

# **Comparison Study of Neural Network and Deep Neural Network on Repricing GAP Prediction in Indonesian Conventional Public Bank**



**By : Hendri Karisma and Dwi Hendratmo Widyantoro**

International Conference  
On System Engineering and Technology (ICSET)  
2016

# Introduction

- To determine of the economy condition of a country, Bank have become an important component (duygun, 2009).
- A healthy banking system affected by public confidence. The public confidence can be cause non-systemic and systemic problems, or causing a crisis.

# Background and Problem

- Bank of Indonesia (BI) and Deposit Insurance Corporation of Indonesia (LPS) as supervisor institution that assess the condition of the bank performance, doing bank performance prediction
- Many approaches are used to analyze the condition of the bank, and is still being developed.
- X-Efficiency is the degree of the condition of the efficiency of a company such as a bank, one of the many ways to judge the efficiency of a company is to look at the condition of risk by looking the repricing gap.
- Requires an approach that is more effective and efficient.
- Size of the bank report data and the structures is not consistent.

# Goal and Scope

- Goal

Knowing the accuracy performance of the implementation of deep neural network in repricing gap using historical data of monthly report of the conventional public bank of Indonesia.

- Scope

- The historical data that used in the experiments is monthly report data from 2003 until 2013 from Bank of Indonesia site.
- Dataset was constructed using aggregation ratio result (38 ratio) from 11 years historical report data (2003 to 2013).

# Bank Performance

- Health condition of the bank.
- Efficiency condition of the financial area.
- The condition of liquidity, risk, and another ration.
- The economic condition of the country.

# Economic Ratios

- There are so many ratios that can be used in bank performance assessment:
  - ROA, ROE, CR, RR, LDR, LAR, ATMR, CAR, etc.

# Repricing Gap

- Interest rate risk which is the risk of gain or capital arising from movements in interest rates (periodic).
- gap between assets and liabilities of a financial institution such as a bank.

# Two Model of Bank Performance Assessment Approach

- Credit Risk Rate
- Statistic & Stochastic Method or Machine Learning



# CAMEL

- **Capital Adequacy** : Expected capital to maintain the balance with the existing risk
- **Assets** : Assessment of asset quality
- **Management Capability** : Quality management, capabilities of board directors & managements.
- **Earnings** : Income quality, including public trust, and absorb the losses due to loan.
- **Liquidity** : The need for adequate sources of liquidity to meet the needs of today and the future.

# RBBR

The standard that used by Bank of Indonesia :

- Risk Profile
- Good Corporate Governance
- Earning
- Capital



# Related Research

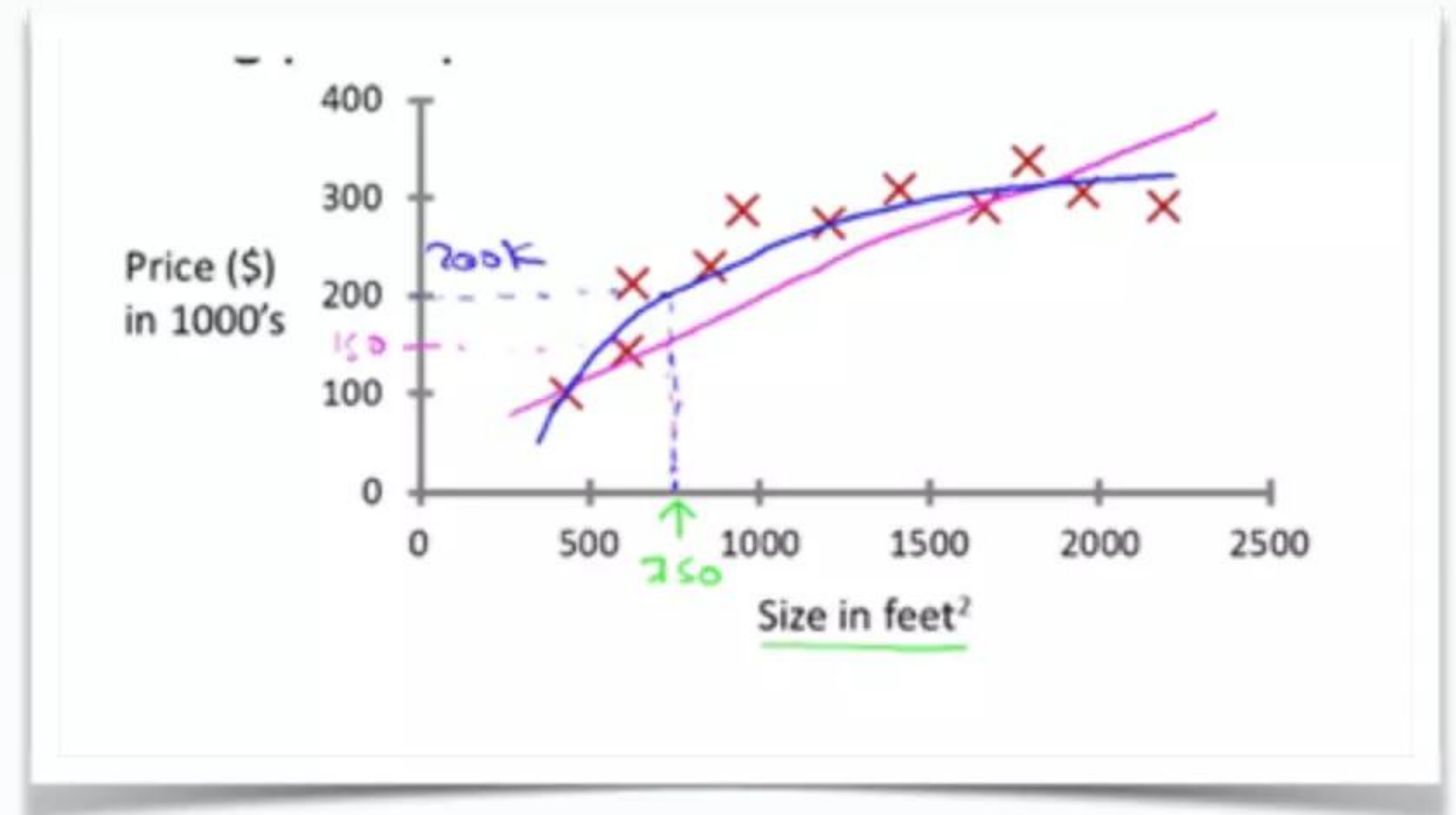
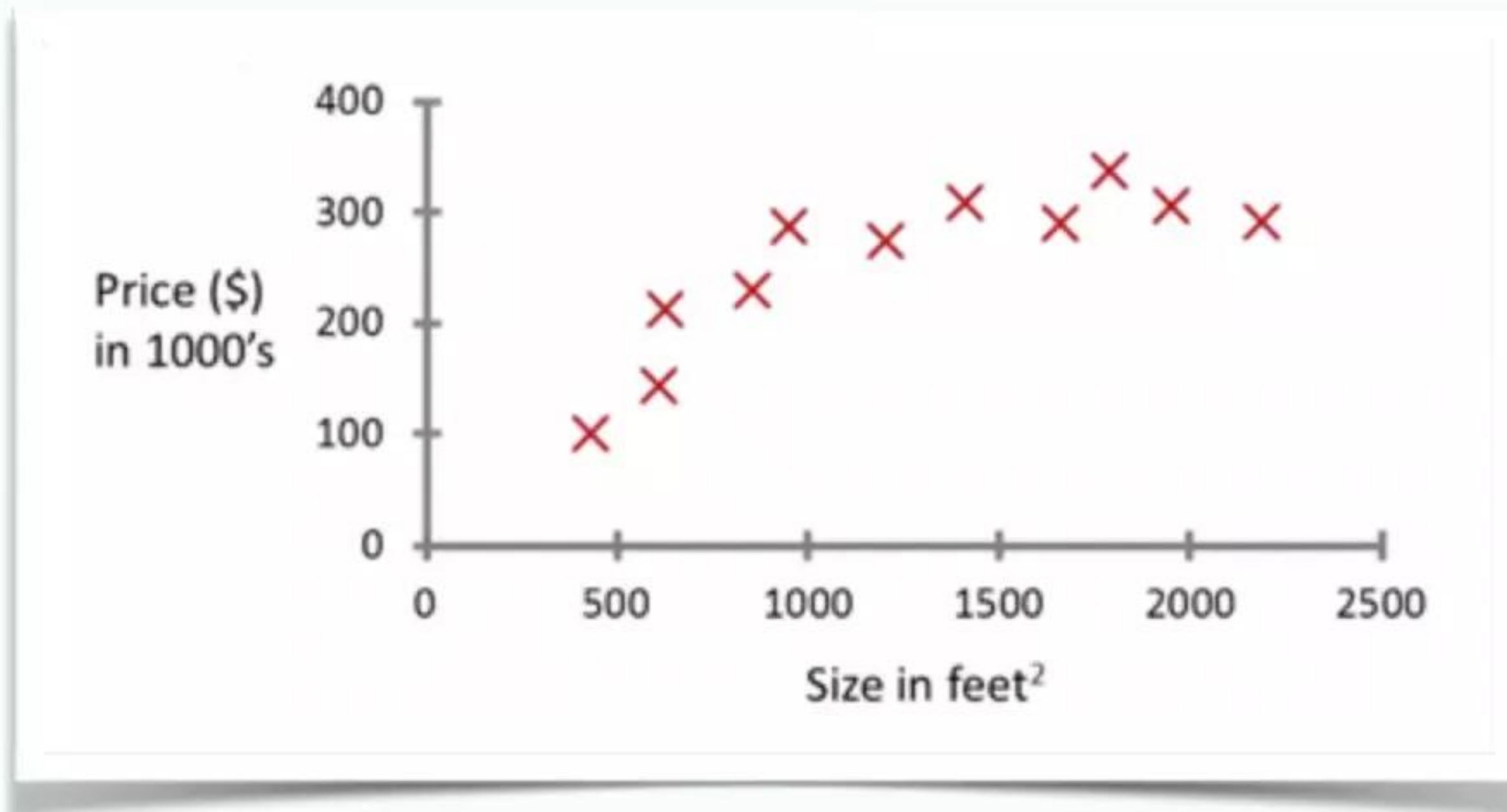
- There are several research about bank performance analysis or prediction, financial early warning, and bankruptcy prediction :
  - C. Devasena, 2015, analyzed the credit risk using iBK which is one of kNN algorithm, than the result was compared to k-NN, k-Star, and LWL classifier,
  - W. Ching, 2006, predicting GDP using SVM and neural network. While T. Papadimitriou 2013, conducted research on forecasting of bankruptcy condition in a company in US.
  - D. Kumar, 2013, conducted research on bank performance forecasting using stock market data and using neural network algorithm. A. Karaa, 2012 conducted research on credit risk assesement using neural network and svm as comparison research with the data of commercial bank of Tunisia.
  - C. Worrell, 2012, conducted research on comparison methods of SVM, C4.5, CN2, naive bayes, logistic regression to rank the bank that have possibility to fail using quarterly report data.
  - R. Zhang, 2013 dan Z.-Yuan. Yu, 2011 both research was risk analyzing of bank using neural network backpropagation with adding PCA and with different features.
  - etc.

# The Methods that Compared

- Standard Backpropagation
- Deep Neural Network with Autoencoder



# Regression



$$y = mx + c$$

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1$$

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

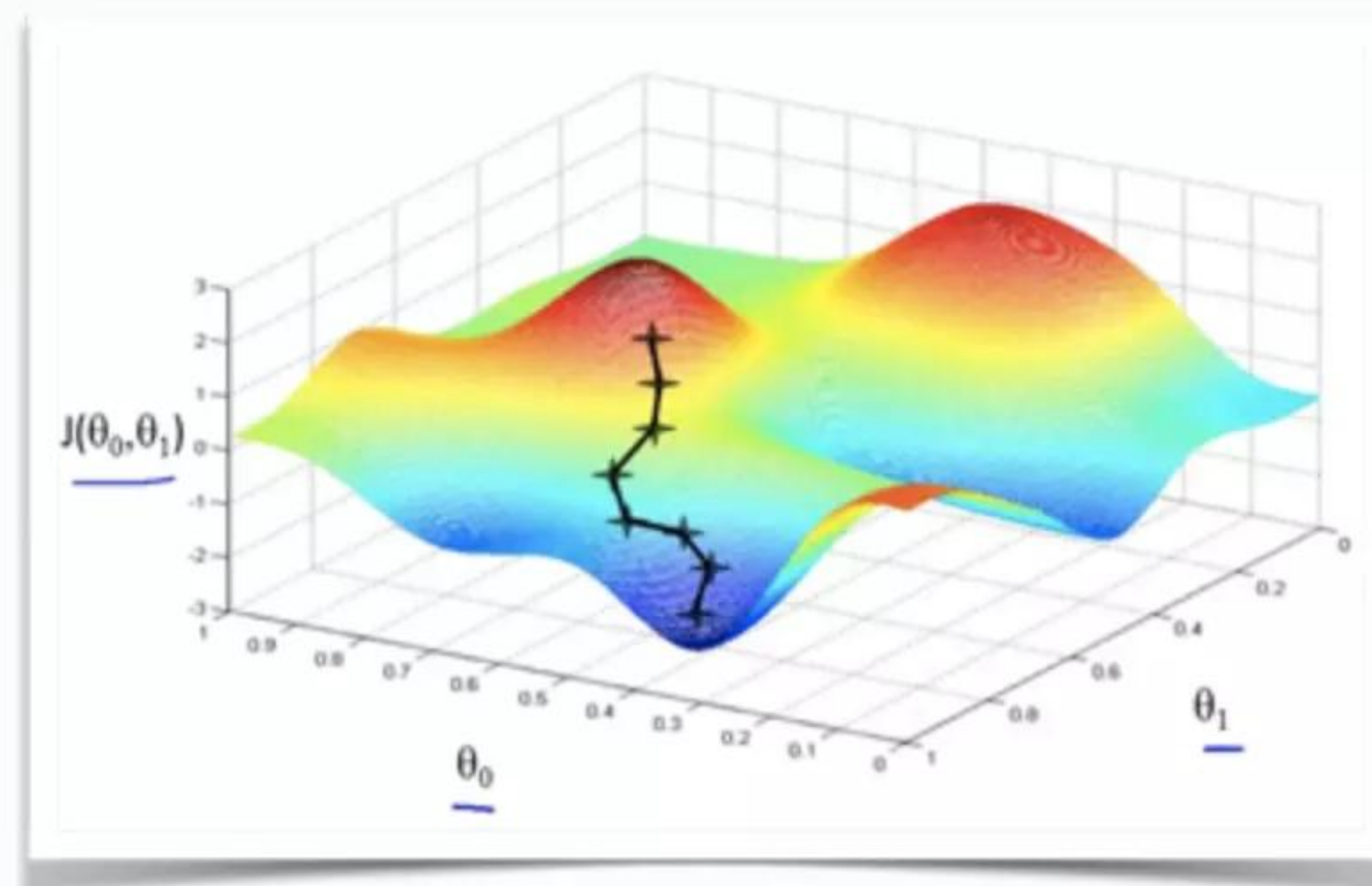
# Stochastic Gradient Descent

- Cost Function :

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

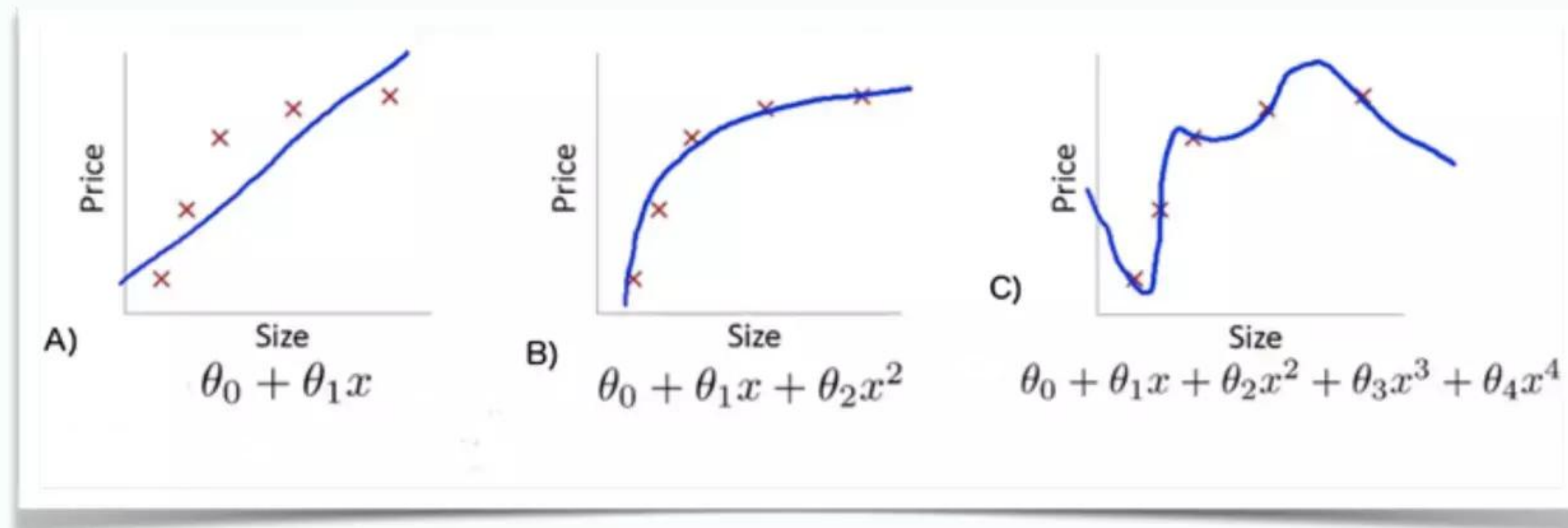
- Update parameters :

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$



$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$
$$\theta_{j=0}: \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$
$$\theta_{j=1}: \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(1)}$$

# Overfitting



- Cost Function :

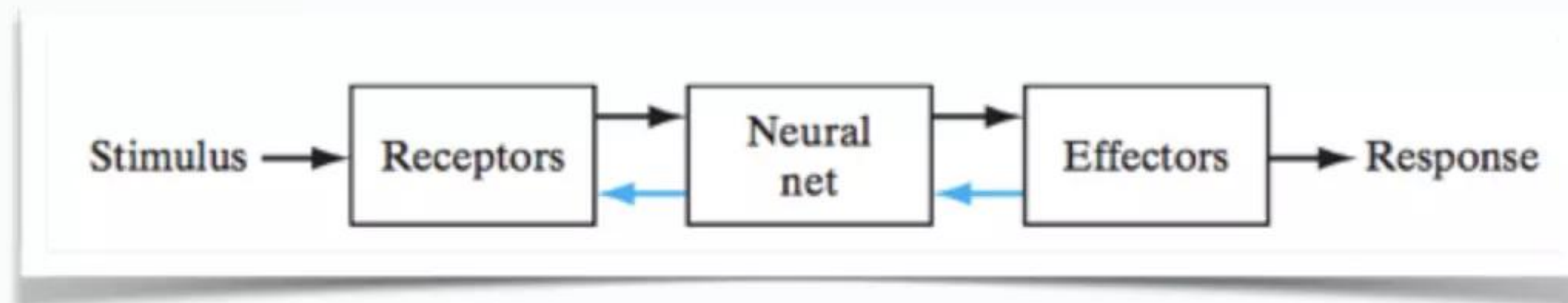
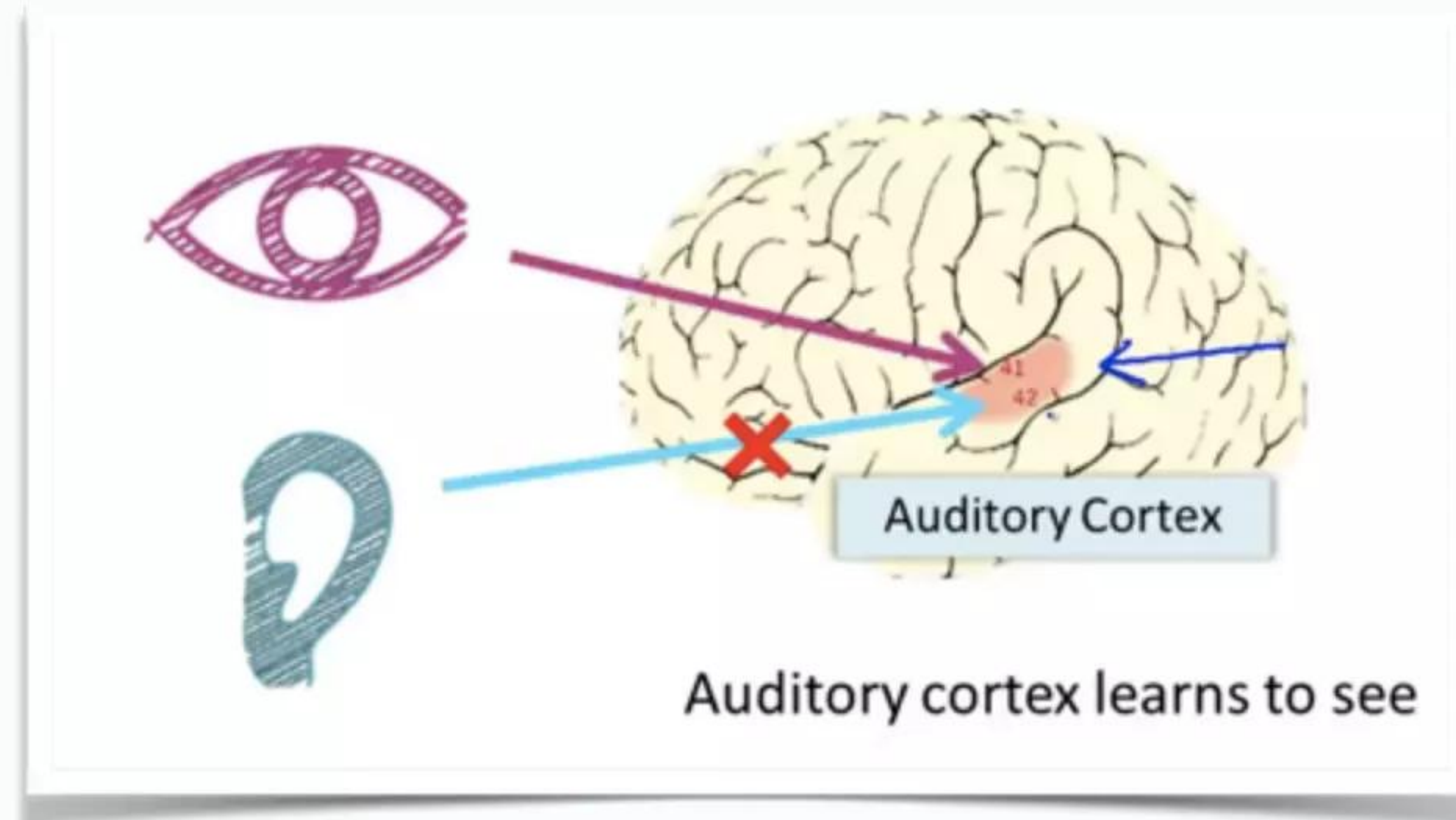
$$J(\theta) = \frac{1}{2m} \left[ \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right]$$

- Update Parameter :

$$\theta_j := \theta_j - \alpha \frac{1}{m} \left[ \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)} + \lambda \sum_{j=1}^n \theta_j \right]$$

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

# Multi Layer Perceptron (Neural Network)



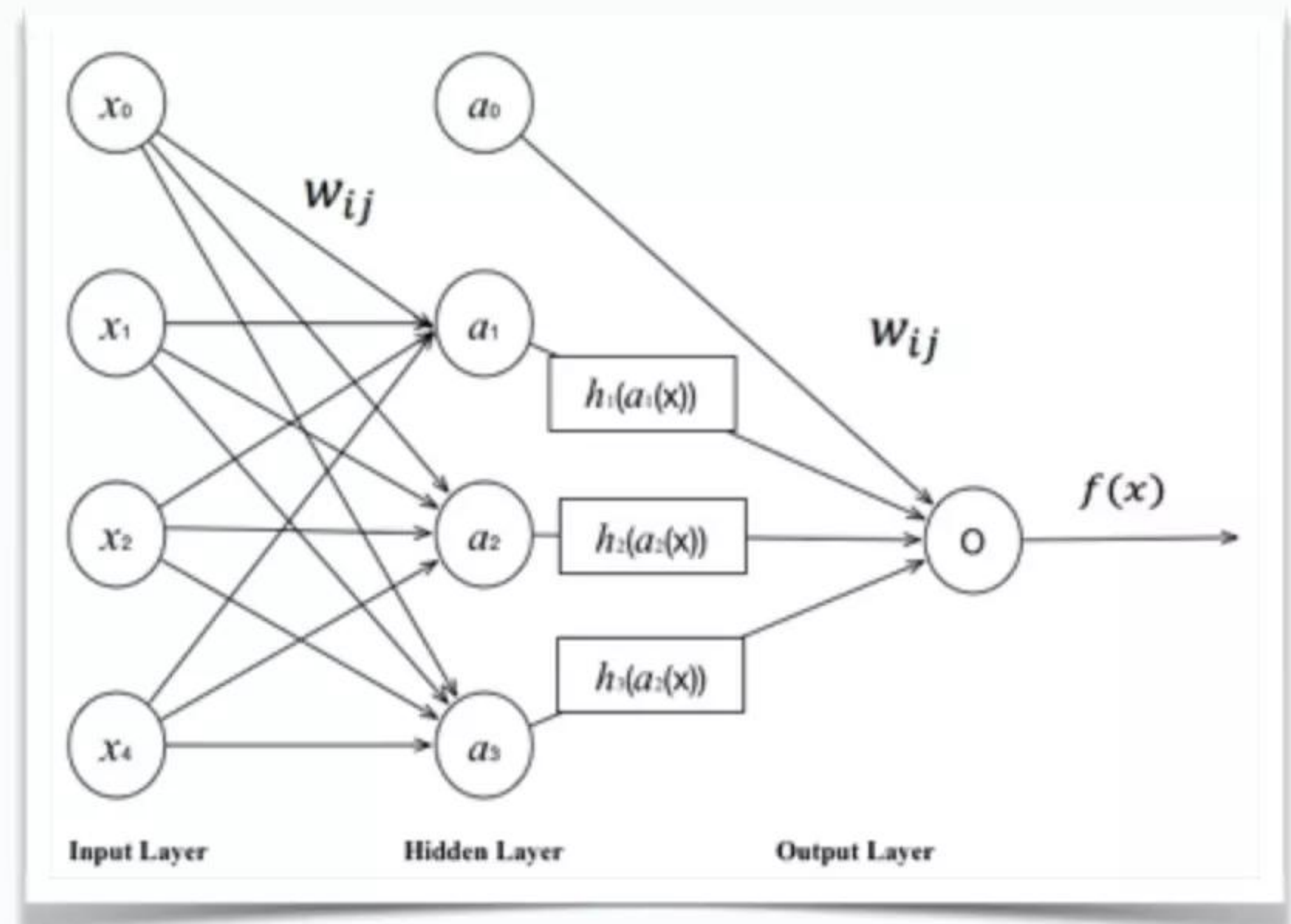
# Multi Layer Perceptron (Neural Network)

$$a(x) = b + \sum_i w_i x_i = b + W^T x$$

$$h(x) = \varphi(a(x)) = \varphi(b + \sum_i w_i x_i)$$

$$\varphi(a(x)) = \frac{1}{1 + e^{-a(x)}}$$

$$f(x) = o(b^{(2)} + W^{(2)T} h^{(1)} x)$$

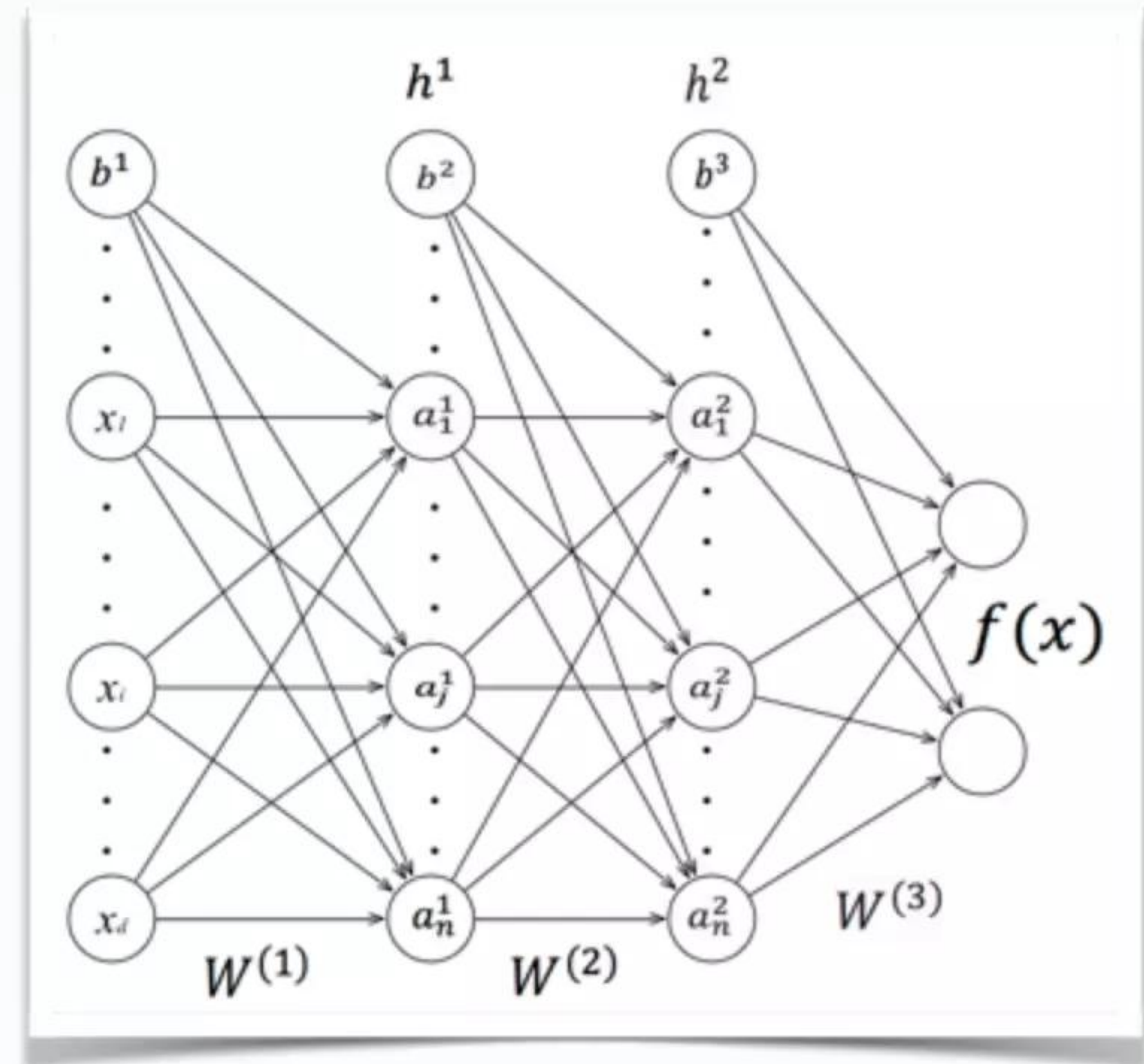


# Multi Layer Perceptron (Neural Network)

$$a^{(k)}(x) = b^{(k)} + W^{(k)}h^{(k-1)}(x)$$

$$h^{(k)}(x) = g(a^{(k)}(x))$$

$$h^{(L+1)}(x) = o(a^{(L+1)}(x)) = f(x)$$



# Cost Function ANN

$$J(\theta) = \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m \text{Cost}(h(x^{(i)}), y^{(i)}) + \lambda \sum_{j=1}^n \theta_j^2$$

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log (h_{\theta}(x^{(i)}))_k + (1 - y_k^{(i)}) \log (1 - (h_{\theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\theta_{ji}^{(l)})^2$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log (h_{\theta}(x^{(i)}))_k$$

$$\text{Cost}(h(x^{(i)}), y^{(i)}) = \sum_{k=1}^K y_k^{(i)} \log (h(x^{(i)}))_k$$

$$= -\log (h(x^{(i)}))_y$$

- Partial Derivative :

$$\frac{\partial}{\partial f(x)_c} - \log f(x)_y = \frac{-1_{(y=c)}}{f(x)_y}$$

- gradient :

$$= \frac{-e^y}{f(x)_y}$$

# Algoritma Lengkap

```
Repeat until convergence {  
    Initialize  $\theta$  ( $\theta \equiv \{W^{(1)}, b^{(1)}, \dots, W^{(L+1)}, b^{(L+1)}\}$ )  
    For N iterations  
        For each training example  $(x^{(t)}, y^{(t)})$   
             $\Delta = -\nabla_{\theta} \text{cost}(f(x^{(t)}; \theta), y^{(t)}) - \lambda \nabla_{\theta} \Omega(\theta)$   
             $\theta \leftarrow \theta + \alpha \Delta$   
    }
```

```
}
```

$$\theta \leftarrow \theta + \alpha \Delta$$

# Backpropagation

Repeat until convergence {

Initialize  $\theta$  ( $\theta \equiv \{W^{(1)}, b^{(1)}, \dots, W^{(L+1)}, b^{(L+1)}\}$ )

For  $i=1$  to  $m$  iterations

Set  $a^{(1)} = x^{(i)}$

Perform a feed forward propagation to compute  $a^{(k)}$  for  $k = 1, 2, 3, \dots, L$

Using  $y^{(i)}$ , compute  $\delta^{(L)} = a^{(L)} - y^{(i)}$

Compute  $\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(1)}$

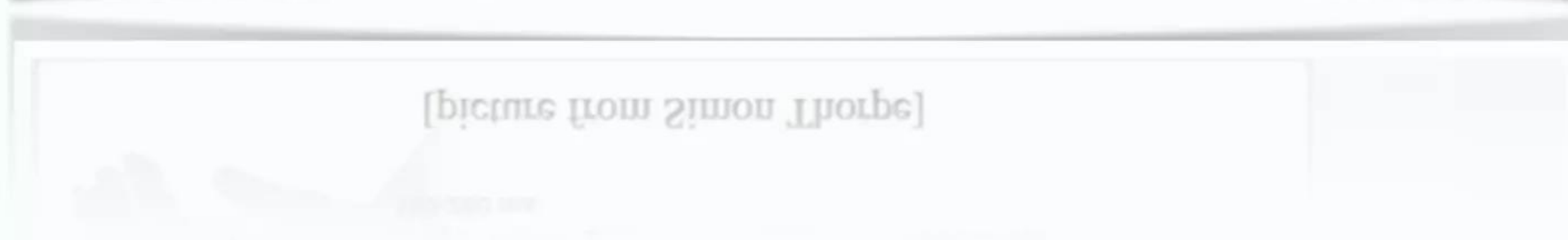
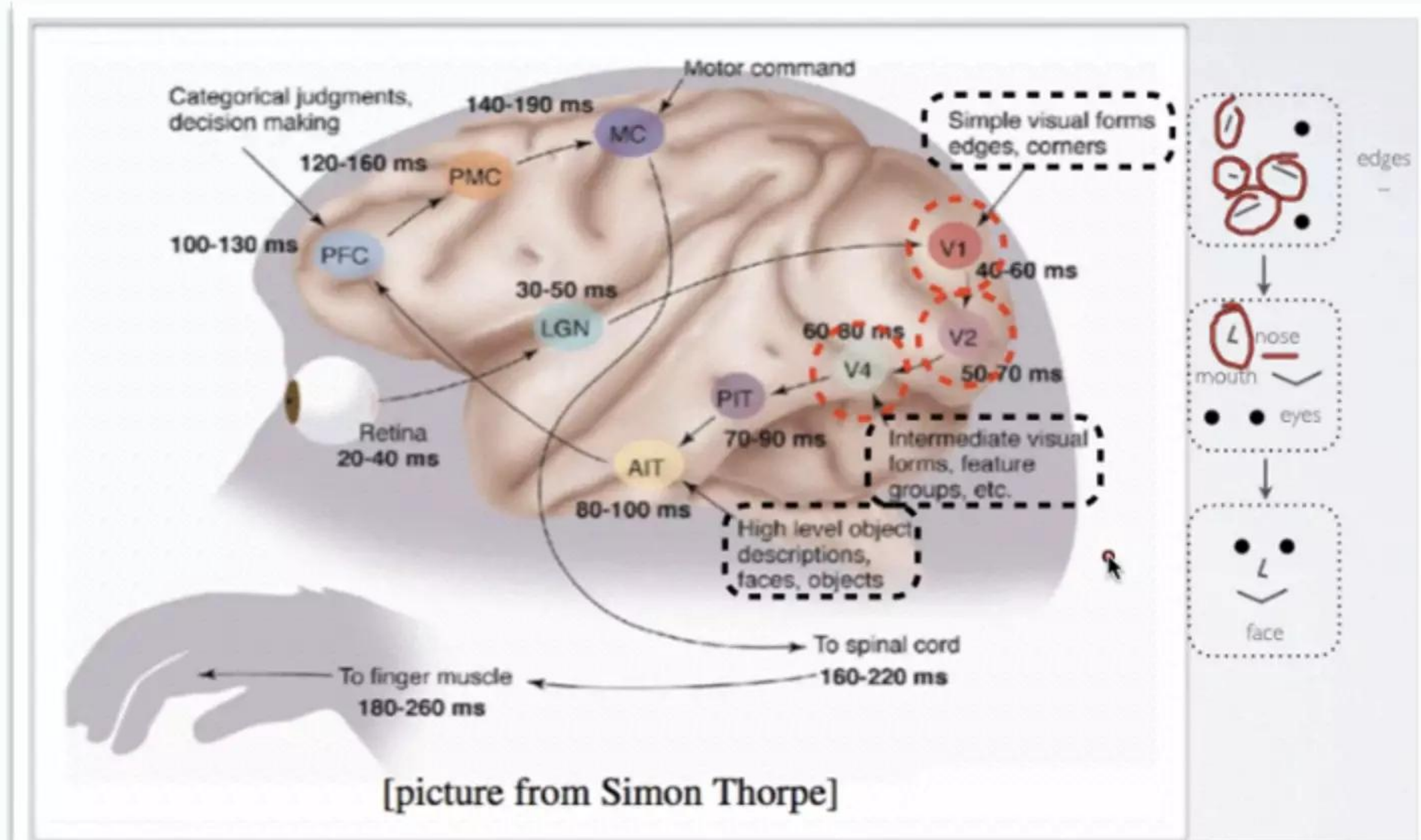
$$\Delta_{ij}^{(k)} := \Delta_{ij}^{(k)} + a_j^{(k)} \delta_i^{(k+1)}$$

}

}

$$\nabla_{(k)}^{(l)} := \nabla_{(k)}^{(l)} + \sigma_{(k)}^{(l)} \delta_{(k+1)}^{(l)}$$

# Deep Learning



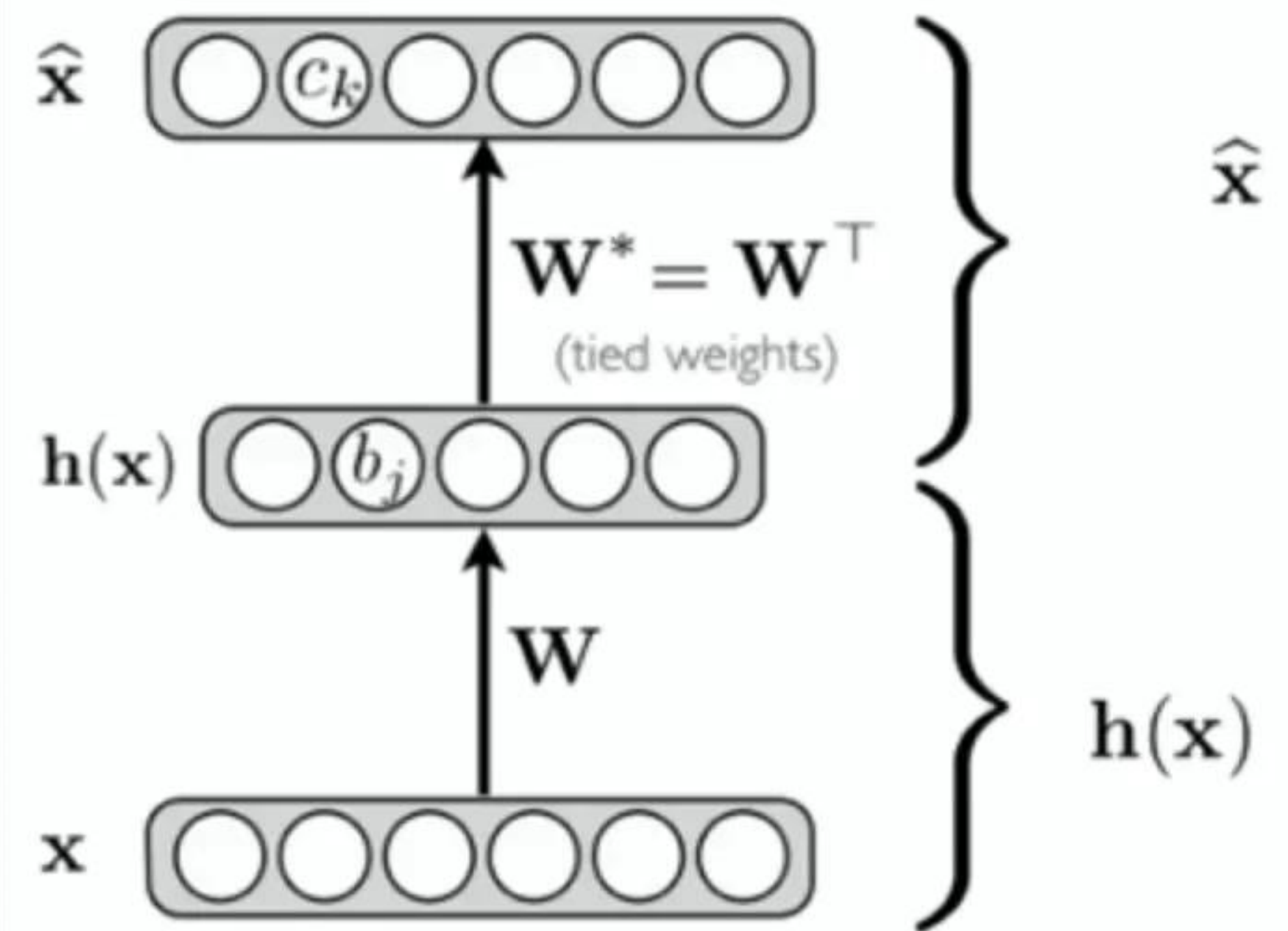
# Autoencoder

Decoder :

$$\begin{aligned}\hat{x} &= o(\hat{a}(x)) \\ &= \text{sigm}(x + W * h(x))\end{aligned}$$

Encoder :

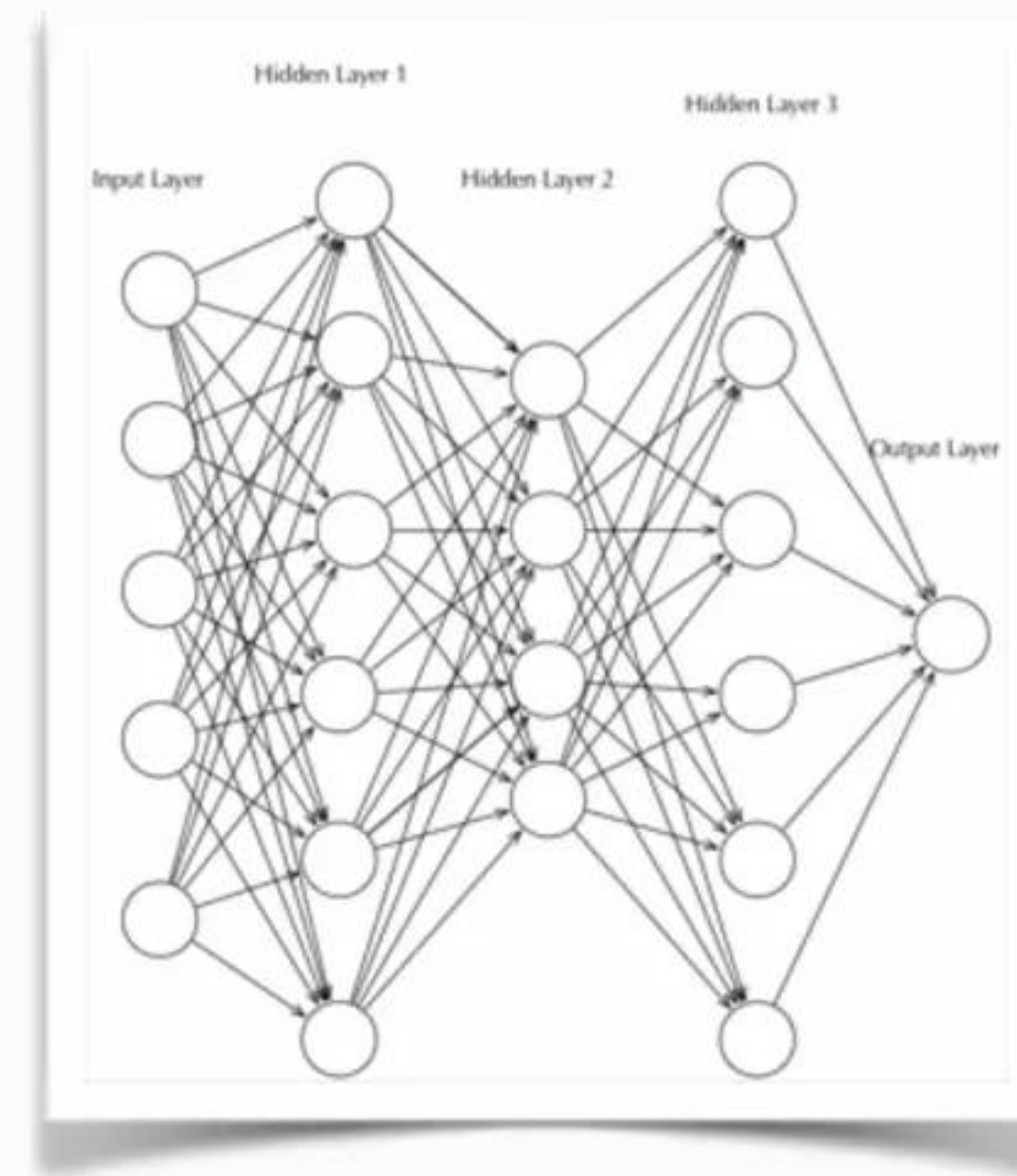
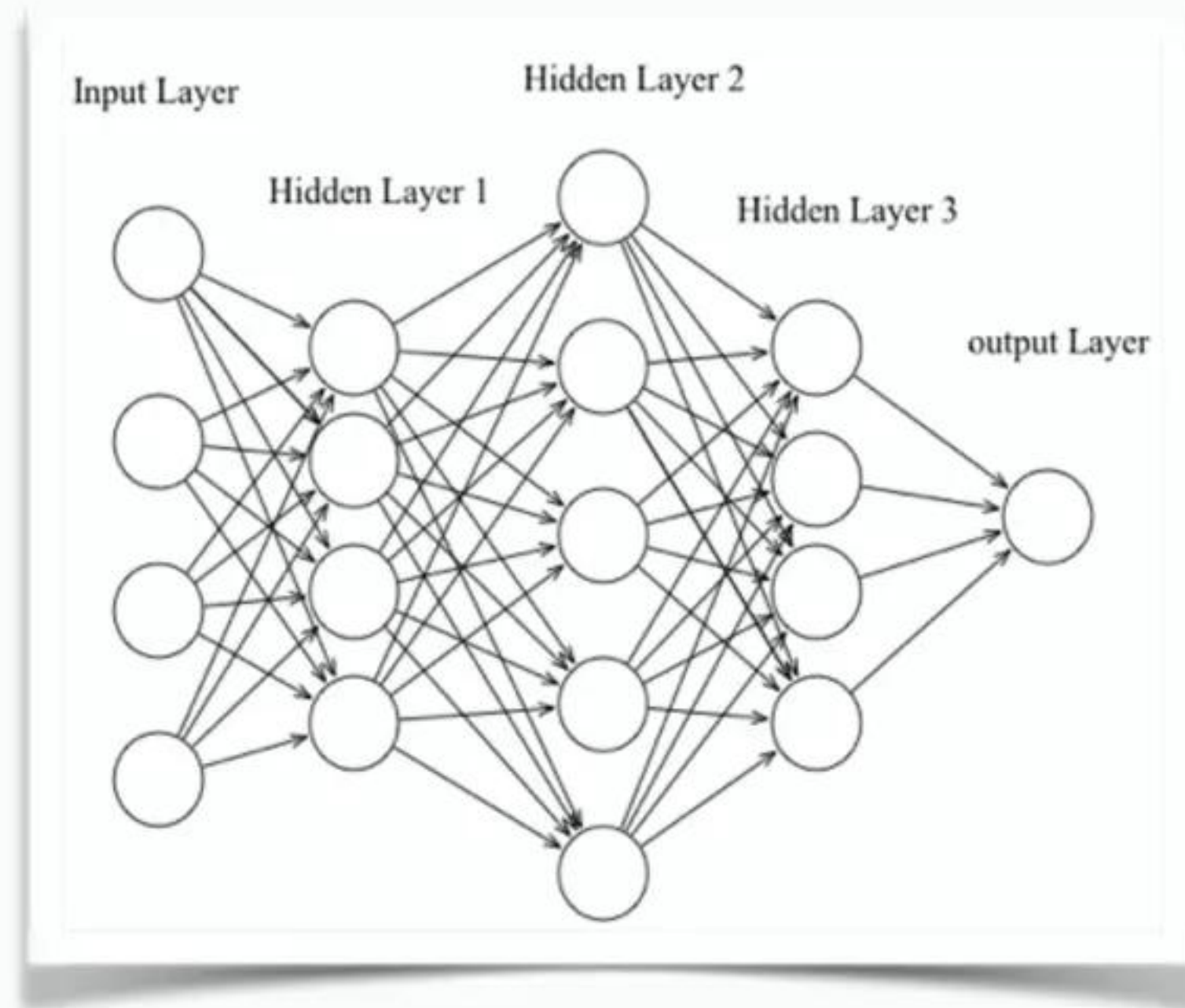
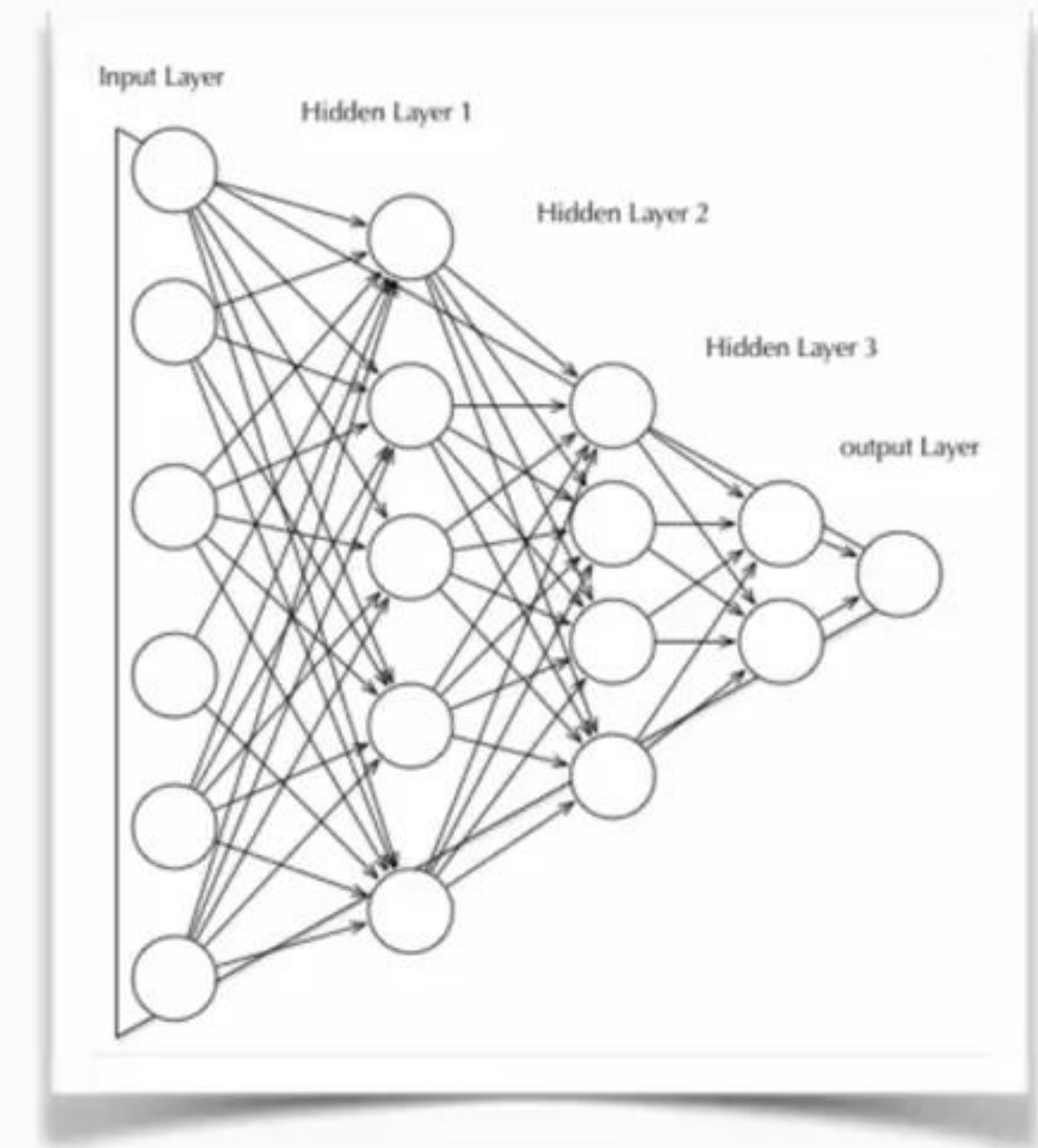
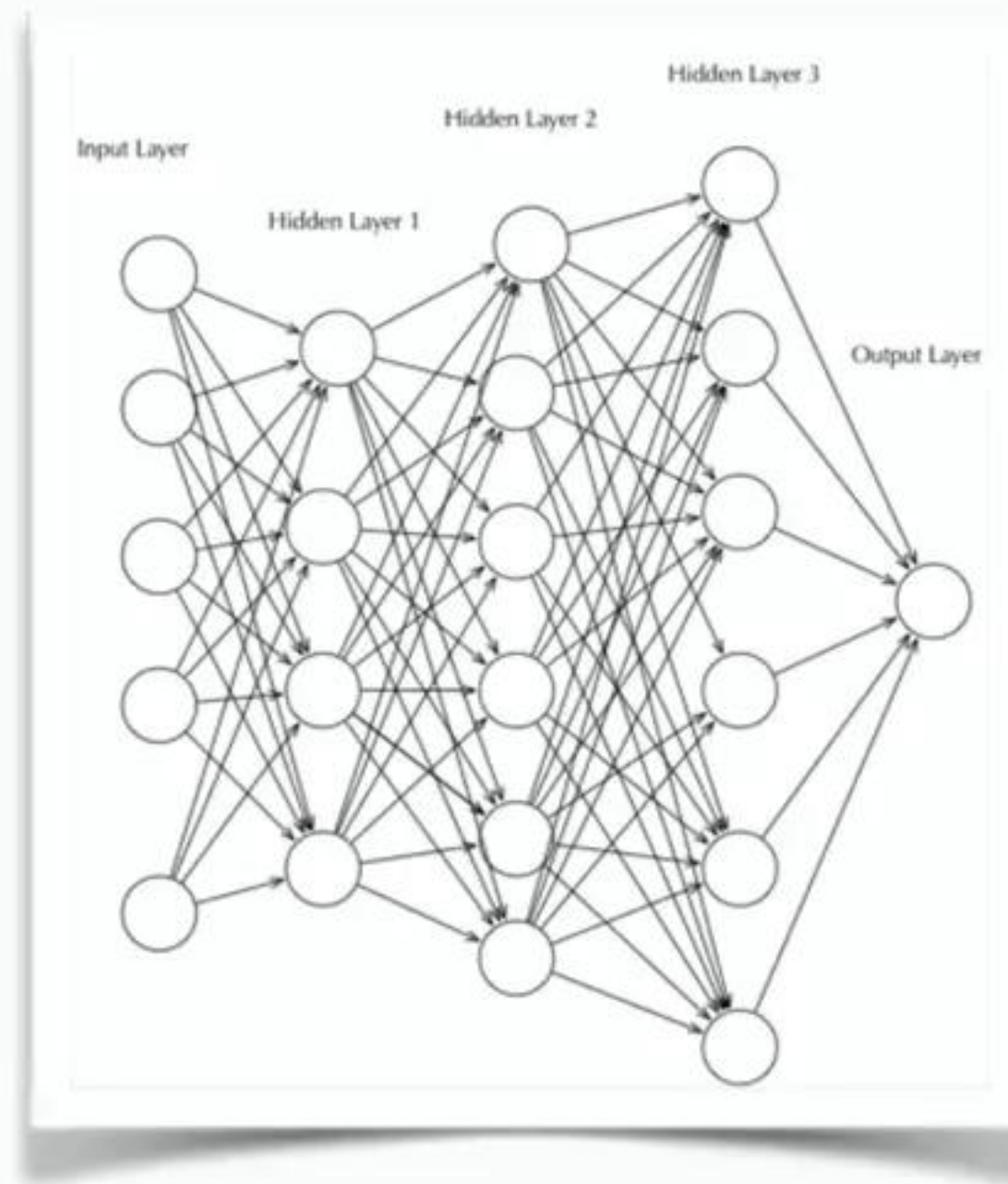
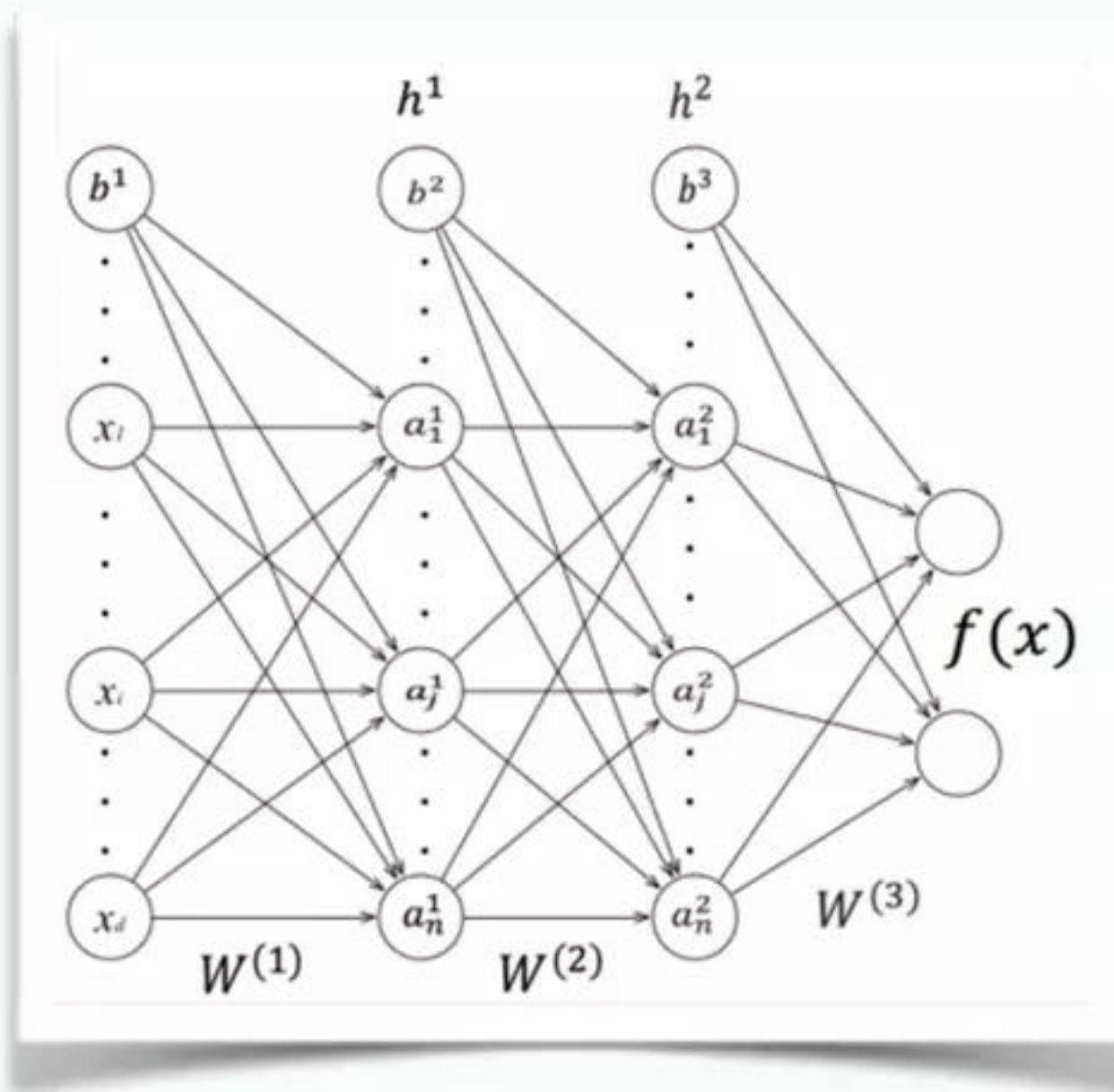
$$\begin{aligned}h(x) &= g(\hat{a}(x)) \\ &= \text{sigm}(b + Wh)\end{aligned}$$



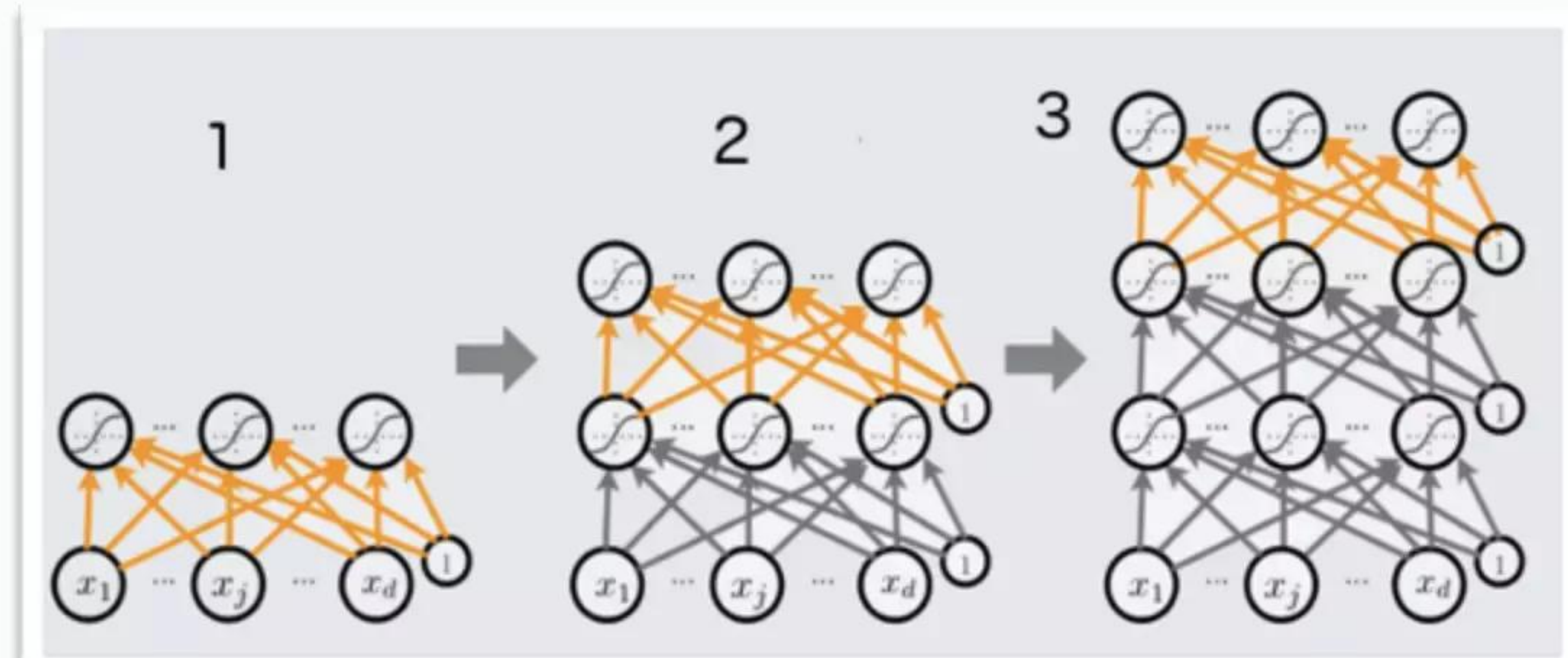
$$\begin{aligned}v(x) &= g(\hat{a}(x)) \\ &= \text{sigm}(b + Wh)\end{aligned}$$



# Deep Neural Network



# Deep Neural Network



# Algoritma Deep Neural Network Secara Penuh

```
//Pre-training
For l=1 to L
    Build unsupervised training set (with  $h^{(0)}(x) = x$ ):
        
$$D = \{h^{(l-1)}(x^{(l)})\}_{l=1}^T$$

    Train autoencoder on  $D$ 
    Use hidden layer weights and biases of greedy
    module to initialize the deep network parameters
     $W^{(l)}, b^{(l)}$ 

Initialize  $W^{(L+1)}, b^{(L+1)}$  randomly (as usual)

//train backpropagation (supervised) with stochastic
//gradient descent
//training
Repeat until convergence {
    For i=1 to m iterations
        Set  $a^{(1)} = x^{(i)}$ 
        Perform a feed forward propagation to compute
         $a^{(k)}$  for  $k = 1, 2, 3, \dots, L$ 
        Using  $y^{(i)}$ , compute  $\delta^{(L)} = a^{(L)} - y^{(i)}$ 
        Compute  $\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(1)}$ 
        
$$\Delta_{ij}^{(k)} := \Delta_{ij}^{(k)} + a_j^{(k)} \delta_i^{(k+1)}$$

    }
}
```

# Data Laporan Bank Umum Konvensional

The screenshot shows an Excel spreadsheet with columns labeled: NOUR\_NO, Q\_BANK, NAMA\_BANK, TAHUN, BULAN, Q\_LAPORAN, NO\_LURUT, KOMPONEN, KOMPONEN\_NOMINAL, FORMAS, Q\_LAPORAN, and JENIS\_FORM. The data rows list various banks like PT BANK RAKYAT INDONESIA (PERSERO) Tbk. and their financial components such as Aktiva, Kas, Penempatan pada Bank Indonesia, Giro Bank Indonesia, and Surat Berharga yang Dimiliki.

```
> db.getCollectionNames();  
[  
  "bank_dataset",  
  "bank_report_2_dataset",  
  "bank_report_3_dataset",  
  "bank_report_3_dataset_keys",  
  "bank_report_dataset",  
  "bank_training_dataset",  
  "datasetrepricing_gap_1",  
  "datasetrepricing_gap_2",  
  "datasetrepricing_gap_3",  
  "my_collection_keys",  
  "repricing_gap",  
  "system.indexes",  
  "system.js"  
]  
  
> db.bank_dataset.count();  
4064212  
>
```

- Data laporan perbankan 11 tahun dari 103 bank umum konvensional.
- Total distinct features by name : 651 features.
- Dengan fitur yang memiliki missing value dibawah 20%-30% : 10 fitur.

# Data Laporan Bank Umum Konvensional

Key	Value	Type
▼ (1) Objectid("55c2fc9bd4c611ffb6d99cb9")	{ 12 fields }	Object
_id	Objectid("55c2fc9bd4c611ffb6d99cb9")	Objectid
join_date	2008-12-01 00:00:00.000Z	Date
▶ komitmen_kontinjensi	{ 35 fields }	Object
▶ kewajiban_asing	{ 0 fields }	Object
# year	2008	Int64
▶ kewajiban	{ 39 fields }	Object
▶ neraca	{ 106 fields }	Object
▶ kualitas_aktiva	{ 25 fields }	Object
" " bank_name	PT BANK RAKYAT INDONESIA (PERSERO) Tbk.	String
# bank_id	2	Int64
▶ laba_rugi_saldo	{ 41 fields }	Object
# month	12	Int64

Total rows dataset RAW : 4064212

Setelah ekstrak berdasarkan waktu dan bank : 19140

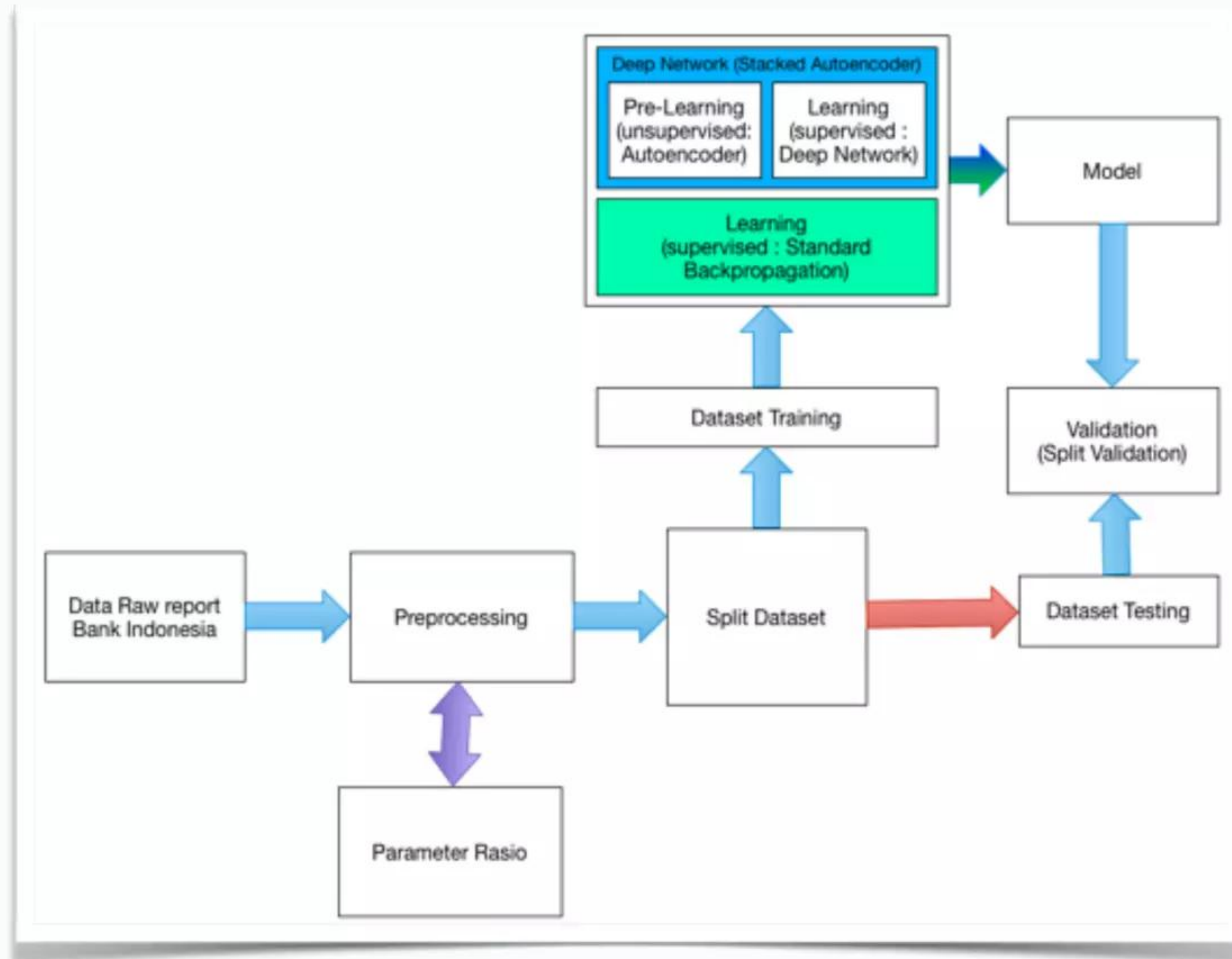
# Preprocessing

- The preprocessing that we used was to calculate some variables in economics for equity, liabilities, asset and the ration. The following is variables that used to construct the dataset (features):
- Illiquid Assets, Illiquid Liabilities, LTR, Giro, Savings, Deposit, DPK, CASA, Core Deposits, Credit, Financing GAP, Total Assets, ATMR, RWA, CAR, Total Equity, EQTA, Earning and Loss for the year, ROA, ROE, LRP, LLR, Operation Cost, Total Income, CIR, Int Rev, Int Cost, Int Rev (ANN), Repricing Gap (estimation target).
- Total fitur : 30 fitur + 2 fitur (waktu dan kode bank).

# Dataset Construction

- Dataset divide to be 2 kind of dataset:
  - Dataset 1 : Using quarterly periodic to estimate the repricing gap.
  - Dataset 2 : Using semester periodic to estimate the repricing gap.
- So for dataset 1 will adding 3 more units (neuron/perceptron/features) for quarterly calculation repricing gap estimation, while for dataset 2 adding 6 more units (neuron/perceptron/features) for semester calculation repricing gap estimation.

# Experiment Flow



# Standard Parameters

- Iterasi Maks : 1000
- Minimum MSE :  $10^{-9}$  (for SB and DNN) and  $10^{-5}$  (for autoencoder)
- Learning Rate : 0.75 (for SB and DNN) and  $10^{-5}$  (for autoencoder)
- Momentum : 0.5 (for SB and DNN) and 0.75 (for autoencoder)
- Activation Function : Sigmoid Function

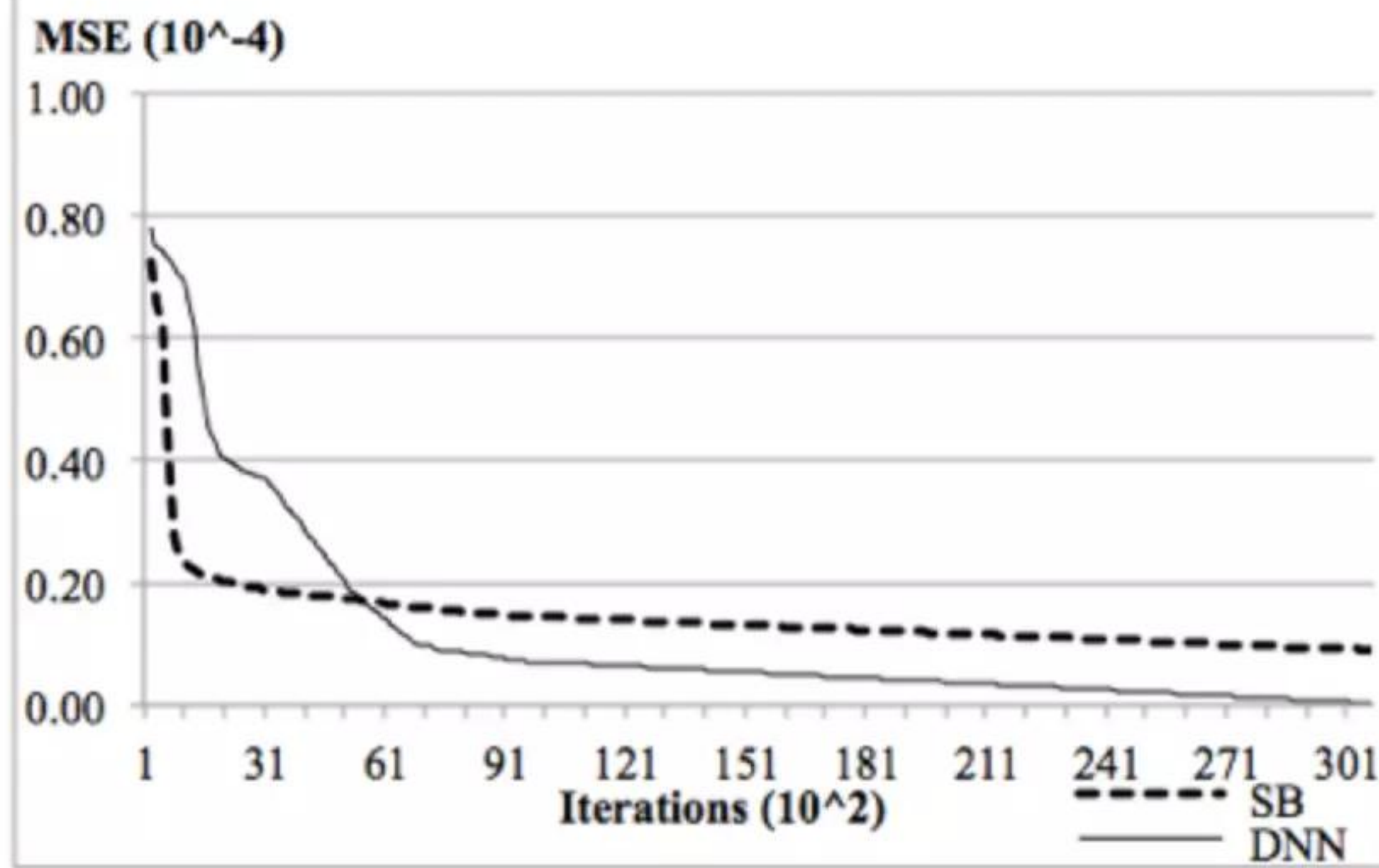
# Standard Parameters

- Backpropagation
  - Iterasi Maks : 1000
  - Minimum MSE :  $10^{-9}$
  - Learning Rate : 0.75
  - Momentum : 0.5
  - Topologi Network : Single Hidden Layer
  - Activation Function : Sigmoid Function

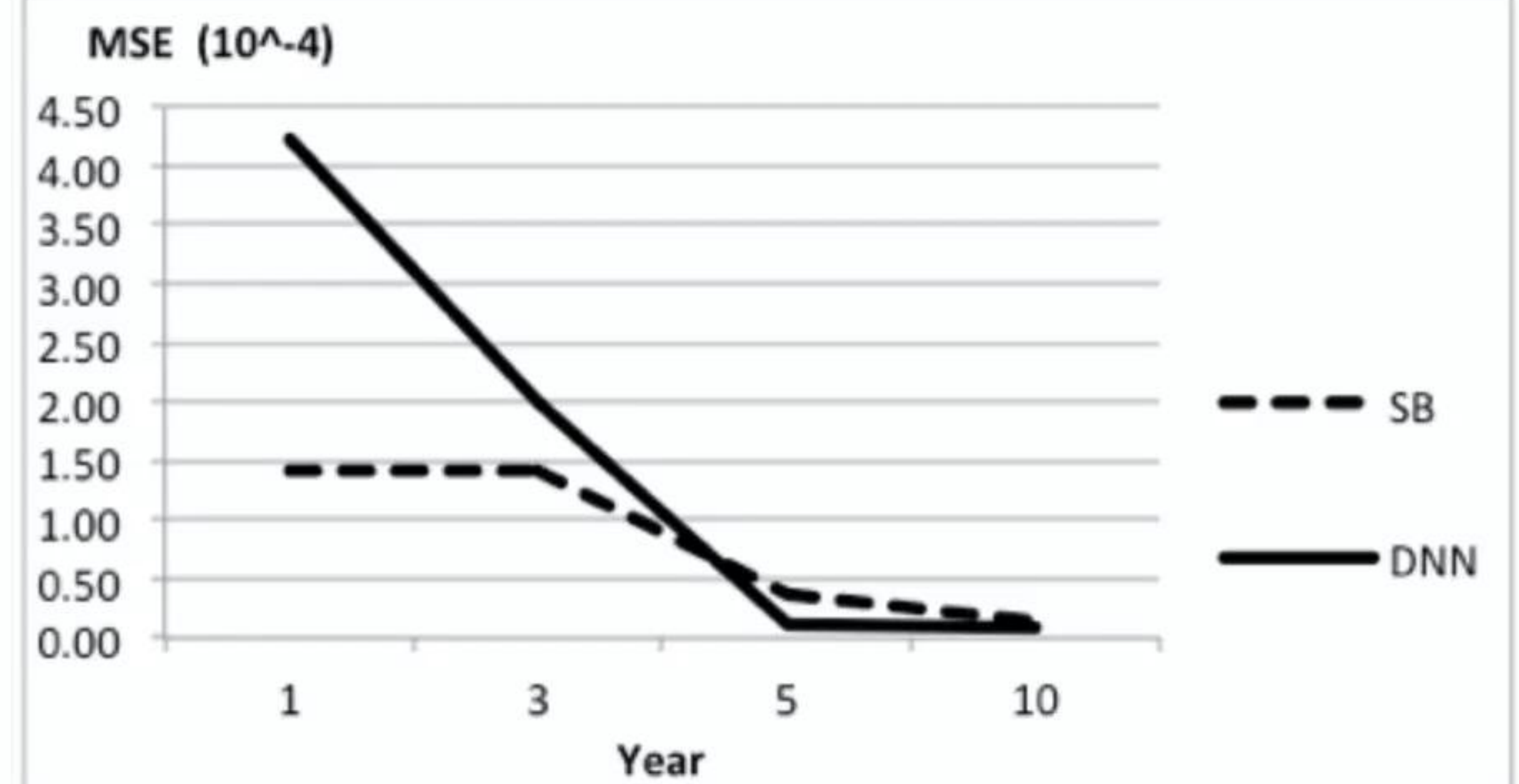
# Experiments phase 1

**Table 1** Experiment using 5 different network topology using DNN with 1000 iterations

No	Architecture	Hidden Layer	MSE ( $10^{-4}$ )
1	bow ties	26,30,26	0.70
2	diamond	26,30,35	0.69
3	increment	35,30,26	0.70
4	decrement	30,26,30	0.70
5	<b>Equal</b>	<b>26,26,26</b>	<b>0.62</b>



**Figure 9** Convergence Experiment of SB algorithm and DNN



**Figure 10** MSE Convergence experiment SB and DNN algorithm using some number of data condition, 1, 3, 5 and 10 years data.

# Conclusion and Discussion

- DNN has better performance than SB handle datasets with missing value and condition of non-linearity is very high.
- On the use of a small number of iterations DNN has a convergence speed slower than SB, at the same iteration condition. But when using larger number of iterations, the SB algorithm is slowing down the speed of convergence, more slowly than the DNN, wherein the DNN consistently pressed MSE value better than SB.
- The main factor of better performance results of DNN is the amount of gradient greater with larger architectural and also the topology of the network is good enough for the case, with sustained an initial value of each gradient is better, where the value is the knowledge of the results of the algorithm autoencoder, thus helping maintain the value gradient keep moving convergent and more consistently suppress the error value.
- Further development for use DNN in predicting repricing gap for conventional bank data in Indonesia is use larger iterations will give better MSE (smaller), setting the network topology more complex or the network architecture, and the implementation of pre-learning algorithms such as restricted Boltzmann machine .

Thank you