

5. 하이퍼파라미터 튜닝

날짜 @2023년 11월 16일

▼ 목차

튜닝 프로세스

1 Hyperparameters

2 Try random values: Don't use a grid

3 Coarse to fine

적절한 척도 선택하기

1 Picking hyperparameters

2 Appropriate scale for hyperparameters

 파이썬으로 구현하기

 파이썬으로 구현하기

하이퍼파라미터 튜닝 실전

1 Re-test hyperparameters occasionally

2 Searching hyperparameters

출석퀴즈 오답노트: 100!

튜닝 프로세스



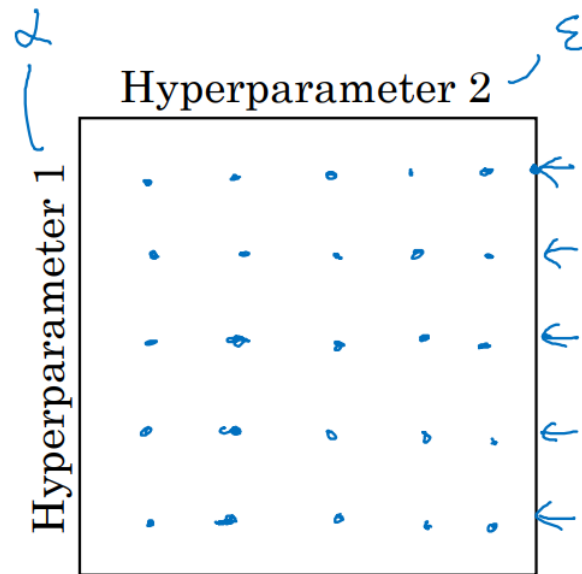
지금까지 배웠듯 신경망을 학습시킬 때에는 여러 하이퍼파라미터들이 관여하게 된다. 이러한 하이퍼파라미터를 효율적·체계적으로 튜닝하는 방법은 무엇일까?

1 Hyperparameters

- 심층신경망: 다뤄야 할 하이퍼파라미터가 많아 어려움
 - **학습률**, **모멘텀**, 층의 수, **은닉 유닛의 수**, 학습률 감쇠 이용 여부, **미니 배치의 크기**
 - Adam 최적화 알고리즘의 하이퍼파라미터($\beta_1, \beta_2, \epsilon$)는 일반적으로 튜닝하지 않음

2 Try random values: Don't use a grid

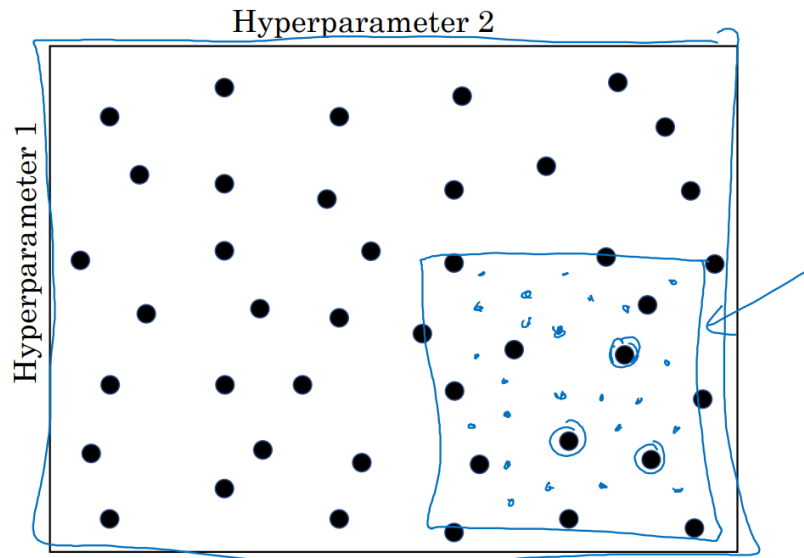
- 전통적인 머신러닝:
 - 2개 하이퍼파라미터가 있을 때, 다음과 같이 격자점을 탐색하여 가장 좋은 하이퍼파라미터를 정함



- 하이퍼파라미터의 수가 적을 때 이용할 수 있음
- 딥러닝:
 - 어떤 하이퍼파라미터가 문제 해결에 더 중요한지 미리 알 수 없으므로, 무작위로 점을 선택하여 일반적으로 중요도가 높다고 판단되는 하이퍼파라미터(e.g. 학습률)의 다양한 값을 탐색하는 것이 좋음
 - 정밀화 접근을 이용할 수 있음

3 Coarse to fine

- 가장 좋은 성능을 보이는 점을 찾았을 때, 그 근방의 점 또한 좋은 성능을 보임
- 정밀화 접근에서는 해당 영역을 확대하여 더 조밀하게 점을 선택함
 - 전체 사각형을 탐색한 뒤 파란 사각형으로 범위를 좁혀 재탐색하게 됨



적절한 척도 선택하기

📌 딥러닝에서는 무작위로 하이퍼파라미터를 찾는 것이 더 효율적이다. 그러나 이는 모든 하이퍼파라미터에 대하여 가능한 값들 중 공평하게 뽑는 것이 아닌, 적절한 척도를 정하여 그 기준 아래에서 뽑는 것이다. 그 척도를 정하는 방법은 무엇일까?

1 Picking hyperparameters

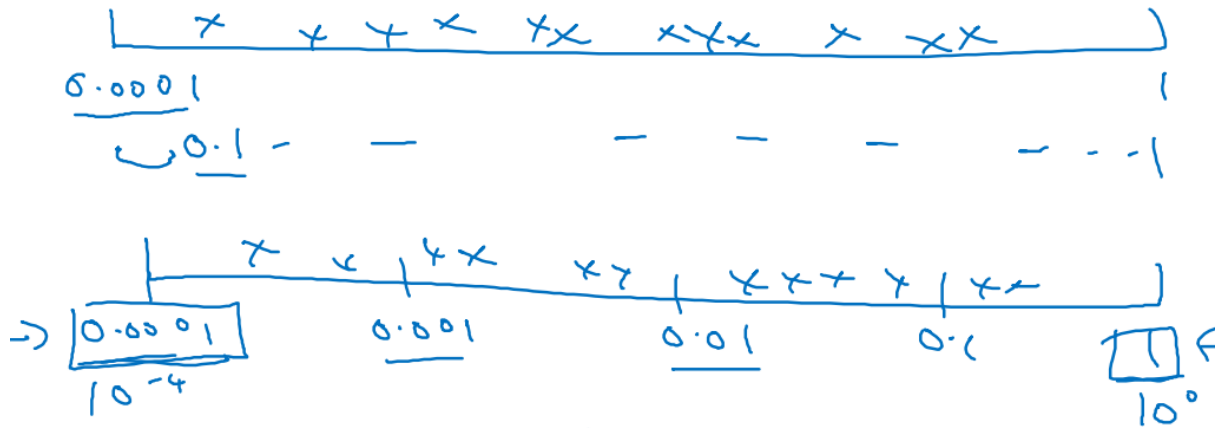
- 균일한 간격의 수직선에서 무작위로 값을 고르는 경우
 - 어떤 레이어 l 에 대하여 은닉 유닛의 수 n_l 를 정할 때 (범위: 50~100)
 - 신경망에서 레이어의 수 L 을 정할 때 (범위: 2~4)
 - 이 경우는 숫자가 작으므로 격자점을 사용해도 문제가 없음
- 선형 척도 대신 로그 척도의 수직선에서 무작위로 값을 고르는 경우
 - 학습률 α 를 정할 때 (범위: 0.0001~1)
 - 지수가중평균의 하이퍼파라미터 β 를 정할 때 (범위: 0.9~0.999)

2 Appropriate scale for hyperparameters

- 위 학습률 예시에서 0.0001~1 범위를 균일하게 무작위 탐색하는 경우, 약 90%의 샘플이 0.1과 1 사이에 위치하게 되어 비합리적임

- 로그 척도를 사용할 경우 0.0001과 0.001 사이, 0.001과 0.01 사이, 0.01과 0.1 사이, 0.1과 1 사이를 탐색할 때 동일한 양의 자원을 쓸 수 있음

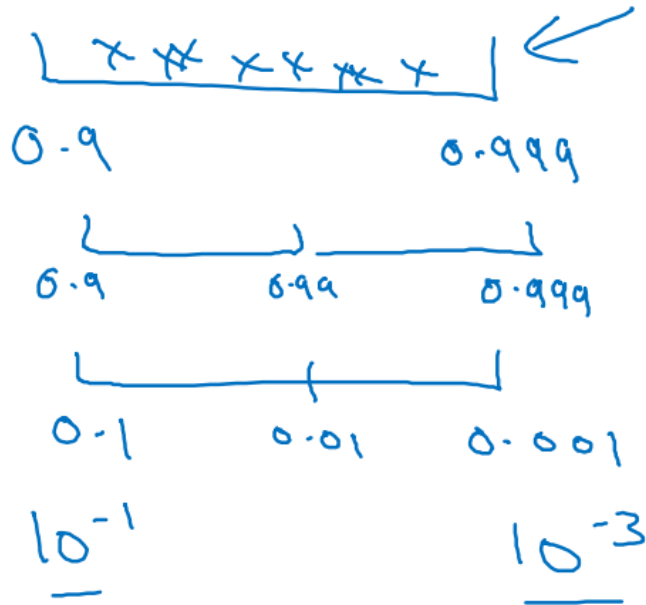
$$\alpha = 0.0001, \dots, 1$$



🚀 파이썬으로 구현하기

```
r = -4 * np.random.rand()
alpha = 10 ** r
```

- 지수가중평균 예시에서 0.9~0.999 범위의 값을 고를 때 또한 균일하게 무작위 탐색하는 것은 효율적이지 않으며, $1 - \beta$ 에 대하여 값을 탐색하는 것이 좋음
 - 결과적으로 0.001과 0.1 사이를 탐색하게 되며, 학습률 예시와 같이 로그 척도를 사용하여 β 값으로 0.9와 0.99 사이, 0.99와 0.999 사이를 탐색할 때 동일한 양의 자원을 쓸 수 있음



🚀 파이썬으로 구현하기

```
r = np.random.rand()
beta = 1 - 10 ** (-r - 1)
```

- ◆ 하이퍼파라미터 값이 1에 가까울수록 결과가 영향을 더 많이 받을 경우, 선형 척도 대신 로그 척도를 사용하면 가능한 결과 공간을 탐색할 때 더 효율적으로 샘플을 추출할 수 있음
- 만약 β 가 1에 가깝다면, β 가 0.9에서 0.9005로 바뀌는 것보다 0.999에서 0.9995로 바뀌는 것이 알고리즘의 결과에 더 큰 영향을 줌
- 적절한 척도를 사용하지 않았더라도, 정밀화 접근을 사용하여 반복함으로써 유의미한 하이퍼파라미터 범위를 얻을 수도 있음

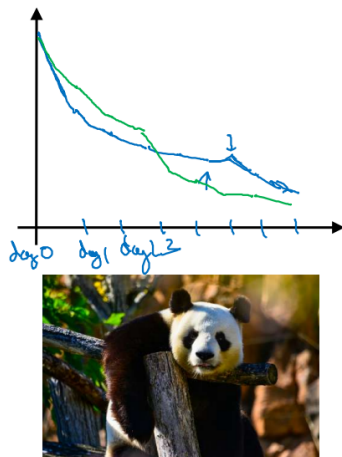
하이퍼파라미터 튜닝 실전

1 Re-test hyperparameters occasionally

- 특정 어플리케이션에서 얻은 하이퍼파라미터에 대한 직관은 다른 영역에 적용될 수도 있음
 - CV 분야의 컨브넷과 레스넷이 음성과 NLP에서도 사용됨
- 그러나 하이퍼파라미터를 찾는 과정은 그렇지 못함
 - 알고리즘의 발전, 데이터의 변화, 데이터 센터 서버의 업그레이드 등으로 최적의 하이퍼파라미터는 계속 바뀜

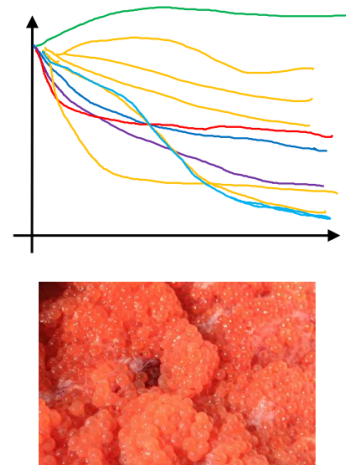
2 Searching hyperparameters

Babysitting one model



Panda ←

Training many models in parallel



Caviar ←

Andrew Ng

- 하나의 모델 돌보기 (Panda)
 - 하나의 모델에 집중. 며칠, 몇 주에 걸쳐 매일 성능을 지켜보며 하이퍼파라미터를 조금씩 바꾸는 방식
 - 컴퓨터 자원이 충분하지 않아 여러 모델을 동시에 학습시킬 수 없는 경우 이용
 - 온라인 광고, CV 등 많은 데이터가 쓰이는 곳에 해당
 - 한 모델이 잘 작동하는지 확인한 뒤 2~3주 후 다른 모델을 초기화하여 다시 돌보게 될 수 있음
- 여러 모델을 동시에 학습시키기 (Caviar)
 - 현재 가진 하이퍼파라미터를 여러 모델에 적용하여 며칠에 걸쳐 스스로 학습하도록 하고, 최고 성능을 보이는 것을 선정하는 방식

- 각 모델이 서로 다른 하이퍼파라미터를 다루게 되므로 여러 하이퍼파라미터를 시험해볼 수 있음
- 여러 모델을 동시에 학습시키기에 충분한 컴퓨터를 갖고 있는 경우 이용

출석퀴즈 오답노트: 100!