Dong-A Univ. (ISPL)

동아대학교 DONG-A UNIVERSITY

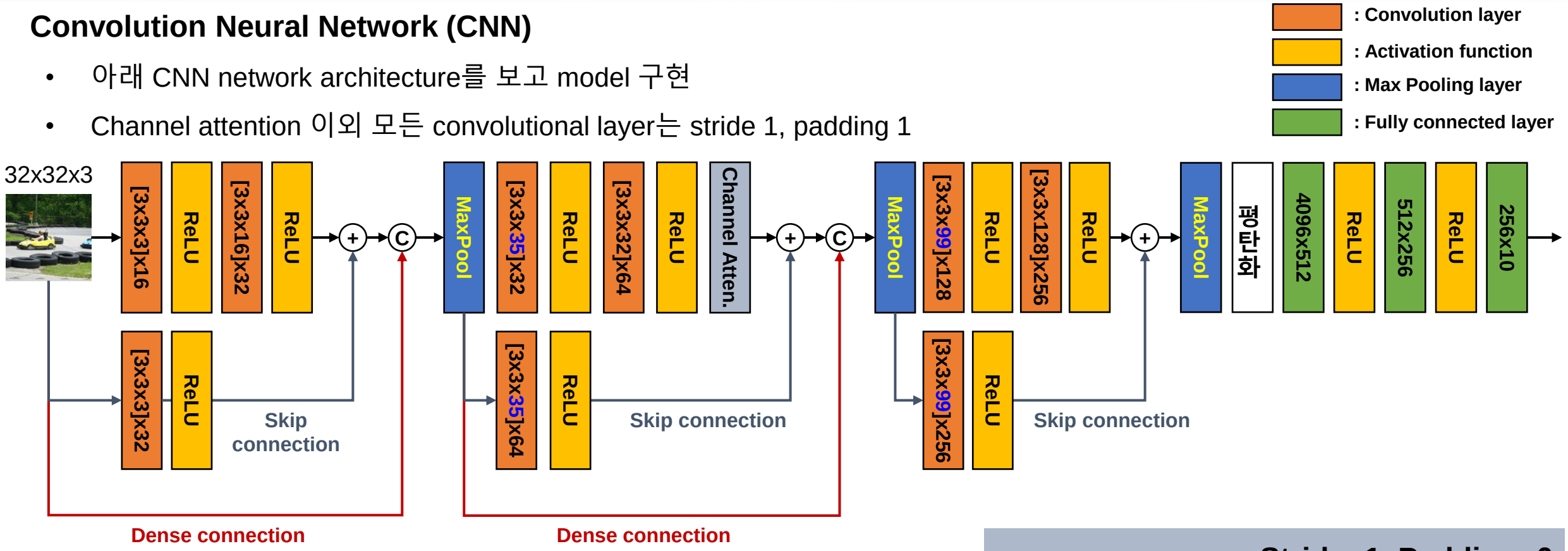
기말고사 대비문제 – 설명 추가

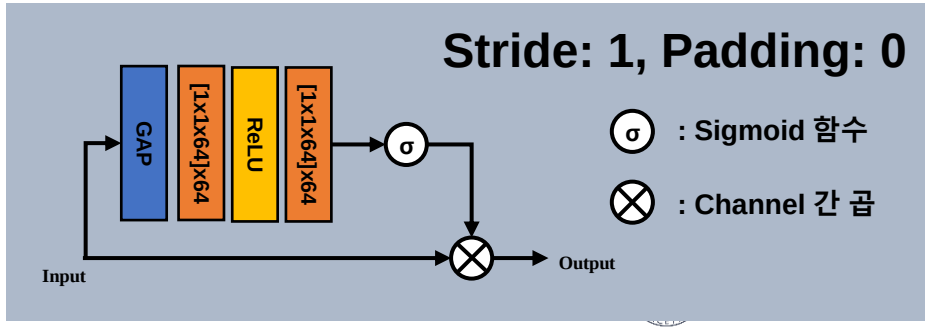
컴퓨터AI공학부

2024년 2학기 머신러닝

기말고사 연습 문제 [1/4]

- **Convolution Neural Network (CNN)**
  - 아래 CNN network architecture를 보고 model 구현
  - Channel attention 이외 모든 convolutional layer는 stride 1, padding 1



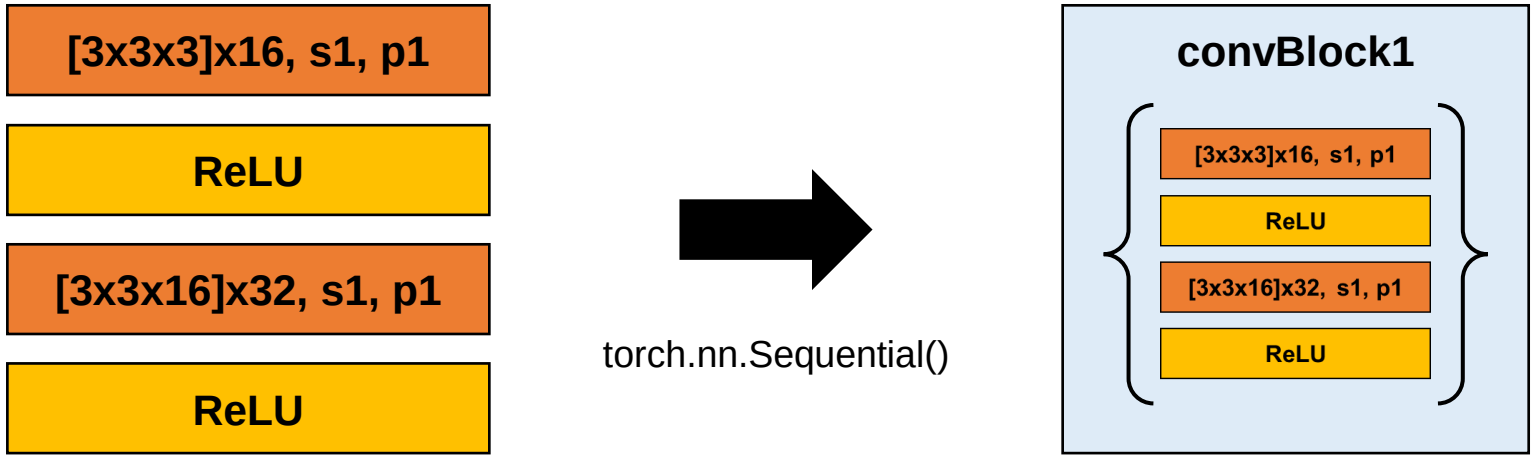


❖ **주의사항(1): Skip connection은 Width, Height, Channel이 모두 같아야 사용 가능**

❖ **주의사항(2): Dense connection (torch.cat)은 width, height이 동일해야 적용 가능**

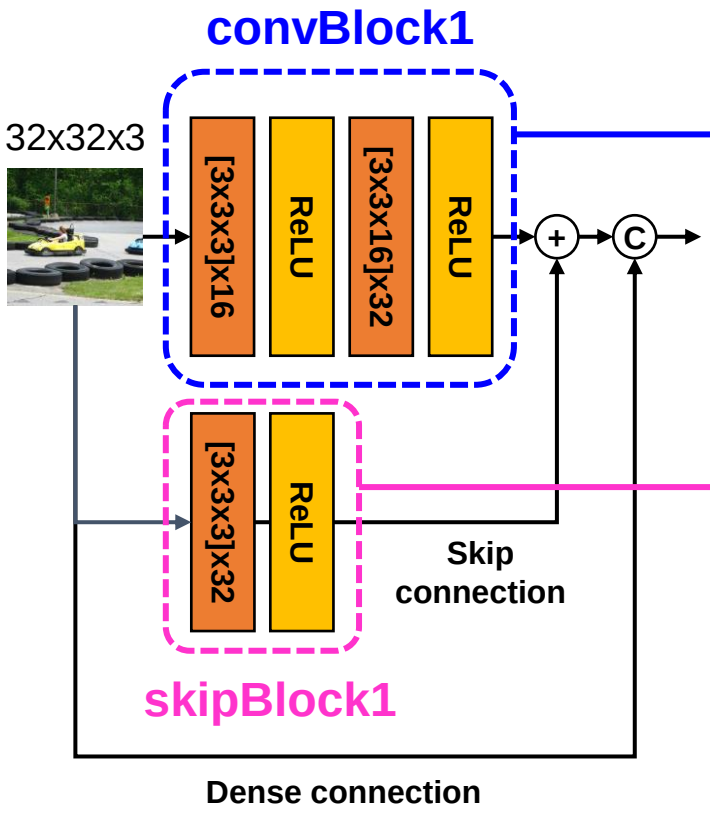
기말고사 연습 문제 [1/4] - 참고사항

- **Convolution Neural Network (CNN)**
  - 신경망 구조 그림을 제시하고 pytorch 라이브러리로 구현하는 문제 유형 입니다.
  - 실습에서 배웠던 신경망 모듈들 (Conv layer, FC layer, Skip 등)을 조합할 수 있습니다.
  - 신경망 층이 깊어지면 소스코드 및 변수명이 복잡해서 구현에 어려울 수 있습니다.
  - 하지만 수업 중 구두로 설명한 내용대로 nn.Sequential() 함수를 이용해 복수개의 계층을 그룹화 할 수 있습니다.
  - 그룹핑을 수행할 단위, 변수 명은 각자가 헷갈리지 않도록 정해주시면 됩니다.
  - nn.Sequential() 함수를 사용하지 않더라도 파라미터 수, 성능은 동일하게 나오며, 활용하지 않아도 상관 없습니다.



기말고사 연습 문제 [1/4] - 참고사항

- **Convolution Neural Network (CNN)**
  - torch.nn.Sequential() 함수를 통해 복수개의 신경망 계층들을 그룹화 가능
  - 아래 1개 블록에 대한 예시



❖ **__init__() 함수 – 신경망 파라미터 정의**

```
self.convBlock1 = nn.Sequential(
        nn.Conv2d(in_channels=3, out_channels=16, kernel_size=3, stride=1, padding=1),
        nn.ReLU(),
        nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, stride=1, padding=1),
        nn.ReLU(),
    )
```

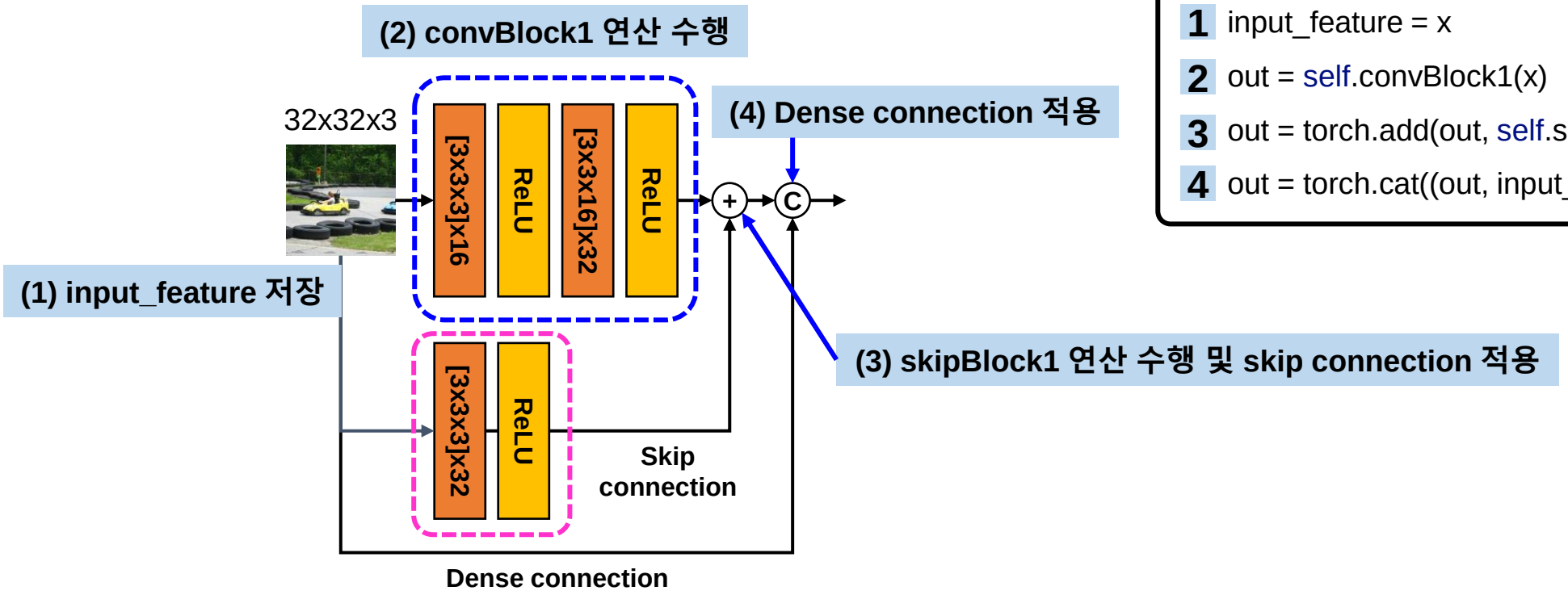
```
self.skipBlock1 = nn.Sequential(
        nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, stride=1, padding=1),
        nn.ReLU(),
    )
```

기말고사 연습 문제 [1/4] - 참고사항

- **Convolution Neural Network (CNN)**
  - torch.nn.Sequential() 함수를 통해 복수개의 신경망 계층들을 그룹화 가능
  - 아래 1개 블록에 대한 예시

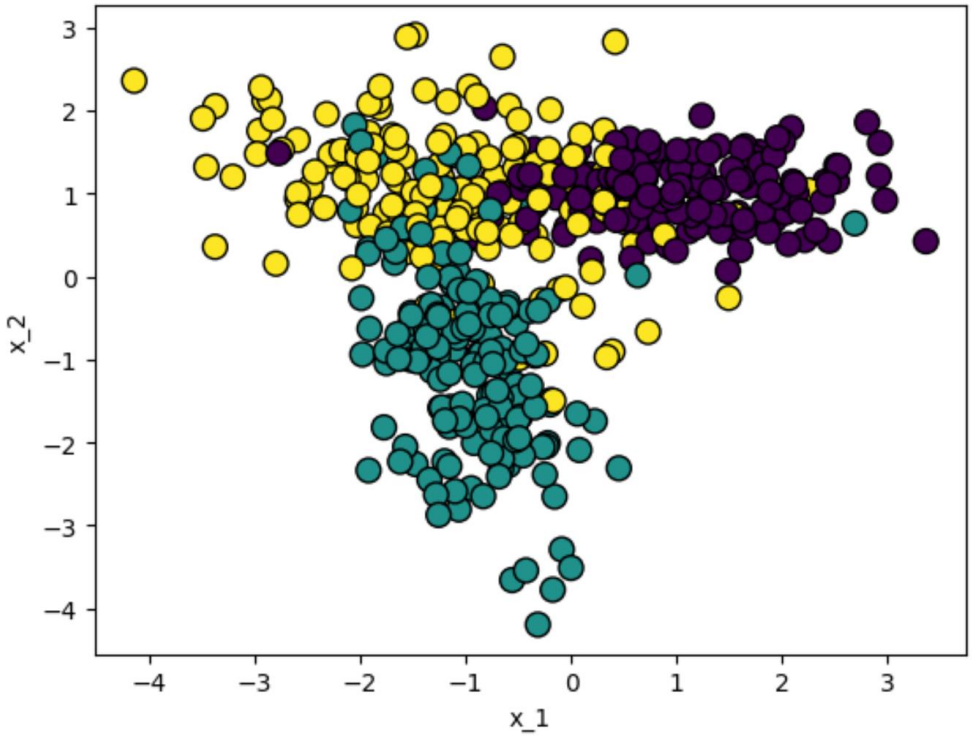
❖ **forward() 함수 – forward 순서 및 입출력 정의**

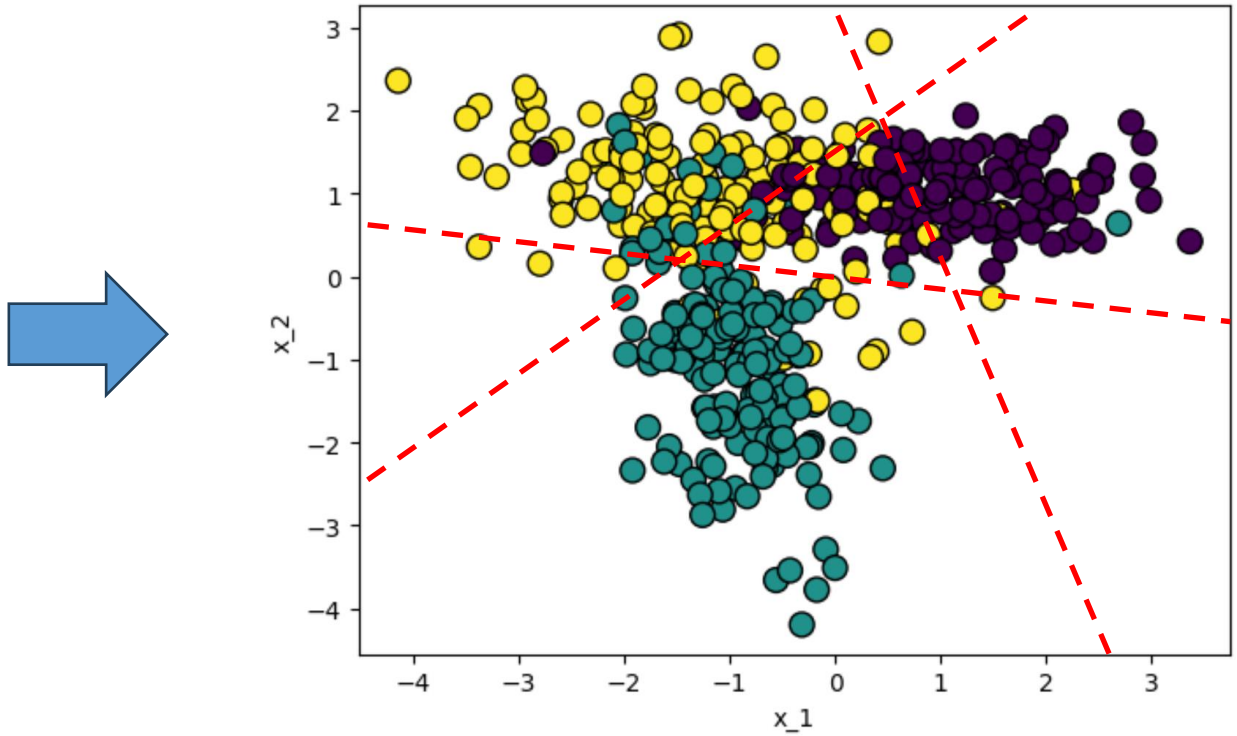
```
def forward(self, x):
1  input_feature = x
2  out = self.convBlock1(x)
3  out = torch.add(out, self.skipBlock1(input_feature))
4  out = torch.cat((out, input_feature), dim=1)
```



기말고사 연습 문제 [2/4]

- **Support Vector Machine (SVM)**
  - 3개 클래스를 가지는 데이터에 대해 분류를 수행하는 SVM 모델 생성

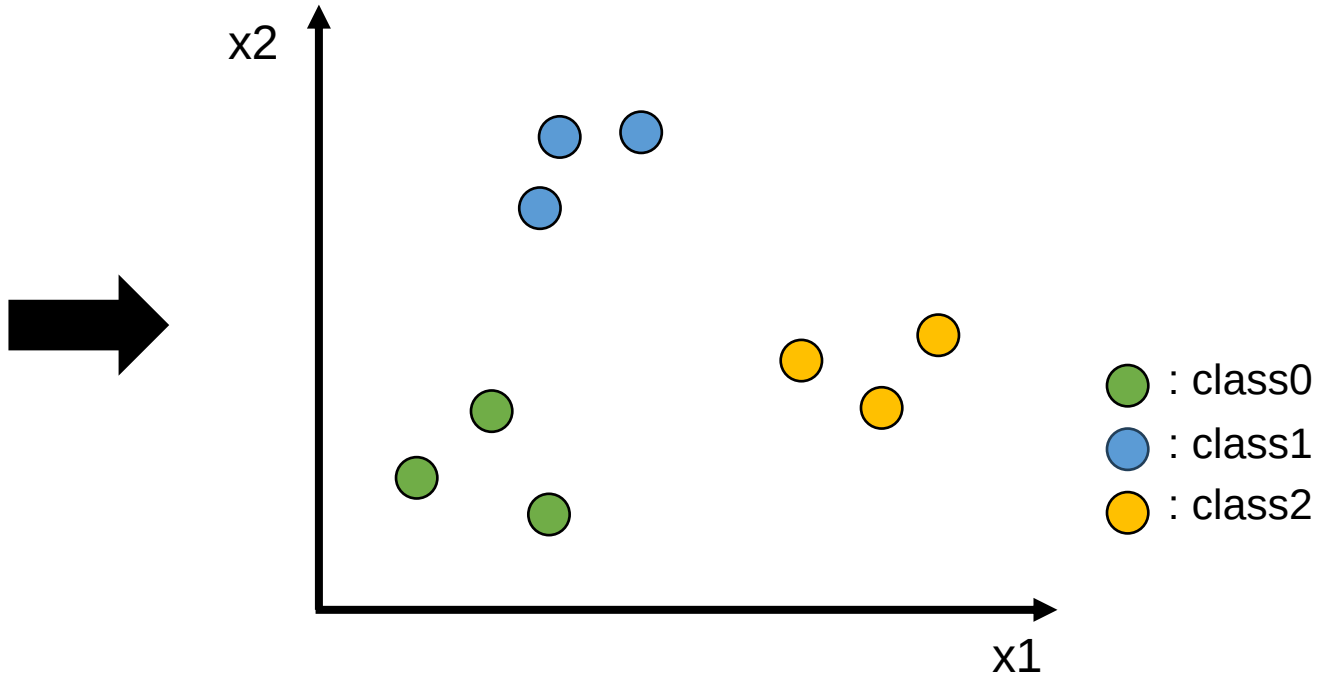




기말고사 연습 문제 [2/4] - 참고사항

- **Support Vector Machine (SVM)**
  - 일반적으로 SVM은 2개 클래스를 가지는 데이터만 분류할 수 있습니다. (Binary classification)
  - 본 문제에서는 3개 클래스를 가지는 데이터를 분류해야 하므로,
    여러 개의 SVM (Binary classification) 모델을 함께 사용해 구현할 수 있습니다.

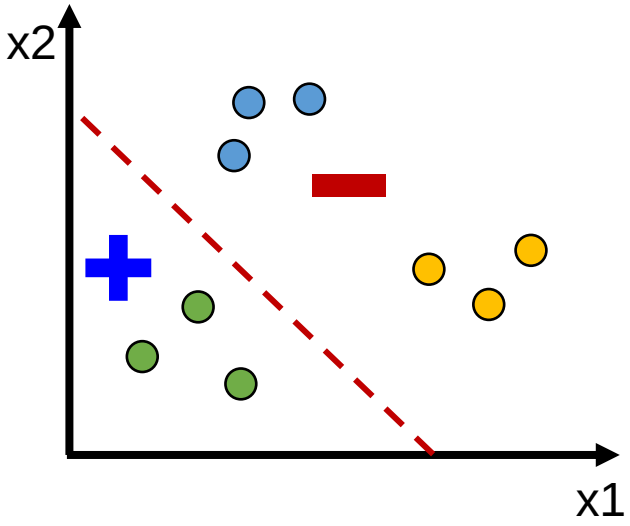
X (features)		y (class)
$x_{1,0}$	$x_{2,0}$	y_0
$x_{1,1}$	$x_{2,1}$	y_1
$x_{1,2}$	$x_{2,2}$	y_2
$x_{1,3}$	$x_{2,3}$	y_3
$x_{1,4}$	$x_{2,4}$	y_4
…	…	…
$x_{1,8}$	$x_{2,8}$	y_8



기말고사 연습 문제 [2/4] - 참고사항

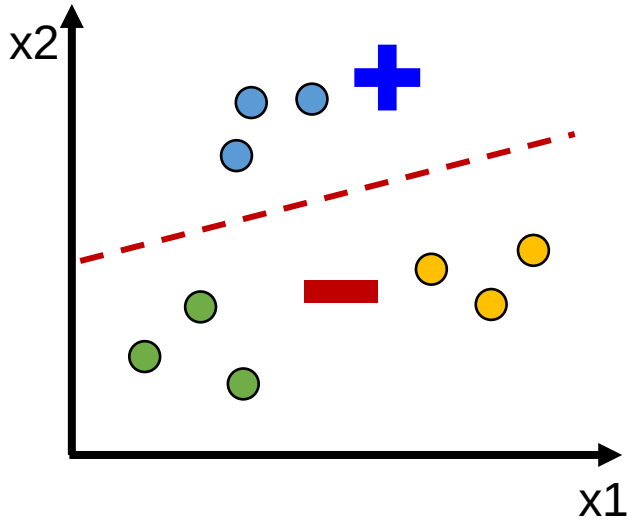
- **Support Vector Machine (SVM)**
  - 일반적으로 SVM은 2개 클래스를 가지는 데이터만 분류할 수 있습니다. (Binary classification)
  - 본 문제에서는 3개 클래스를 가지는 데이터를 분류해야 하므로,
    여러 개의 SVM (Binary classification) 모델을 함께 사용해 구현할 수 있습니다.

: class0
: class1
: class2



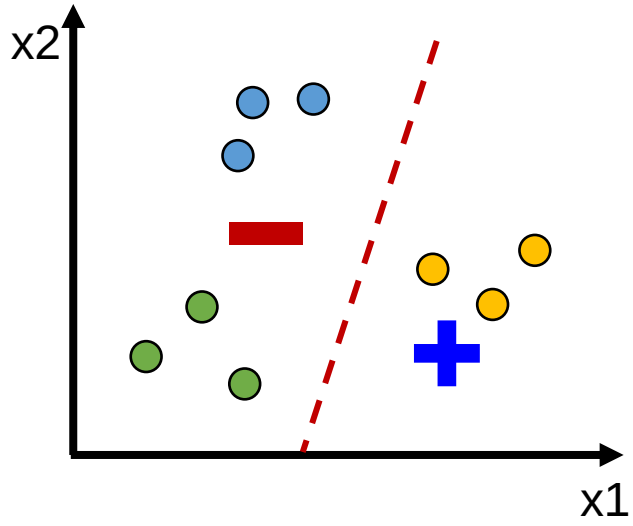
SVM1: Class0 분류

Class0 → positive sample
Class1/2 → negative sample



SVM2: Class1 분류

Class1 → positive sample
Class0/2 → negative sample



SVM3: Class2 분류

Class2 → positive sample
Class0/1 → negative sample

기말고사 연습 문제 [2/4] - 참고사항

- **Support Vector Machine (SVM)**
  - 서로 다른 class를 분류하는 SVM을 따로 학습하기 위해,
    fit() 함수를 임의의 class에 대해서만 학습하도록 변경할 수 있습니다.

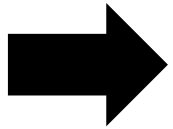
❖ **기존 실습에서 사용한 SVM fit() 함수**

```
def fit(self, X, y):
    n_samples, n_features = X.shape

    # 레이블을 -1 또는 1로 변환
    y_modified = np.where(y <= 0, -1, 1)

    # Weight 및 bias 초기화
    self.weights = np.zeros(n_features)
    self.bias = 0

    # Gradient Decent
    …
```

❖ np.where() 함수 설명 블로그 자료

❖ **2개 클래스를 가지는 데이터 예시**

y (class)
$y_0 = 0$
$y_1 = 1$
$y_2 = 1$
$y_3 = 0$
$y_4 = 1$
…
$y_8 = 1$



y (class)
$y_0 = -1$
$y_1 = +1$
$y_2 = +1$
$y_3 = -1$
$y_4 = +1$
…
$y_8 = +1$

기말고사 연습 문제 [2/4] - 참고사항

- **Support Vector Machine (SVM)**
  - 서로 다른 class를 분류하는 SVM을 따로 학습하기 위해,
    fit() 함수를 임의의 class에 대해서만 학습하도록 변경할 수 있습니다.

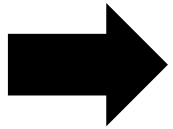
❖ **특정 class만 학습하도록 변형한 SVM fit() 함수**

```
def fit(self, X, y, target_class):
    n_samples, n_features = X.shape

    # 레이블을 -1 또는 1로 변환
    y_modified = np.where(y != target_class, -1, 1)

    # Weight 및 bias 초기화
    self.weights = np.zeros(n_features)
    self.bias = 0

    # Gradient Decent
    …
```

❖ np.where() 함수 설명 블로그 자료

❖ **3개 클래스를 가지는 데이터 예시**

y (class)
$y_0 = 2$
$y_1 = 1$
$y_2 = 2$
$y_3 = 0$
$y_4 = 0$
…
$y_8 = 1$



target_class=2

y (class)
$y_0 = +1$
$y_1 = -1$
$y_2 = +1$
$y_3 = -1$
$y_4 = -1$
…
$y_8 = -1$

기말고사 연습 문제 [2/4] - 참고사항

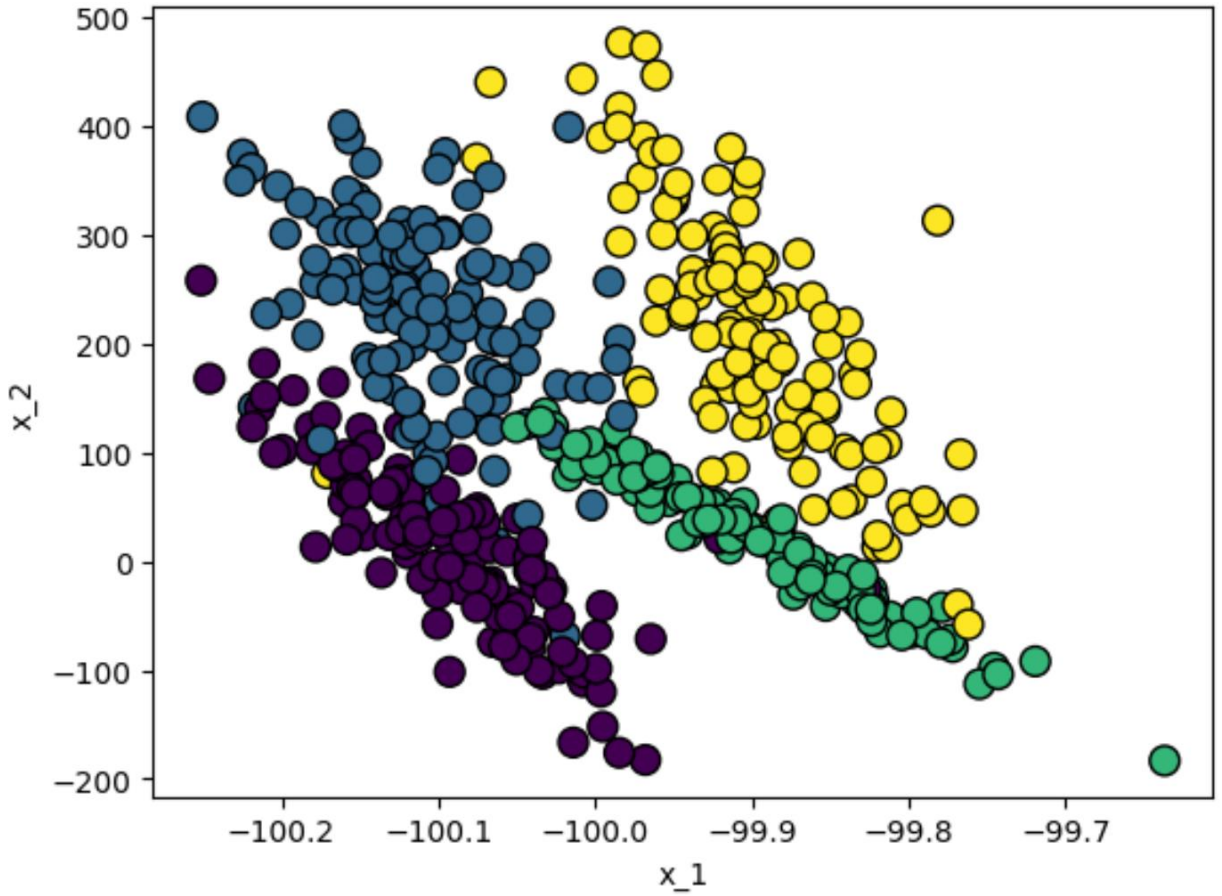
- **Support Vector Machine (SVM)**
  - 이후 서로 다른 class를 분류하는 SVM 모델을 복수개 선언하고, 각각의 class를 예측하도록 학습을 수행하면 됩니다.

❖ **SVM 모델 학습 및 hyperplane (w, b) 출력**

```
# model1 -> class0 분류
model1 = SVM()
margin_log = model1.fit(X, y, target_class=0)
print(model1.weights, model1.bias)

# model2 -> class1 분류
model2 = SVM()
margin_log = model2.fit(X, y, target_class=1)
print(model2.weights, model2.bias)

# model3 -> class2 분류
model3 = SVM()
margin_log = model3.fit(X, y, target_class=2)
print(model3.weights, model3.bias)
```

기말고사 연습 문제 [3/4]

- **K Nearest Neighbors (KNN)**
  - x1, x2 데이터에 대해 각각 Gaussian 정규화 수행 후 KNN 학습 및 예측 수행



Gaussian 정규화

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

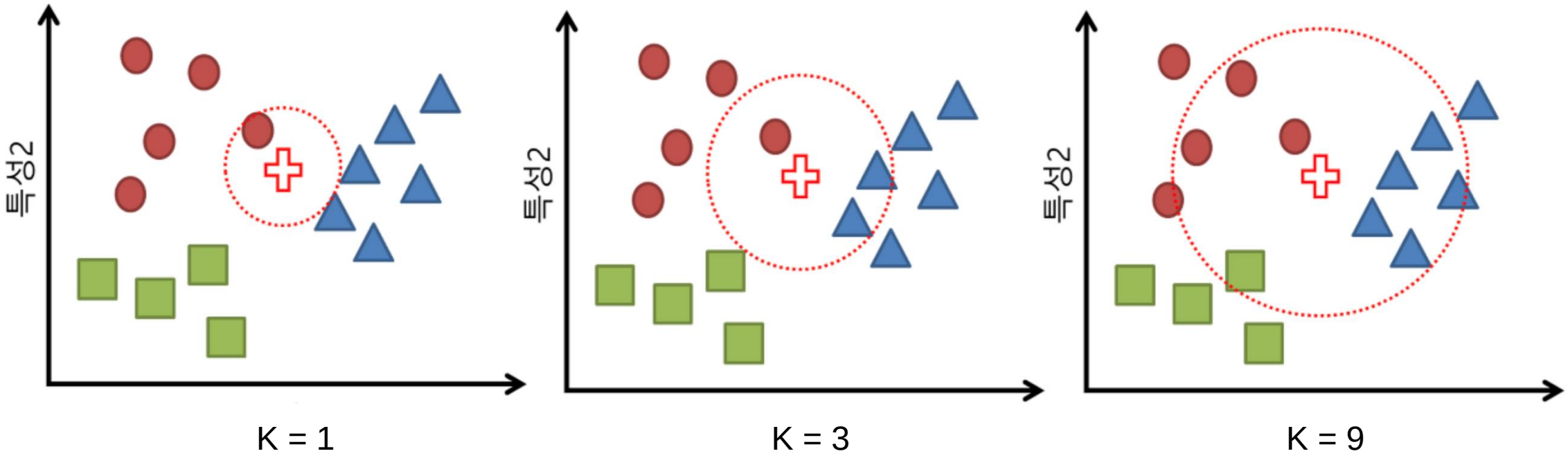
- μ: 평균
- σ: 표준편차

기말고사 연습 문제 [3/4] - 참고사항

- **K Nearest Neighbors (KNN)**
  - **x1, x2 데이터**
    - ✓ 현재 basecode에는 X 변수로 x1, x2 데이터에 각각 접근할 수 있습니다.
    - ✓ X는 2차원 numpy array로 타입으로, 데이터 개수 x feature 개수로 구성이 되어있습니다.
    - ✓ 따라서, x1 데이터는 X[: , 0] 으로 접근, x2 데이터는 X[: , 1]로 접근할 수 있습니다.
  - **Gaussian 정규화**
    - ✓ Linear regression 실습과 같이 np.mean() 함수로 평균, np.std() 함수로 표준편차를 계산할 수 있습니다.
    - ✓ x1 데이터에 대한 gaussian 정규화 예시
      - ➢ **X[: , 0] = (X[: , 0] - np.mean(X[: , 0])) / np.std(X[: , 0])** ⟵ $x_i' = \frac{x_i - \mu}{\sigma}$
  - **KNN 학습 및 예측 수행**
    - ✓ 14주차 화요일 강의자료, 강의영상 또는 직접 작성한 KNN 코드를 활용해주세요.

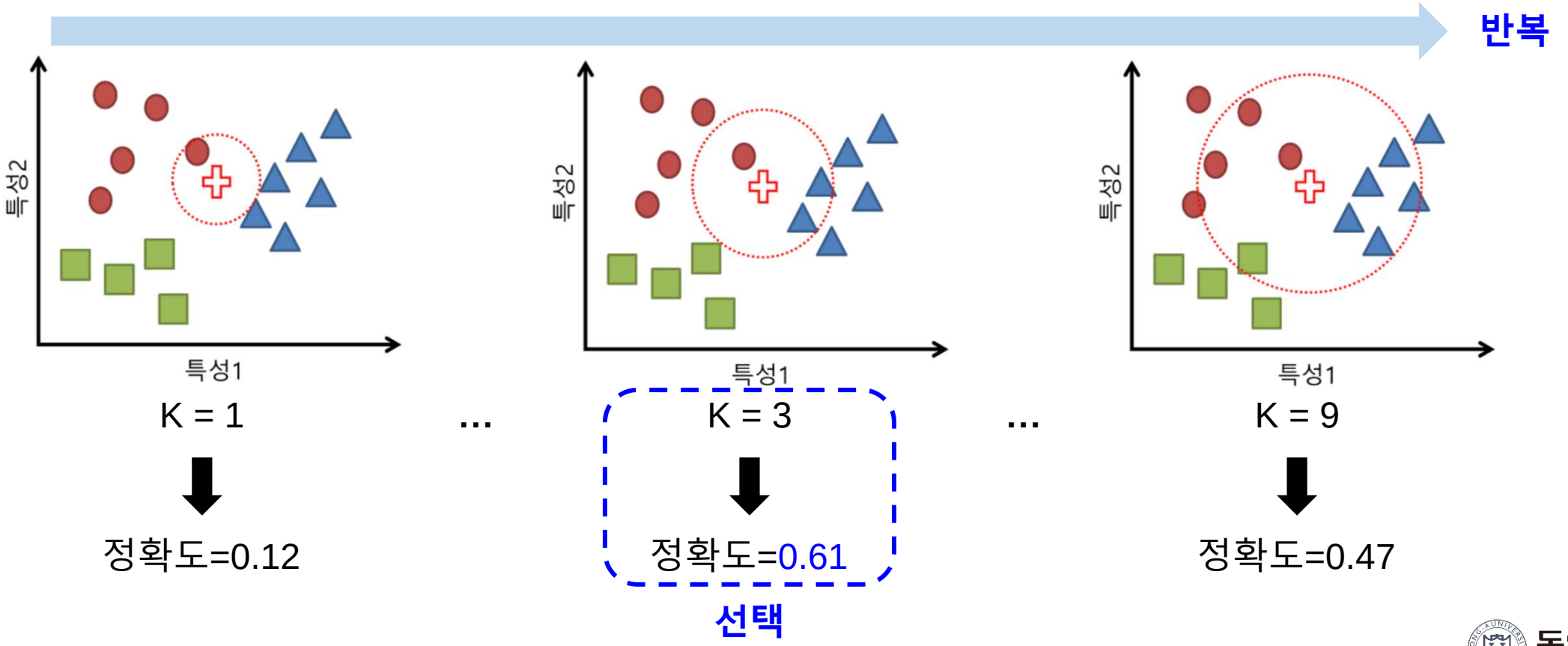
기말고사 연습 문제 [4/4]

- **K Nearest Neighbors (KNN)**
  - Training dataset을 80% 학습 데이터 / 20% 테스트 데이터셋으로 분류 후, 최적의 k값 도출



기말고사 연습 문제 [4/4] - 참고사항

- **K Nearest Neighbors (KNN)**
  - K 값에 대한 반복 (Ex. 1~9)을 수행하면서 각각에 대한 성능 확인 및 가장 높은 성능 가지는 K 값을 찾으면 됩니다.
  - 아래 그림은 예시입니다.



수업 관련 질의응답

- **본 자료 중 오류가 있거나 이론/실습 수업 관련 질문은 아래 경로를 통해 해주시기 바랍니다.**
  - **이메일 문의 (수업 조교)**
    - ✓ 박병주 석사과정: bjpark@donga-ispl.kr
    - ✓ 김선재 박사과정: sjkim@donga-ispl.kr
  - **학생연구실 방문**
    - ✓ 위치: 승학캠퍼스 공대1호관 (S03) 210호 영상신호처리연구실
    - ✓ 방문 전 미리 이메일 또는 연구실 전화 (051-200-5593) 로 약속을 잡아주세요.

- **시험 내용과 관련한 질문은 12월 13일 오후 6시 까지만 받도록 하겠습니다.**

Questions & Answers

Dongsan Jun (dsjun@dau.ac.kr)
Image Signal Processing Laboratory (www.donga-ispl.kr)
Dept. of Computer Engineering
Dong-A University, Busan, Rep. of Korea