11주차

날짜	@2024년 5월 21일
과제	강의 요약 출석 퀴즈
세부내용	[딥러닝 2단계] 5. 하이퍼파라미터 튜닝 6. 배치 정규화

하이퍼파라미터 튜닝

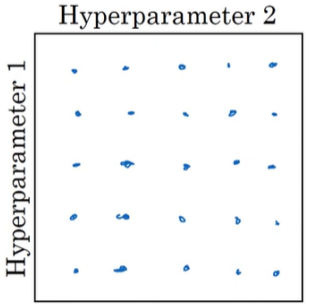
1️⃣ 튜닝 프로세스

Hyperpatameters (빨강 → 노 → 보라 순으로 중요)

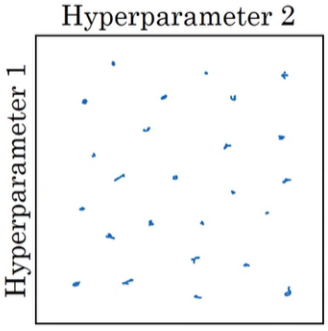
- α (learning rate)
- β (for Momentum) ~ 0.9
- β_1, β_2, ε (for Adam) = 0.9, 0.999, 10^-8 (not tune)
- # of layers
- # of hidden units
- learning rate decay
- mini-batch size

Try Random Values: Don't Use a Grid

- 초기 머신러닝 : hyperparam1 과 hyperparam2의 격자점을 구해서 탐색
  → 이중 최고의 hyperparameter 선택



- 딥러닝 : 무작위로 점을 선택 → 이 중 최고의 하이퍼파라미터 선택



- 어떤 하이퍼파라미터가 더 중요한지 미리 알 수 없기 때문에 random하게 선택
- 실제로는 2차원 이상의 공간에서 탐색하게 될 것

Coarse to Fine (정밀화 접근)

① 전체 사각형에서 무작위로 점을 선택하여 최고의 하이퍼파라미터 선택

② 1번에서 선택한 하이퍼파라미터를 중심으로 더 작은 사각형으로 범위를 좁힘

③ 작은 사각형에서 다시 무작위로 점을 선택하여 최고의 하이퍼파라미터 선택

2️⃣ 적절한 척도 선택하기

Picking Hyperparameters at Random

- 무작위로 선택하는 것이 합리적인 하이퍼파라미터
  - 은닉 유닛의 수 : $n^{[l]}$ → 50 ~ 100 사이의 값을 무작위로 선택
  - 층의 수 : L → 2 ~ 4 사이의 값을 무작위로 선택 or 격자점 사용도 가능

⇒ 모든 하이퍼파라미터가 해당하지는 않음

Appropriate Scale for Hyperparameters

- α : 0.0001 ~ 1 사이의 값을 무작위로 선택
  - 선형척도에서 탐색 : 0.1 ~ 1 사이에 sample의 90%가 존재 → 비합리적
  - 로그척도에서 탐색 : 0.0001 ~ 0.001 ~ 0.01 ~ 0.1 ~ 1 에서 균일하게 선택 → 합리적
- in python) `r=-4*np.random.randn()` → $r \in [-4, 0]$

  `alpha = 10^r` → $\alpha \in [10^{-4}, 10^{0}]$

Hyperparameters for Exponentially Weighted Averages

- β : 0.9 ~ 0.999 사이의 값을 무작위로 선택
  - 0.9 : 마지막 10개의 값의 평균과 유사
  - 0.999 : 마지막 1000개의 값의 평균과 유사
  - 선형척도에서 탐색 → 비합리적
- 1-β 에서 탐색 : 0.1 ~ 0.01 ~ 0.001에서 균일하게 선택 → 합리적
  - $r \in [-3, -1]$
  - $1 - \beta = 10^{r}$ → $\beta = 1 - 10^{r}$
- 선형척도에서의 탐색이 비합리적인 이유
  - 1에 가까울수록 값이 조금만 바뀌어도 결과가 많이 바뀌게 됨
  - ex) β가 0.9000 → 0.9005 : 결과에 영향 없음
  - ex) β가 0.9990 → 0.9995 : 결과에 큰 영향을 줌

3️⃣ 하이퍼파라미터 튜닝 실전

Babysitting One Model

- 데이터는 방대하지만 컴퓨터 자원이 많이 필요하지 않아서 적은 수의 모델을 한 번에 학습 시킬 수 있을 때 사용

- 학습 과정에서 babysit 진행 : 하나의 모델로 매일 성능을 지켜보면서 학습 속도를 조금씩 바꾸는 방식

Training Many Models in Parallel

- 컴퓨터 자원이 충분히 많아 여러 모델을 한 번에 학습 시킬 수 있을 때 사용
- 서로 다른 모델을 동시에 학습시켜 여러 하이퍼파라미터 설정의 성능을 확인

⇒ 컴퓨터 자원의 양과 함수 관계에 따라 두 접근 중 하나를 선택

배치 정규화

1️⃣ 배치 정규화

Normalizing Inputs to Speed Up Learning

- 로지스틱 회귀 모델 : input X를 정규화하면 학습 속도가 빨라짐
- 심층 신경망 : input X 외에도 Z값을 정규화하면 w나 b를 빠르게 학습시킬 수 있음

Implementing Batch Norm

- Given some intermediate values in NN → $z^{(1)}, \cdots, z^{(m)}$: l번째 층의 은닉 유닛의 값 ([l] 생략
  - $\mu = \frac{1}{m}\sum_i z^{(i)}$
  - $\sigma^2 = \frac{1}{m}\sum_i (z^{(i)} - \mu)^2$
  - $z^{(i)}_{norm} = \frac{z^{(i)}-\mu}{\sqrt{\sigma^2+\varepsilon}}$ → 모든 z의 평균은 0, 분산은 1

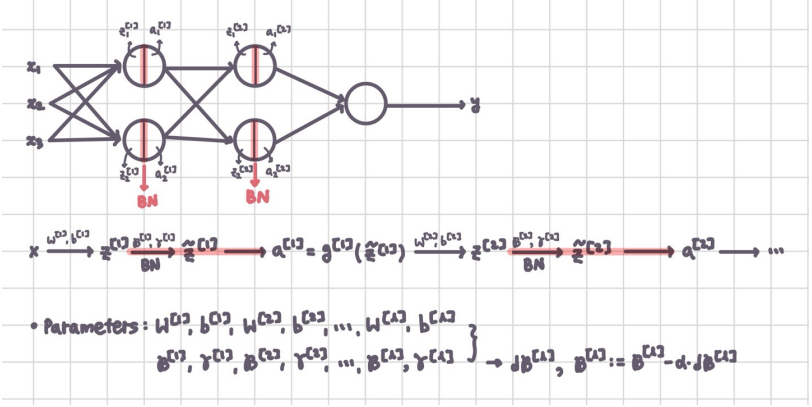
    ⇒ but 은닉 유닛의 값은 다양한 분포를 가져야 좋음
  - $\tilde{z}^{(i)} = \gamma z^{(i)}_{norm} + \beta$ → γ, β는 경사하강법으로 학습시킬 수 있는 변수
- $\gamma = \sqrt{\sigma^2 + \varepsilon},\ \beta = \mu$이면 $\tilde{z}^{(i)} = z^{(i)}$
- γ, β가 다른 값을 가지면 은닉 유닛 값이 서로 다른 평균과 분산을 가지도록 할 수 있음
- use $\tilde{z}^{[l](i)}$ instead of $z^{[l](i)}$

⇒ 입력층만 정규화하는 것이 아니라, 신경망 내부의 은닉층 값까지 정규화하는 것

⇒ γ와 β를 조정하여 서로 다른 평균과 분산을 가지도록 정규화

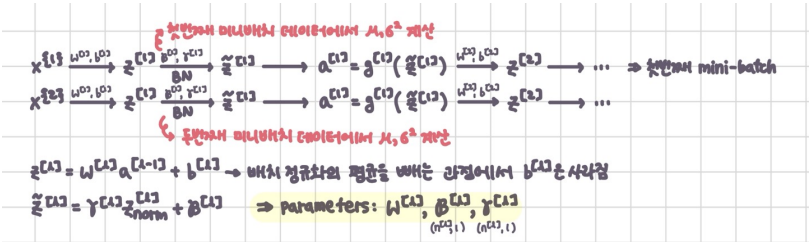
2️⃣ 배치 정규화 적용시키기

Adding Batch Norm to a Network



- tensorflow : `nn.batch_normalization` 로 구현 가능

Working with mini-batches



Implementing Gradient Descent

- for t=1, ... , # of mini-batches:
  compute forward prop on $X^{\{t\}}$
  In each hidden layer, use BN to replace $z^{[l]}$ with $\tilde{z}^{[l]}$
  use backprop to compute $dw^{[l]}$, $d\beta^{[l]}$, $d\gamma^{[l]}$
  update parameters $w^{[l]} := w^{[l]} - \alpha dw^{[l]}$, $\beta^{[l]} := \beta^{[l]} - \alpha d\beta^{[l]}$, $\gamma^{[l]} := \gamma^{[l]} - \alpha d\gamma^{[l]}$
- 경사하강법 외에도 momentum, RMSprop, Adam을 이용해서 업데이트할 수 있음

3️⃣ 배치 정규화가 잘 작동하는 이유는 무엇일까요?

Learning on Shifting Input Distribution

- 검정 고양이 이미지만으로 학습된 신경망
  - 다양한 색깔의 고양이 이미지에 적용하면 좋은 성능을 내지 못함
- Covariate Shift : 데이터 분포 변화 → X의 분포가 바뀌면 다시 학습해야 함

Why This Is a Problem with NNs ?

- 심층 신경망에서 입력값의 분포가 바뀌면 다음 층의 활성값과, 다음층의 파라미터도 영향을 받음 ⇒ Covariate Shift 문제
- 배치정규화 : 은닉층 값들의 분포가 변화하는 양을 줄여줌
  - 은닉 값이 변하더라도 은닉값의 평균과 분산은 유지되도록 함
  - 이전 층과 다음 층의 매개변수 간 상관관계를 줄여줌 → 학습속도 향상

Batch Norm as Regularization

- 각각의 미니배치에 대해서 평균과 분산을 계산 → noise를 가지고 있음
- $z^{[l]}$을 $\tilde{z}^{[l]}$로 변환하는 과정에도 noise 존재
- 드롭아웃과 같이 은닉층에 잡음을 추가해서 약간의 일반화 효과를 보여줌
- 큰 미니배치를 사용하면 일반화 효과가 오히려 감소
  ⇒ 일반화 목적으로 사용하기에는 적합하지 않음

4️⃣ 테스트 시의 배치 정규화

Batch Norm at Test Time

- μ, σ^2 : 미니배치 안에서 계산
  - 테스트 시에는 미니배치가 없으므로 다른 처리 방법이 필요
  - 학습 시 사용된 미니배치들의 지수가중평균을 추정치로 사용
- 미니배치로 학습, 계산하지만 테스트 시에는 한 번에 샘플 하나씩 처리