


```

# 결정 경계 그래프
from matplotlib.colors import ListedColormap

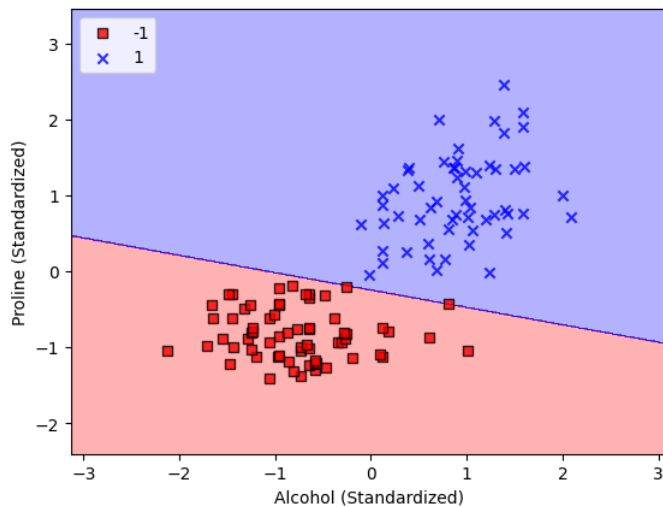
# 결정 경계 그래프 함수 정의
def plot_decision_regions(X, y, classifier, resolution=0.01):
    markers = ('s', 'x', 'o', '^', 'v')
    colors = ('red', 'blue', 'lightgreen', 'gray', 'cyan')
    cmap = ListedColormap(colors[:len(np.unique(y))])

    x1_min, x1_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    x2_min, x2_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx1, xx2 = np.meshgrid(np.arange(x1_min, x1_max, resolution), np.arange(x2_min, x2_max, resolution))
    Z = classifier.predict(np.array([xx1.ravel(), xx2.ravel()]).T)
    Z = Z.reshape(xx1.shape)

    plt.contourf(xx1, xx2, Z, alpha=0.3, cmap=cmap)
    plt.xlim(xx1.min(), xx1.max())
    plt.ylim(xx2.min(), xx2.max())
    for idx, cl in enumerate(np.unique(y)):
        plt.scatter(x=X[X[y == cl, 0], y=X[y == cl, 1], alpha=0.8, c=colors[idx], marker=markers[idx], label=cl, edgecolor=None if idx == 1 else

# 결정 경계 그래프 출력
plot_decision_regions(X, y, classifier=ppn)
plt.xlabel('Alcohol (Standardized)')
plt.ylabel('Proline (Standardized)')
plt.legend(loc='upper left')
plt.show()

```



Linear classifier

$$f(x, W) = Wx + b$$

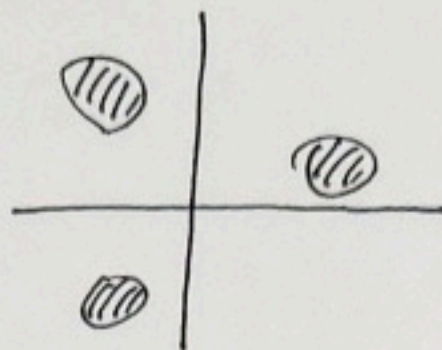
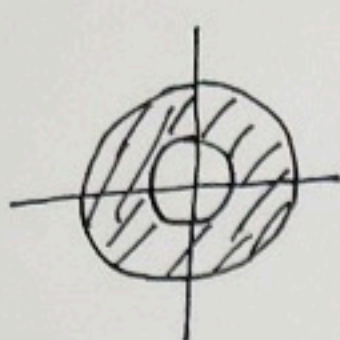
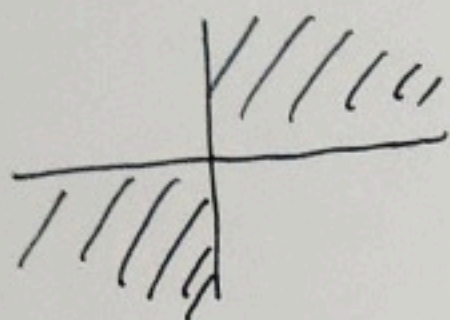
W : 가중치

x : classes

W 를 잘 만들어야 잘 예측

W 를 주어진 데이터로 학습

- 선형이기 때문에 특징에 대해 다음과 같이 그려진다면 예측이 어려움



⇒ 모든 특징을 사용하지 않고
중요한 특징만 선택하면 유리

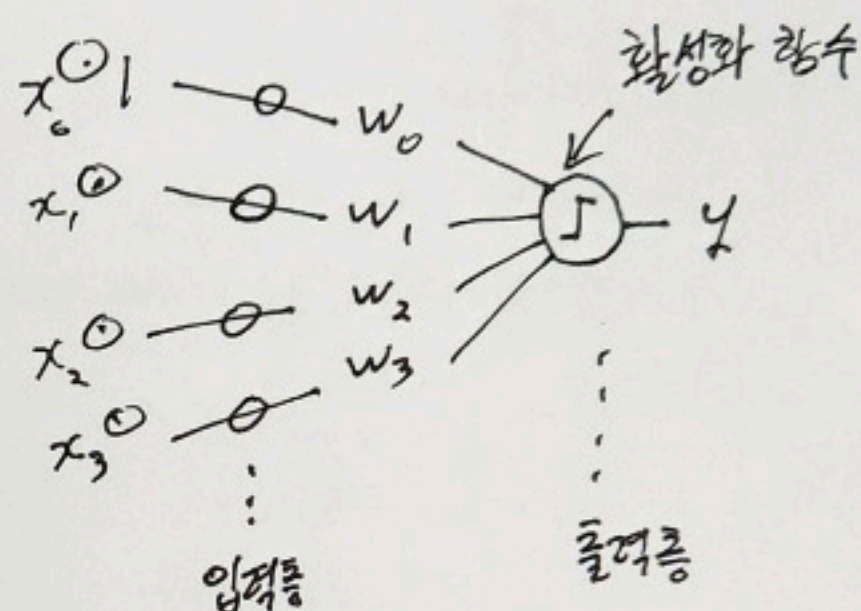
퍼셉트론

- 사람의 뉴런과 유사

- 입력층과 출력층을 가짐

- 특징값과 가중치를 곱한 후에 활성화 함수를 적용

- 최종 출력 값은 +1 or -1



Gradient : 경사하강법

$$J(w) = \sum_{x_k \in Y} -y_k (w^T x_k)$$

$$\frac{\partial J(w)}{\partial w_i} = \sum_{x_k \in Y} \frac{\partial (-y_k (w_0 x_{k0} + w_1 x_{k1} + \dots + w_d x_{kd}))}{\partial w_i}$$

$$= \sum_{x_k \in Y} -y_k x_{ki}, \quad i = 0, 1, \dots, d$$

배치모드 : Batch

- 배치단위로 업데이트

에포크 : epoch (세대)

- 모델이 한번 학습하는 것

Stochastic 모드

- 특정한 샘플이 발생하면 즉시 갱신

- 샘플 순서를 섞음

→ 선형 분리 불가능하면 무한 반복

목적함수 : $J(\theta)$, $J(w)$

- 퍼셉트론의 목적함수 : 오분류된 샘플의 집합

- 조건

- $J(w) \geq 0$
- w 가 최적이면 $J(w) = 0$
- 특정한 샘플이 많으면 $J(w)$ 는 큰 값

$$J(w) = \sum_{x_k \in Y} -y_k (w^T x_k)$$

이미지 분류

- 문제
- View point variation
 - Illumination
 - Background clutter
 - Occlusion
 - Deformation
 - Intra class variation
 - Context

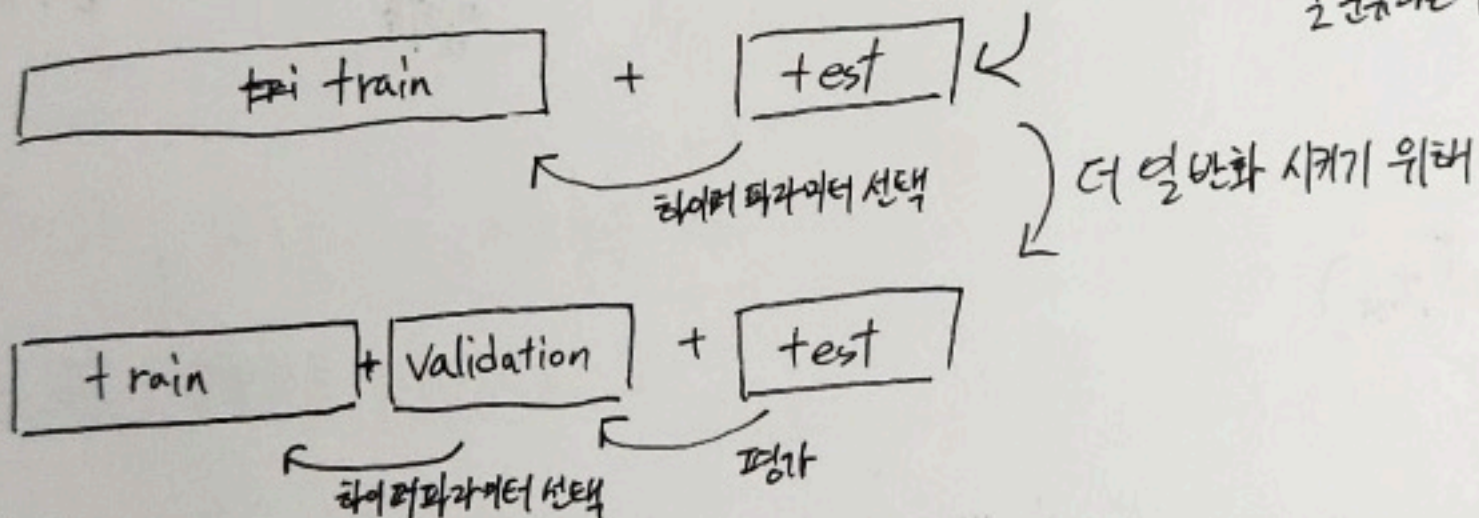
Nearest Neighbor

- 데이터와 가장 비슷한 값으로 선택
- image data 일 때는 image의 픽셀값의 차이를 가지고 트레이닝 이미지를 선택
- 무식한 방법 (여러 예외 사항이 발생)

K-Nearest Neighbor

- 하이퍼 파라미터 : K , distance
- 어떤 파라미터가 좋을지 평가 정수가 높은 파라미터 선택

Setting Hyperparameters



Cross-Validation : 표준단위로 분류

