Chatper 11 심층 신경망 훈련하기

그레디언트 소실과 폭주문제

그레디언트 소실: 하위층이 진행될 수록 그레디언트가 점점 작아지는 현상

그레디언트 폭주: 하위층이 진행될수록 그레디언트가 점점 발산하는 현상

✓ **해결방안**: 초기화와 활성화 함수를 잘 조절해보자

- ### 글로럿과 He 초기화

예측을 할 때 정방향으로, 역전파 할때는 역방향으로 양방향으로 신호가 적절하게 흘러야한다.

적절하게 잘 흐르기 위해서는 각 층의 출력에 대한 분산과 입력에 대한 분산이 같아야 한다고 주장

- ### 배치 정규화

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \qquad \text{// mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

  - 초기화를 통해 그레디언트 소실과 폭주의 문제를 감소할 수 있지만 이것은 훈련 초기단계에만 해당이 되고 훈련 진행되는 동안 이러한 문제들이 다시 발생하지 않으리란 보장이 없다.
  - **배치 정규화**: 각 층의 활성화 함수의 출력값 분포가 골고루 분포되도록 강제하는 방법으로 각 층에서의 활성화 함수 출력값이 정규분포를 이루도록 하는 방법
  - 배치정규화는 규제와 같은 역할을 하여 다른 규제 기법의 필요성을 줄여준다.
  - 그러나 시간이 많이 걸린다,

- ### 그레디언트 클리핑

역전파 될때 일정 임계값을 넘어서지 못하게 그레디언트를 잘라내는 것

✓ 이용: 옵티마이저를 만들때 `clipnorm`과 `clipvalue`를 조정한다.

사전훈련된 층 재사용하기

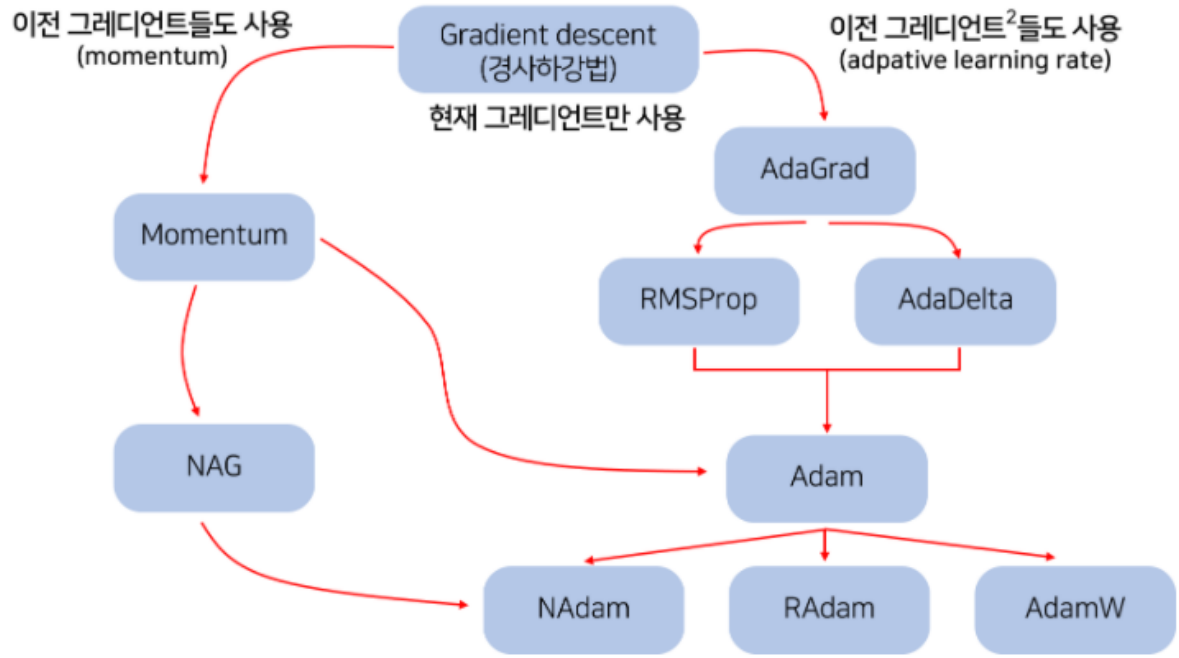
✓ **전이 학습**: 기존 비슷한 유형의 문제를 처리한 신경망이 있을 때, 그 신경망의 **하위층**을 재사용하는 것.

- 훈련속도를 높여주고, 필요한 훈련데이터도 훨씬 작다.

- ### 비지도 사전훈련

레이블 된 훈련데이터가 많지 않은 복잡한 문제에 대하여 레이블이 없는 데이터들을 비지도 학습하여 특성을 추출하고 이를 이용하여 학습시킨다.

고속 옵티마이저



- ## 모멘텀 최적화

경사하강법은 이전 그레디언트에 대한 반영이 없는 반면 모멘텀 최적화 방식은 이전 그레디언트에 대한 반영이 있다.

> 비유를 하자면 미끄럼틀을 내려올때 빗면의 그레디언트가 남아있어 앞으로 쭉 나갈 수 있게 한다. 즉 지역 minimum값을 피할 수 있다.

이때 이전에 있던 모든 그레디언트들을 그대로 고려한다면 아주 긴 평평한 땅에서도 절대 멈추지 않기 때문에 이전의 영향력을 매 업데이트마다 γ배씩 감소시킨다.

- ## 네스테로프 가속경사(NAG)

현재의 위치가 아니라 모멘텀의 방향으로 조금 앞선 곳에서 비용함수의 그레디언트를 계산하는 방식

> 즉 앞을 미리보고 현재의 관성을 조절하여 업데이트의 크기를 바꾸는 방식
>
> 비유를 하자면 u자형 협곡에서 NAG가 예측을 하여 방향을 알고 경사하강을 하기 때문에 성능이 뛰어나다.

✓ 관성에 의해 수렴지점에서 진동하는 것을 방지해준다.

- ## AdaGrad

가장 가파른 **차원(축)**을 따라 그레디언트 벡터의 스케일을 감소시킨다.

> cf) 경사하강법은 가장 가파른 경사를 따라 하강한다.

✓ 학습률이 너무 감소되어 전역 최적점에 도착하기 전에 알고리즘이 완전히 멈춤.

✓ 학습률 파라미터를 덜 튜닝해도 되는 점이 장점

- ## RMSProp

모든 그레디언트가 아닌 가장 최근 반복에서 비롯된 그레디언트만 누적하여 AdaGrad의 문제점을 해결하였다.

- ## Adam과 Nadam최적화
    - **Adam**: 모멘텀 최적화 + RMSProp의 결과
    - **Nadam**: NAG + Adam
- ## 학습률 스케줄링

> 학습률을 경사를 내려가는 step의 크기라고 생각하면 편하다.

✓ **학습 스케줄링**: 큰 학습률로 시작하여 학습 속도가 느려질때 학습률을 낮춰 최적의 솔루션을 더 빨리 발견하는 방법

규제를 사용해 과대적합 피하기

- ## l1, l2 규제

신경망의 연결 가중치를 제한하기 위해 l2 규제를 사용하거나 희소모델(많은 가중치가 0인) 희소모델을 만들기 위해 l1 규제를 사용할 수 있다.

- ## DropOut

매 훈련 스텝에서 각 뉴런은 임시적으로 드롭아웃될 확률P를 가지며 동작. 이번 훈련 스텝에는 완전히 무시되지만 다음 스텝에서는 활성화될 수 있습니다.

모델이 과대적합되면 Dropout 비율을 늘리고 과소적합되면 Dropout비율을 늘린다.

Dropout은 일정확률로 뉴런들이 작동을 안하기 때문에 수렴을 상당히 느리게 만드는 경향이 있다.

- ## 맥스 - 노름 규제

각각의 뉴런에 대해 입력의 연결 가중치 W가 $\|W\|_2 \leq 2$ 가 되도록 제한하는 것