Lecture 2.
Linear Classifier

Review

1. **What is Machine Learning?**
   - Definition
   - Fields of ML
   - Narrow down to Image Classification

2. **Making a Model I**
   - Narrow down to Iris Classification
   - Data-Driven Approach
     - NN, K-NN Algorithms

3. **How to Test a Model**
   - Hyperparameter
   - Cross Validation

4. **Making a Model II**
   - Limitaions of Data-Driven Approach
   - Parametric Approach
   - Linear Classifier : Algebraic & Geometric

Review

feature → predict ans : 1 ---- 86

but, actually ans : 2 --- 77

3

Review

Questions

- **Train이 잘 되었는지 판단할 수치적 척도 필요**

: define a **Loss Function** that quantifies our unhappiness with the scores across the training data

- **Parameter를 update하는 algorithm 필요**

: come up with a way of efficiently finding the parameters that minimize the **Loss Function**

21

Today's Contents

Loss

How is a Model Optimized / Updated?

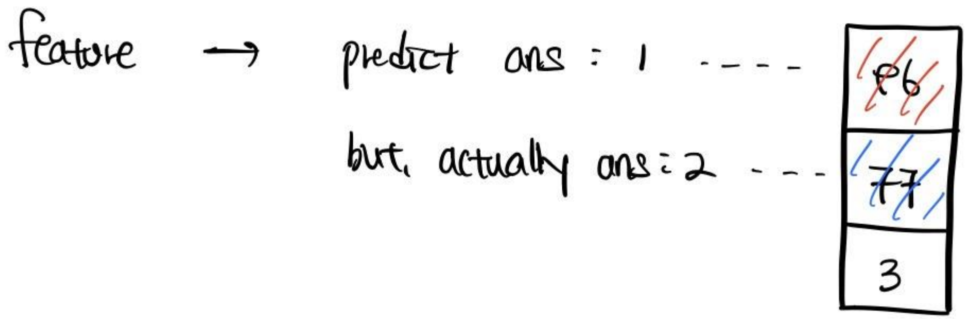
1. Training Set의 Data들을 Linear Classifier에 통과시켜서, 그 결과들을 정답과 비교
2. 1에서의 결과를 바탕으로 Parameter의 값들을 Update
3. 다시 1로

Loss

2. 1에서의 결과를 바탕으로 Parameter의 값들을 Update

정답과의 차이가 "많다"면?

정답과의 차이가 "작아 " 서 거의 비슷하다면?



Loss

Loss Over Dataset : 현재 Model의 Parameter가 주는 결과가 얼마나 부정확한지를 숫자로 표현

We want to, (MINMIZE / MAXIMIZE) the LOSS

Loss

How do we calculate the Loss Over (Training) Dataset?

Training dataset $\{(x_i, y_i)\}_{i=1}^{N}$ 에 대해,

feature label

Loss over the dataset 은,

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W, b), y_i)$$

loss function

ith data의 prediction 결과

ith data의 정답

Loss Function

Training dataset $\{(x_i, y_i)\}_{i=1}^{N}$ 에 대해,

feature label

Loss over the dataset 은,

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W, b), y_i)$$

loss function

ith data의 prediction 결과

ith data의 정답

We calculate the Loss via,

Loss Function

Loss Function

Loss Function indicates how incorrect the parameters are, for a given data

How do we define a Loss Function?

How do we define a "Good" Model?

Loss Function

Two Popular Loss Functions,

1. Multiclass SVM Loss

2. Cross-Entropy Loss

Loss Function : Multiclass SVM Loss

The SVM Loss is set up so that
the SVM wants the correct class for each input
to have a score higher than the incorrect classes by some fixed margin △.

정답 label의 score가 나머지 incorrect label의 score보다 △만큼 크기를 바람.

Loss Function : Multiclass SVM Loss

$$L_i = \sum_{j \neq y_i} max(0\ ,\ s_j - s_{y_i} + \triangle)$$

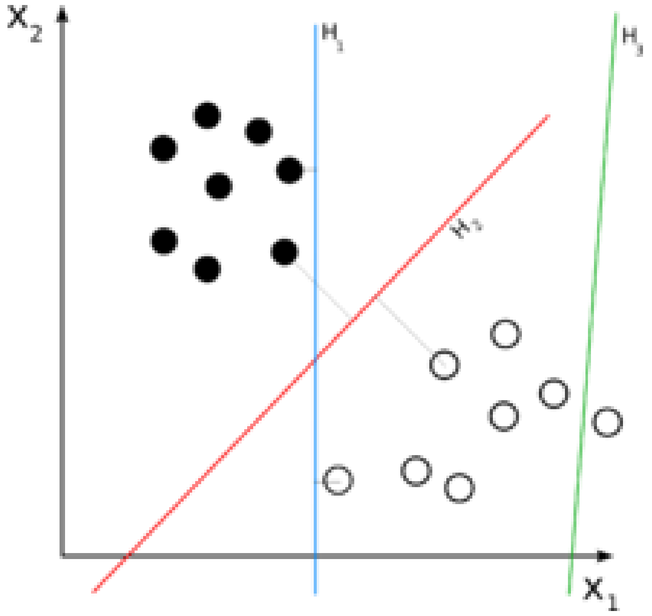
$$L = \frac{1}{N} \sum_{i=1}^{N} L_i$$

Loss Function : Multiclass SVM Loss

More on SVM (Optional)

Multiclass SVM Loss의 objective는,

Try to find the decision hyperplane with max-margin property



** For more detail,
https://ko.wikipedia.org/wiki/%EC%84%9C%ED%8F%AC%ED%8A%B8_%EB%B2%A1%ED%84%B0_%EB%A8%B8%EC%8B%A0
cs229.stanford.edu/notes/cs229-notes3.pdf

Loss Function : Cross-Entropy Loss

An approach based on probability

Softmax function maps the scores to probabilities

$$\text{softmax function} : \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$

정답 score

e^{score} 의 sum

Loss Function : Cross-Entropy Loss

softmax function : $\dfrac{e^{f_{y_i}}}{\sum_j e^{f_j}}$

$e^{f_{y_i}}$ ~ 정답 score

$\sum_j e^{f_j}$ ~ e^{score} 의 sum

↓

interpret as, $P(y_i | x_i ; W, b) = \dfrac{e^{f_{y_i}}}{\sum_j e^{f_j}}$

(parameter는 W, b. x_i가 input일 때, y_i가 나올 확률)

Loss Function : Cross-Entropy Loss

$$p_i = \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$

$$L_i = -\,log(p_i)$$

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i$$

Loss Function : Cross-Entropy Loss

Loss Function : Cross-Entropy Loss

In reality,

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}}$$

$$= \frac{Ce^{a_i}}{C\sum_{k=1}^{N} e^{a_k}}$$

$$= \frac{e^{a_i + \log(C)}}{\sum_{k=1}^{N} e^{a_k + \log(C)}}$$

$$log(C) = -max(a)$$

Loss Function : Cross-Entropy Loss

이때, Unknown Parameter W와 b를 어떻게 estimate?

via **MLE(Maximum Likelihood Estimation)**

More on MLE (Optional)

conditional probability를 계산할 때,

unknown parameter : W, b

We want to find W, b that maximizes the (Log) Likelihood Function,

so define the loss as,

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right).$$

** For more detail, https://en.wikipedia.org/wiki/Multinomial_logistic_regression / http://jaejunyoo.blogspot.com/2018/02/minimizing-negative-log-likelihood-in-kor-3.html
https://ratsgo.github.io/deep%20learning/2017/09/24/loss

Actually, these are....

Hyperparameters!

Regularization

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i$$

But is this enough…?

loss L을 최소화하는 weight W를 찾는 것이 목적.

이때, L = 0이 되게 하는 W'가 있다면,

임의의 실수 k > 1에 대해, kW*도 L = 0을 만족.

따라서, W is **"NOT" uniquely determined**!

Regularization

among all kW′, small ones are preferred (reasons discussed later)

so, add regularization term to discourage large weight

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W, b), y_i) + \underbrace{\lambda R(W)}_{\text{Regularization Term}}$$

λ : regularization strength

$$R(W) = \sum_k \sum_l |W_{kl}| \quad \text{-- L1}$$

or,

$$= \sum_k \sum_l W_{kl}^2 \quad \text{-- L2}$$

Regularization

더 작은 W가 갖는 이점?

ex) $x = [1, 1, 1, 1]^T$

$w_1 = [1, 0, 0, 0]$, $w_2 = [1/4, 1/4, 1/4, 1/4]$

면, $w_1 x = w_2 x = 1$

but, $R(w_1) = 1 > R(w_2) = 1/4$

Regularization

Regularization을 통해,

No input dimension can have a very large influence on the scores all by itself

Regularization

If an input dimension has a very large influence on the scores all by itself,

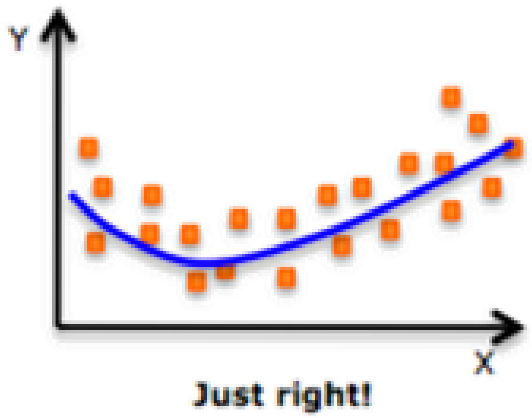
then, model이 training data의 noise까지 학습해버릴 수 있다!

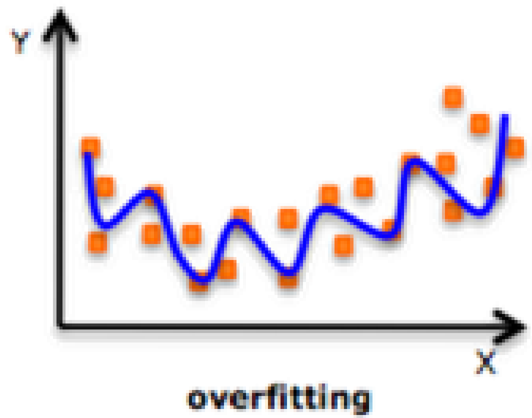
Model performing TOO WELL on training data

Regularization

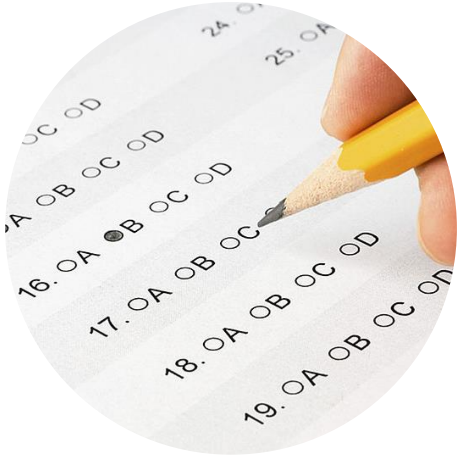
We are building a **GENERAL** model for classification

Thus, doing **TOO WELL** on the training data is not desired



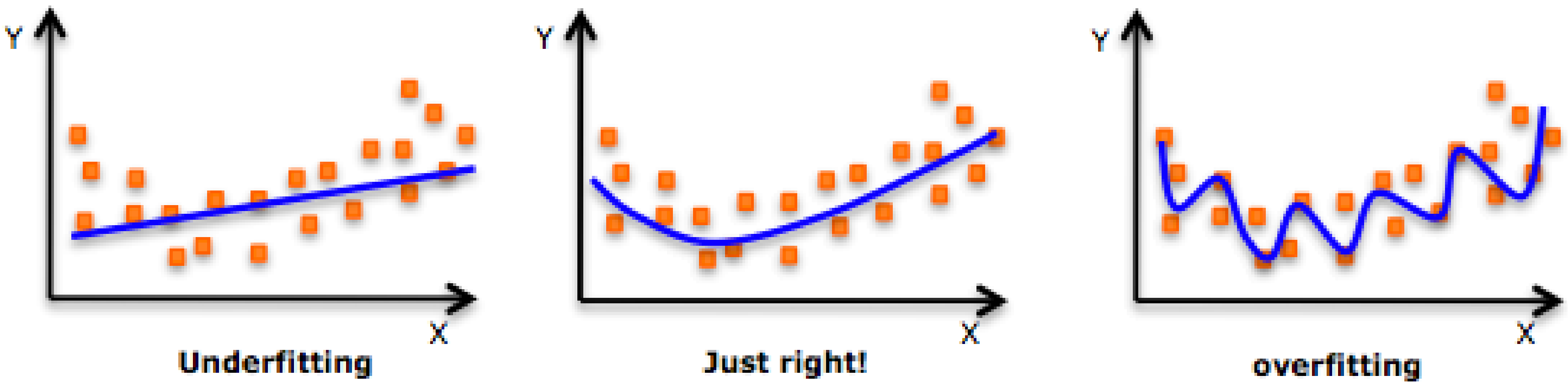


Regularization



Regularization : Overfitting

Regularization



Regularization

$$R(W) = \sum_k \sum_l {W_{kl}}^2$$

L2 Regularizer favors W that is spread out

Regularization : Overfitting

by $R(W)$, we discourage large W = prevent some input dim. from

having too much influence on the output

= prevent the model from

learning the noises

= prevent overfitting

(to some extent)

Review

1. **Loss**
2. **Loss Function**
   - Multiclass SVM Loss
   - Cross-Entropy Loss
3. **Regularization**
   - Why it is needed : to discourage large weight matrix
   - Overfitting

Preview on Next Class

Questions

- **Train이 잘 되었는지 판단할 수치적 척도 필요**

: define a **Loss Function** that quantifies our unhappiness with the scores across the training data

- **Parameter를 update하는 algorithm 필요**

: come up with a way of efficiently finding the parameters that minimize the **Loss Function**

Optimization and Backpropagation

Loss Function은 Scalar Function

Thus, Loss Function의 input W, b에 대해

Gradient를 구해 주어 W, b에서 빼면,

Loss가 줄어드는 방향으로 학습하지 않을까?

21