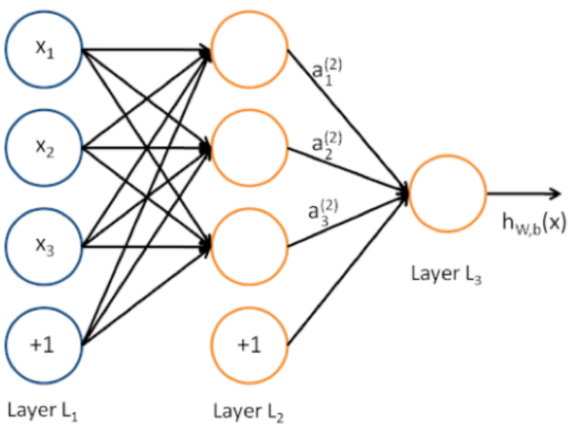
2. Fully-connected layers
    - A general function approximator
    - We can approximate any function (relation) if we have enough nodes and layers
    - Universal approximation theorem
    -

2.3 Internal nodes



2.4 The number of parameters





Parameter=3

Parameters:4x3+4=16

2.5 Deep neural networks

- The function is much more complicated
- The number of parameters is very large
- We may use it resolve complex problems with a huge amount of data