Neural Network

Multi-Layer Perceptron

2018.08.17

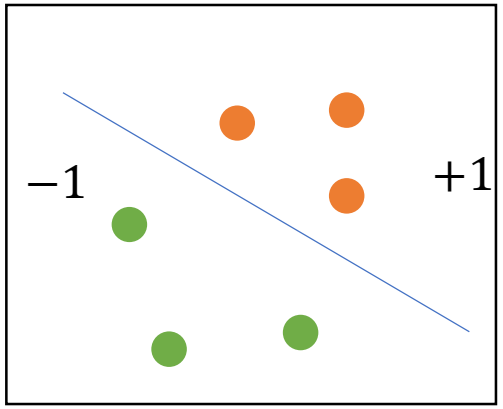
Wonbin Kim

Introduction

Perceptron

- Originally appeared to solve the two-class model.

$$y = f(\boldsymbol{W}^T x + \boldsymbol{b}), \qquad f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0. \end{cases}$$

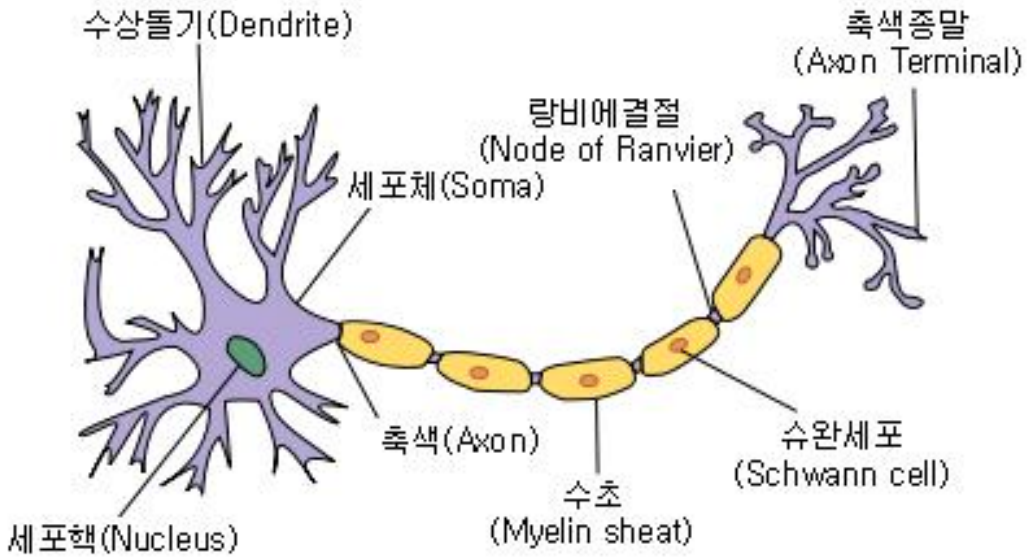


- Since the gradient is zero almost everywhere, So alternative error function is considered, perceptron criterion.

$$\mathbb{E}_p(w) = -\sum_{n \in \mathcal{M}} w^T x_n y_n$$

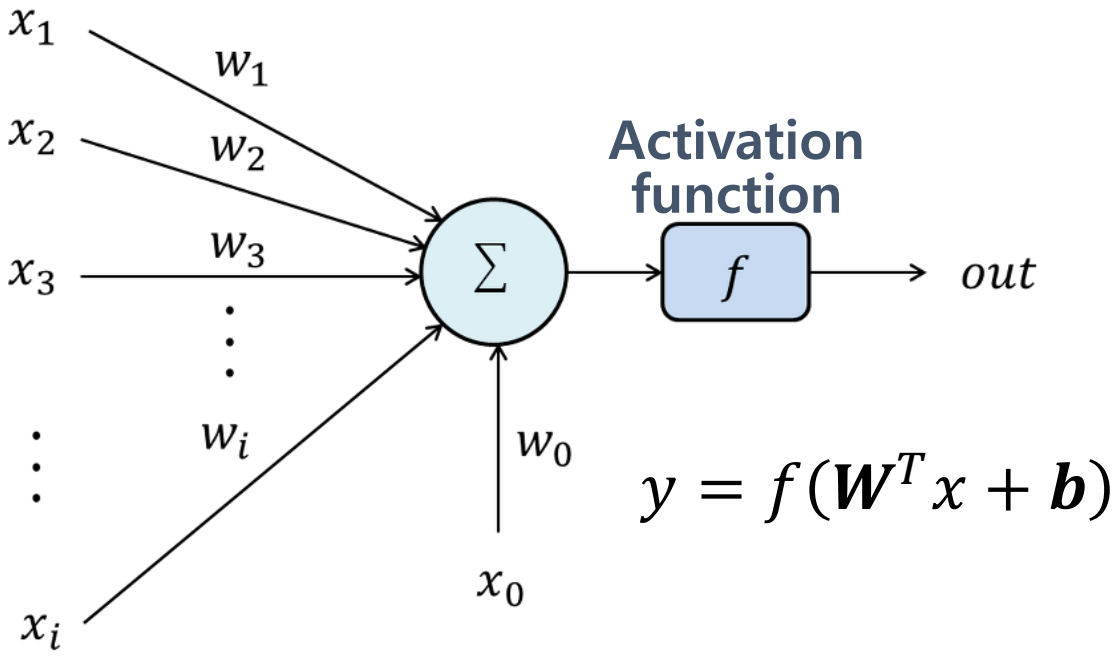
Biological motivation

Biological Neuron



[Figure 1]

Mathematical Model



Simple neuron model as a Linear classifier

From Regression…

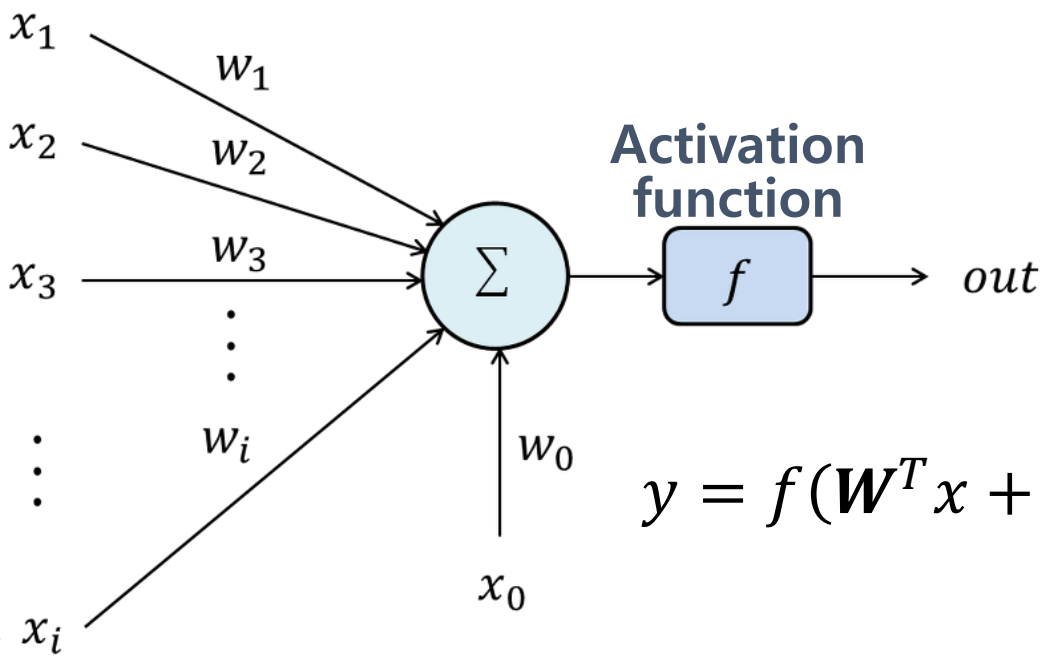
- $X \in \mathbb{R}^{n \times d}$: Data input.

Linear Regression

- The model predict target value y by a linear classifier with $\boldsymbol{W}$ and $\boldsymbol{b}$.

Logistic Regression

- Our model predict classification scores $\boldsymbol{y}$ by linear classifier with $\boldsymbol{W}$ and $\boldsymbol{b}$, following softmax function.



$$y = f(\boldsymbol{W}^T x + \boldsymbol{b})$$

Activation

- Non-Linearity
- Normalizing
- Boundary

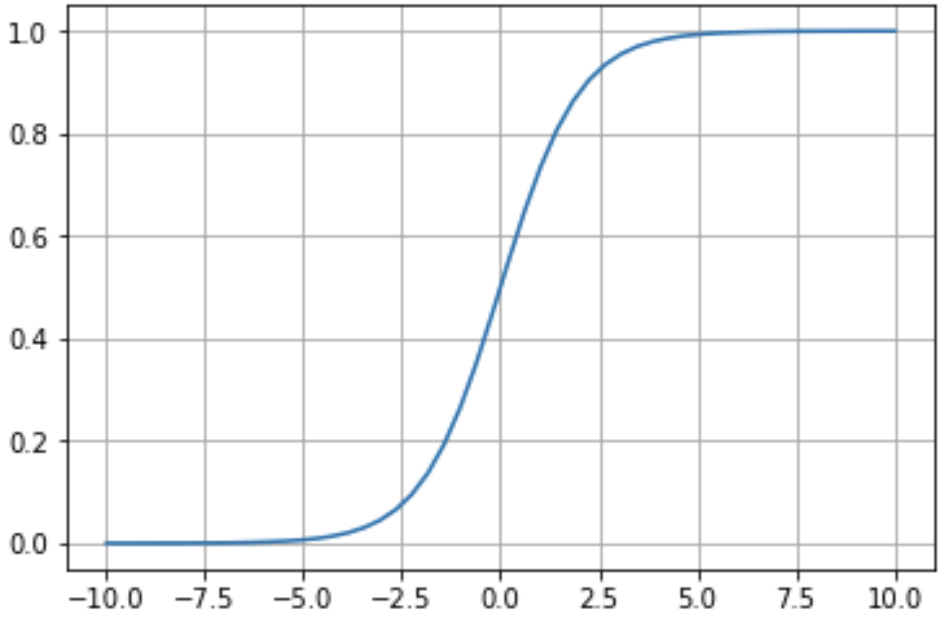
Ex) Sigmoid, Tanh, ReLU, Leaky ReLU, ELU, etc.

Activation – Sigmoid function

Sigmoid function σ

- Mathematical form :

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



- A real-valued number is squashed into [0,1].

Drawbacks

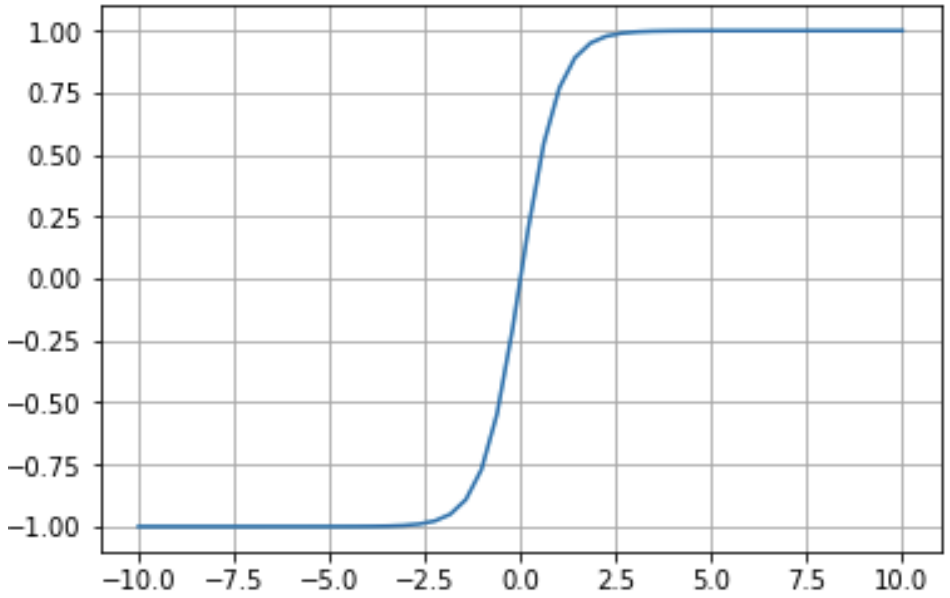
- Vanishing Gradient
- Not-zero centered output

[Implementation.1]

Activation – Tanh (Hyperbolic Tangent)

- Mathematical form :

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = 2\sigma(2x) - 1$$



- A real-valued number is squashed into [-1,1].

Drawbacks

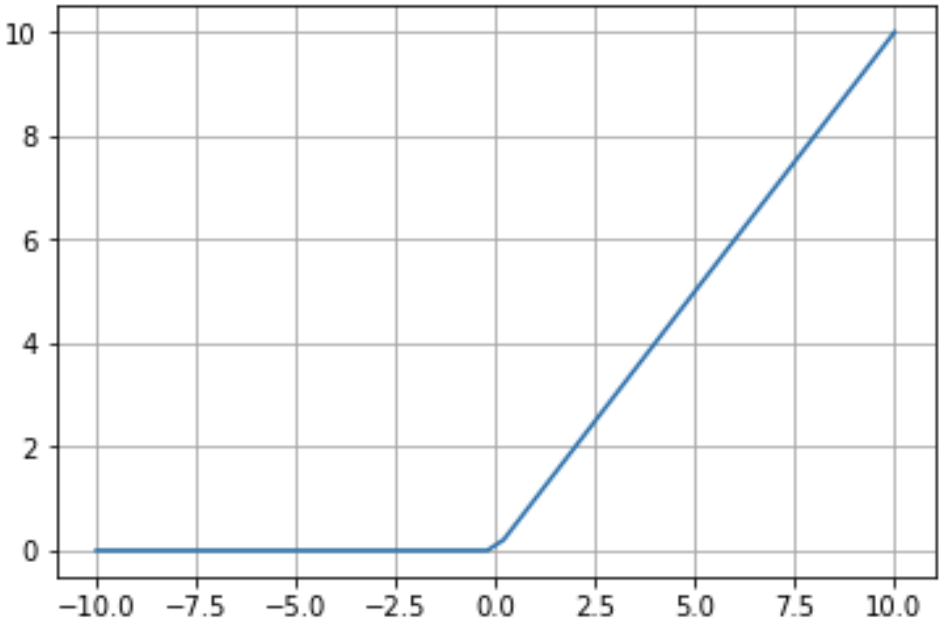
- Vanishing Gradient

[Implementation.2]

Activation – ReLU (Rectified Linear Unit)

- Mathematical form :

$$\text{relu}(x) = \max(x, 0) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$



- A negative number is squashed into 0.

Drawbacks

- Gradient flooding

[Implementation.3]

Activation – Leaky ReLU

- Mathematical form :

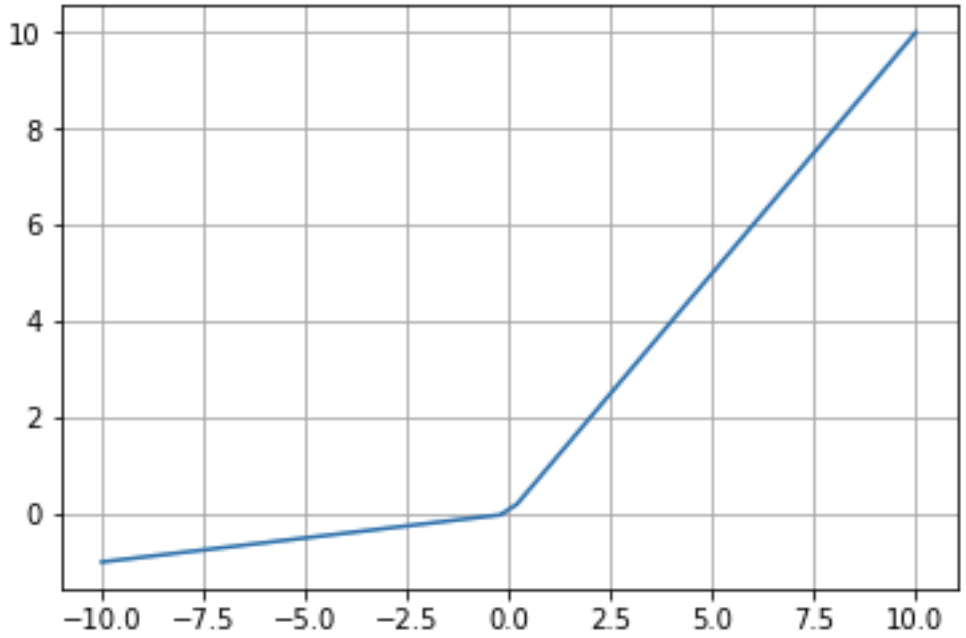
$$\text{lrelu}(x) = \max(\text{x}, \alpha x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases}$$

- A negative number is considered.

Drawbacks

- Gradient flooding

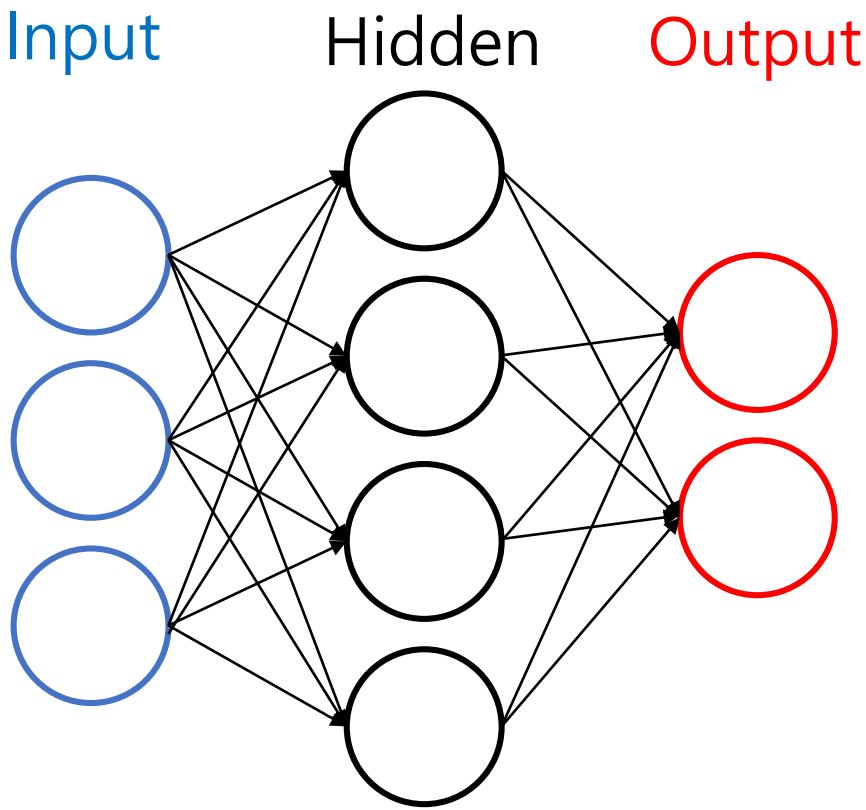
[Implementation.4]



Architecture

Architecture

- Acyclic Graph
- These architectures called as Fully-Connected Layer, Artificial Neural Networks or Multi-Layer Perceptrons.
- A neural network which have n-1 hidden layers called **N-Layer** Neural Network



Ex) 2-Layer Neural Network

Size of neural networks

- Let l and L denote the order of layers and the number of layers, respectively. i.e., input layer is the 0^{th} layer and output layer is the L^{th} layer.
- Let $n(l)$ denotes the number of node of the l^{th} layer. Then
- The number of weight of the neural network is written as

$$\sum_{l=1}^{L} n(l-1) * n(l)$$

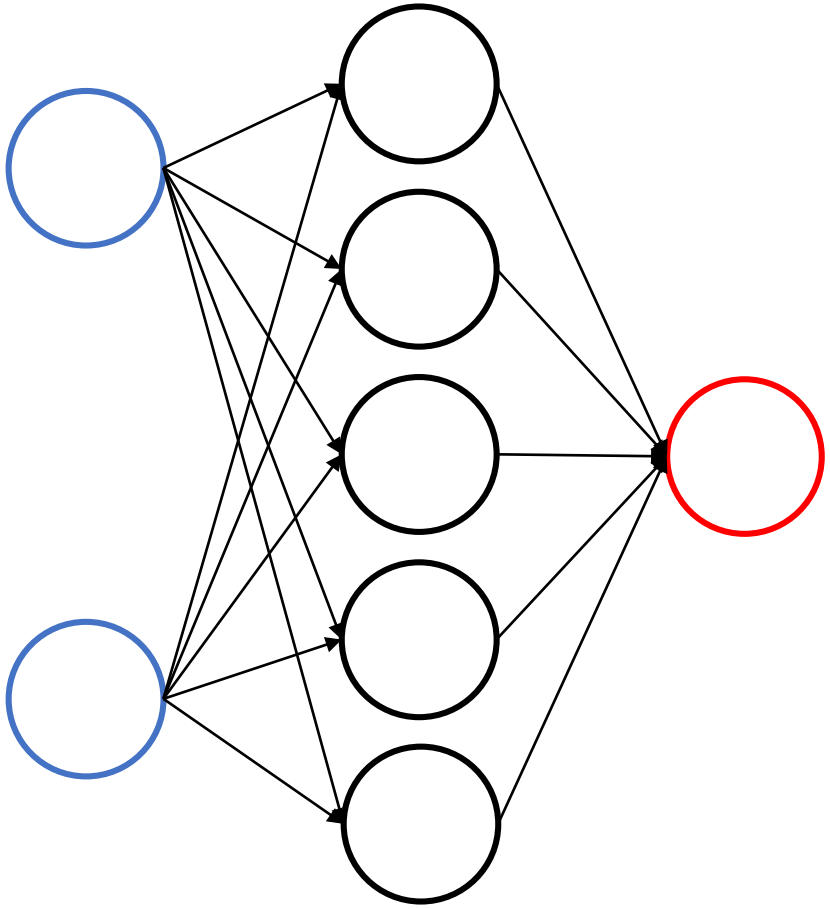
- The number of bias of the neural network is written as

$$\sum_{l=1}^{L-1} n(l+1)$$

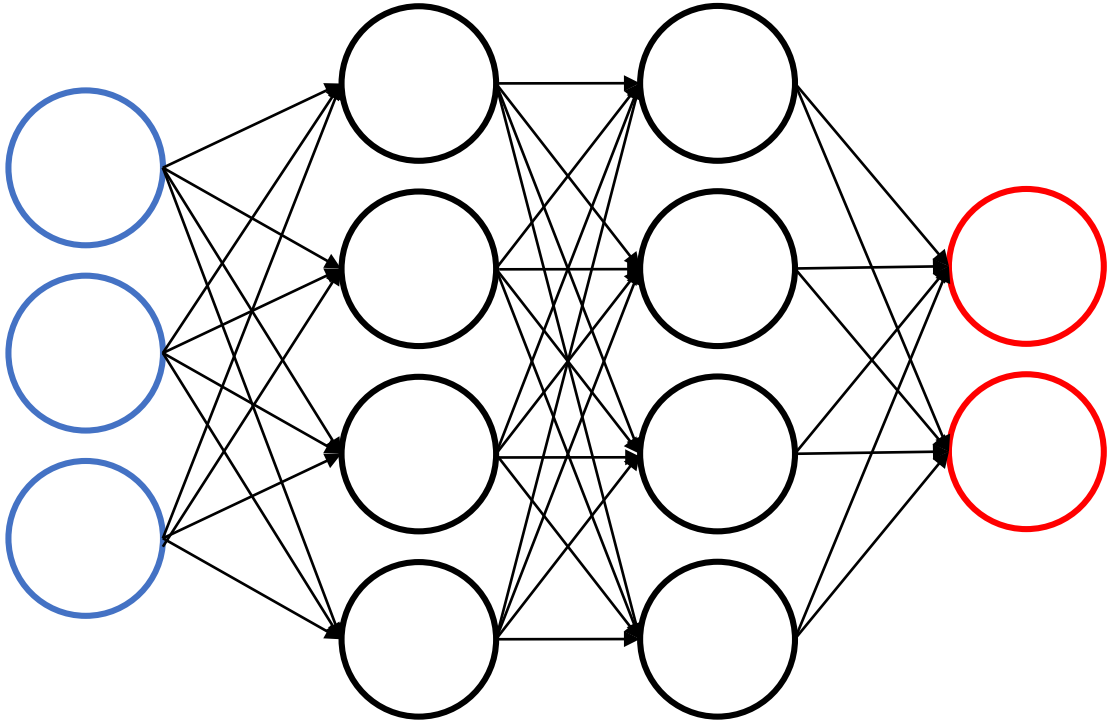
Quiz?

- How many weights and biases are in the networks below?

[1]

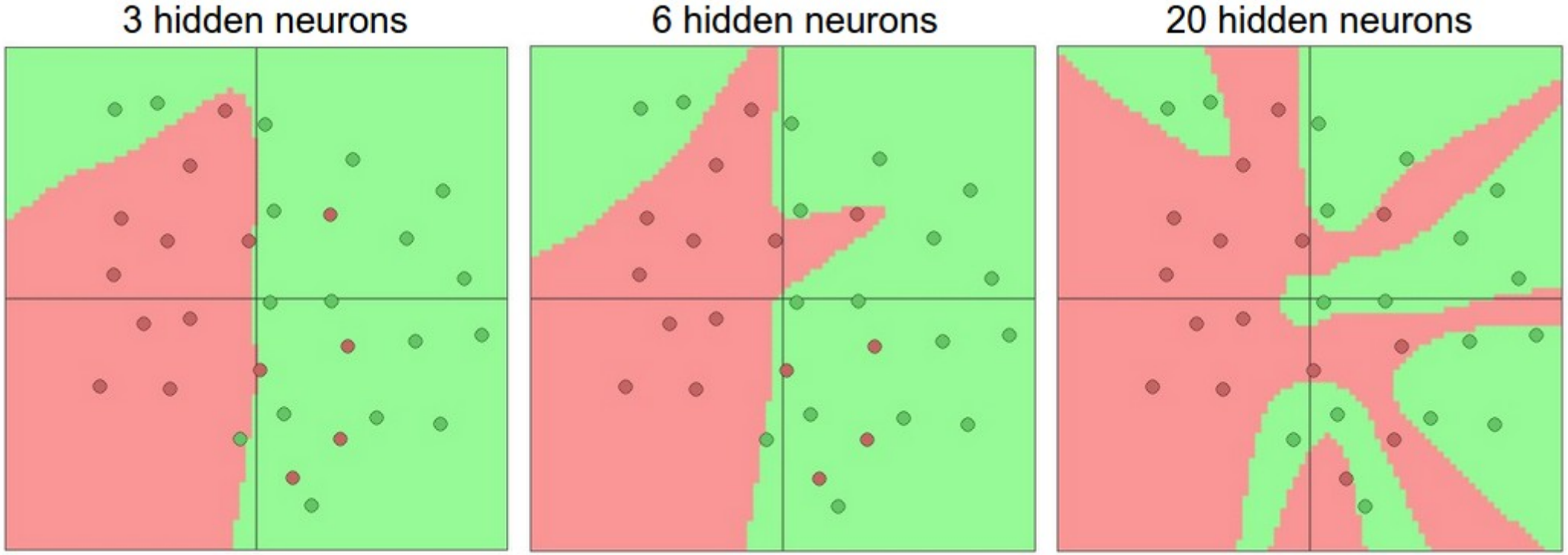


[2]



Representation power and configuration

- How many weights and biases are appropriate?



[Figure 2]

Data Preprocessing

Basic – Mean subtraction

- To remove some bias from data, subtract the mean of training data across individual feature in data.
- Depending on the data type, you can select the mean value calculation method.
- E.g. The mean of RGB images can be computed by each channel.

[Implementation.5]

Basic - Normalization

In order to fit data dimensions to the same scale, normalizing it. There is two common methods for normalization.

- Dividing each dimension by its standard deviation

- Scaling so that the min and max along the dimension is -1 and 1 respectively.

[Implementation.6]

Basic – PCA and Whitening

In order to decorrelate or whiten, using PCA.

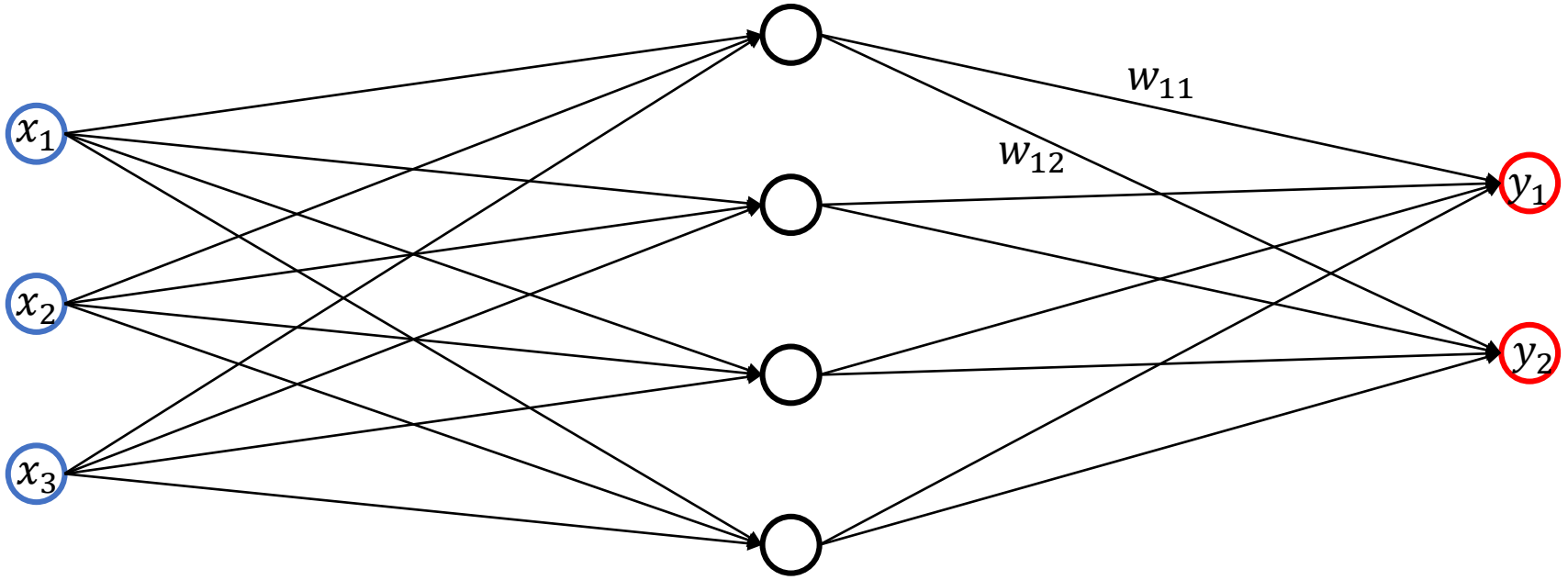
- First, Decorrelate data by eigen-vector of the covariance matrix.
- Second, Whitening data by singular-value of the covariance

[Implementation.7]

Initialization

Simple way - Initialized by zero

- Let's start from zero.



- What are the expected values of y_1 and y_2?
- What are the expected values of w_{11} and w_{12}, after first update occur?

[Implementation.8]

Initialized by small random numbers

- Initialize weights by random value to prevent correlating between parameters. (i.e. *symmetry breaking*.)

[Implementation.9]

- If the number of input of the layer is large?

- Consider the inner product s= $\sum_i^n w_i x_i$, where w and x is zero mean and activation function is ReLU. And n denotes the number of input node.

Scaling random initialization value.

$$Var(S) = Var\left(\sum_i^n w_i x_i\right) = \sum_i^n Var(w_i x_i)$$

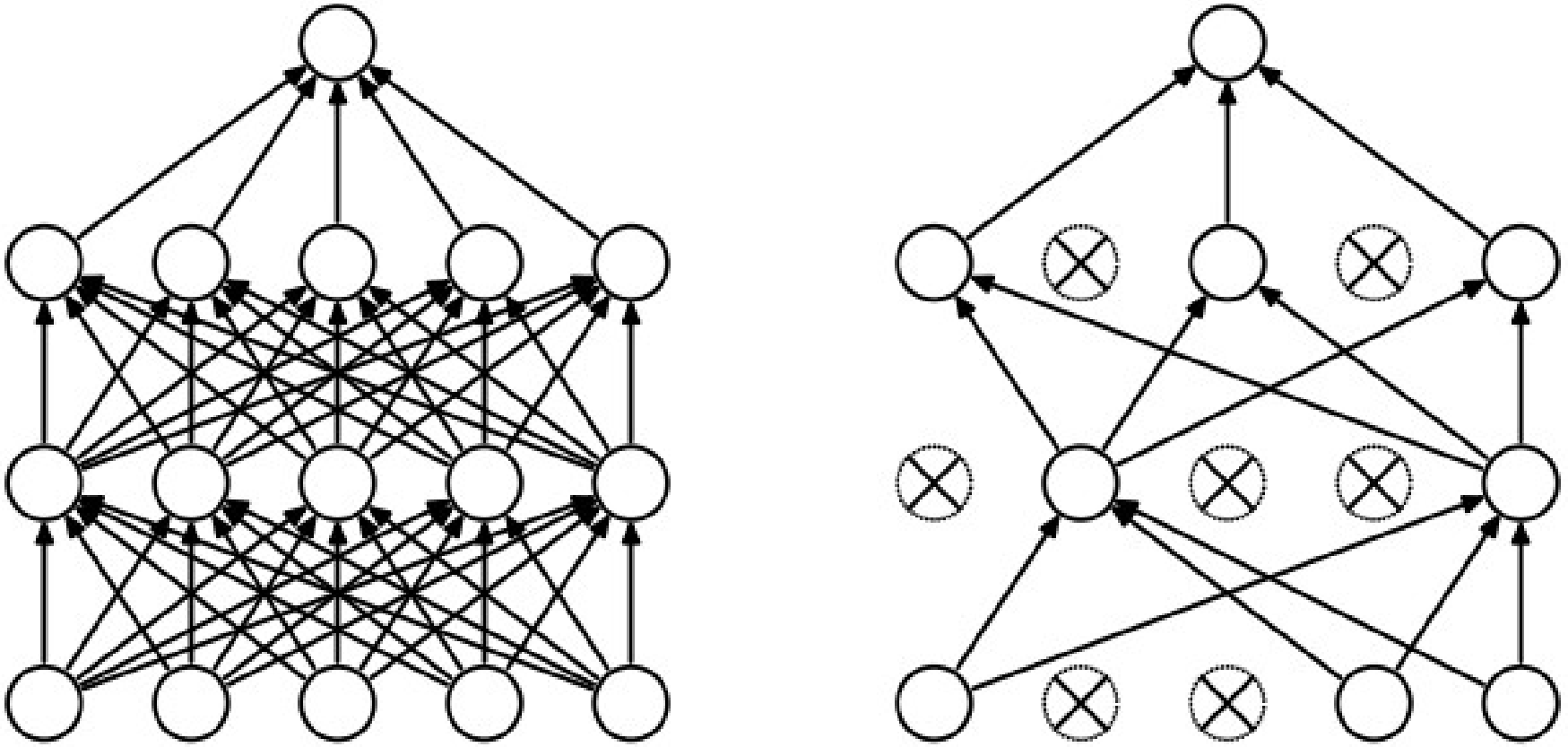
$$= \sum_i^n [E(w_i)]^2 Var(x_i) + E[(x_i)]^2 Var(w_i) + Var(x_i)Var(w_i)$$

$$= \sum_i^n Var(x_i)Var(w_i) = \big(n \cdot Var(w)\big)Var(x)$$

- We can choose $1/n$ (Xavier Initialization) as scaling factor so that the variance of our initialization become the variance of input.
- Likewise, Scaling Factor $2/n$ [2] will perform better, in according to the author, because the scaled variance takes account into negative values although ReLU considers only positive values.

Regularization

DropOut[3]



(a) Standard Neural Net

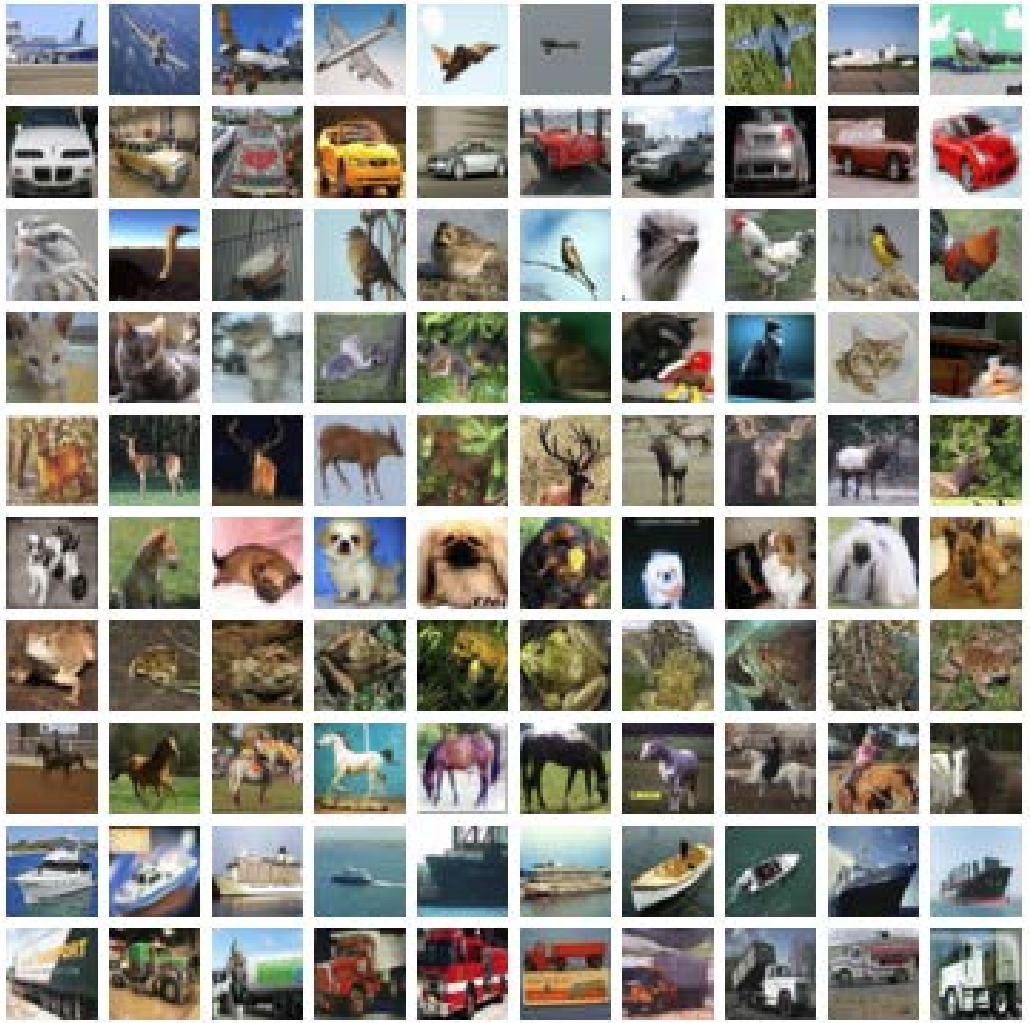
(b) After applying dropout.

Multi-Layer Perceptron

First Practice with MNIST

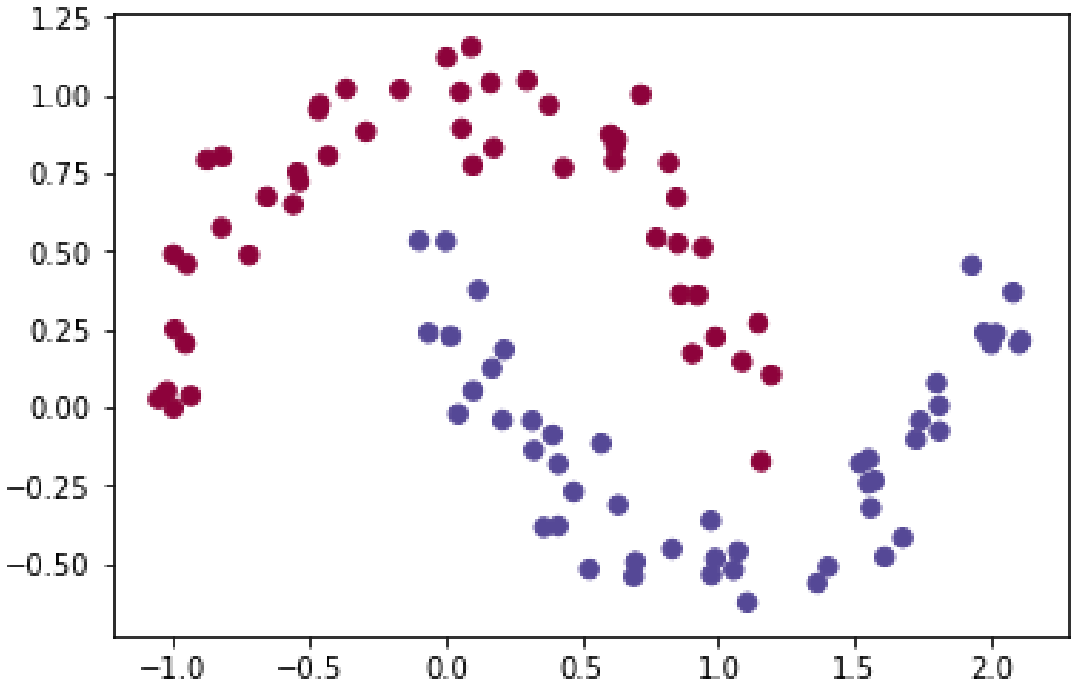
- Hand-written digits
- It has 60,000 grayscale images for training set. And 10,000 grayscale images for test set.
- The size of a image is (28,28)

Second Practice with CIFAR-10



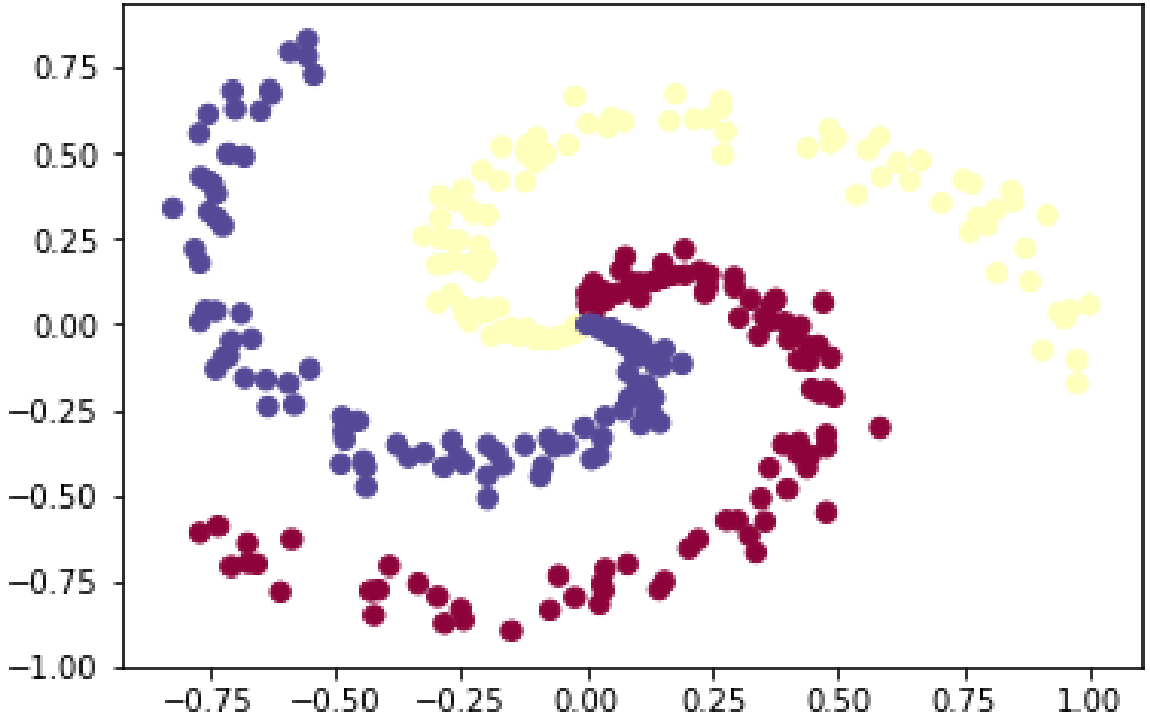
- 10 classes small images
- It has 60,000 colour images for training set. And 10,000 colour images for test set.
- The size of a image is (32,32,3)

Third Practice with complex toy data



Two moon Data

Spiral Data



Thank!

Reference

- [1] Figure.1 : http://www.kormedi.com/dictionary/Medical/View.aspx?idx=5820
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. "*Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*", arxiv 1502.01852
- [3] Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan Salakhutdinov. "*Dropout: A Simple Way to Prevent Neural Networks from Overfitting*", JMLR, 2014, 15, 1929-1958.
- [4] Figure.2 : http://cs231n.github.io/neural-networks-1/