



11주차_하이퍼파라미터 튜닝

≡ 링크

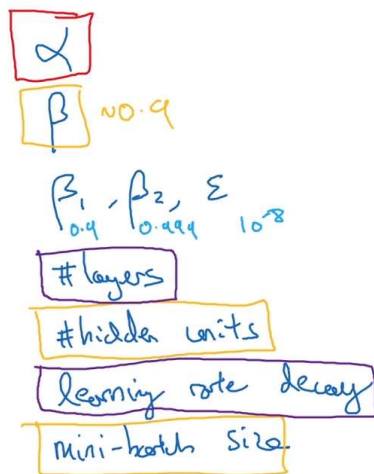
<https://velog.io/@pehye89/Euron-11주차-하이퍼파라미터-튜닝>

✓ 1 more property

100 출석퀴즈

튜닝 프로세스

1. 학습률 α
2. 모멘텀 알고리즘의 $\beta = 0.9$
3. Adam Optimization의 $\beta_1, \beta_2, \epsilon$
4. 은닉층의 개수
5. 은닉 유닛의 수
6. 학습률 감쇠
7. 미니배치 사이즈



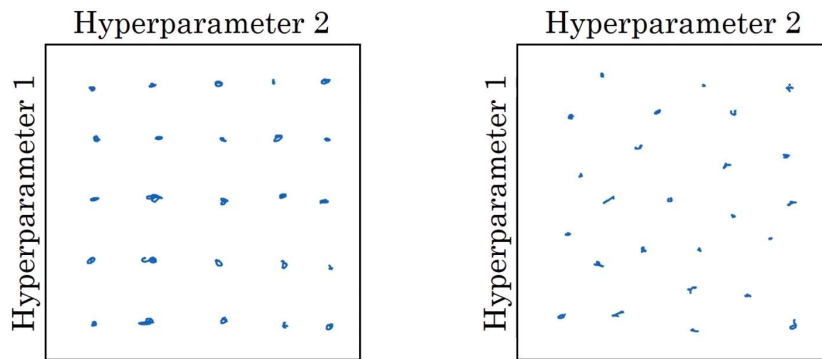
하이퍼파라미터들의 중요도 순위

- 가장 중요한 하이퍼파라미터 : α
- 다음으로 중요한 하이퍼파라미터 : β , 은닉유닛의 개수, 미니배치의 갯수
- 다음으로 중요한 하이퍼파라미터 : 은닉층의 개수, 학습률 감쇠
- Adam Optimization의 $\beta_1, \beta_2, \epsilon$ 값들은 대부분 고정되어있다.

튜닝을 할 때 어떤 값을 탐색할지 어떻게 조정하나?

- 만약 2개의 하이퍼파라미터가 있을 때, 과거에왼쪽에 있는 것처럼 격자점에 있는 값들을 탐색했다. 이 예시는 하이퍼파라미터의 수가 적을 때 효과적이다.

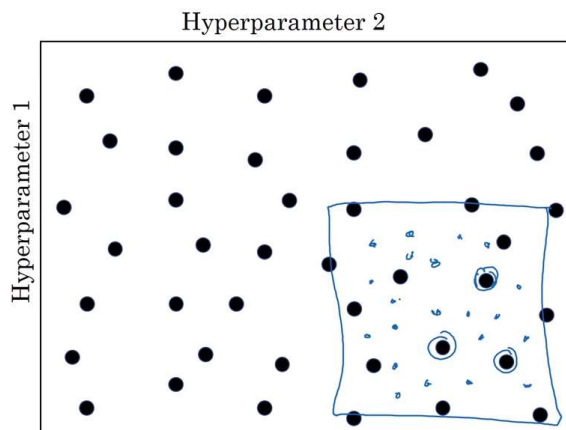
- 하지만 하이퍼파라미터들이 많은 딥러닝의 경우에는, 그리드에서 탐색(Grid Search)하는 것보다 **무작위로 선택하는 것이** 더 효과적이다. 이렇게 하는 이유는, 어떤 하이퍼파라미터가 문제해결에 더 효과적인지 알 수 없고, 또 위에 봤던 것처럼 더 중요한 하이퍼파라미터가 있을 수 있기 때문이다.



- 실제로는 2~3개보다 더 많은 하이퍼파라미터를 탐색한다. 또한 해당 모델에서 어떤 하이퍼파라미터가 중요한지 미리 알기 어렵다. 그렇기에 랜덤하게 값을 탐색할 때 더 다양한 값들을 탐색할 수 있다.

정밀화 접근 Coarse to Fine

- 만약 아래 동그라미쳐진 3개의 값들이 가장 좋은 값들임을 구했을 때, 해당 값들에서 더 작은 영역으로 범위를 좁혀 다시 한번 랜덤하게 값들을 골라서 값들을 골라 탐색해본다면 더 효과적인 값을 얻을 수 있다.



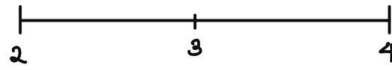
적절한 척도 선택하기

무작위로 값을 탐색하는 것이 효과적일 때가 있다. 만약 은닉 유닛의 개수나 은닉층의 수를 정할 때는 랜덤하게 값을 추출하는 것이 효과적이다.

$n^{[e]}$ 의 14개
50, ..., 100 사이



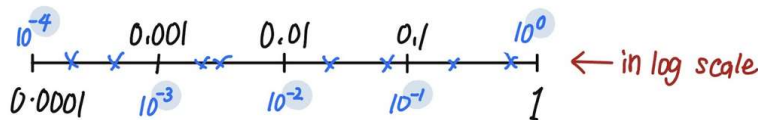
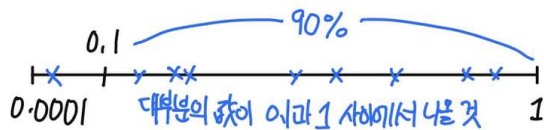
레이어의 수 L
2, ..., 4개 사이



하지만 학습률의 경우, 1과 0.0001중 랜덤하게 값을 고르게 되면, 대부분의 경우 0.1과 1 사이의 값이 나오기 때문에 공평하게 값이 나오지 않는다.

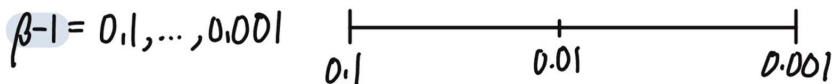
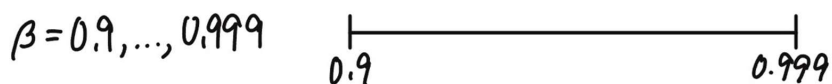
그렇기 때문에 선형척도 대신, 로그척도(log scale)에서 균일하게 뽑는 것이 더 합리적이다.

$\alpha = 0.0001, \dots, 1$ 사이 *scale의 차이!



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

지수가중평균을 계산할 때의 β 의 경우도 비슷하다. 0.9와 0.999사이를 탐색하는 것은 공평하다고 할 수 없기 때문에, $(1 - \beta)$ 를 취해준 후, 위 예시처럼 로그척도에서 값을 랜덤하게 탐색한다.



그렇다면 왜 선형척도에서 샘플을 뽑는 것이 효과적이지 않은 것인가? 만약 β 가 1에 가까우면, 작은 값의 변화에도 큰 영향을 받게 된다.

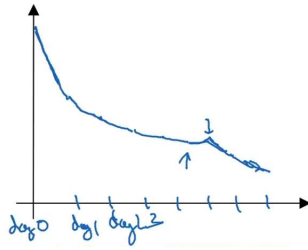
- 만약 β 가 0.9에서 0.9005으로 바뀐다면 알고리즘에 큰 영향을 미치지 못할 것이다.
- 하지만 만약 β 가 0.999에서 0.9995으로 바뀐다면, 알고리즘에 큰 영향을 줄 것이다.

하이퍼파라미터 튜닝 실전

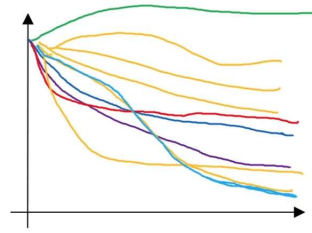
딥러닝은 현재, NLP, Computer Vision 등, 다양한 분야에서 적용되고 있다. 하지만 하이퍼파라미터에 대한 직관은 아무리 같은 딥러닝이여도 모든 분야에서 동일하게 적용되지 않을 수 있다.

하이퍼파라미터들을 찾을 때 2가지 방법을 사용한다.

Babysitting one model



Training many models in parallel



1. 판다 접근

모델 돌보기 (Babysitting one model)

- 데이터는 방대하지만, CPU나 GPU 등 컴퓨터적 자원이 충분하지 않을 때 주로 사용하는 방법
- 하나의 모델의 성능을 보고 하이퍼파라미터를 조율해가면서 학습 성능을 향상 시키는 것

2. 캐비어 접근

동시에 여러 모델 훈련 (Training many models in parallel)

- 컴퓨터의 자원이 충분해서 다양한 하이퍼파라미터 값을 갖고 있는 여러 모델들을 동시에 학습시키는 방법