5주차

Hyperparameter tuning

좋은 하이퍼파라미터 찾기

♥ Tuning process

심층 신경망 - 다뤄야 할 하이퍼파라미터가 많음

- 중요한 하이퍼파라미터
  - learning rate α
  - momentum (default : 0.9)
  - mini-batch size
  - hidden units
  - layers
  - learning rate decay
  - $\beta_1, \beta_2, \epsilon$ (Adam : 0.9, 0.999, 10^(-8))

- 머신러닝이 생긴지 얼마 안됐을 때 → Use grid(Grid Search)
  - 하이퍼파라미터의 수가 적을 때 사용
- 딥러닝 → Try random values
- Coarse to fine (정밀화 접근)
  - 더 작은 영역으로 확대, 더 조밀하게 점들을 선택
  - 범위를 좁혀나가기

1. **Use random sampling, not a grid search**
2. **Use coarse to fine if you want**

♥ Using an Appropriate Scale

무작위 → 적절한 척도를 정하기!

- 선형 척도 < 로그 척도
  - 로그 척도에서 균일하게 무작위로 값을 뽑는 것
  - 0.0001, 0.001, 0.01, 0.1, 1

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

  - $\Rightarrow 10^{-4} < \alpha < 10^{0}$

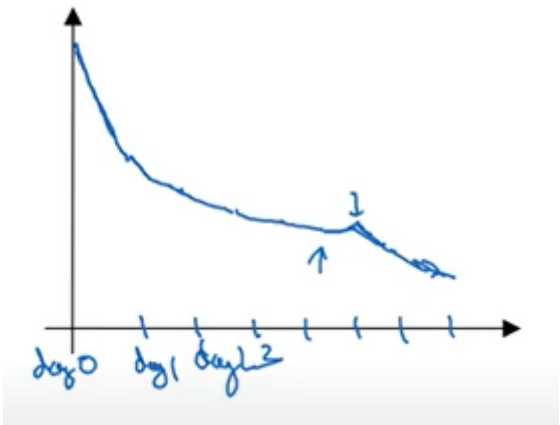
♥ Hyperparameter Tuning in Practice

1. **Babysitting one model**

   데이터는 방대하지만 컴퓨터 자원이 많이 필요하지 않아 적은 숫자의 모델을 한 번에 학습시킬 수 있을 때 사용

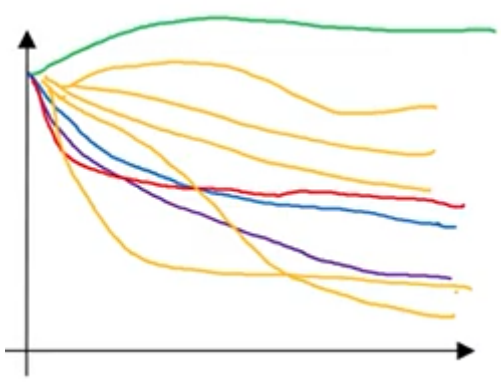
   여러 모델을 동시에 학습시킬 컴퓨터 자원이 충분치 않을 때 사용

   - 며칠, 몇 주에 걸쳐 매일 모델을 돌보며 학습시키는 것
   - 성능을 잘 지켜보다가 학습 속도를 조금씩 바꾸기



2. **Training many models in parallel**

   - 서로 다른 모델을 동시에 학습시키기



♥ Normalizing Activations in a Network

Batch Normlization

신경망과 하이퍼파라미터의 상관관계를 줄여 하이퍼파라미터가 잘 작동하도록 한다.

깊은 심층 신경망이라도 쉽게 학습할 수 있도록 도와준다.

Implementing Batch Norm

Given some intermediate values in NN, $z^{(1)}, \ldots, z^{(m)}$

$\mu = \frac{1}{m} \sum_i z^{(i)}$
$\sigma^2 = \frac{1}{m} \sum_i (z_i - \mu)^2$
$z^{(i)}_{norm} = \frac{x^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$

⇒ 은닉 유닛은 다양한 분포를 가져야하기 때문에 항상 평균 0, 표준편차 1을 갖는 것이 좋지만은 않다.

< 하나의 은닉층에 배치 정규화를 구현하기>

$\tilde{z}^{(i)} = \gamma * z_{norm}^{(i)} + \beta,$

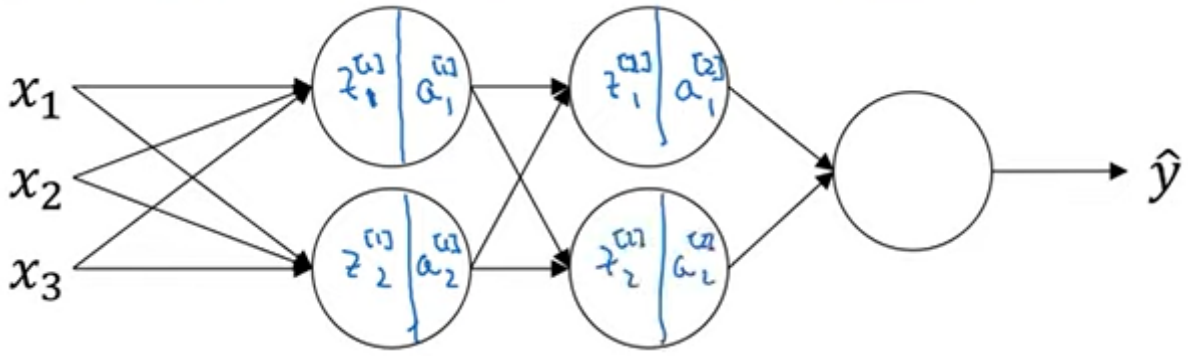
$\gamma \ \& \ \beta$ are learnable parameters of model

⇒ 평균과 분산이 두 변수 $\gamma \ \& \ \beta$에 의해 조절된다.

⇒ 은닉 유닛 값 $z^{(i)}$의 평균과 표준편차를 특정한 평균과 분산을 갖도록 정규화하는 것

♥ Fitting Batch Norm into Neural Networks

Adding Batch Norm to a network



x - $w^{[1]}, b^{[1]}$ → $z^{[1]}$ - Batch Norm(BN) (평균 0, 분산 1을 갖도록 정규화 한 뒤, $\gamma^{[1]}, \beta^{[1]}$를 이용해 값을 조정해주는 것) → $\tilde{z}^{[1]}$ → $a^{[1]} = g^{[1]}(\tilde{z}^{[1]})$

- $w^{[2]}, b^{[2]}$ → $z^{[2]}$ - Batch Norm(BN) $\gamma^{[2]}, \beta^{[2]}$ → $\tilde{z}^{[2]}$ → $a^{[2]} = g^{[2]}(\tilde{z}^{[2]})$

⇒ z 계산과 a 계산 사이에 배치 정규화가 이뤄진다.

⇒ 비정규화된 z 값 대신 정규화된 값 $\tilde{z}$ 사용

Implementing gradient descent

배치 정규화를 사용하여 경사하강법 구현하기

for t = 1, …, m (# of mini-batches)

compute forward prop. on X^t

In each hidden layer, use BN to replace $z^{[l]}$ with $\tilde{z}^{[l]}$

Use back prop. to compute $dW^{[l]}, \cancel{db^{[l]}}, d\beta^{[l]}, d\gamma^{[l]}$

Update parameters

$W^{[l]} := W - \alpha \, dW^{[l]}$
$\beta^{[l]} := \beta^{[l]} - \alpha \, d\beta^{[l]}$
$\gamma^{[l]} := ...$

+ can works with momentum, RMSpop, Adam

♥ Why Does Batch Norm Work?

1. 입력 특성 X에 대해 비슷한 범위를 갖도록 정규화하여 학습 속도를 높인다.
2. 신경망에서 깊은 층의 가중치가 앞쪽 층의 가중치의 변화에 영향을 덜 받는다.

“Covariance shift” (공변량 변화): X, Y 간의 대응을 학습시킬 때, X의 분포가 바뀌면 학습 알고리즘을 다시 학습해야 한다.

⇒ 배치 정규화는 은닉층 값들의 분포가 변화하는 양을 줄여준다. (학습하게 될 값의 분포를 제한-안정화)

Batch Norm as regularization

⇒ 드롭아웃처럼 은닉층에 잡음을 추가해서 아주 약간의 일반화(regularization) 효과를 보여준다.

효과가 크지는 않음

큰 미니배치를 사용하면 잡음이 줄어들며 regularization 효과도 줄어든다.

배치 정규화를 regularization 목적으로 사용하는 것은 안좋음

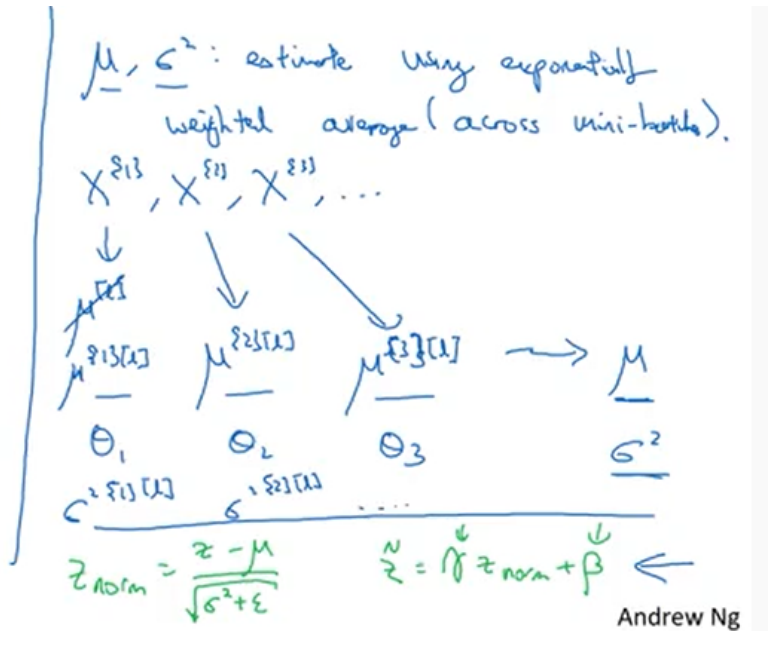
의도치 않은 부수효과 정도로 생각하기

♥ Batch Norm At Test Time

배치 정규화는 한 번에 하나의 미니배치 데이터를 처리하지만 테스트에서는 한 번에 샘플 하나씩을 처리해야 한다.

이를 위해 신경망을 어떻게 학습시켜야 할까?

⇒ 각각 독립된 μ 와 σ^2 의 추정치를 사용하기



♥ Softmax Regression

이진이 아닌 다중?

⇒ 소프트맥스 회귀(로지스틱 회귀를 일반화)

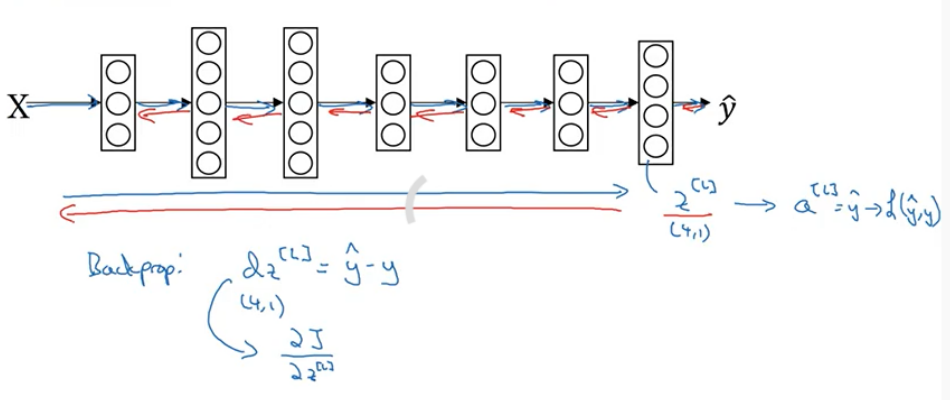
활성화 함수 g^{[L]} : softmax activation function

⇒ t를 합이 1이 되도록 정규화

♥ Training Softmax Classifier

Softmax regression : 클래스가 둘 이상인 경우 로지스틱 회귀를 일반화 한 것

Gradient descent with softmax



♥ The Problem of Local Optima

Optimization algorithm

local optimum 에 빠지는 문제

⇒ 충분히 큰 신경망을 학습시키기

안정지대 → 학습 속도를 느려지게 함

⇒ Momentum, RMSprop, Adam 등 알고리즘의 도움 받기

⇒ Adam과 같은 최적화 알고리즘이 안정지대 내에서 움직이거나 벗어나는 속도를 올릴 수 있다.

♥ TensorFlow

Deep-Learning Framework 중 하나