[Week11]_이원주

하이퍼 파라미터 튜닝

종류, 중요도

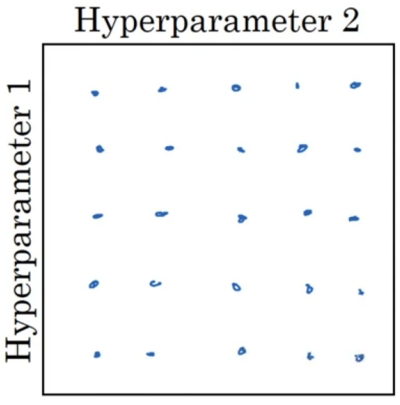
🌟 **[요약]**

하이퍼 파라미터 list

Aa 하이퍼 파라미터	설명
α	learning rate
β	모멘텀 알고리즘의.
β_1, β_2, ϵ	Adam 알고리즘의.
# layers	은닉층의 갯수
# hidden unit	은닉 유닛의 수
learning rate decay(감쇠) 정도	-
mini-batch size	미니배치 크기

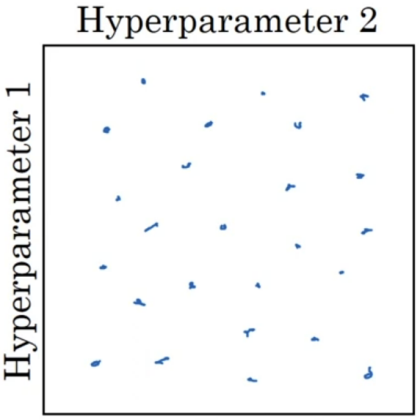
튜닝 프로세스

- **격자**(위) **말고 무작위**(아래)**로** 정하기
  - 예를 들어
    하이퍼 파라미터1 = α
    하이퍼 파라미터2 = ε
    라고 하면
  - α는 진짜 중요하고, ε는 영향X이므로
  - 격자(위)
    - (α 5개) * (ε 5개)
    - ε값은 바뀌어도 결과에 영향X,
      실질적으로 α 값만 5개 해본 셈.



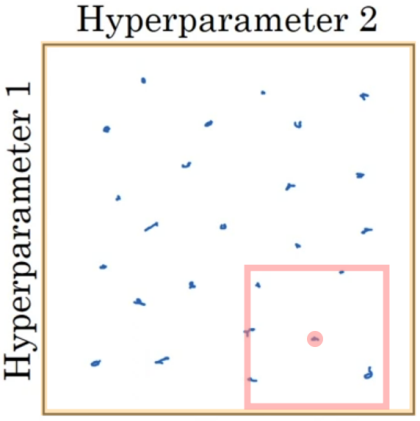
격자

- 무작위(아래)
  - 개수는 똑같이 25개, 무작위로 점 찍기
  - α 값을 25개 해본 셈.
- 사실 이 경우라면 ε은 바꿀 필요 없고 α 값만 바꿔서 해보면 되지만,
  실제로는
  어떤 파라미터가 가장 중요한지 미리 알기는 어렵기 때문에 무작위로 하는 게 좋다!



무작위

- 원한다면 **정밀화 접근** 이용 가능
  - 전체에서 한 번 무작위로 25개 찍었지. (노랑)
  - 그중 최적인 1개 주변에서 다시 25개 찍어서
    한 번 더 해보는 것.
    (분홍)



무작위로 고르는 방법

- 어떤 하이퍼 파라미터는 무작위로 고를 때 → (ex) 학습률
  - 균일하게 (X)
    - 0.0001 ~ 1 중에 완전 랜덤하게!
      근데 그러면 대부분의 값은 0.1 ~ 1이 나올걸.
    - = 선형 척도
  - **적절한 척도 선택해서** (O)
    - 0.0001 ~ 0.001 ~ 0.01 ~ 0.1 ~ 1
      각 구역 확률 동일하게 → 로그 척도 사용

    ▼ 코드

0.0001	0.001	0.01	0.1	1
10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}

r `=` -4 * `np.random.rand()`

α `=` 10^{r}

- r
  - [-4, 0] 사이 랜덤한 수
  - $\log \alpha$ → 로그 척도
- 근데 너무 부담갖진 말기.
  어차피 정밀화 관점 쓰면 척도 틀려도 ㄱㅊ.

여러 하이퍼 파라미터를 실험하는 방법

- 모델 돌보기 (Babysitting one model)
  - 컴퓨터 자원이 충분하지 않아서 → 한 번에 여러 모델을 동시에 학습시킬 수 없는 경우
  - 한 모델을 가지고 애기 돌보듯이 계속 성능을 지켜보면서 학습시키는 것.
- 여러 모델을 병렬적으로 학습시켜서 → 결과 보고 하나 고르기

Tip

- 다른 영역 → 다른 하이퍼 파라미터
  - 자연어, 비전, 음성 → 다 다름.
- 같은 문제에 대해서도 → 시간이 지나면 하이퍼 파라미터가 여전히 잘 작동하는지 점검 필요.
  - 왜냐면 환경이 조금 달라지면 하이퍼 파라미터도 바뀌어야 하기 때문.

배치 정규화

Batch Normalizing

배경지식

- 정규화 (Normalizing)

필요성 (장점)

- 하이퍼 파라미터에 관계 없이 → 튼튼한 모델을 만들어줌.
  - = 신경망과 하이퍼파라미터의 상관관계를 줄여줌.
  - = 더 많은 하이퍼 파라미터가 잘 작동하게 만들어줌.
- 하이퍼파라미터 탐색을 쉽게 만들어줌.

▼ 이유

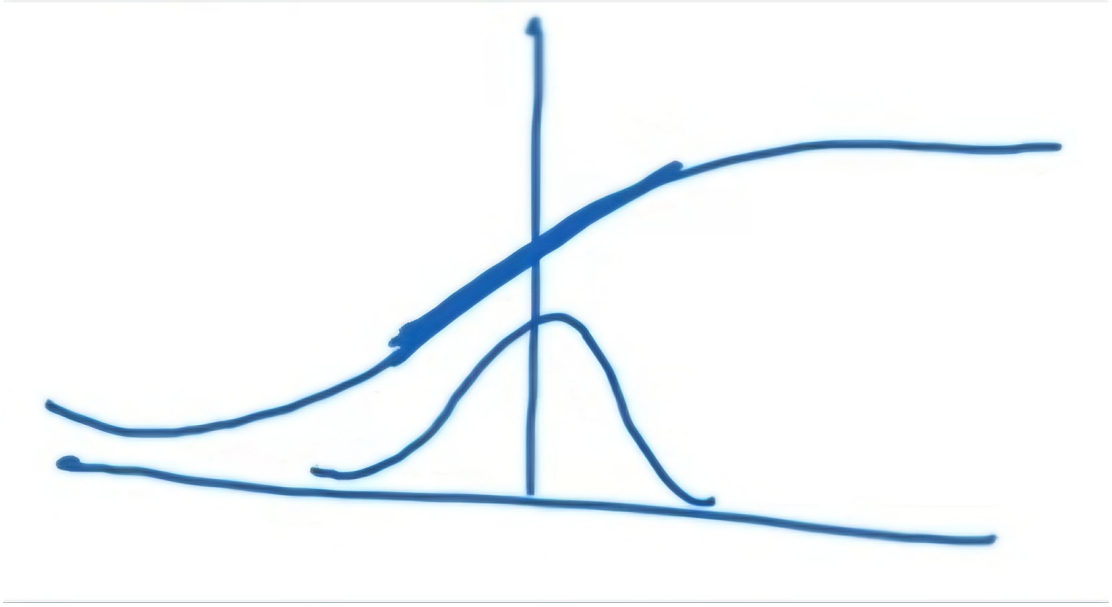
- 각 layer의 input 분포를 어느정도 일정하게 유지해줌
  - 데이터 분포가 변하면 → 모델도 변해야 함.
  - 마찬가지로 hidden layer에서도 학습하면서 input의 분포가 확확 바뀌면 학습 어려워짐.

방법

- ***Aha!***
  - 우리가 전에 배운 건 input layer (= input data)의 정규화
  - 그런데 input layer의 out 말고

    hidden layer의 out (=z, a)에도 정규화를 하면 어떨까?
    둘 다 다음 layer에 input으로 들어가는 애들이잖아.
    - 보통 z를 많이 씀.
    - z 구하고 → 정규화해서 $\tilde{z}$ 구하고 → sigmoid 씌워서 a 구하고
  - 그럼 다음 은닉층의 학습에 도움이 될거야.

- **방법**
  - 전에 배운 방식대로 z_{norm}을 구함. 평균이 0이고 분산이 1인 정규분포를 따르게.
  - 근데 hidden layer에서는 분포를 좀 바꿔야 한단 말야? 정규분포대로면 z에 sigmoid (비선형 함수) 씌웠을 때 결과값이 거기서 거기가 된다고.



- 그래서 $\tilde{z} = z_{norm} * \gamma + \beta$로 평균, 분산 좀 바꿔줌.
- 이 γ, β는 hidden unit마다 다를 수 있음.

- 코드
  - 텐서플로 같은 거 쓰면 이미 만들어놓은 라이브러리 있어서 코드 한 줄로 가능.

- 이제 z 구할 때 b (bias)는 무시해도 됨.
  - 어차피 정규화하면 평균이 0 되니까.
  - 남은 파라미터는 w, γ, β

test 시의 배치 정규화

- 여태 배운 거에선 → mini batch에 대해 배치 정규화를 했음.
  - 따라서 각 unit에 들어오는 data 개수(= batch size)가 여러 개
- But 테스트 시에는 → batch size = 1
  - 정규화에 필요한 μ, σ

    (mini batch에서 data들의 평균과 분산)을 계산할 수 없음. data가 1개니까.
  - 따라서 training 때 얘네 추정치를 구해야 함.
    - mini batch마다 μ, σ 구했을 거 아냐. 걔네의 지수 가중 이동 평균을 사용.
    - 즉 $\mu^{\{1\}}$, $\mu^{\{2\}}$, ... → μ 추정치

      $\beta^{\{1\}}$, $\beta^{\{2\}}$, ... → β 추정치

- 사실 training 끝나고 실제 평균 구해도 되는데, 메모리 아끼려고 training 하면서 지수 가중 이동 평균으로 평균(추정치)도 같이 구하는 것.