

DATA MINING

LECTURE 6

Linear Regression
Logistic Regression
Neural Networks
A Little Bit of Deep Learning

CSC177
Dr. Victor Chen

Regression Problem

- The problem of predicting continuous values is called regression problem
- General approach: find a continuous function that models the continuous points.

Linear Regression with one input

A simple regression has the coefficient β and the constant α . The equation is then:

$$y = \alpha + \beta * x$$

where α is y-intercept and β is slope

Linear Regression with more than one input

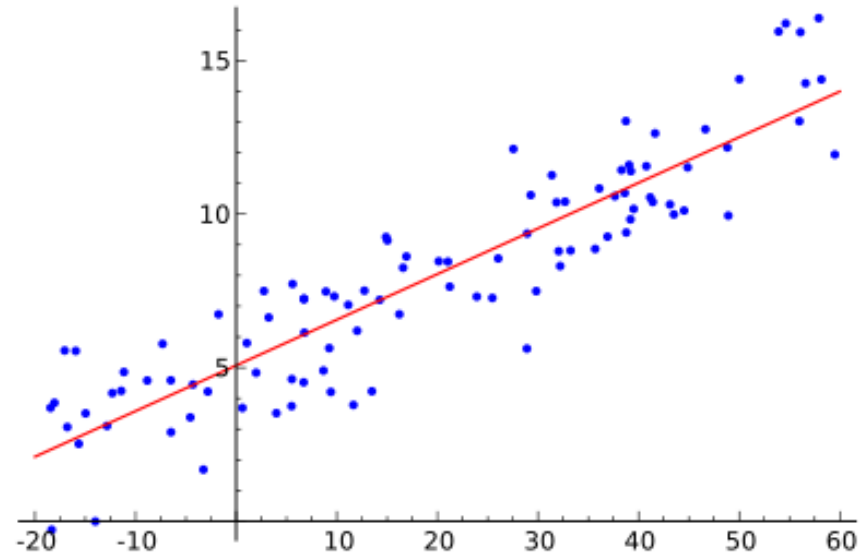
A multiple regression is in the following form:

$$y = \alpha + \beta_1 * X_1 + \beta_2 * X_2 + \dots + \beta_k * X_k$$

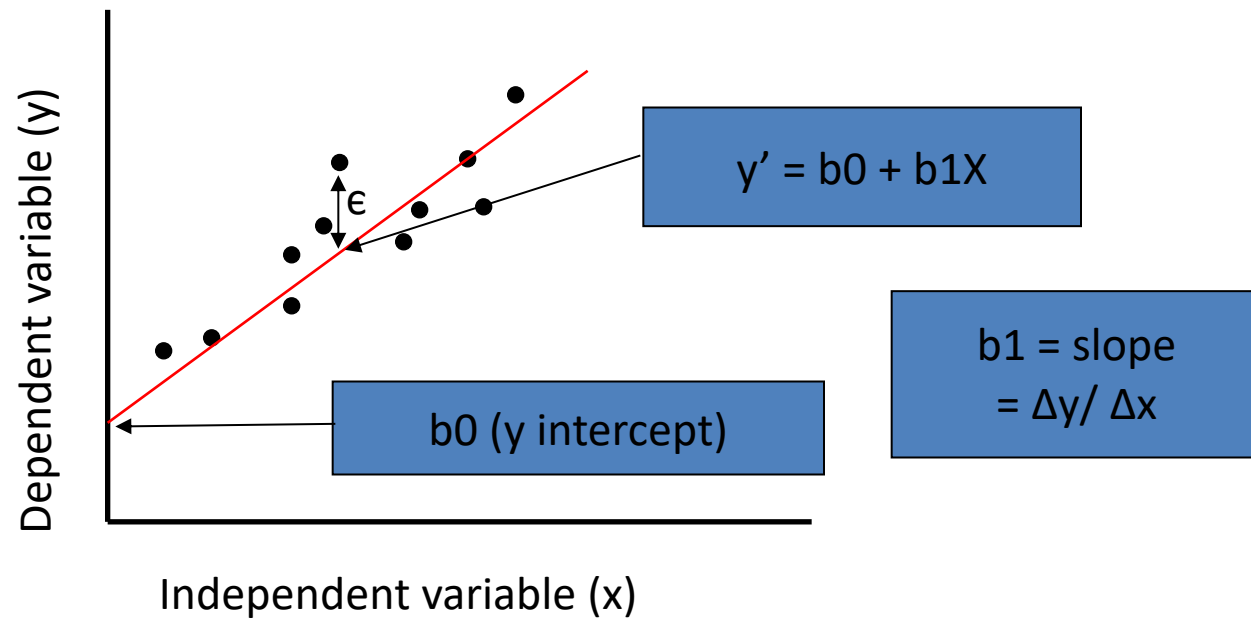
where k is the number of variables, or parameters.

Linear regression for 2D data

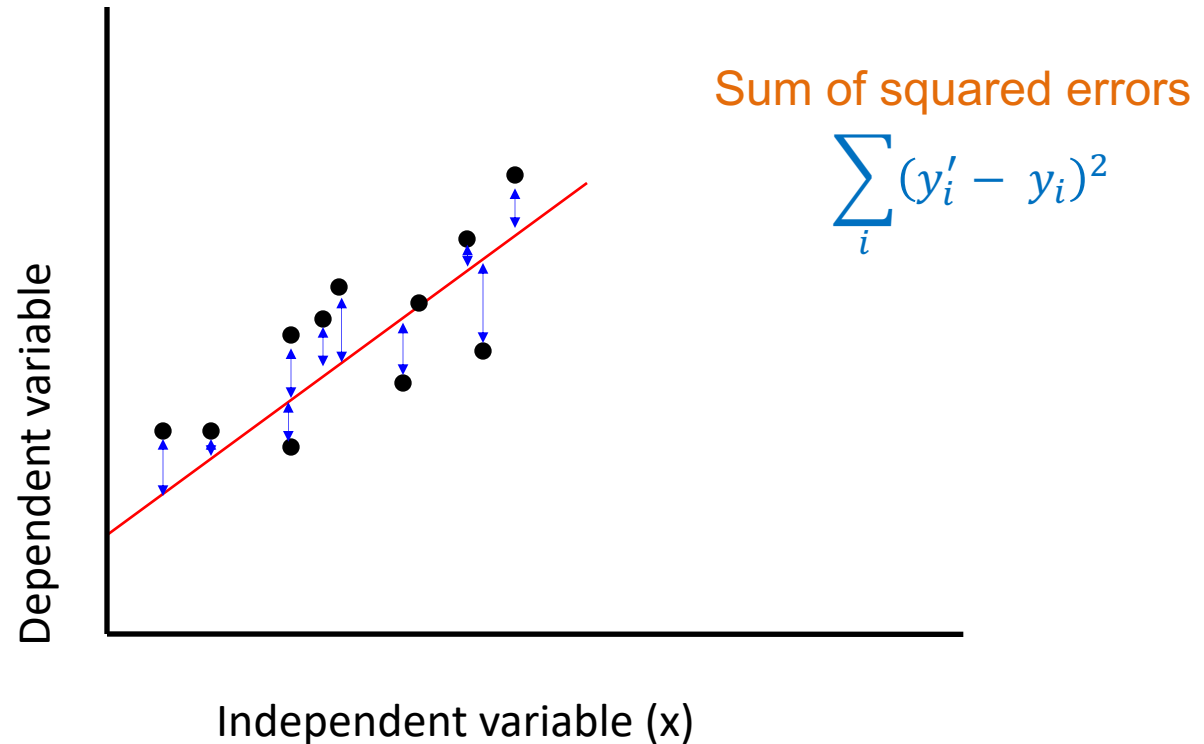
- Given a dataset of the form $\{(x_1, y_1), \dots, (x_n, y_n)\}$, find a **linear continuous function** that **minimizes the Sum of Squares of Error (SSE)**



Linear Regression

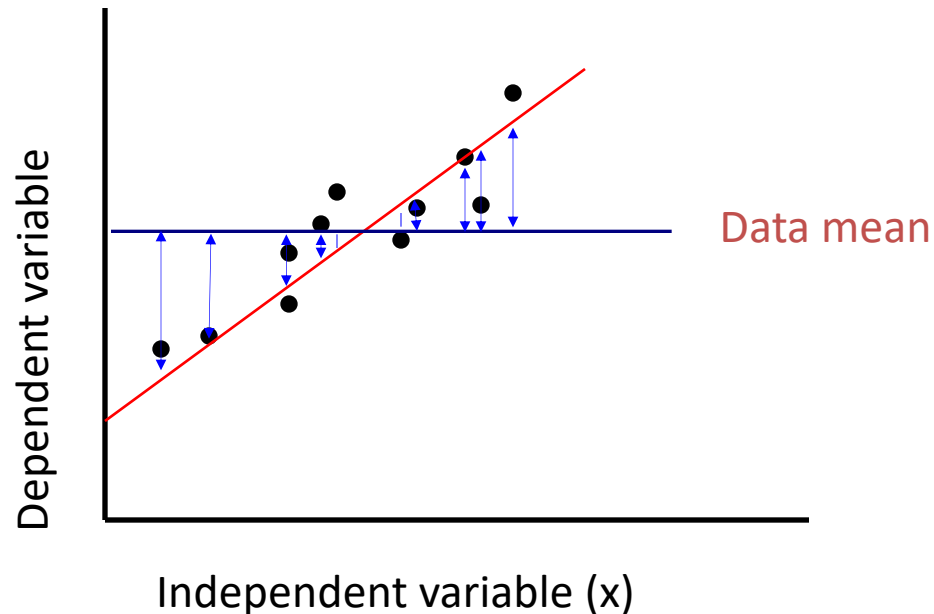


Sum of Squares of Error (SSE)



Sum of Squares of Error (SSE) is the sum of all the squared errors.

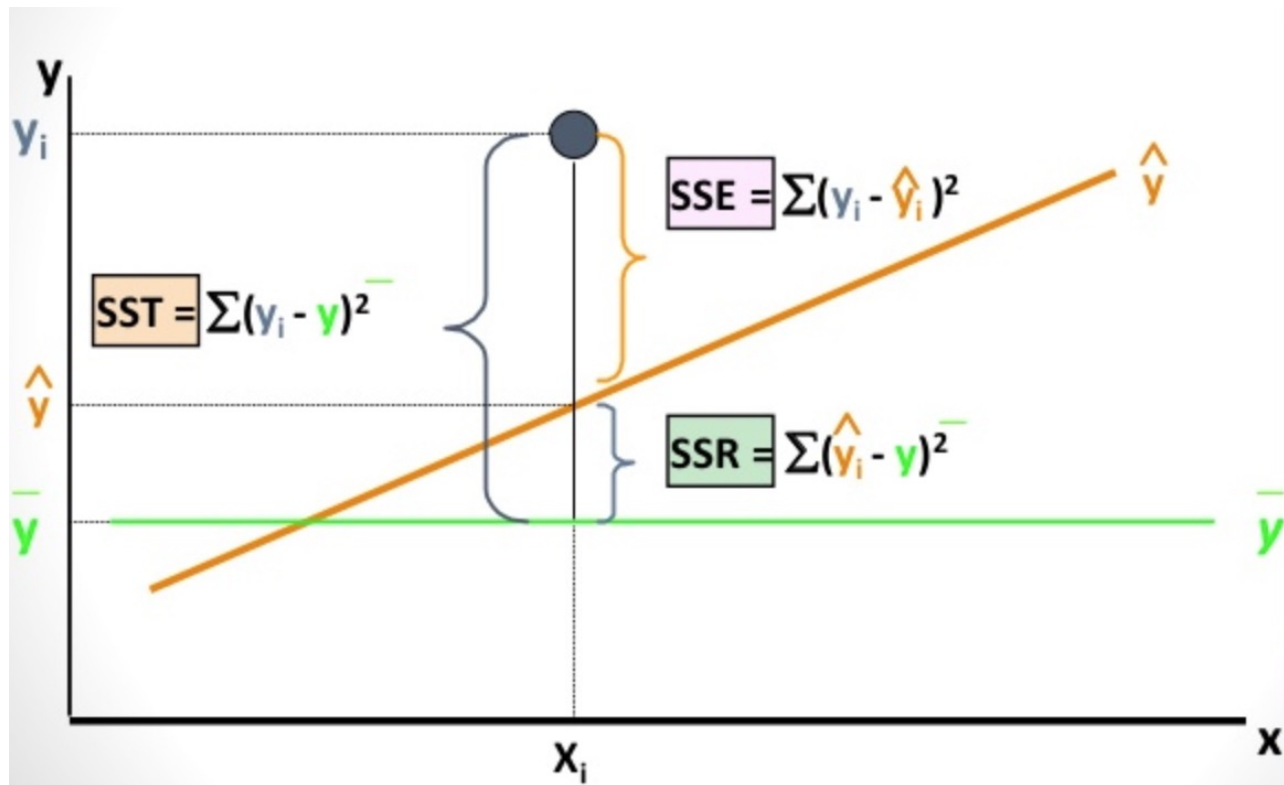
Sum of Squares of Regression (SSR)



Sum of Squares of Regression (SSR) is the sum of the squared differences between each prediction and the mean of data.

Sum of Squares of Total (SST)

$$\text{SST} = \text{SSE} + \text{SSR}$$

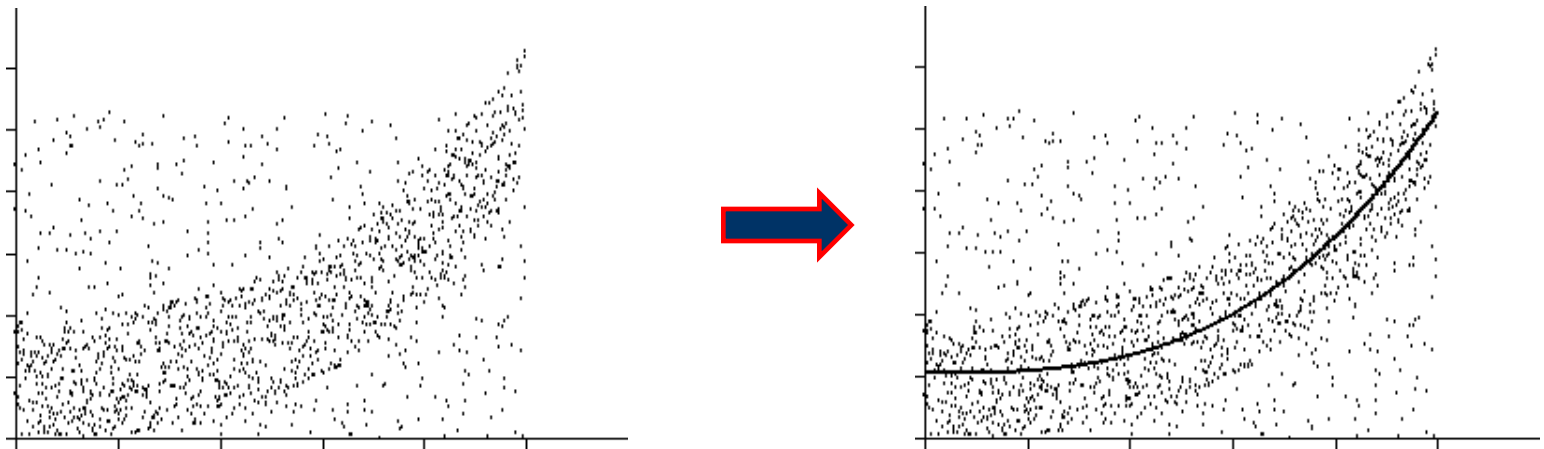


R-Squared score

- R-Squared score can be used as a single summary number to measure the quality of linear regression model
- The value of R^2 can range between 0 and 1.
- **The higher R^2 , the more accurate the regression model is.**

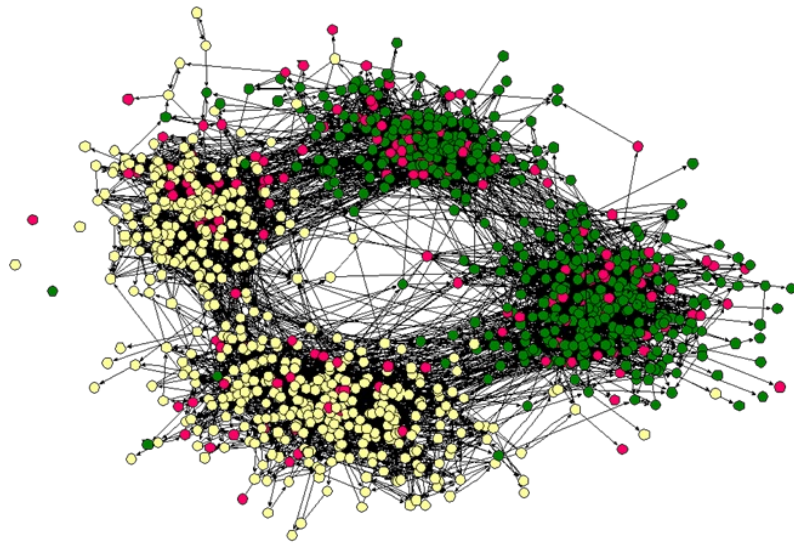
$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Nonlinear functions can also be fit as regressions



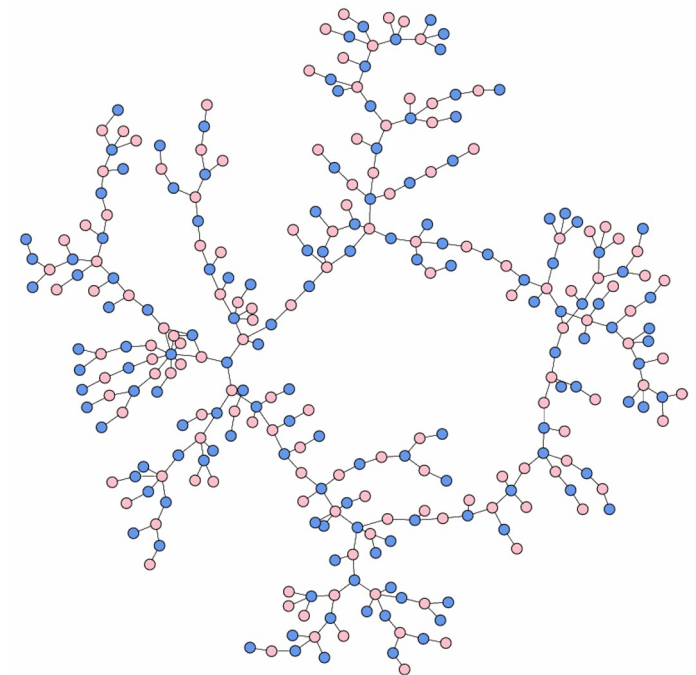
Any nonlinear continuous functions can also be fit as regressions, including power, Logarithmic, Exponential, and Logistic.

A Real Example: Social Network Analysis



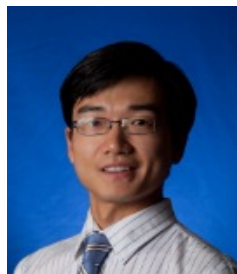
High school friendship

High school dating

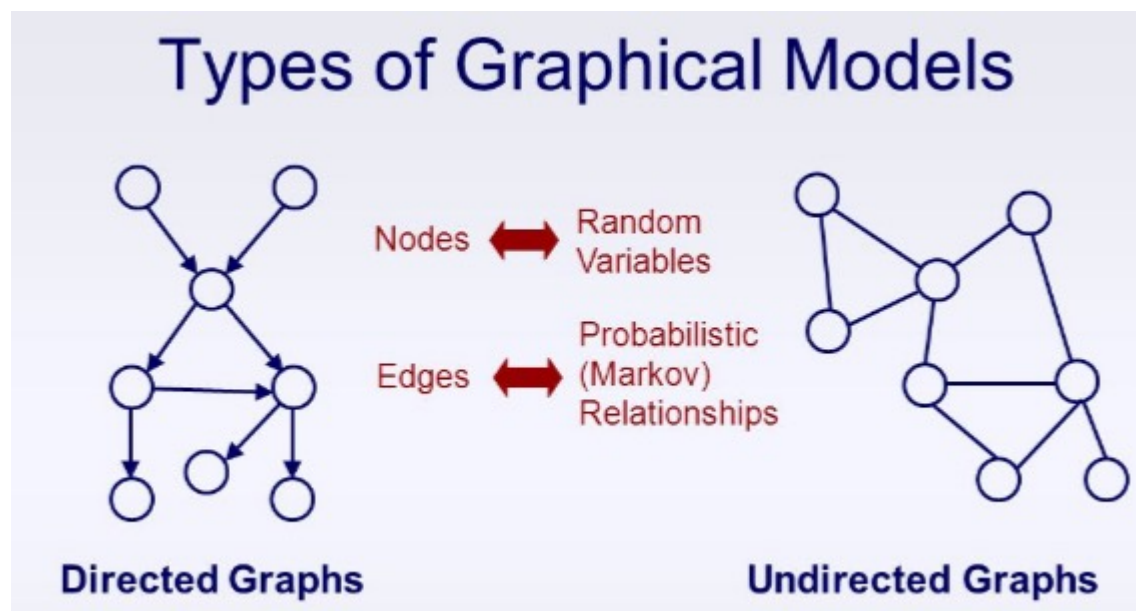


A Real Example: Social Network Analysis

- In link prediction, given any two arbitrary users, we aim to answer:
 - What is the likelihood that a particular link (relationship) exists between them?



Markov Networks v.s. Bayesian Networks



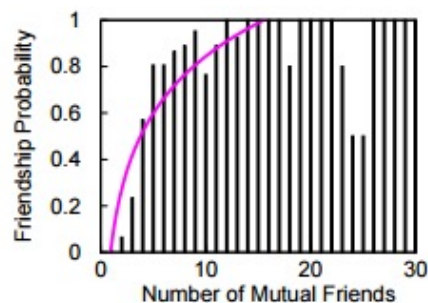
- **Haiquan Chen**, Wei-Shinn Ku, Haixun Wang, Liang Tang, Min-Te Sun:
Scaling Up Markov Logic Probabilistic Inference for Social Graphs. *IEEE Trans. Knowl. Data Eng.* 29(2): 433-445 (2017)

Experimental validation

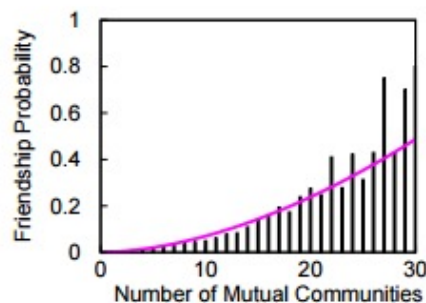
$$\left\{ \begin{array}{l} \text{common_friends}(X, Y, +n) \longrightarrow \text{knows}(X, Y) \\ \text{common_communities}(X, Y, +n) \longrightarrow \text{knows}(X, Y) \end{array} \right.$$

TABLE 1
Datasets used in the experiments.

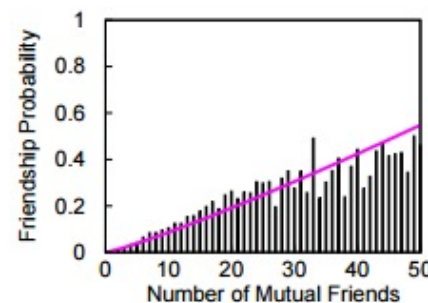
Name	Description	Number of nodes	Number of edges	Number of communities
DBLP	collaboration network	317,080	1,049,866	13,477
LJ	social network	3,997,962	34,681,189	287,512



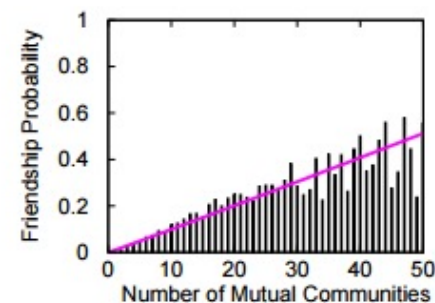
(a) Friendship probability versus # of common friends (DBLP)



(b) Friendship probability versus # common communities (DBLP)

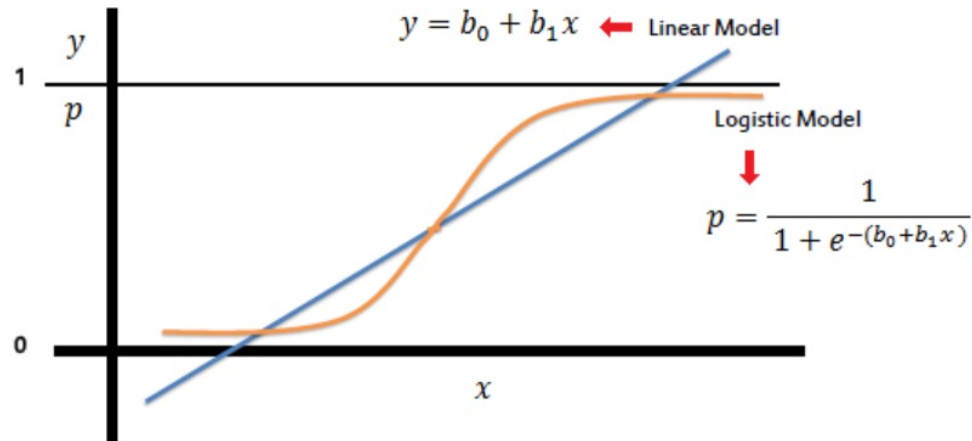


(c) Friendship probability versus # of common Friends (LJ)



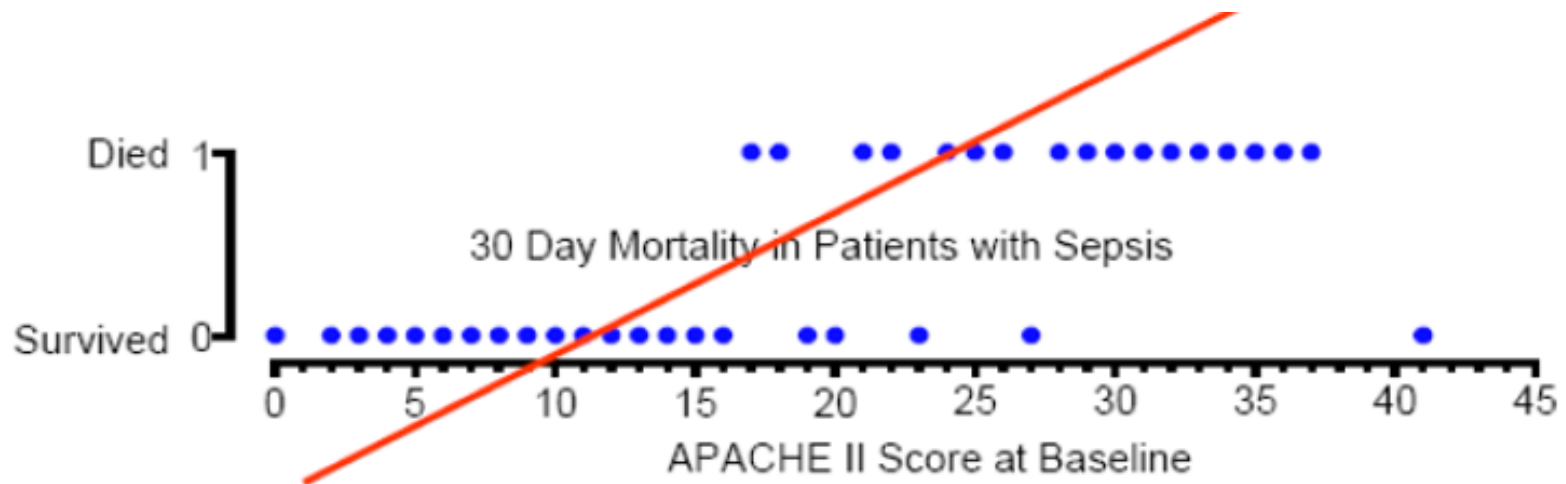
(d) Friendship probability versus # of common communities (LJ)

Now Logistic Regression...



Linear Regression Doesn't Work

- A linear function/regression is not good
 - It may produce probabilities beyond $[0, 1]$



Note: APACHE II is one of several ICU scoring systems.

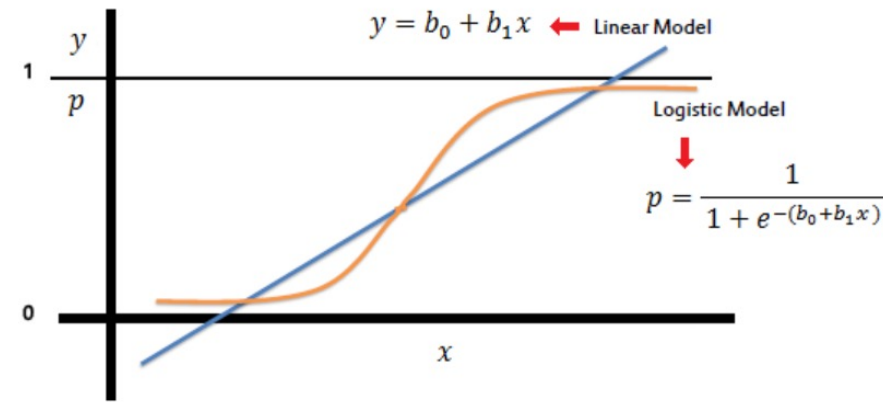
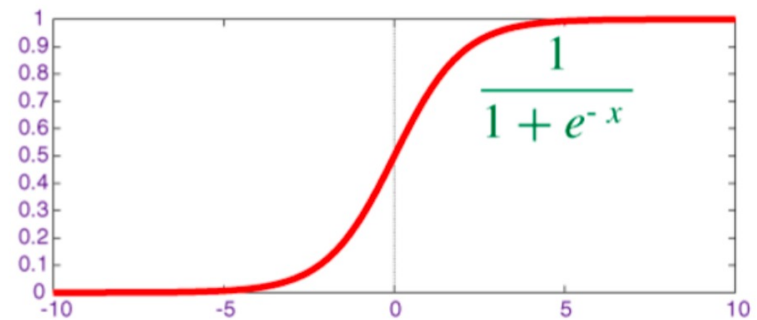
Logistic Regression

Logistic (Sigmoid) function maps any input x between $[0, 1]$

$$f(t) = \frac{1}{1 + e^{-x}}$$

Logistic regression is a linear regression on the **Sigmoid function**

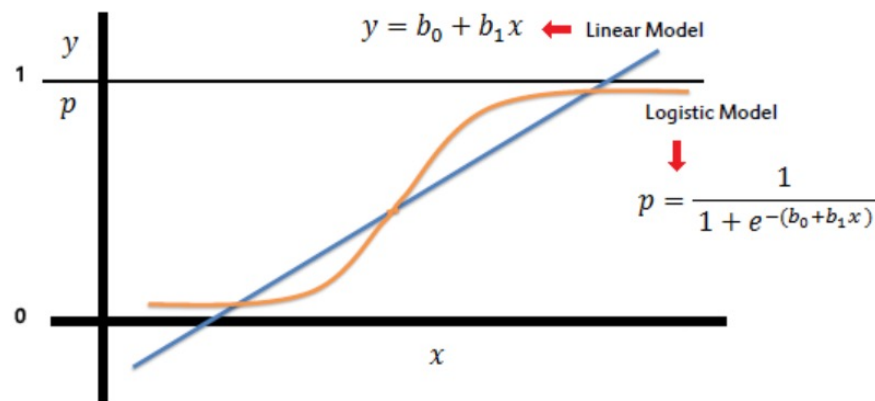
$$P(C|x) = \frac{1}{1 + e^{-(\alpha + \beta \cdot x)}}$$



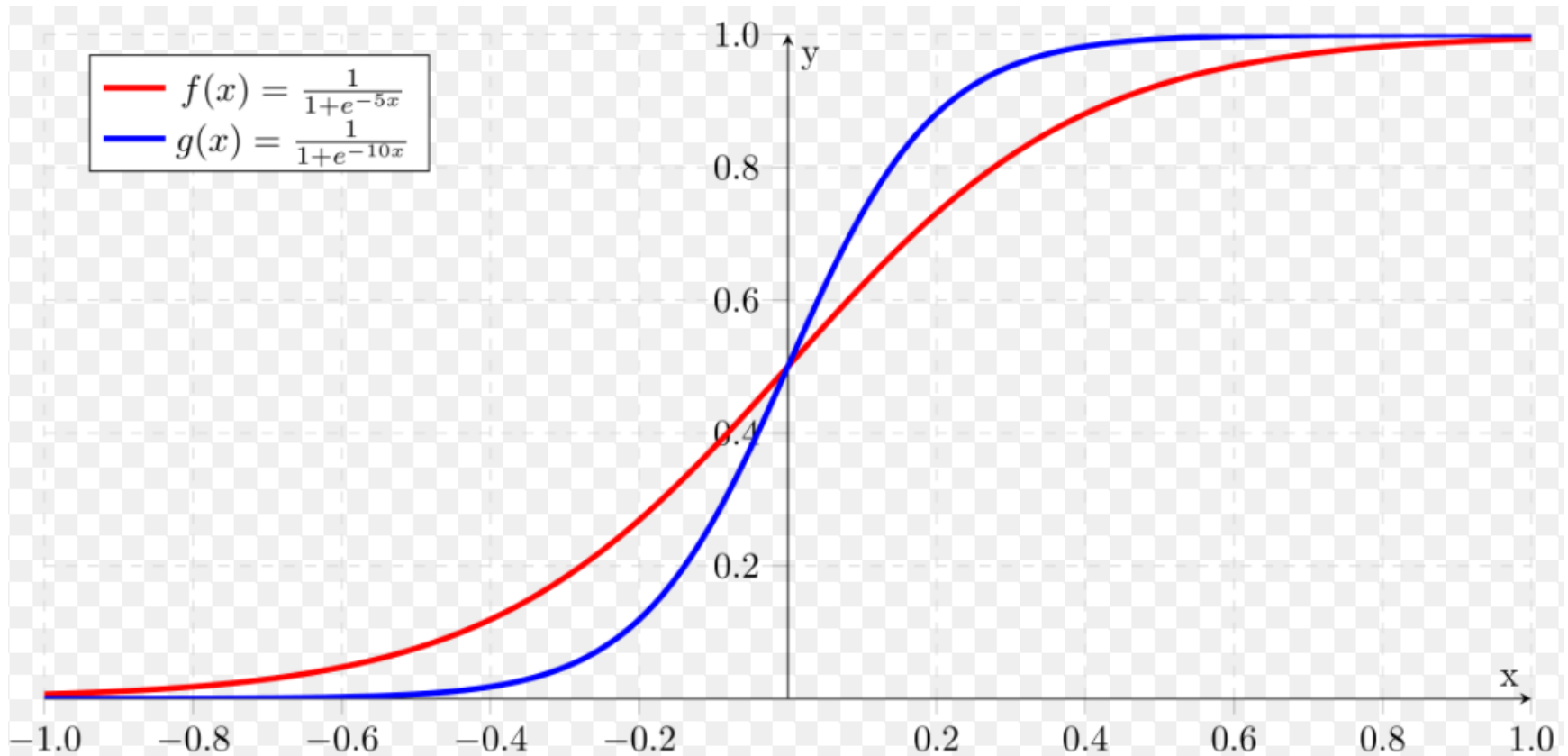
Q: What is the logistic regression model for more than one dimension?

Logistic Regression

- For 2-class problem, the probability threshold is set to 0.5 so
 - If the predicted probability ≥ 0.5 , predict “y = 1”,
 - If the predicted probability < 0.5 , predict “y = 0”,

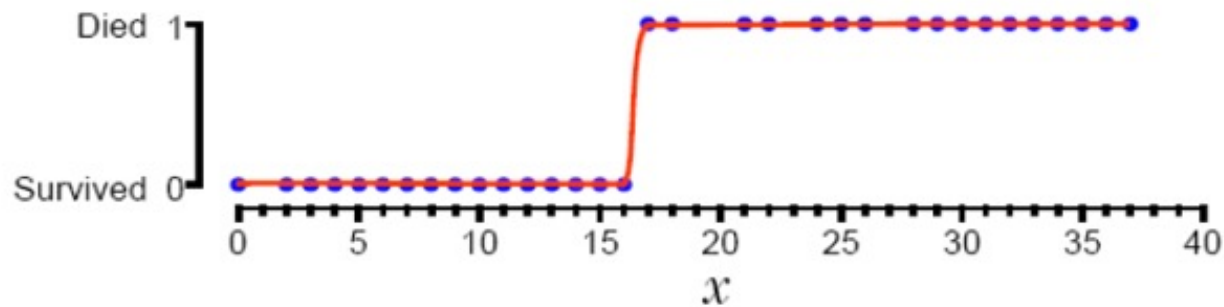


How β affects the model

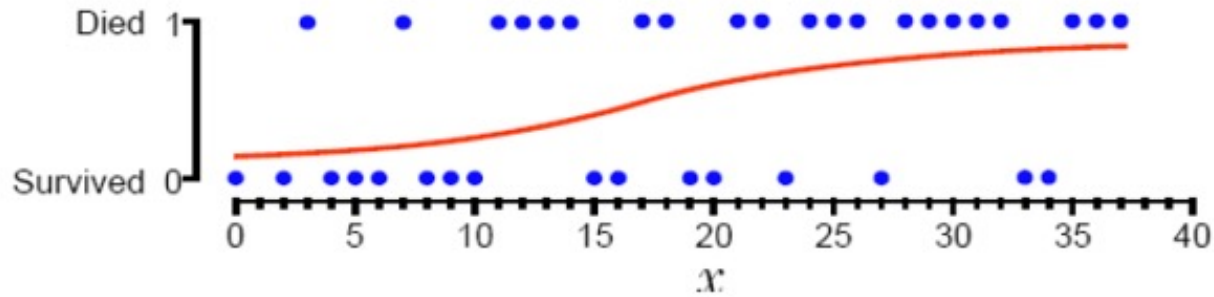


Compare Two Models In One Dimension

Data that has a sharp survival cut off point between patients who live or die should have a large value of β .



Data with a lengthy transition from survival to death should have a low value of β .



Q: What x value makes the model output a probability of 0.5?

Logistic regression in 2D space

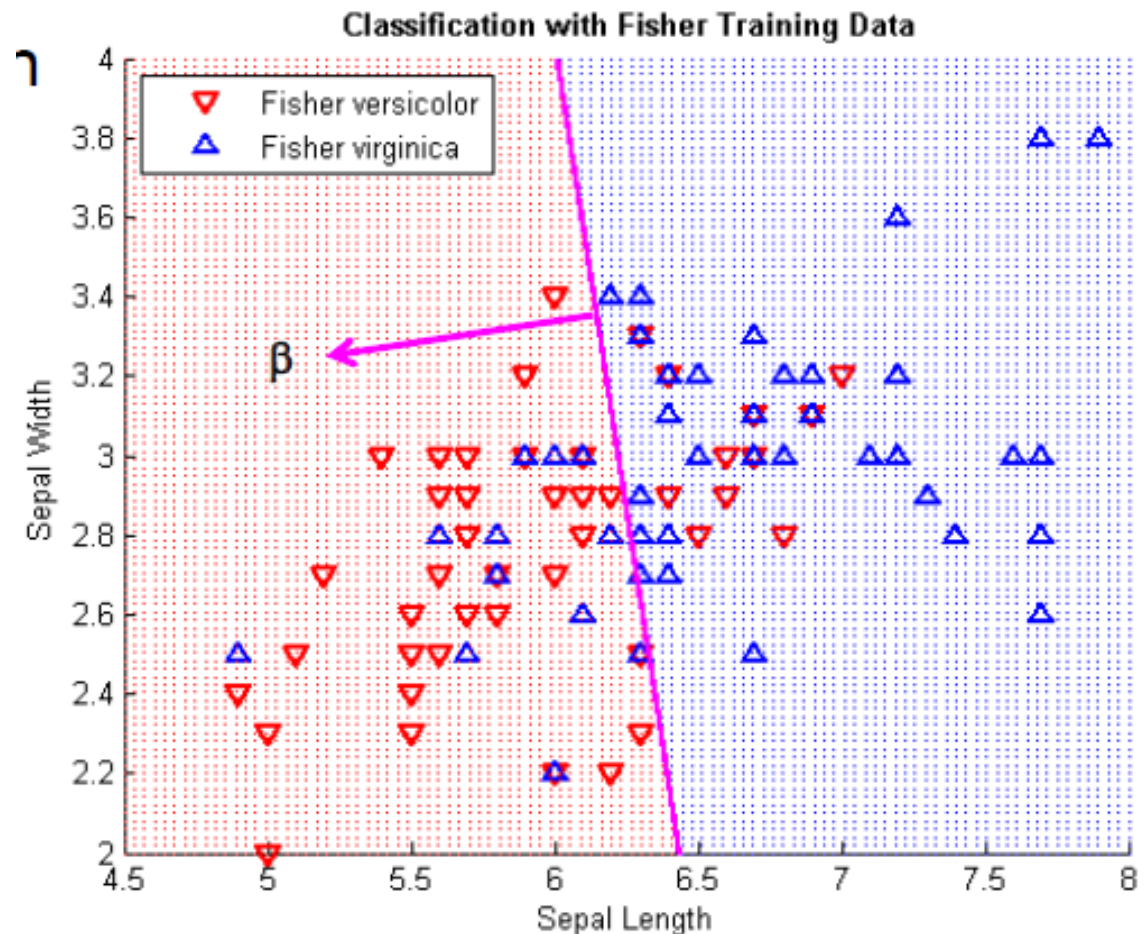
Predict Iris flower species based on **sepal length** and **sepal width** only

Coefficients

$$\beta_1 = -1.9$$

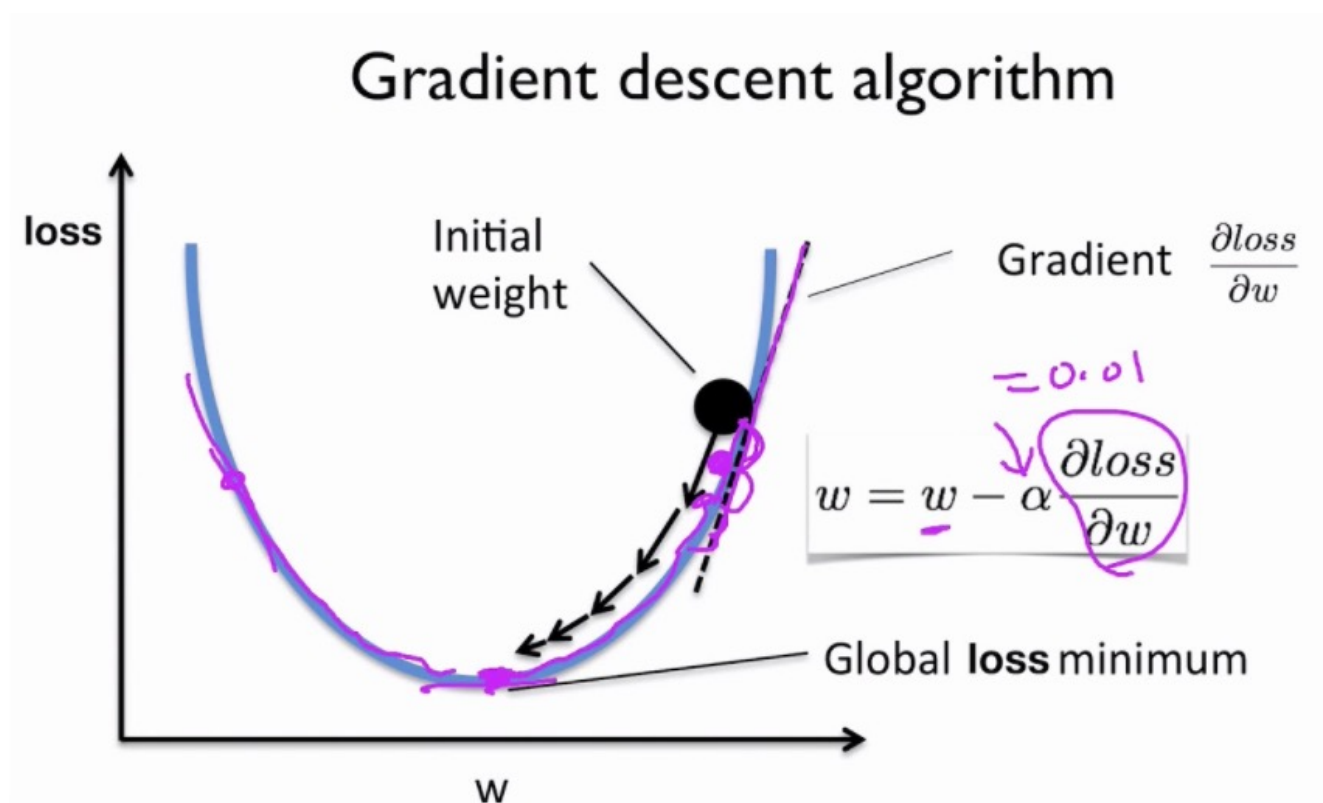
$$\beta_2 = -0.4$$

$$\alpha = 13.04$$

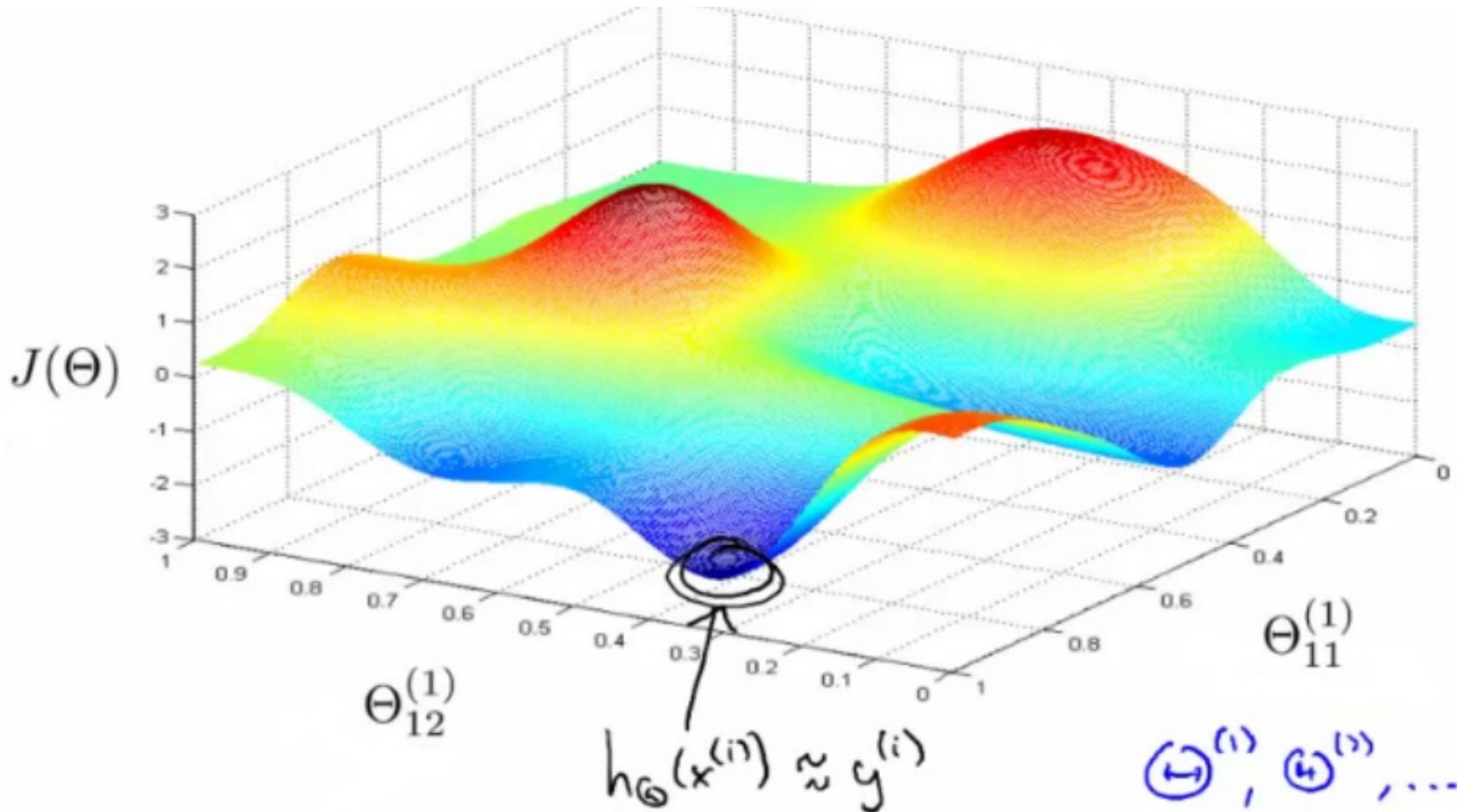


Estimating the coefficients

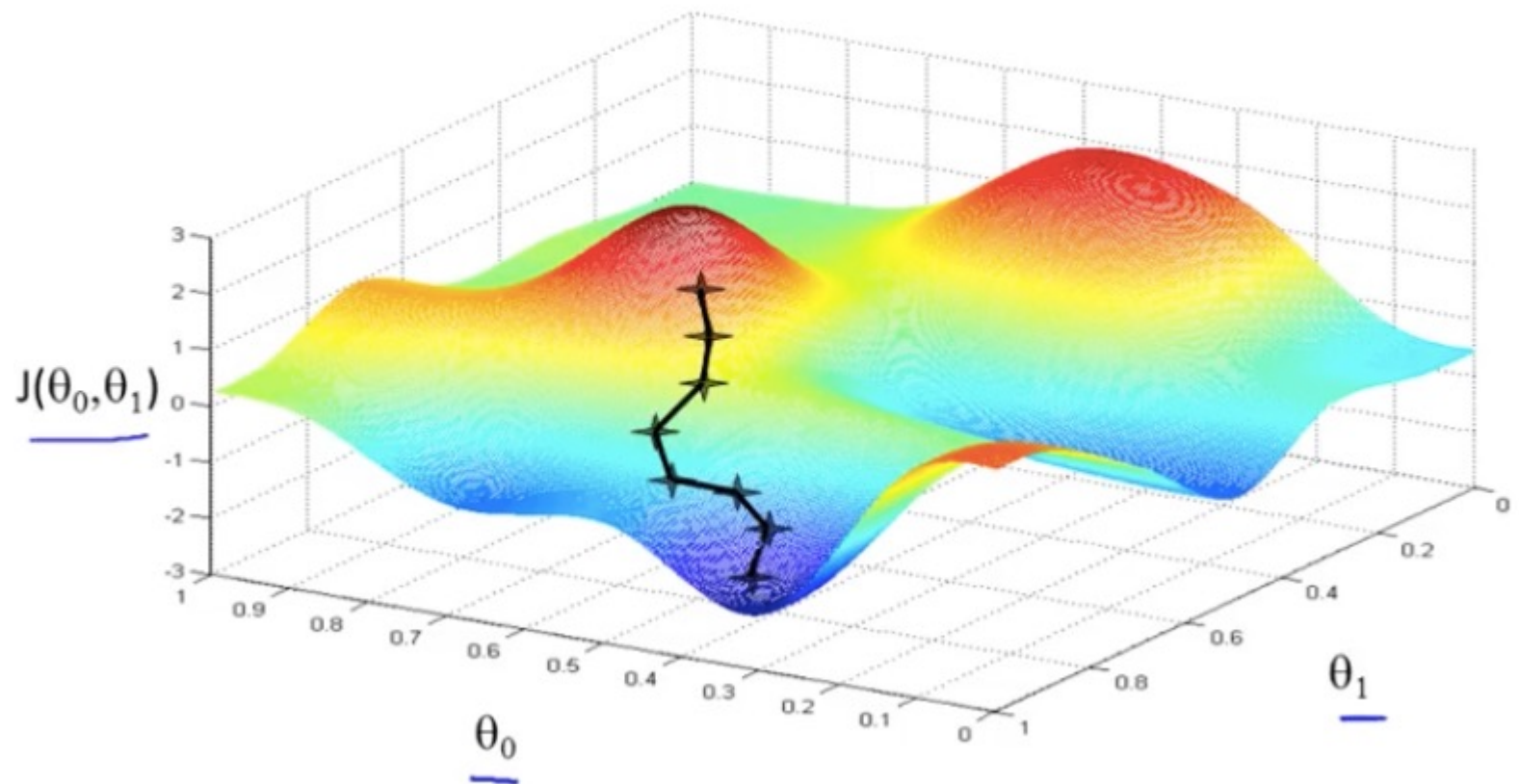
- Use **gradient descent algorithm** to find the near-optimal coefficients for linear/logistic regression



Gradient descent for two parameters



Gradient descent for two parameters



Gradient descent implementation

Batch gradient descent

$$\rightarrow J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Repeat {

$$\rightarrow \theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$\frac{\partial}{\partial \theta_j} J_{train}(\theta)$$

(for every $j = 0, \dots, n$)

}

Stochastic gradient descent

$$\rightarrow \text{cost}(\theta, (x^{(i)}, y^{(i)})) = \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$\rightarrow J_{train}(\theta) = \frac{1}{m} \sum_{i=1}^m \text{cost}(\theta, (x^{(i)}, y^{(i)}))$$

1. Randomly shuffle dataset. \leftarrow

2. Repeat {

for $i = 1, \dots, m$ {

$$\theta_j := \theta_j - \alpha (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

(for $j = 0, \dots, n$)

$$\frac{\partial}{\partial \theta_j} \text{cost}(\theta, (x^{(i)}, y^{(i)}))$$

$$\rightarrow (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \dots$$

Sklearn implementation

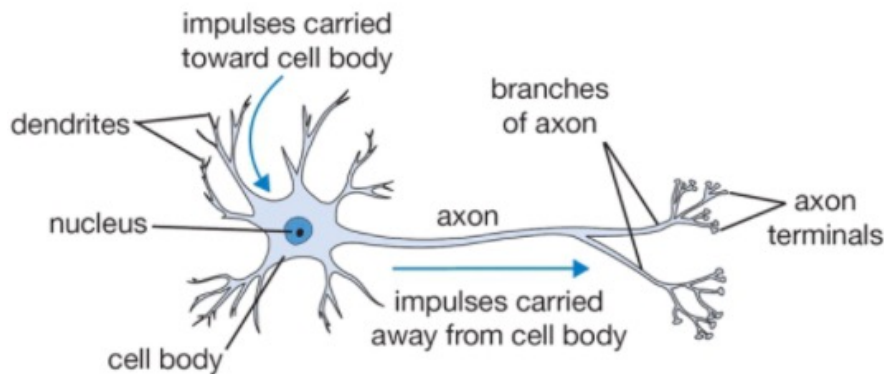
- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Logistic/Linear Regression Advantages

- Linear regression produces a continuous output.
- Logistic regression produces class membership with predicted probability .
- The coefficients can be used for understanding the feature importance.
- Works for relatively large datasets

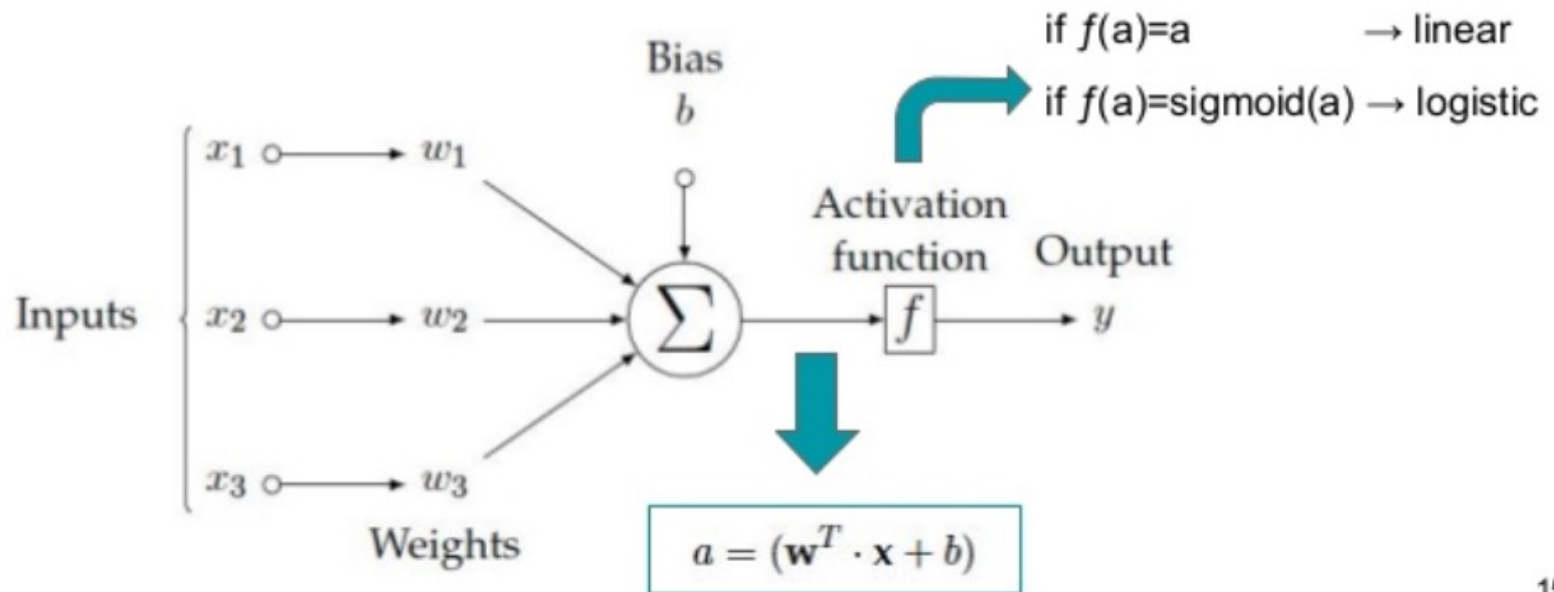
Neural networks

- Logistic regression can be considered as the simplest form of a **neural network**, which is a collection of perceptron.
- Perceptron is seen as an analogy to a biological neuron.

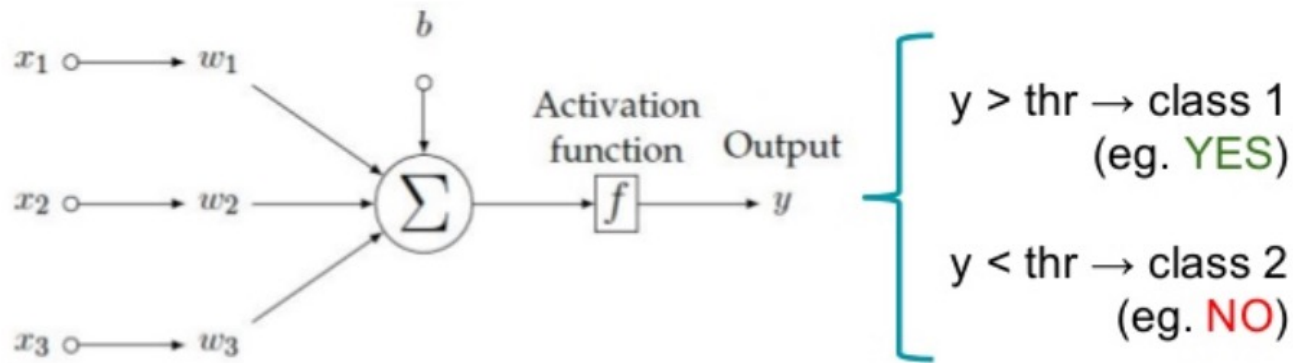


Perceptron (Neuron)

The Perceptron can represent both linear & logistic regression:

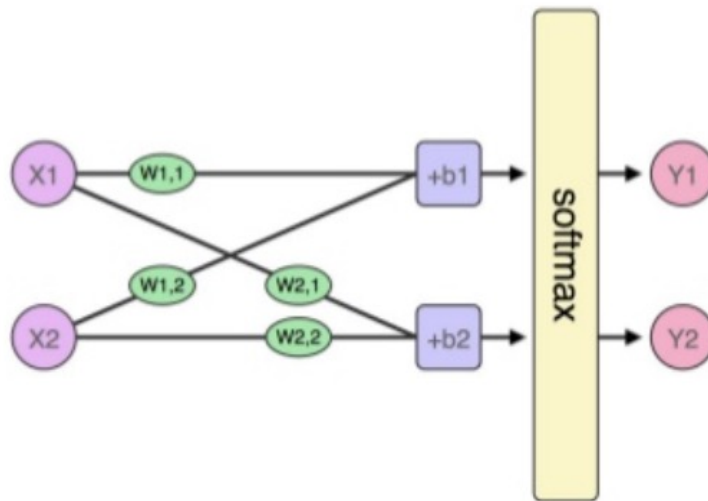


2-class classification with one neuron



2-class classification with two neurons, one for each class

Putting multiple neurons in parallel we can predict multiple classes



Softmax normalization

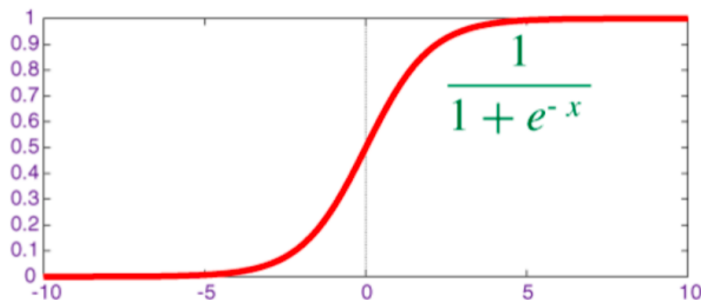
$$P(y = k|\mathbf{x}) = \frac{\exp \mathbf{x}^T \mathbf{w}_k}{\sum_{n=1}^N \exp \mathbf{x}^T \mathbf{w}_n}$$

Normalization factor so that the sum of probabilities sum up to 1.

sigmoid vs softmax

- **sigmoid** function is used for the two-class logistic regression

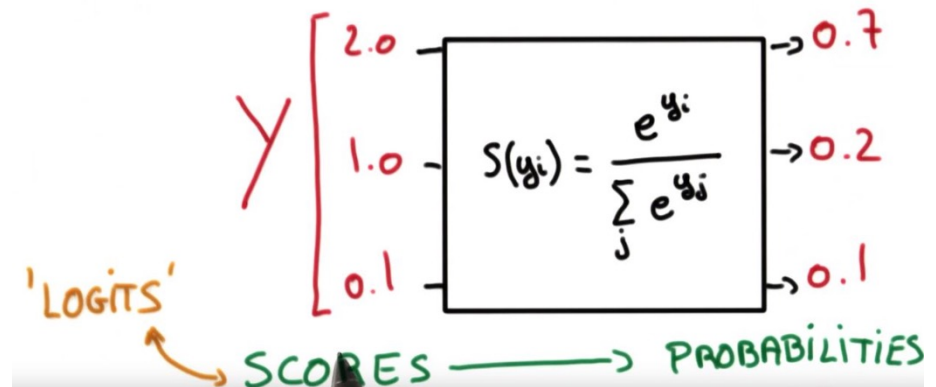
$$\frac{1}{1 + e^{-\beta \cdot X_i}}$$



- **softmax** function is used for the multiclass logistic regression

$$\frac{e^{\beta_k \cdot X_i}}{\sum_{0 \leq c \leq K} e^{\beta_c \cdot X_i}}$$

SOFTMAX



Softmax implementation

What is softmax([1, 2, 3])

```
def softmax(x):  
    """Compute the softmax of vector x."""  
    exps = np.exp(x)  
    return exps / np.sum(exps)
```

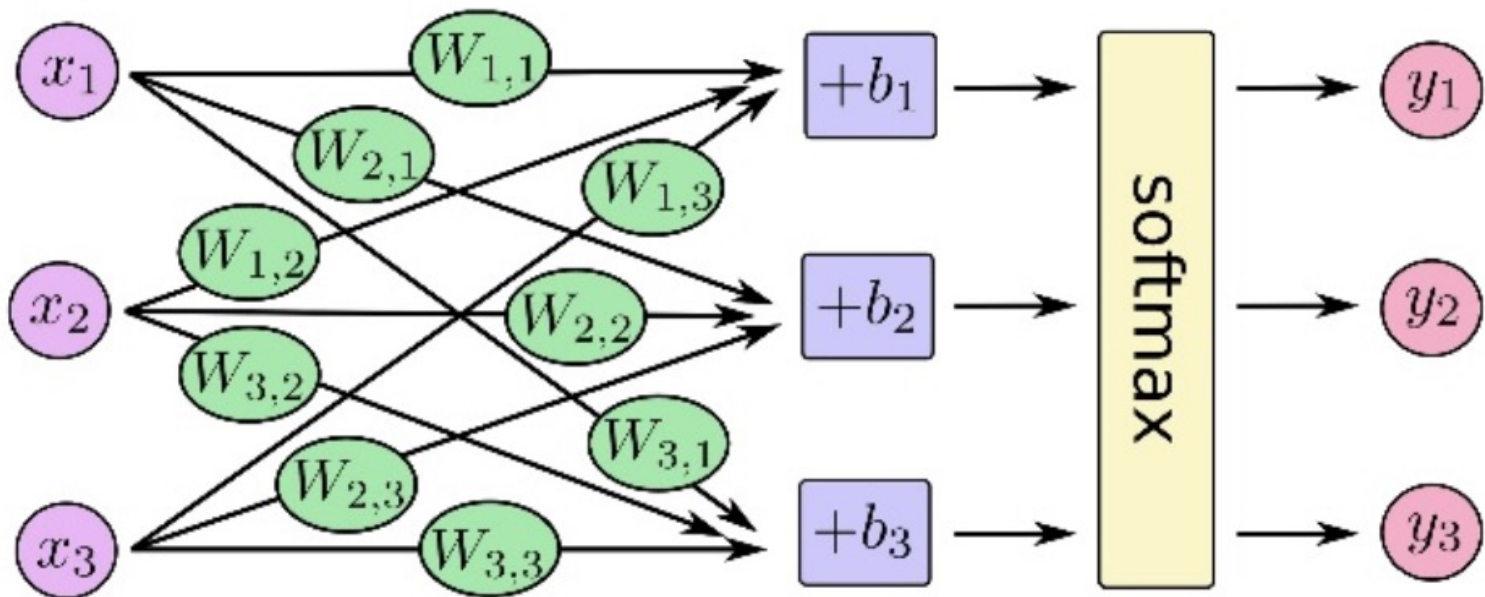
- $y_1 = \frac{e^1}{e^1 + e^2 + e^3} = 0.09$
- $y_2 = \frac{e^2}{e^1 + e^2 + e^3} = 0.24$
- $y_3 = \frac{e^3}{e^1 + e^2 + e^3} = 0.67$

Output:

- [0.09, 0.24, 0.67]

3-class classification with three neurons

Putting multiple neurons in parallel we can predict multiple classes



Neural Network = Multi Layer Perceptron

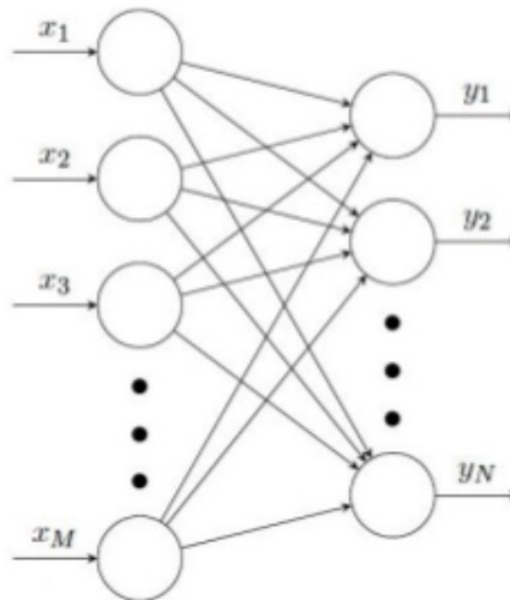
- Multiple classes can be predicted by putting many neurons in parallel.



raw pixels
unrolled img

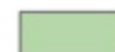


Input
layer



Ouput
layer

0.3 "dog"



0.08 "cat"



...

0.6 "whatever"



Sklearn implementation

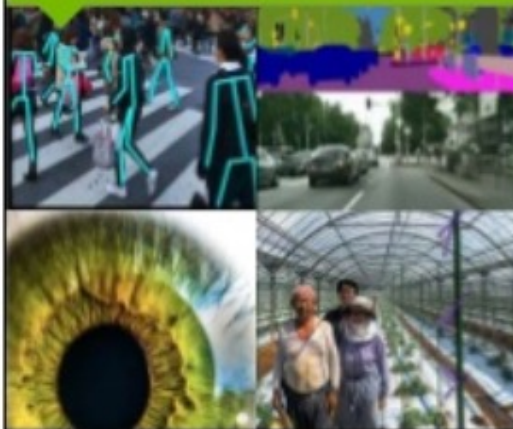
- http://scikit-learn.org/stable/modules/neural_networks_supervised.html#

Deep learning

AI APPLICATIONS

Image Classification Object Detection

COMPUTER VISION



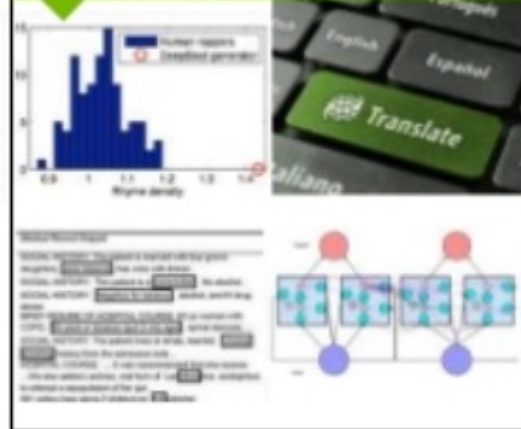
Voice Recognition Language Translation

SPEECH & AUDIO



Recommendation Engines Sentiment Analysis

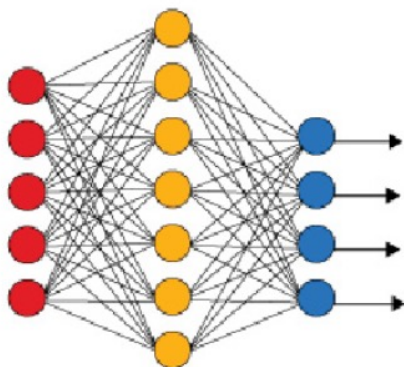
NATURAL LANGUAGE PROCESSING



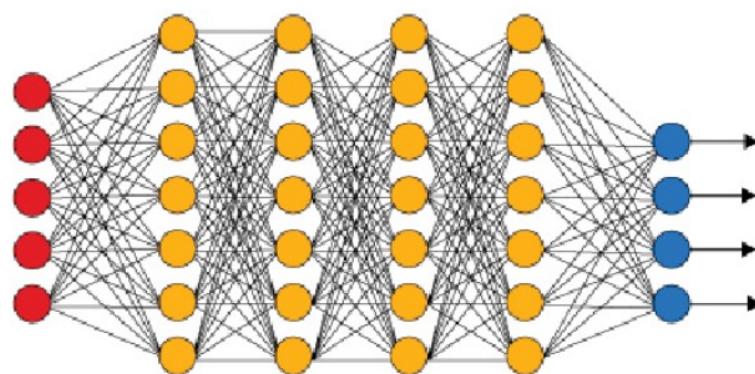
Neural networks vs deep learning neural networks

- "Normal" neural networks usually have **one to two hidden layers** and are used for **SUPERVISED classification**.
- Deep learning neural network have **more hidden layers** and can be used for **both UNSUPERVISED and SUPERVISED** learning tasks.

Simple Neural Network



Deep Learning Neural Network



● Input Layer

● Hidden Layer

● Output Layer

An example deep learning neural network

