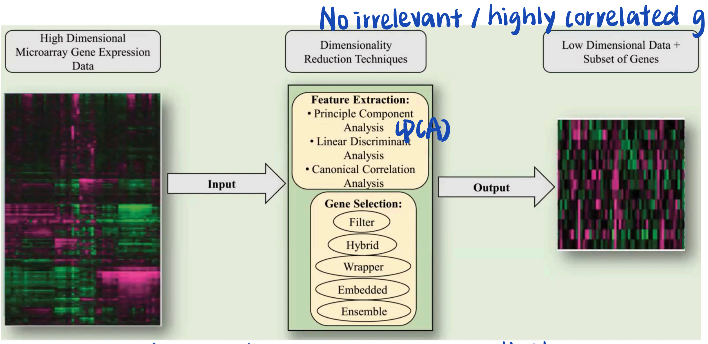
Course Code : BMEG 3105
Course Title : Data Analytics for Personalized Genomics and Precision medicine
Lecture Topic: NN
Scribed by Hu Jingjie (1155157229@link.cuhk.edu.hk)

Contents


## Feature selection and dimension reduction

No irrelevant / highly correlated genes




choose the best subset genes from all the genes

## Principal components analysis

1st capture max variance

2nd capture max amount of residual variance, at orthogonal to the first

❖Filter
- ➢Classification performance is not involved in the selection loop
- ➢Variance thresholds: Features with a higher variance contain more useful information
  - Age, Height
- ➢Information gain: Features should be different

❖Wrapper
- ➢Using the classification performance to guide selection
- ➢Computational expensive
- ➢Recursive feature elimination
- ➢Sequential feature selection

| Person | Height(m) | Weight(kg) |
|---|---|---|
| P1 | 1.79 | 75 |
| P2 | 1.64 | 54 |
| P3 | 1.70 | 63 |
| P4 | 1.88 | 78 |
| P5 | 1.75 | 70 |

❖Suppose we have a $n$ by $d$ data matrix, $X$. We first normalize each feature to make the average of each feature 0. Then, we get $X'$

❖Then, we calculate the covariance matrix of $X'$
- ➢$\Sigma = \frac{1}{n-1} X'^T X'$, $\Sigma$: a $d$ by $d$ matrix

❖Find the eigenvectors and eigenvalues of $\Sigma$

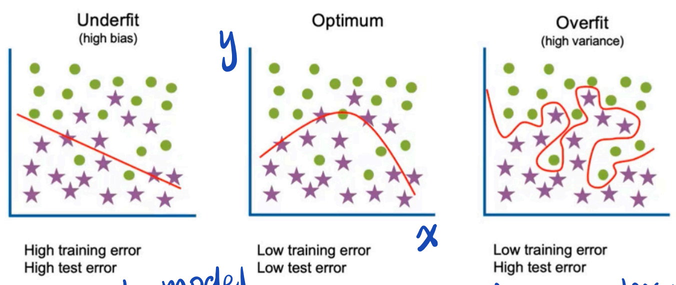
❖M eigenvectors with the M largest eigenvalues
- ➢Principal components

❖Project the data to the M eigenvectors' direction
- ➢$\hat{X} = X'P$

| Person | Height (m) | Weight (kg) | Age |
|---|---|---|---|
| P1 | 1.79 | 75 | 20 |
| P2 | 1.64 | 54 | 20 |
| P3 | 1.70 | 63 | 20 |
| P4 | 1.88 | 78 | 20 |
| P5 | 1.75 | 70 | 20 |

## Neural networks

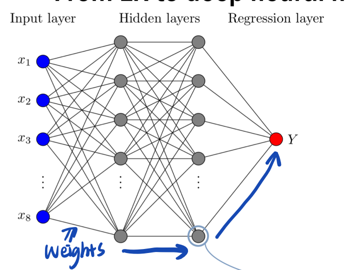
the relationship between different variables is much more complicated than simple linear combination



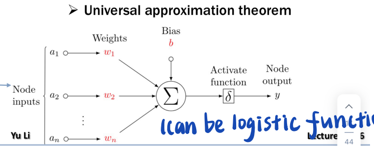

too simple model with too difficult problems

too complex model

## From LR to deep neural networks




- To resolve complicated problems
  - Increase the number of nodes
  - Increase the number of layers
  - Add non-linear function
- Fully-connected layers
  - A general function approximator
  - We can approximate any function (relation) if we have enough nodes and layers
  - Universal approximation theorem




(can be logistic function)

NN Yu Li Lecture 5

**Universal approximation theorem:**[26][27] Let $[a, b]$ be a finite segment of the real line, $s = b - a$ and $\lambda$ be any positive number. Then one can algorithmically construct a computable sigmoidal activation function $\sigma\colon \mathbb{R} \to \mathbb{R}$, which is infinitely differentiable, strictly increasing on $(-\infty, s)$, $\lambda$ -strictly increasing on $[s, +\infty)$, and satisfies the following properties:

1) For any $f \in C[a, b]$ and $\varepsilon > 0$ there exist numbers $c_1, c_2, \theta_1$ and $\theta_2$ such that for all $x \in [a, b]$

$$|f(x) - c_1\sigma(x - \theta_1) - c_2\sigma(x - \theta_2)| < \varepsilon$$

2) For any continuous function $F$ on the $d$-dimensional box $[a, b]^d$ and $\varepsilon > 0$, there exist constants $e_p$, $c_{pq}$, $\theta_{pq}$ and $\zeta_p$ such that the inequality

$$\left| F(\mathbf{x}) - \sum_{p=1}^{2d+2} e_p \sigma\left( \sum_{q=1}^{d} c_{pq}\sigma(\mathbf{w}^q \cdot \mathbf{x} - \theta_{pq}) - \zeta_p \right) \right| < \varepsilon$$

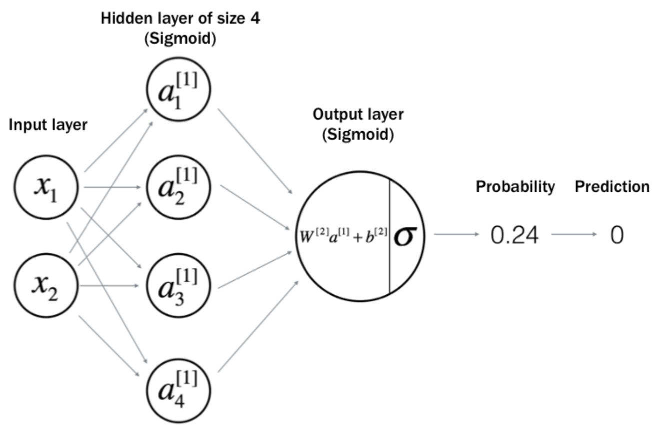
holds for all $\mathbf{x} = (x_1, \ldots, x_d) \in [a, b]^d$. Here the weights $\mathbf{w}^q$, $q = 1, \ldots, d$, are fixed as follows:

$$\mathbf{w}^1 = (1, 0, \ldots, 0), \quad \mathbf{w}^2 = (0, 1, \ldots, 0), \quad \ldots, \quad \mathbf{w}^d = (0, 0, \ldots, 1).$$

In addition, all the coefficients $e_p$, except one, are equal.

Here "$\sigma\colon \mathbb{R} \to \mathbb{R}$ is $\lambda$-strictly increasing on some set $X$" means that there exists a strictly increasing function $u\colon X \to \mathbb{R}$ such that $|\sigma(x) - u(x)| \le \lambda$ for all $x \in X$. Clearly, a $\lambda$-increasing function behaves like a usual increasing function as $\lambda$ gets small. In the "*depth-width*" terminology, the above theorem says that for certain activation functions depth-$2$ width-$2$ networks are universal approximators for univariate functions and depth-$3$ width-$(2d + 2)$ networks are universal approximators for $d$-variable functions ($d > 1$).

image ⟶ label
why is this a dog (function)
(combination of lot of logistic regression)




the weight between $a_1, a_2, a_3, a_4$ and $x, y$ is different

$\Rightarrow$ increase complexity of function

C. Add two bias nodes in the input layer X

D. Add an additional feature in the input layer

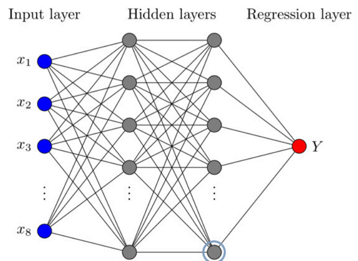
E. Change the linear function to non-linear activation function

linear $\Rightarrow$ just one layers

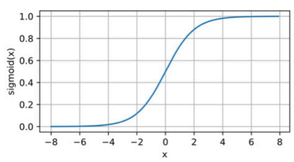
## Activation functions

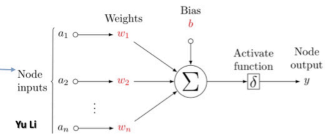
usefull!

### Different activation functions-Sigmoid




$$Sigmoid(x) = \frac{1}{1+\exp(-x)}$$

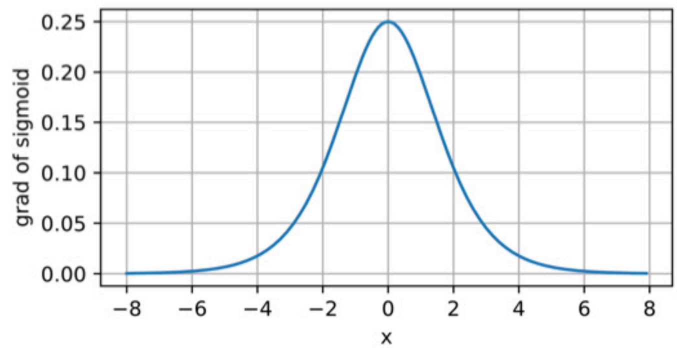


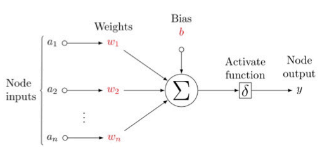

IN Yu Li Lec

$$Sigmoid(x) = \frac{1}{1+\exp(-x)}$$

$$\frac{d}{dx}\text{sigmoid}(x) = \frac{\exp(-x)}{(1+\exp(-x))^2}$$
$$= sigmoid(x)(1 - sigmoid(x))$$




### Different activation functions-ReLU

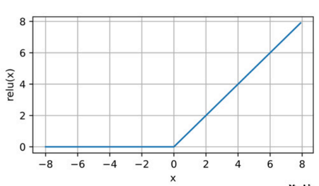


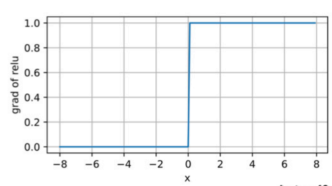

***rectified linear unit (ReLU)***

The most commonly used one deep learning field!

$$ReLU(x) = \max(x, 0)$$

$$pReLU(x) = \max(0, x) + \alpha\min(0, x)$$




one-hot encoding

| | | | |
|---|---|---|---|
| dog | 1 | 0 | 0 ← (0.8, 0.1, 0.1) |
| cat | 0 | 1 | 0 |
| bird | 0 | 0 | 1 |

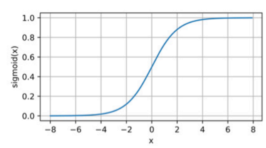
cross-entropy loss

$CE(Y, d) = -\sum_i^K d_i \log y_i$

Sklearn

### Different activation functions-Softmax

❖Sigmoid: regression or binary classification




$$Sigmoid(x) = \frac{1}{1+\exp(-x)}$$

❖Softmax: multi-class classification

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}}$$

input