

R E P O R T

머신러닝 기말고사



학 과 머신러닝 02분반

교수님 전동산

학 과 컴퓨터공학과

학 번 2254747

이 름 유우림



동아대학교
DONG-AUNIVERSITY

9주차 1차시

중간고사 리뷰

9주차 2차시

1. CNN의 개요 및 특징

- CNN의 필요성:
 - MLP(다층 퍼셉트론)는 이미지 데이터를 평탄화(flattening)하여 공간적(형상) 정보를 잃음.
 - CNN은 2D 이미지 데이터를 처리하여 공간적 구조를 유지하며 특징을 추출함.
 - CNN은 필요한 가중치 수가 MLP에 비해 적어 학습 효율이 높음.

2. CNN의 주요 구성 요소

(1) Convolution (합성곱 연산)

- 목적: 입력 이미지의 특징을 필터(커널)를 사용해 추출.
- 계산 과정:
 - 필터는 입력 이미지의 특정 부분과 요소별 곱셈 후 합산.
 - 결과값을 출력 특징 맵(Feature Map)으로 생성.
- 중요 매개변수:
 - 필터 크기: 커널의 가로, 세로 크기 (예: 3x3, 5x5).
 - Stride(보폭): 필터가 이동하는 간격. Stride가 클수록 출력 크기 작아짐.
 - 패딩(Padding): 입력 데이터 경계에 값을 추가해 크기를 유지.

출력 크기 계산 공식:

$$Output_Height = \frac{Input_Height + 2P - FH}{S} + 1$$
$$Output_Width = \frac{Input_Width + 2P - FW}{S} + 1$$

- PP: 패딩 크기
- FH, FWFH, FW: 필터의 높이와 너비
- SS: 스트라이드

(2) 풀링 (Pooling)

- **목적:** 입력 데이터의 차원을 축소하여 연산량 감소 및 과적합 방지.
- **유형:**
 - Max Pooling: 필터 영역 내 최대값 추출.
 - Average Pooling: 필터 영역 내 평균값 추출.

3. CNN 활용 사례

1. 이미지 분류: 이미지넷 챌린지(ILSVRC) 등에서 사용.
2. 초해상도 복원(SR): 저해상도 이미지를 고해상도로 변환.
3. 비디오 복원: 영상의 왜곡을 제거하고 선명도를 높임.

10주차 1차시

1. LeNet-5 정의: CNN 대표모델

(1) 간단한 CNN 모델 구조

모델 구성:

1. 입력: 이미지.
 $32 \times 32 \times 3$
2. **Conv1**: $5 \times 5 \times 3$ 필터, 64개, stride=1, padding=0.
 $28 \times 28 \times 6$
3. **Pooling**: 2×2 , stride=2 Max Pooling.
 $14 \times 14 \times 6$
4. **Conv2**: $5 \times 5 \times 64$ 필터, 16개, stride=1, padding=0 .
 $10 \times 10 \times 16$
5. **Pooling**: 2×2 , stride=2 Max Pooling .
 $5 \times 5 \times 16$
6. **Flattening & Fully Connected Layer**: 평탄화 후 분류.



필터: 3×3 , stride=1, padding=1 \Rightarrow input feature map과 같은 크기의 output feature map이 됨

10주차 2차시

1. CNN 대표 모델

LeNet-5

- 초창기 CNN 구조로 문서 인식에 사용.
- 구조:
 - Convolution → Pooling → Flatten → Fully Connected.

ResNet (2015) - Residual Networks

- 잔차 학습(**Skip Connection**)을 통해 더 깊은 네트워크 학습 가능.
- element by element Sum을 해줌.
- 특징:
 - 성능 저하 없이 더 깊은 구조 구현.
 - Global Average Pooling(GAP)을 통해 출력 크기 축소.

DenseNet (2017)

- 각 층의 출력을 모든 후속 층과 연결.
- feature map 뒤에 feature map을 붙임.
- 파라미터 효율성을 유지하며 학습 성능 향상.

SENet (2018) - Squeeze and Excitation Networks

- 채널 주의 메커니즘(**Channel Attention**)을 도입해 각 채널의 중요도 학습..
- element by element Multiplication을 해줌

11주차 1차시

Channel-Attention(CA)

- Feature map의 각 채널에 대해 중요도를 부여(weight값을 각각 따로 부여)

Attention 과정

일반적인 feature map에서 각 채널에 대한 평균값을 만들 (ppt상의 1x1xC인 Feature map을 의미)

이게 MLP input으로 들어감(FC- layer)

ReLU 적용

output을 1x1xC Feature map으로 출력함

여기서 얻은 weight값(스칼라)을 일반적인 feature map에 곱해줌(각 channel에)

→ Attention을 통해 중요도가 부여된 feature map생성

Skip Connection - ResNet

- 기존 데이터의 채널을 out과 같은 채널로 맞춤 → skip
- 여러 layer를 거친 feature map → out
- element by element Sum을 해줌
- 단, Width, Height, Channel 모두 같아야 사용가능

$out = out + skip$ / $out = torch.add(out, skip)$

Dense Connection - DenseNet

- 기존 데이터 → x
- 여러 layer를 거친 feature map → out

- out 뒤에 그냥 붙임
- 단, Height, Width가 같아야됨

`out = torch.cat([x, out], dim=1)`

CA (Channel Attention) - SENet

- 여러 layer를 거친 feature map → out
- 위에 값인 out을 FC-layer후 1x1xC Average Pooling → weight
- element by element Multiplication을 해줌

`out = out * weight`

11주차_2차시-SVM_이론

1. unit vector $\Rightarrow \rightarrow x^{\rightarrow}$

2. 내적(inner product): 함수

$$\langle x, y \rangle = \|x\| \cdot \|y\| \cdot \cos\theta = x^T \cdot y$$

$$\rightarrow \langle x, x \rangle = x^T \cdot x = \|x\|_2^2$$

$$\rightarrow \|x\|_2 = \sqrt{x^T \cdot x}$$

$$\rightarrow \mathbf{x} \perp \mathbf{y} \Rightarrow \langle x, y \rangle = 0$$

1. SVM의 기본 개념

- 목적:
 - 두 클래스를 가장 잘 분리하는 **최적의 초평면(Optimal Separating Hyperplane)** 찾기.
 - **Margin**을 최대화하여 일반화 성능을 높임.
- 초평면(Separating Hyperplane):

$$f(X) = W^T X + b$$

- W : 가중치 벡터(Weight Vector).
 - b : 절편(Bias).
- 지원 벡터(Support Vector):
 - 초평면과 가장 가까운 데이터 포인트.

2. Margin 계산

- Margin 정의:
 - Margin은 Support Vector 간의 거리로, 다음과 같이 계산됨:

$$\rho = \frac{1}{\|W\|}$$

- $\|W\|$: W 벡터의 유클리드 노름.
- 초평면과의 거리:
 - 데이터 포인트 X_i 가 초평면과의 거리는 다음과 같음:

$$\text{Distance} = \frac{|W^T X_i + b|}{\|W\|}$$

3. SVM의 최적화 문제

- 목적 함수:
 - Margin을 최대화하는 W 와 b 를 찾음:

$$\text{Margin} = \frac{2}{\|W\|}, \text{MaxMargin} = \max \frac{2}{\|W\|} \therefore \min \frac{1}{2} \|W\|^2$$

- 제약 조건:

$$y_i(W^T X_i + b) \geq 1 \quad \forall i$$

- y_i : 데이터의 레이블 (+1 또는 -1).

4. 손실 함수 (Hinge Loss)

- Hinge Loss 정의:

$$\text{Loss} = \max(0, 1 - y_i(W^T X_i + b))$$

- 조건:
 - $y_i(W^T X_i + b) \geq 1$: 분류 성공, 손실 없음 ($\text{Loss} = 0$).
 - $y_i(W^T X_i + b) < 1$: 분류 실패, 손실 발생 ($\text{Loss} > 0$).

5. Gradient Descent를 활용한 학습

- **목적:**

- Hinge Loss를 최소화하기 위해 W 와 b 를 업데이트.

- **Weight와 Bias 업데이트:**

- 손실이 발생하는 경우:

$$W \leftarrow W - \alpha(\lambda W - y_i X_i)$$

$$b \leftarrow b + \alpha y_i$$

- 손실이 없는 경우:

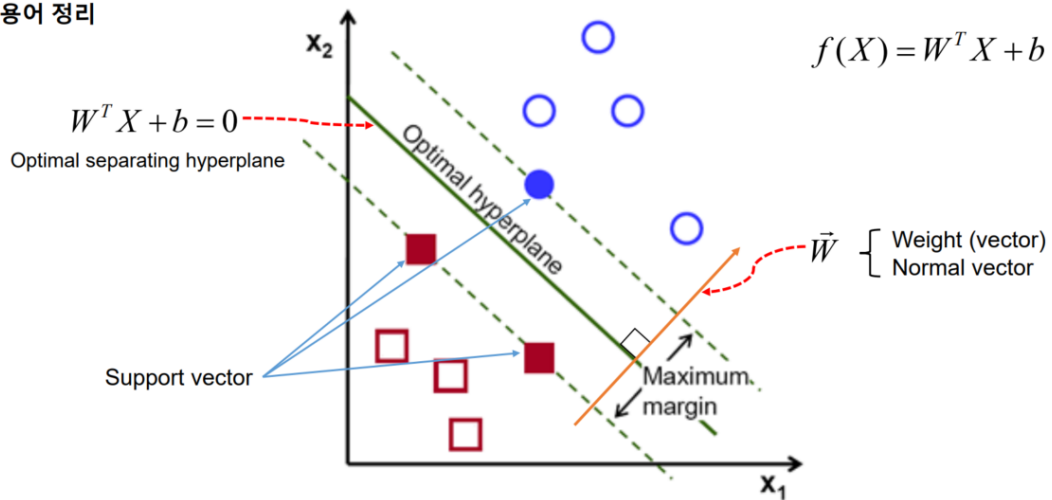
$$W \leftarrow W - \alpha \lambda W$$

- α : 학습률(Learning Rate).
- λ : 정규화 파라미터.

13주차_1차시-SVM 실습

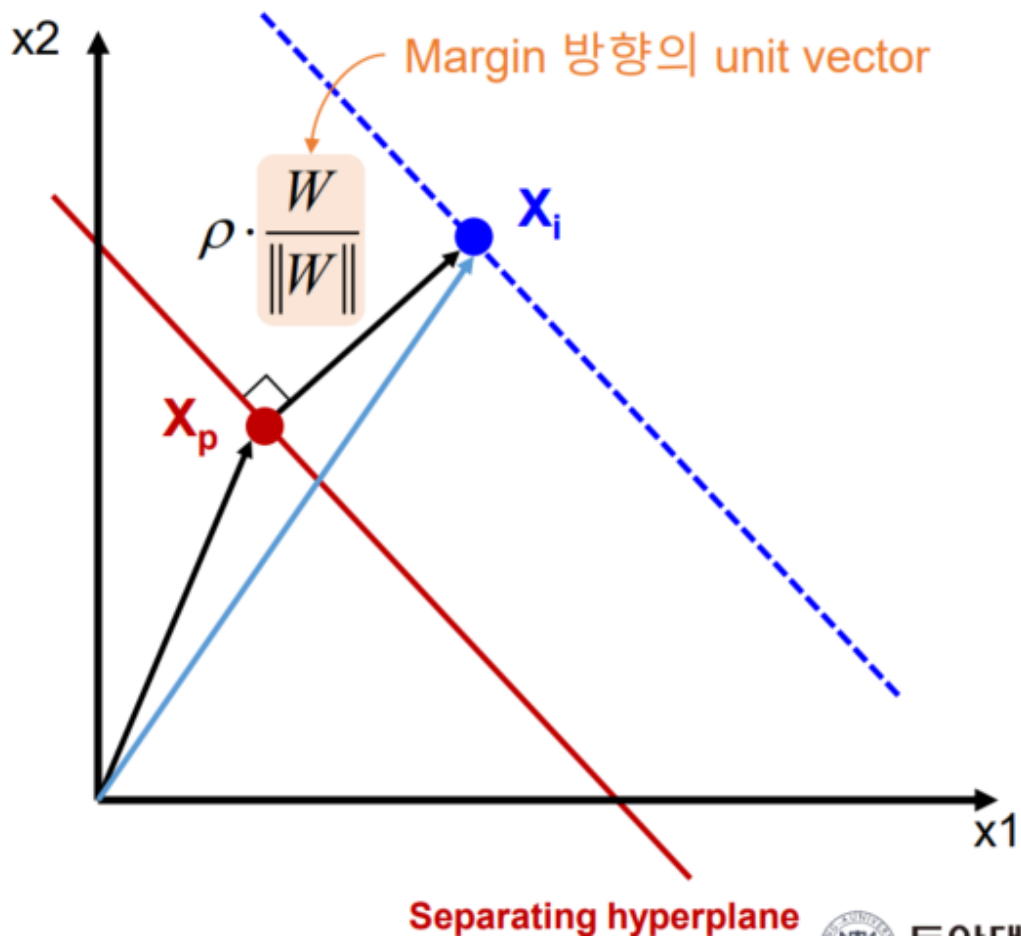
Review – Support Vector Machine

- 용어 정리



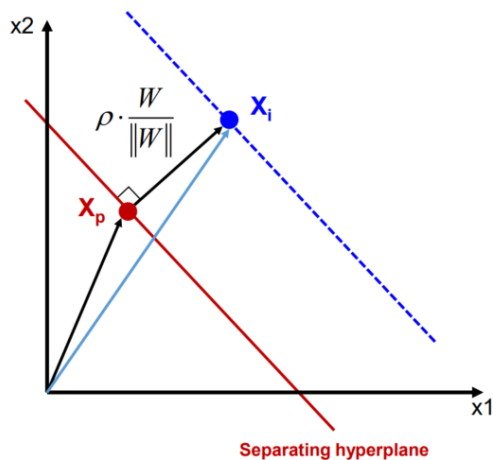
Ref.: <https://towardsdatascience.com/>

5/27

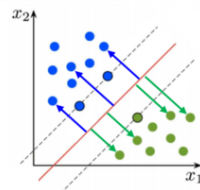


Review – Support Vector Machine

- 임의의 데이터 x_i 에 대해 separating hyperplane과의 거리: ρ



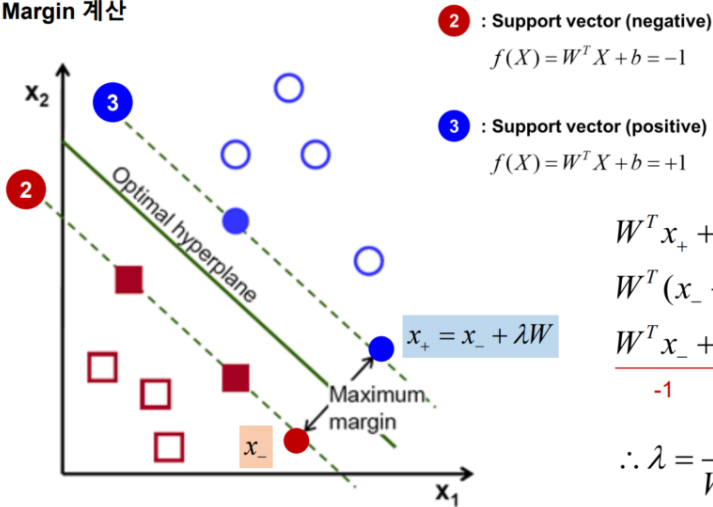
- ① $x_i = x_p + \rho \cdot \frac{W}{\|W\|}$
- ② $f(x_p) = W^T x_p + b = 0$
- ③ $f(x_i) = W^T x_i + b$
 $= W^T \cdot \left(x_p + \rho \cdot \frac{W}{\|W\|} \right) + b$
 $= \underbrace{(W^T x_p + b)}_0 + \rho \cdot \frac{\|W\|^2}{\|W\|}$



- ④ $\therefore \rho = \frac{f(x)}{\|W\|}$

Review – Support Vector Machine

Margin 계산



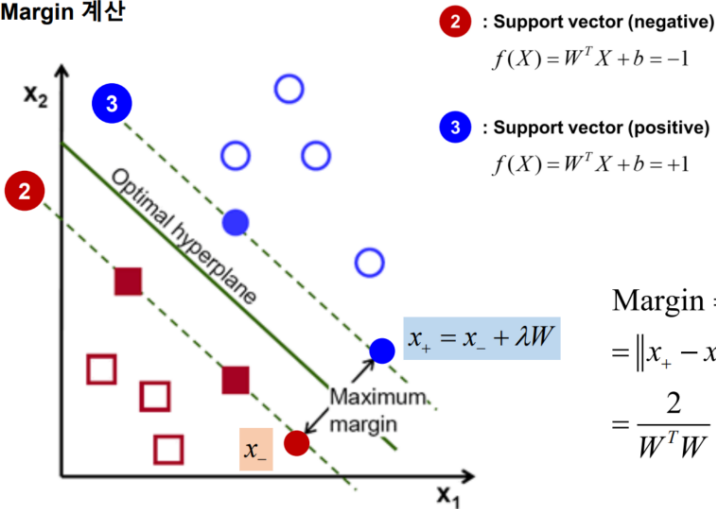
$$\begin{aligned}
 W^T x_+ + b &= 1 \\
 W^T (x_- + \lambda W) + b &= 1 \\
 \frac{W^T x_- + b + W^T \lambda W}{-1} &= 1 \\
 \therefore \lambda &= \frac{2}{W^T W}
 \end{aligned}$$

11/27



Review – Support Vector Machine

Margin 계산



$$\begin{aligned}
 W^T x_+ + b &= 1 \\
 W^T (x_- + \lambda W) + b &= 1 \\
 W^T x_- + b + W^T \lambda W &= 1 \\
 \therefore \lambda &= \frac{2}{W^T W}
 \end{aligned}$$

$$\begin{aligned}
 \text{Margin} &= \text{distance}(x_+, x_-) \\
 &= \|x_+ - x_-\|_2 = \|\lambda W\|_2 \\
 &= \frac{2}{W^T W} \cdot \sqrt{W^T W} = \frac{2}{\|W\|_2}
 \end{aligned}$$

12/27



13주차_2차시-Entropy, KNN

1. 엔트로피(Entropy)

- 정의:
 - 불확실성(Uncertainty)의 정도를 나타내는 척도.
 - 불확실성이 높을수록 엔트로피 값 증가.
- 엔트로피의 활용:
 - 머신러닝에서 사용 사례:
 1. 손실 함수(Loss Function): 모델의 예측 오류를 측정 (예: Cross Entropy Error).
 2. 의사결정트리(Decision Tree): 엔트로피가 낮은 특징부터 분할.
 3. 적극적 학습(Active Learning): 정보량이 많은 샘플을 우선 선택.
- Cross Entropy Error (CEE) 예제:
 - 정답 y : $[0, 0, 1, 0, \dots, 0]$
 - 예측 값 y' : $[0, 0, 0.8, 0, \dots, 0]$
 - CEE 계산:

$$CEE(y, y') = - \sum y_i \log(y'_i) = - \log(0.8) \approx 0.2231$$

$$D_{KL}(P||P) = 0$$

$$\begin{aligned} D_{KL}(P||Q) &= \text{Cross Entropy}(P, Q) - \text{Entropy}(P) \\ &= - \sum P_i \log(q_i) - (- \sum P_i \log(P_i)) \end{aligned}$$



Cross Entropy: 원래 정보량 + 불확실성에 대한 엔트로피
Entropy: 원래 정보량



1. $D_{KL}(P||Q) \geq 0$
2. $D_{KL}(P||Q) \neq D_{KL}(Q||P)$

2. K-최근접 이웃 알고리즘 (K-Nearest Neighbor, KNN)

- 정의:
 - 비지도 학습(Unsupervised Learning) 기법.
 - 데이터 간 거리를 기준으로 최근접 K 개의 이웃 데이터를 참조하여 예측.
- KNN 주요 개념:
 1. KNN 분류(Classification):
 - K 개의 데이터에서 다수결(Majority Voting)로 분류.
 2. KNN 예측(Regression):
 - K 개의 데이터에서 평균값, 중간값 등을 사용해 예측.
- 핵심 이슈:
 1. 최적의 K 결정:
 - K 가 작을수록 과적합(Overfitting), K 가 크면 과소적합(Underfitting).
 2. 거리 측정 방법:
 - L1 Norm (맨해튼 거리):
$$d = \sum |x_i - y_i|$$
 - L2 Norm (유클리드 거리):
$$d = \sqrt{\sum (x_i - y_i)^2}$$
 - 마할라노비스 거리(Mahalanobis Distance).
 - 상관관계 기반 거리(Correlation Distance).