

# 효과적인 네트워크 훈련

정보컴퓨터공학과 권혁동

검증 세트 확보

스케일 적용

벡터화

# 검증 세트 확보

- 딥러닝 모델을 수정할 때는 **하이퍼 파라미터**를 수정
  - 일반 파라미터: bias, weight
  - 하이퍼 파라미터: loss function
  - 딥러닝 설계자가 선택 가능한 항목
- 하이퍼 파라미터 수정을 **튜닝**이라 칭함
- 테스트 세트로 튜닝을 진행할 경우, 실전 성능 하락
- **검증 세트**가 필요

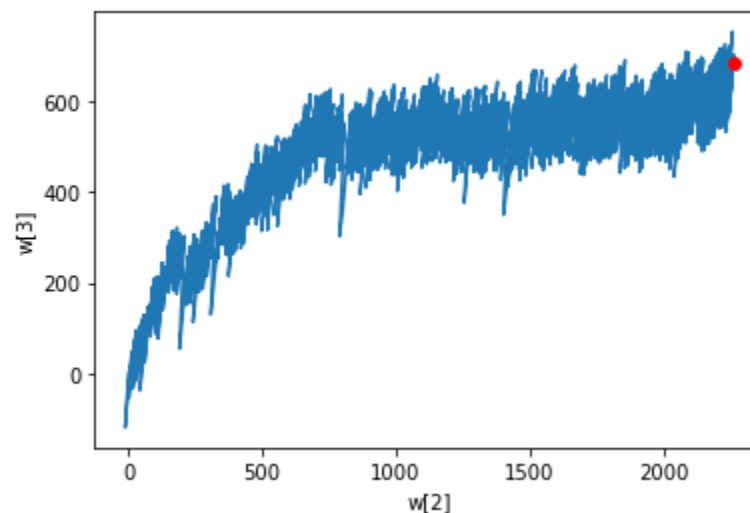
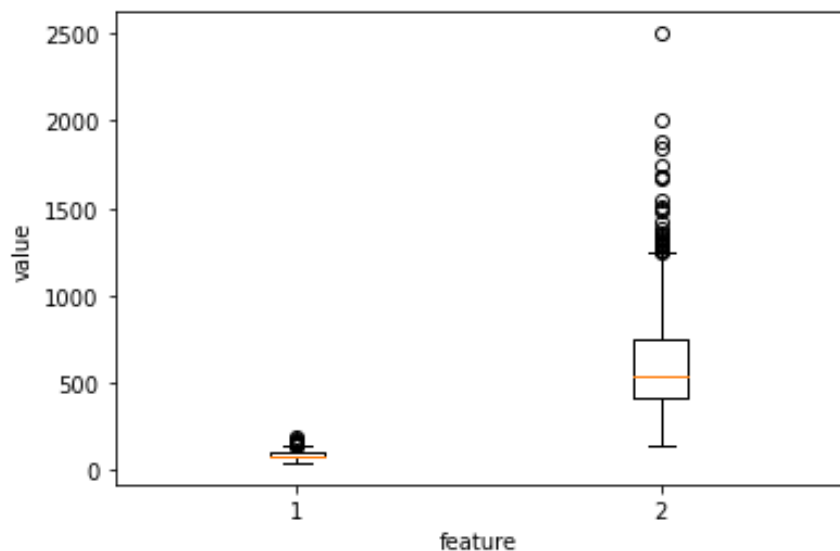
# 검증 세트 확보

- 기존 테스트 데이터를 6:2:2로 분리
  - (훈련 세트) : (검증 세트) : (테스트 세트)
- **모델 평가시에는 검증 세트로 평가**
  - 훈련 세트는 검증 세트에 비해 정확도가 높게 나옴

Test(60)	Val(20)	Test(20)
----------	---------	----------

# 스케일 적용

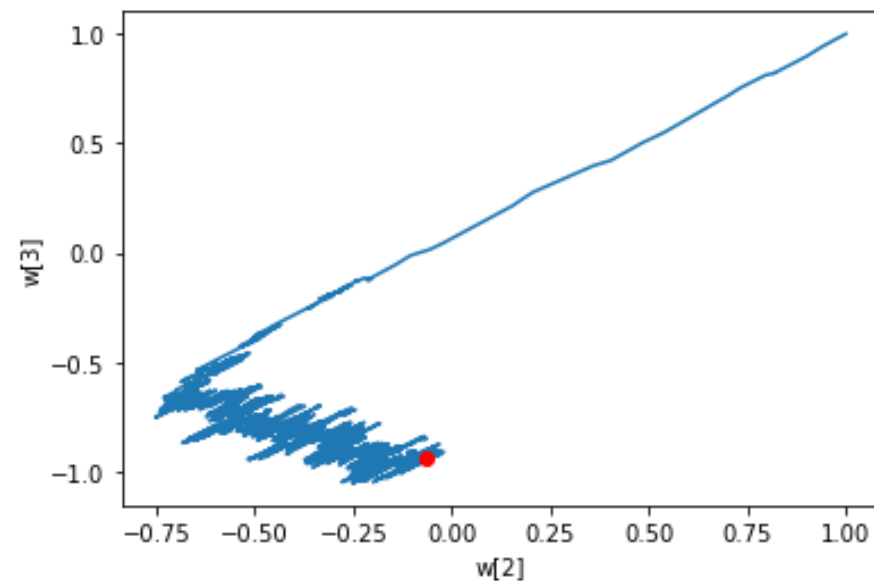
- 스케일: 어떤 특성이 가지고 있는 **값의 범위**
  - 예시의 perimeter는 100~200, area는 200~2000
  - 두 특성의 스케일 차이가 큼
- 스케일을 고려하지 않을 시 모델이 불안정하게 수렴



# 스케일 적용

- 스케일을 조정하여 안정적으로 훈련
- **표준화(standardization)** 사용
- $Z = \frac{x - \mu}{s}$ 
  - Numpy에는 표준(mean)과 표준 편차(std) 메소드 제공
  - 표준화를 손쉽게 진행할 수 있음

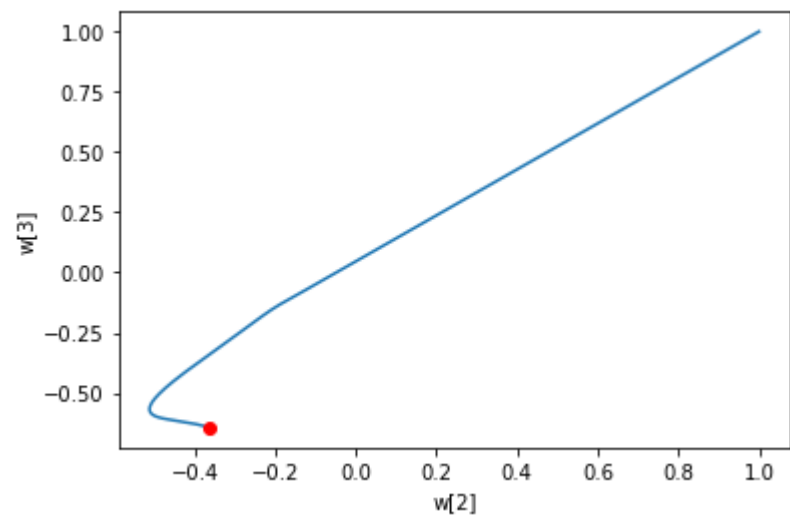
```
train_mean = np.mean(x_train, axis=0)
train_std = np.std(x_train, axis=0)
x_train_scaled = (x_train - train_mean) / train_std
```



# 벡터화

- 배치 경사하강법은 학습에 매우 유리
- 하지만 **모든 샘플을 사용**하므로, **시간이 오래 걸림**
- 학습을 **벡터화**하는 것으로, 빠른 속도로 학습을 가능하게 수정
- 벡터화를 위해 **행렬 곱(dot product)**을 사용
  - numpy의 dot 활용

$$\begin{aligned} \mathbf{XW} &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \\ &= w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3 \end{aligned}$$



Q & A