with tensorflow

구현을 위한 딥러닝

▌ 고려대학교 물리학과 한승희

▌ mod96@naver.com

Contents

1. Types of Problems
2. Theoretical Background
3. Bias – Variance Tradeoff
4. About data
5. Implementation

Contents

1 Types of Problems

Types of Problems - 1

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning

Supervised Learning

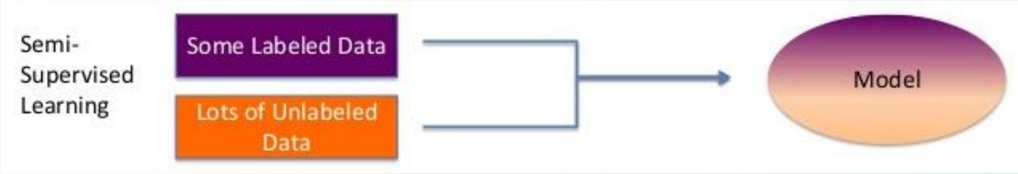
- Majority of algorithms. Machine is trained using well-labeled data. (inputs and outputs are matched)
- Ex> Classification, Regression

Unsupervised Learning

- Learning happens without supervision. Only inputs are used to create a model.
- Ex> Clustering

Semi - Supervised Learning

❖ Some data is labeled, some not. Since clean, perfectly labeled datasets aren't easy to come by, good for real world data.

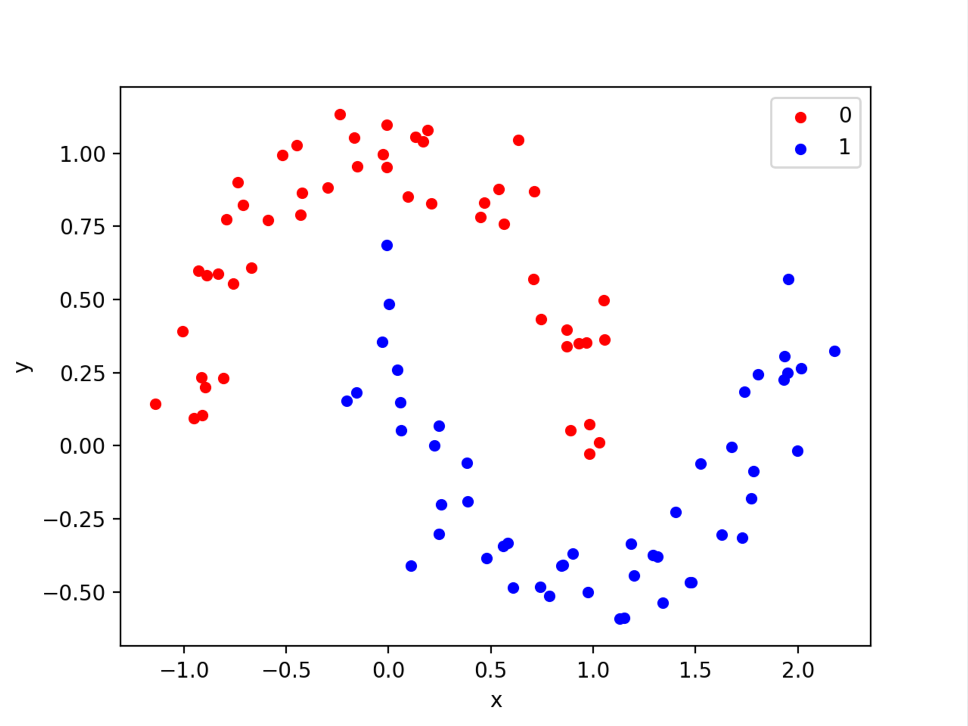


Types of Problems - 2

- Classification
- Regression
- Clustering
- Etc.

Classification

- Predicts discrete number of values. The data is categorized under different labels



Classification

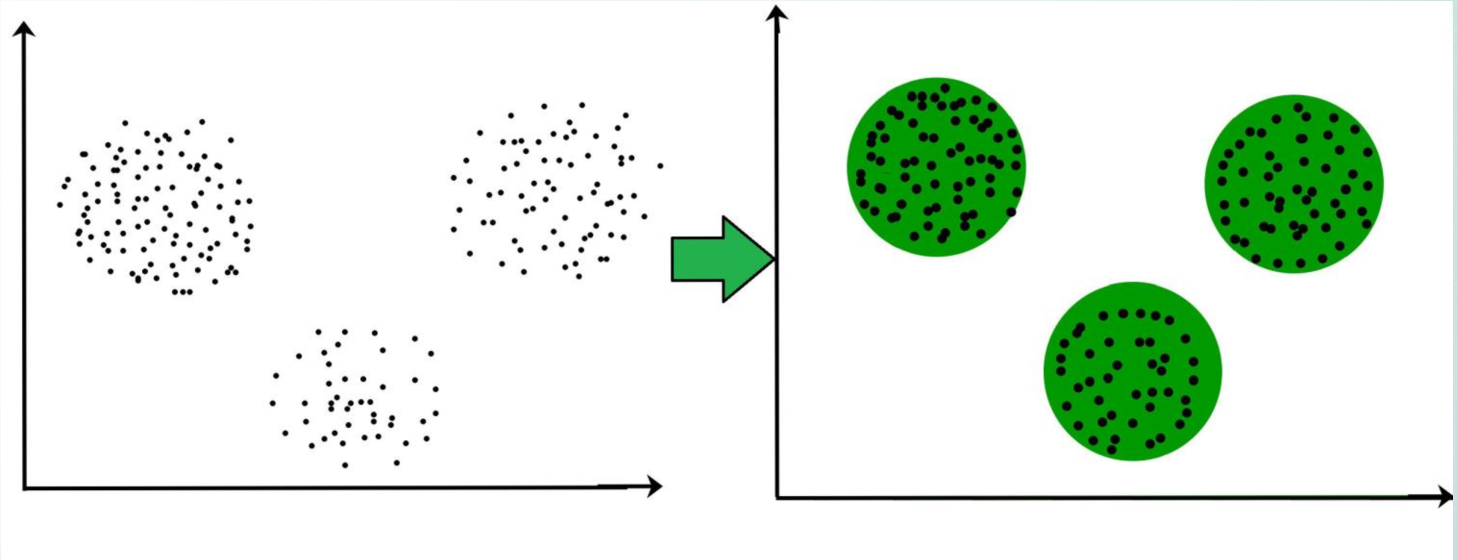
Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.3	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
5.5	2.3	4.0	1.3	*I. versicolor*
6.5	2.8	4.6	1.5	*I. versicolor*
5.7	2.8	4.5	1.3	*I. versicolor*
6.3	3.3	4.7	1.6	*I. versicolor*
4.9	2.4	3.3	1.0	*I. versicolor*
6.6	2.9	4.6	1.3	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*
6.5	3.0	5.8	2.2	*I. virginica*
7.6	3.0	6.6	2.1	*I. virginica*
4.9	2.5	4.5	1.7	*I. virginica*

Regression

- Predicts continuous values output. Mostly analysis using statistical model which is used to predict the numeric data instead of labels.

Clustering

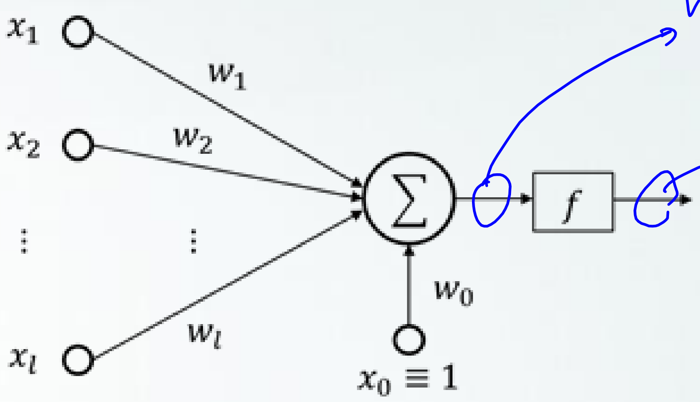
- Task of partitioning the dataset into groups, called clusters. This splits the data in such a way that points within single cluster are very similar and points in different are different.



Contents

2 Theoretical Background

Perceptron (Neuron)



$w_0 x_0 +$

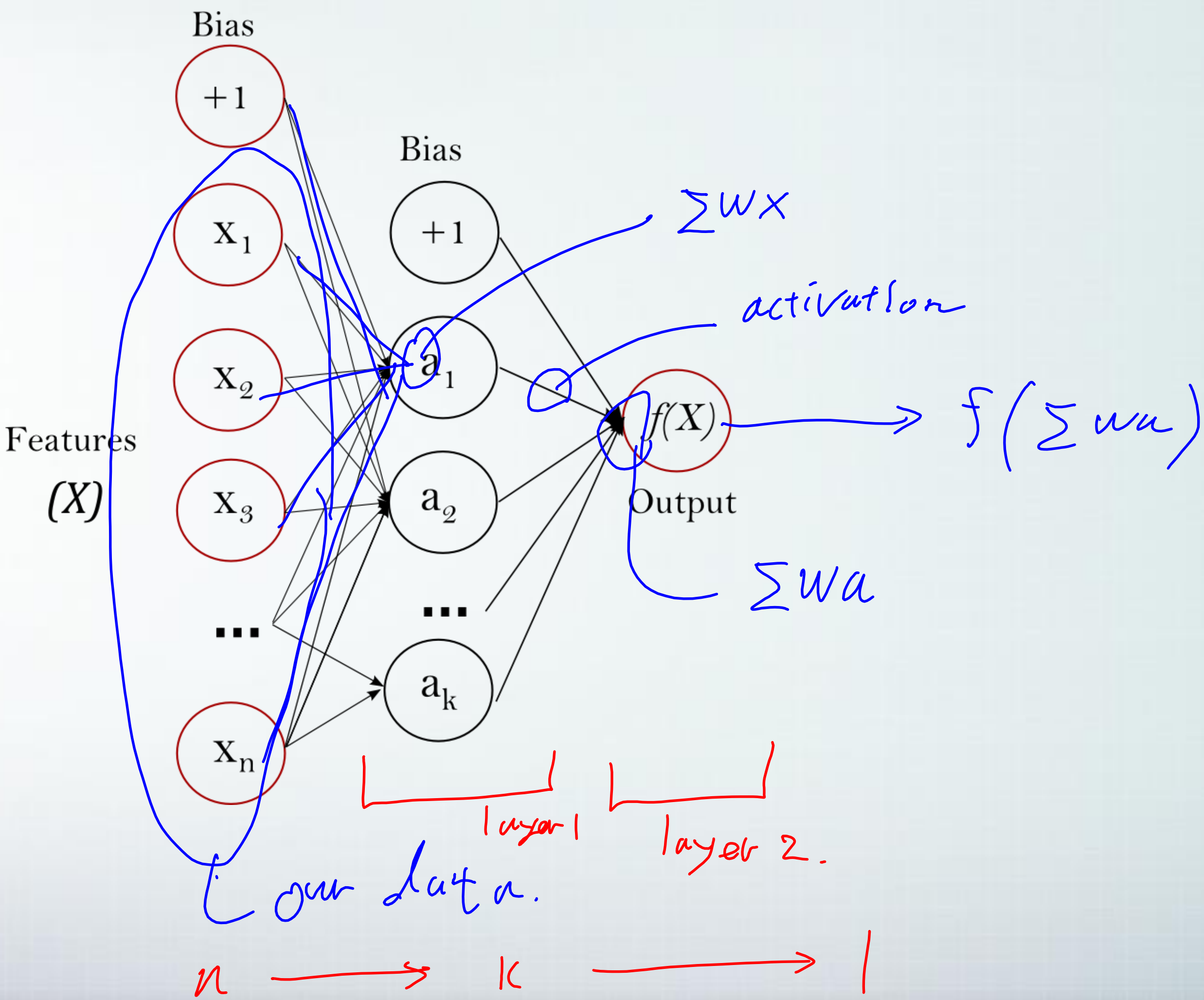
$w_1 x_1 + \cdots + w_\ell x_\ell = \sum_0^\ell w_i x_i$

$f[\ \sim\ ]$

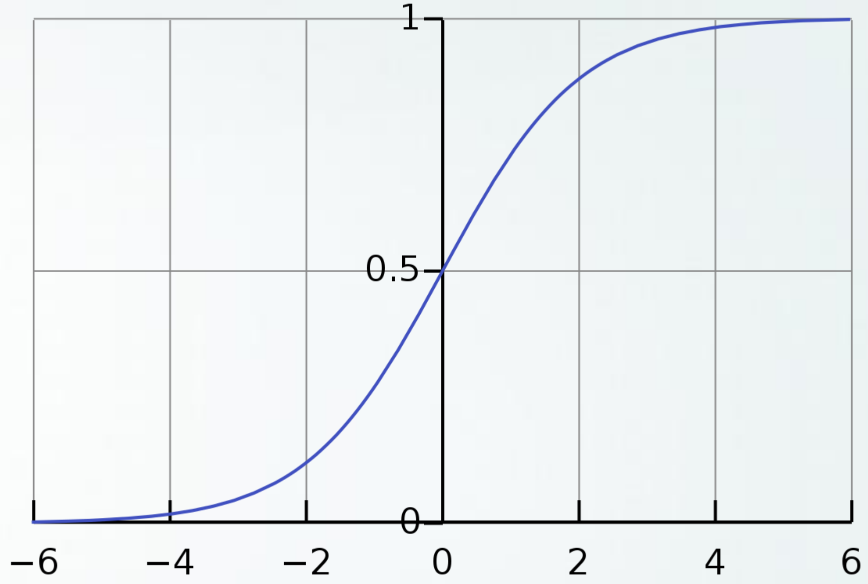
activation ftn

(비선형 함수)

Multi Layer Perceptron



Activation ftn (Sigmoid)

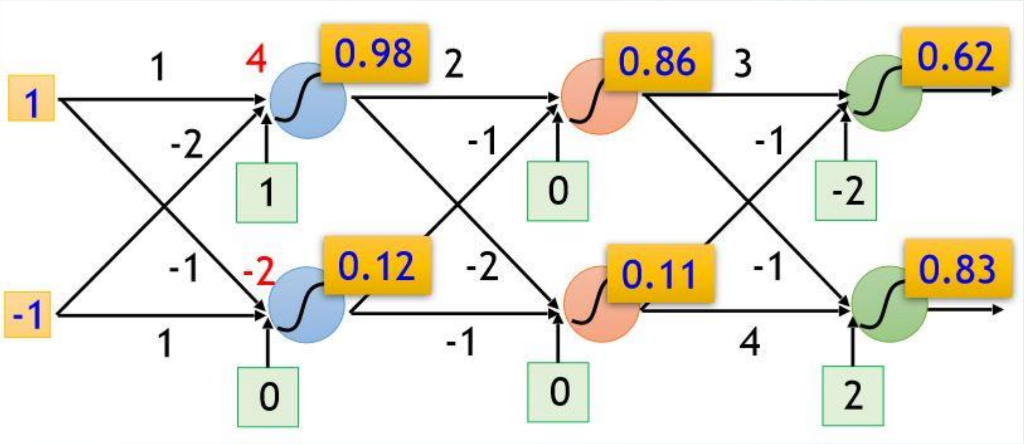


$$a(x) = \frac{1}{1+e^{-x}} \; : \; 0 \sim 1$$

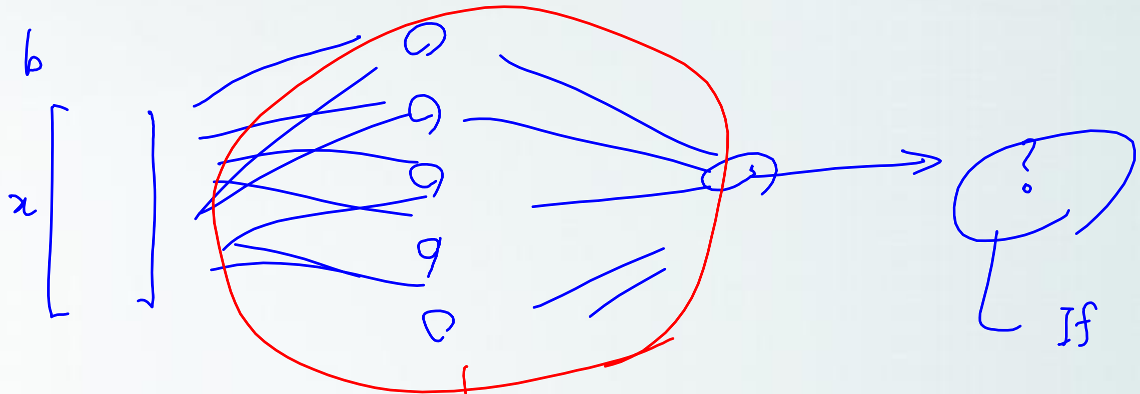
$$a'(x) = -\frac{-e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}$$

$$= a(x)(1-a(x)) \; : \; 0 \sim 1$$

Example



But…Why?



If binary classification

0, 1.

=> let: output = $P(C=1)$

distribution

: continuous.

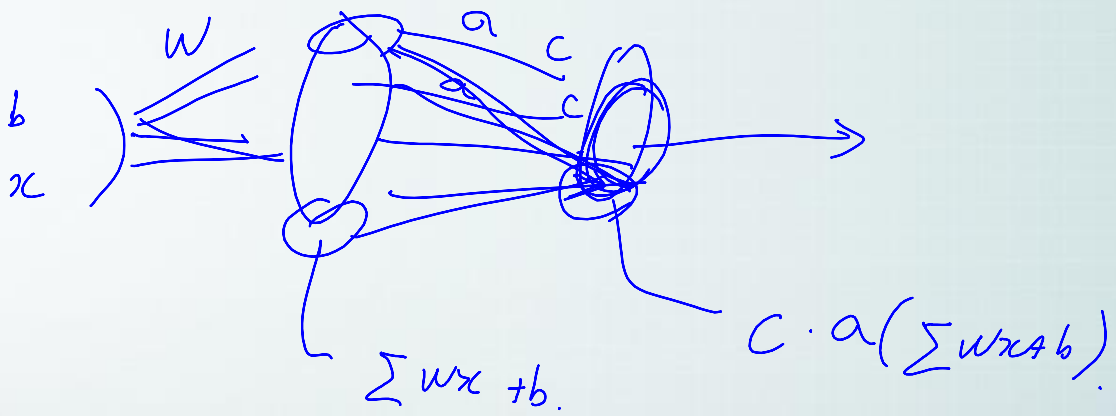
non-linear,

continuous ftn

Universal Approximation thm

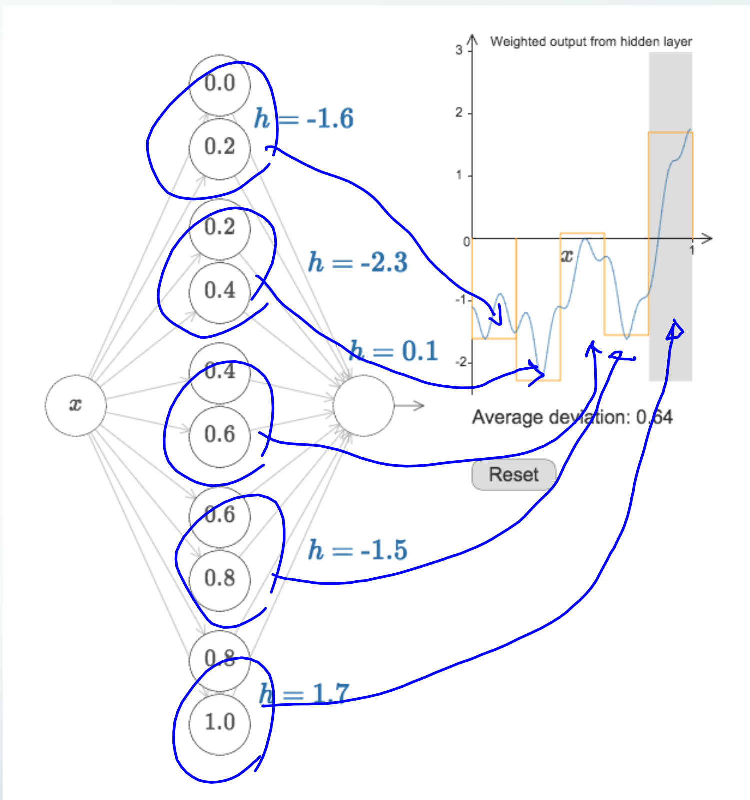
- The sum of the following form can approximate any continuous function F on $[0,1]^n$ to any degree of accuracy:

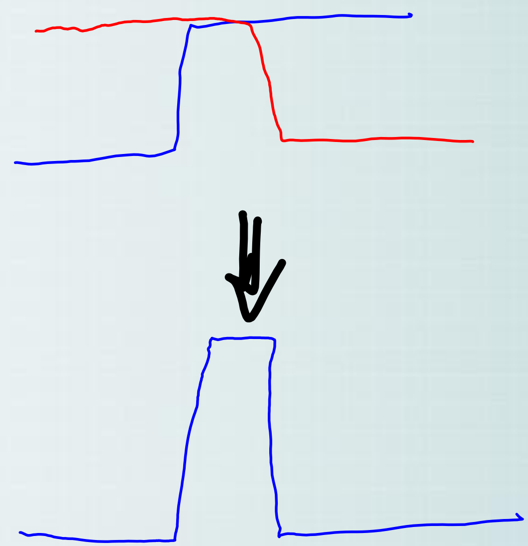
- $F(x) \approx \sum_k c_k \sigma(\sum_{i=1}^{n} w_{ki} x_i + b_k)$



Universal Approximation thm

❖ **See what happens with perceptron**

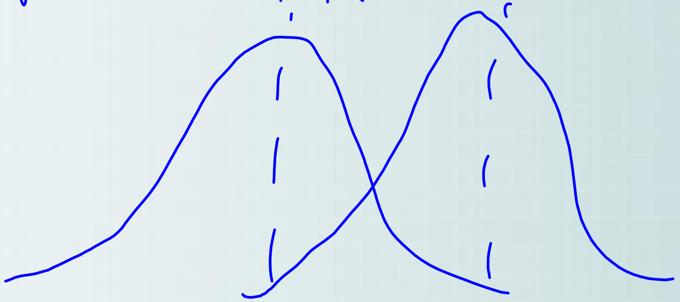




Training – Loss ftn?

- **Loss ftns**
  - Mean Squared Error
  - Mean Absolute Error
  - KL-Divergence
  - Categorical Cross Entropy
  - etc.

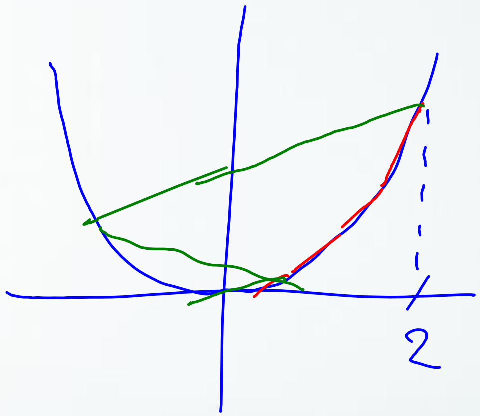
prob. dist. 거리 (not metric)



Training - Optimizer

❖ Gradient Descent & Backpropagation

$$\theta_{n+1} = \theta_n - \eta \cdot \nabla_\theta L(\theta)\Big|_{\theta_n}$$



$y = x^2$

$x_0 = 2.$

$\nabla y\big|_{x=2} = 2x\big|_{x=2} = 4.$

$\eta = 1 \Rightarrow x_1 = -2$

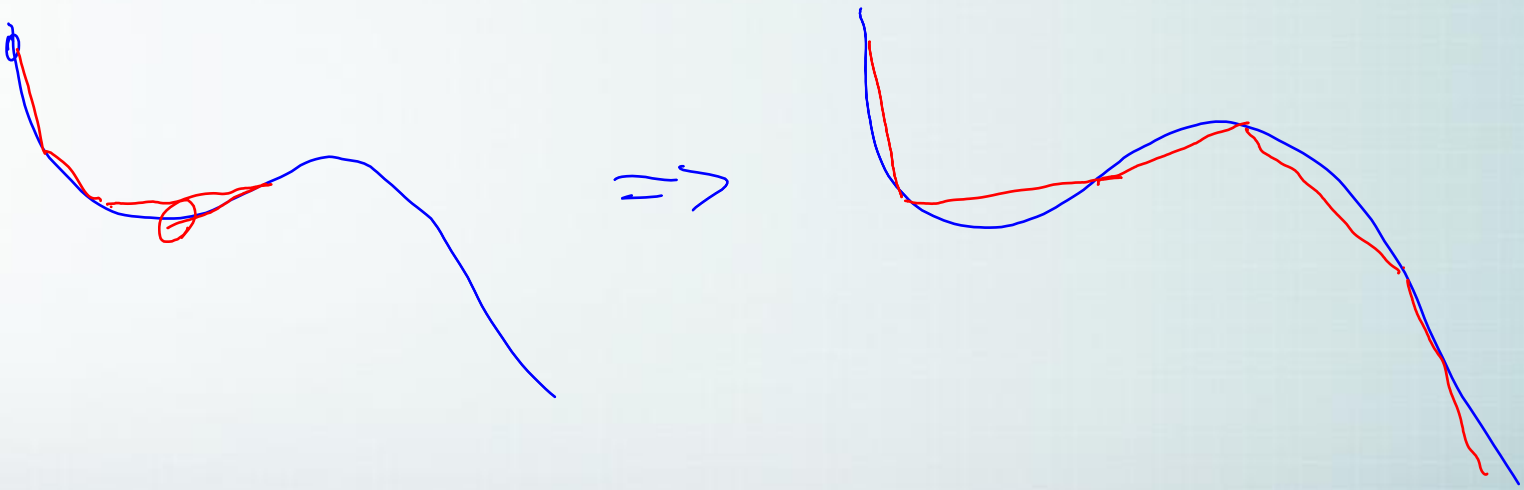
$\eta = 0.5 \Rightarrow x_1 = 0.$

$\eta = \frac{1}{4} \Rightarrow x_1 = 1.$

Training - Optimizers

- **Momentum**

$$\Theta_{n+1} = m \cdot \Theta_n - \eta \cdot \nabla_{\Theta} L(\Theta)\Big|_{\Theta_n}$$

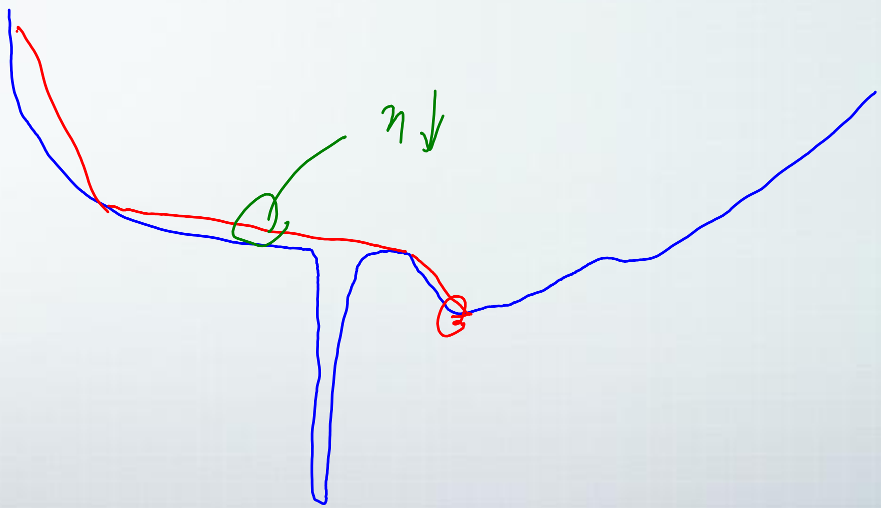


Training - Optimizers

❖ Adagrad (adaptive gradient)

$$\Theta_{n+1} = \Theta_n - \frac{\eta}{\sqrt{G_n + \epsilon}} \nabla_\Theta L(\Theta)\Big|_{\Theta_n}$$

$$G_n = \sum_{i=1}^{n} \left[\nabla_\Theta L(\Theta)\Big|_{\Theta_i} \right]^2$$



Training - Optimizers

- **RMSProp (root mean square propagation)**

$$\Theta_{n+1} = \Theta_n - \frac{\eta}{\sqrt{G_n + \epsilon}} \nabla_\Theta L(\Theta)\Big|_{\Theta_n}$$

$$G_n = \gamma \cdot G_{n-1} + (1-\gamma)\left[\nabla_\Theta L(\Theta)\Big|_{\Theta_n}\right]^2$$

Training - Optimizers

❖ Adam (adaptive moment estimation)

$$\hat{m}_\Theta = \frac{m_\Theta^{n+1}}{1-(\beta_1)^{n+1}} \quad \text{where} \quad m_\Theta^{n+1} = \beta_1 m_\Theta^n + (1-\beta)\nabla_\Theta L(\Theta)\Big|_{\Theta_n}$$

$$\hat{G}_\Theta = \frac{G_\Theta^{n+1}}{1-(\beta_2)^{n+1}} \quad \text{where} \quad G_\Theta^{n+1} = \beta_2 G_\Theta^n + (1-\beta_2)\left[\nabla_\Theta L(\Theta)\Big|_{\Theta_i}\right]^2$$

$$\Theta_{n+1} = \Theta_n - \eta \frac{\hat{m}_\Theta}{\sqrt{\hat{G}_\Theta + \epsilon}}$$

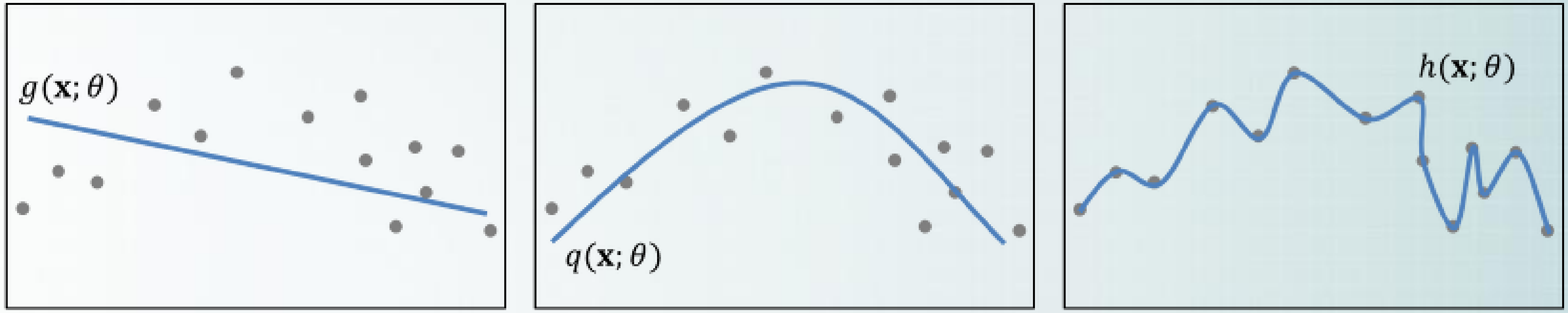
❖ **About learning rate…**

- Google exercise

Contents

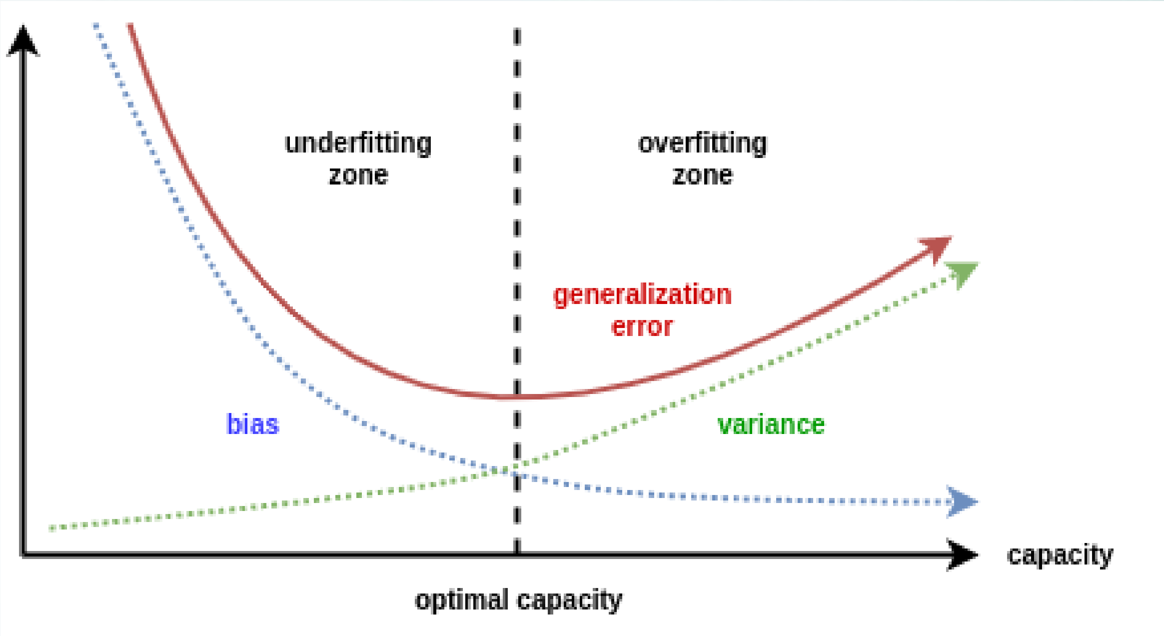
3 Bias – Variance Tradeoff

Bias vs Variance



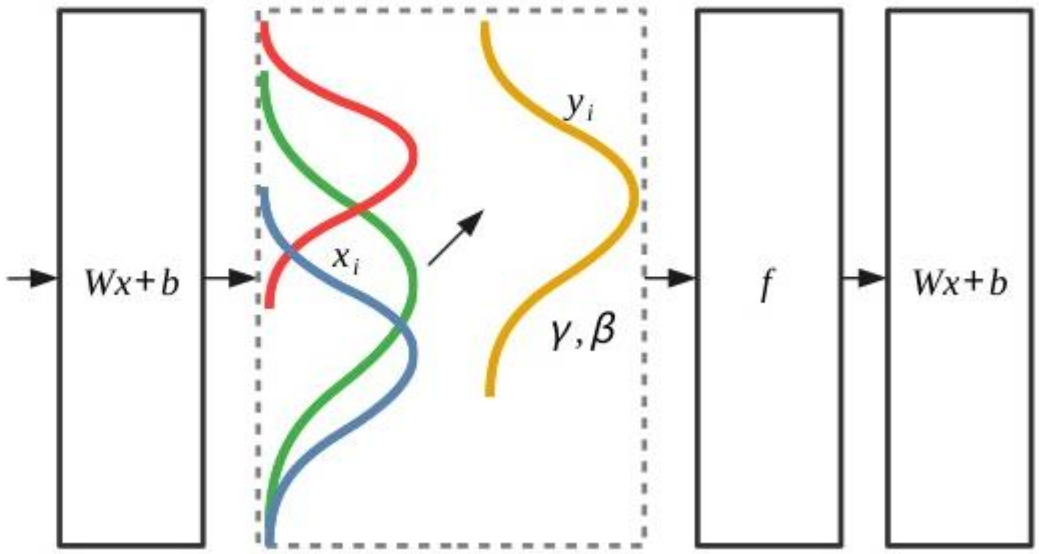
(a) Underfit (b) Ideal fit (c) Overfit

Bias vs Variance

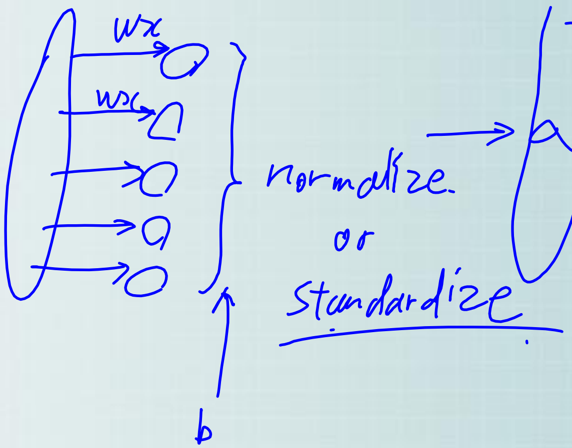


Bias vs Variance : BN

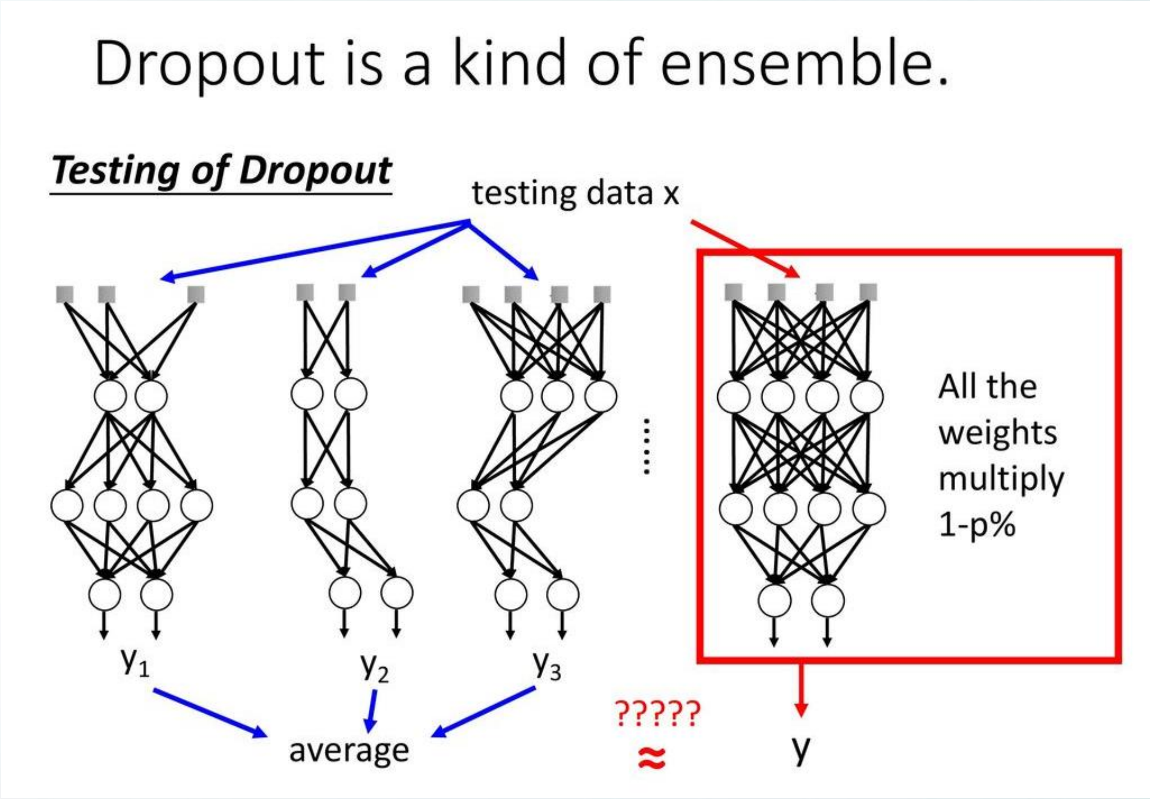
Ensure the output statistics of a layer are fixed.



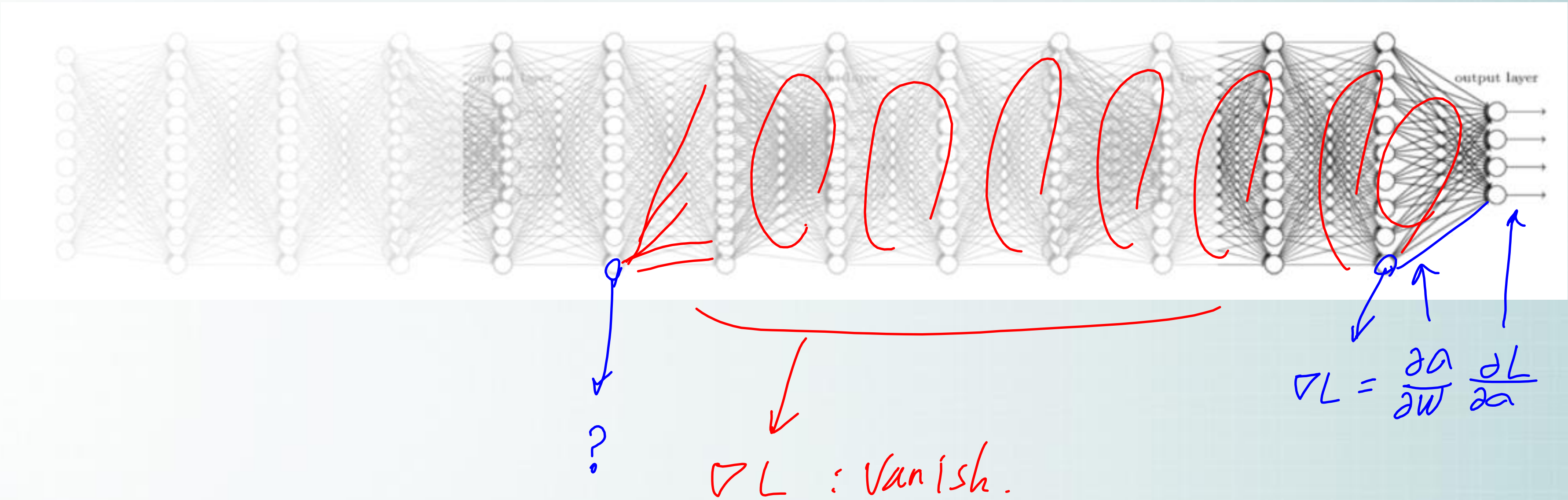
(one perceptron)



Bias vs Variance : dropout

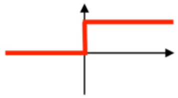
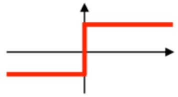
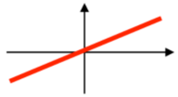
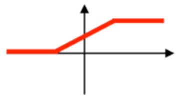
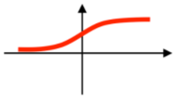
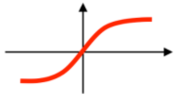
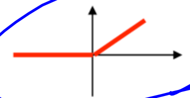
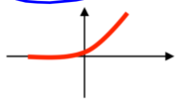


Bias vs Variance : vanishing gradient



Vanishing Gradient : relu

activation ftns

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

$\nabla = 1$

also, rid of negatives.

Contents

4 About data

Normalization vs Standardization

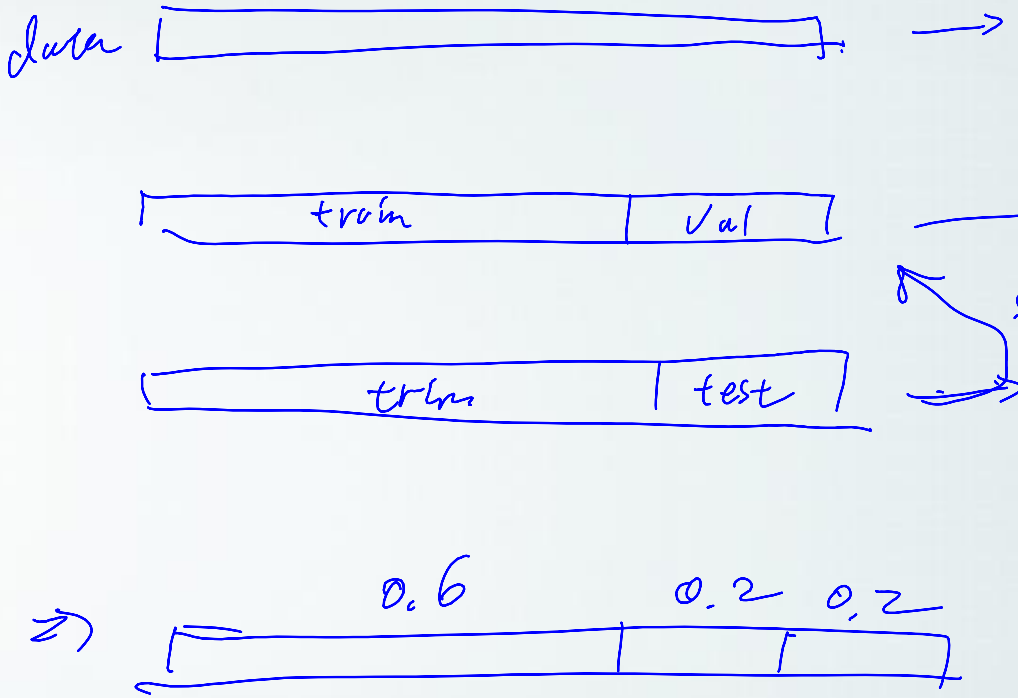
- **Normalization :**

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- **Standardization :**

$$X' = \frac{X - \mu}{\sigma}$$

Train-Valid-Test split



→ train → overfit?

→ what if val have anomaly?

same

Contents

5 Implementation

Pima-Indian Diabetes dataset

- **https://www.kaggle.com/kumargh/pimaindiansdiabetescsv**

Credit Card Fraud Detection

- https://www.kaggle.com/mlg-ulb/creditcardfraud/