CH6. 학습관련기술들

학습할 내용

1) 가중치 매개변수의 최적값을 탐색하는 최적화 방법

2) 가중치 매개변수 초깃값, 하이퍼파라미터 설정 방법

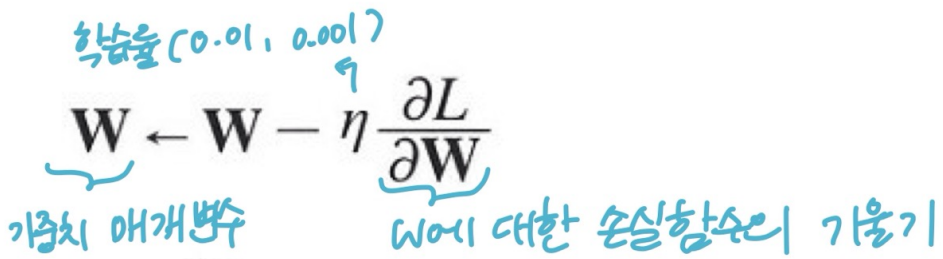
3) 오버피팅의 대응책 - 정규화 1) 가중치 감소 2) 드롭아웃

6.1 매개변수 갱신

매개 변수의 최적값을 찾는 것 : 최적화

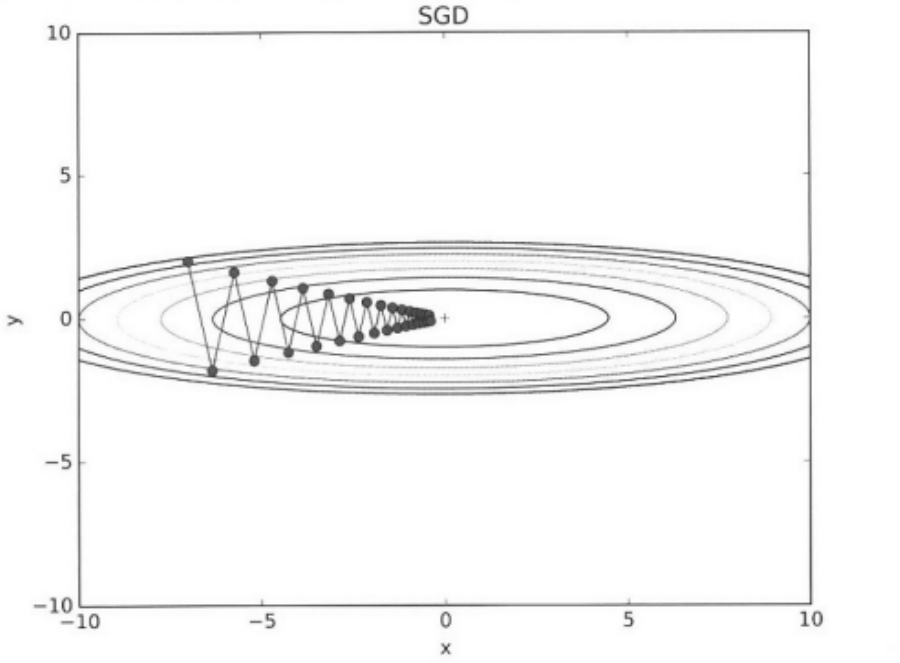
1) 미분 - 확률적 경사 하강법 (SGD)

- 수식



- 단점
  - 비등방성 함수에서는 탐색 경로가 비효율적

그림 6-3 SGD에 의한 최적화 갱신 경로 : 최솟값인 (0, 0)까지 지그재그로 이동하니 비효율적이다.



- 개선 방법 : 모멘텀, AdaGrad, Adam

6.1.4 모멘텀

그림 6-4 모멘텀의 이미지 : 공이 그릇의 곡면(기울기)을 따라 구르듯 움직인다.

- 수식

W에 대한 손실함수의 기울기

속도(velocity)

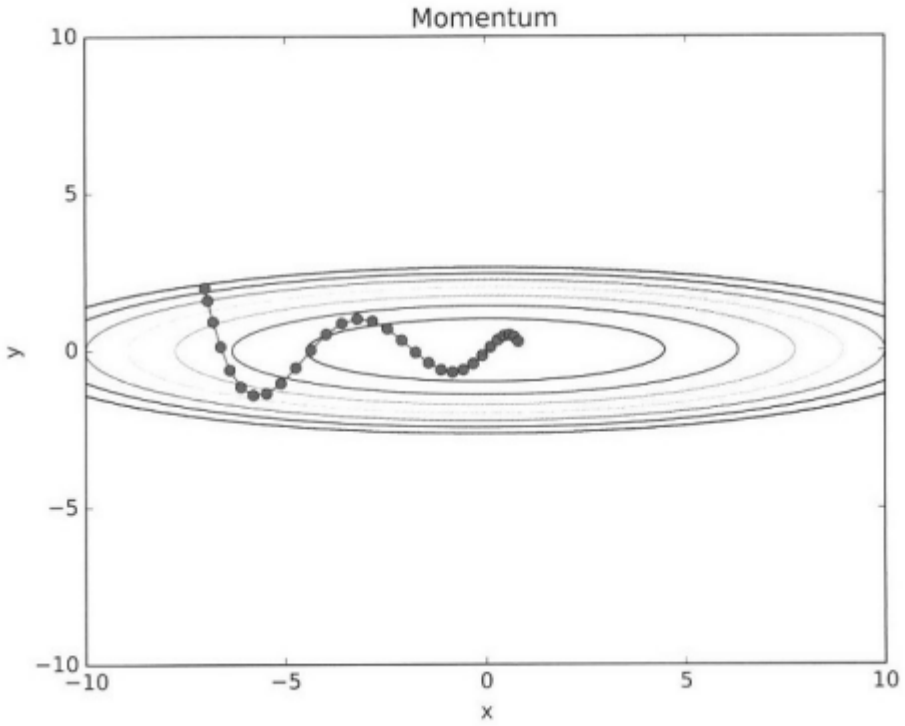
$$\mathbf{v} \leftarrow \alpha\mathbf{v} - \eta\frac{\partial L}{\partial \mathbf{W}} \quad \text{[식 6.3]}$$

물체가 아무런 힘을 받지 않을 때 하강시킴 (α=0.9)

$$\mathbf{W} \leftarrow \mathbf{W} + \mathbf{v} \quad \text{[식 6.4]}$$

    - 식 6.3은 기울기 방향으로 힘을 받아 물체가 가속된다는 물리 법칙을 나타냄.
- 최적화 갱신 경로

그림 6-5 모멘텀에 의한 최적화 갱신 경로



    - SGD와 비교하면 지그재그 정도가 덜함.

6.1.5 Adagrad

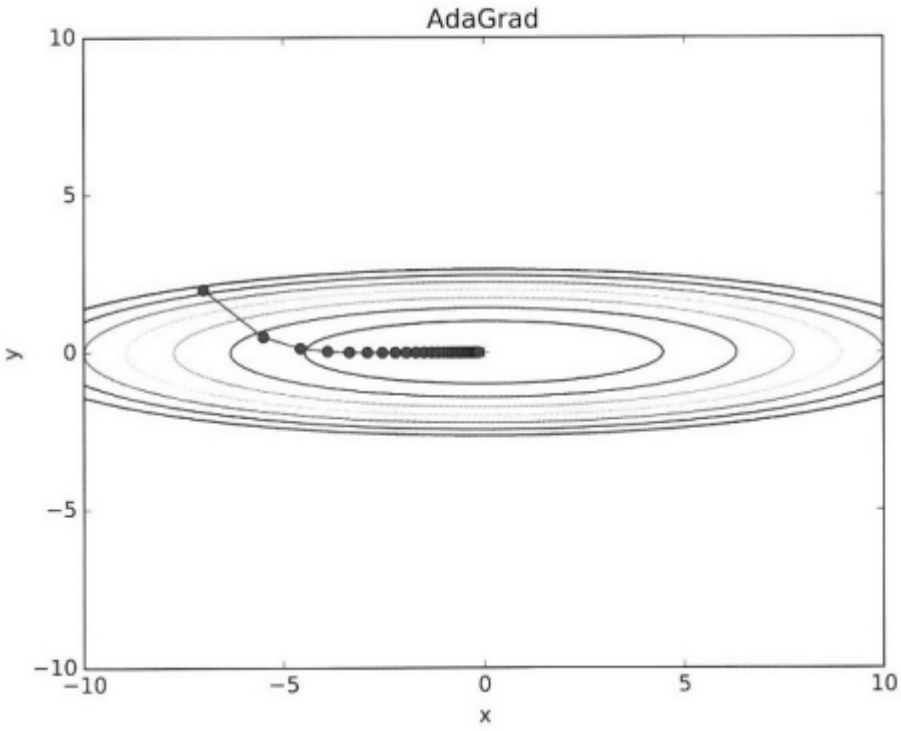
- 각각의 매개변수에 맞춤형 값을 만들어줌.
- 개별 매개변수에 적응적으로 학습률을 조정하면서 학습을 진행
    - 학습률이 너무 작으면 학습시간이 길어지고, 너무 크면 발산하여 학습이 제대로 이루어지지 않음
- 수식

$$\mathbf{h} \leftarrow \mathbf{h} + \frac{\partial L}{\partial \mathbf{W}} \odot \frac{\partial L}{\partial \mathbf{W}} \quad \text{[식 6.5]}$$

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{1}{\sqrt{\mathbf{h}}} \frac{\partial L}{\partial \mathbf{W}} \quad \text{[식 6.6]}$$

- 매개변수의 원소 중 많이 움직인 (크게 갱신된 = 기울기 변화가 큰) 원소는 학습률이 낮아짐.
- 최적화 갱신 경로

그림 6-6 AdaGrad에 의한 최적화 갱신 경로

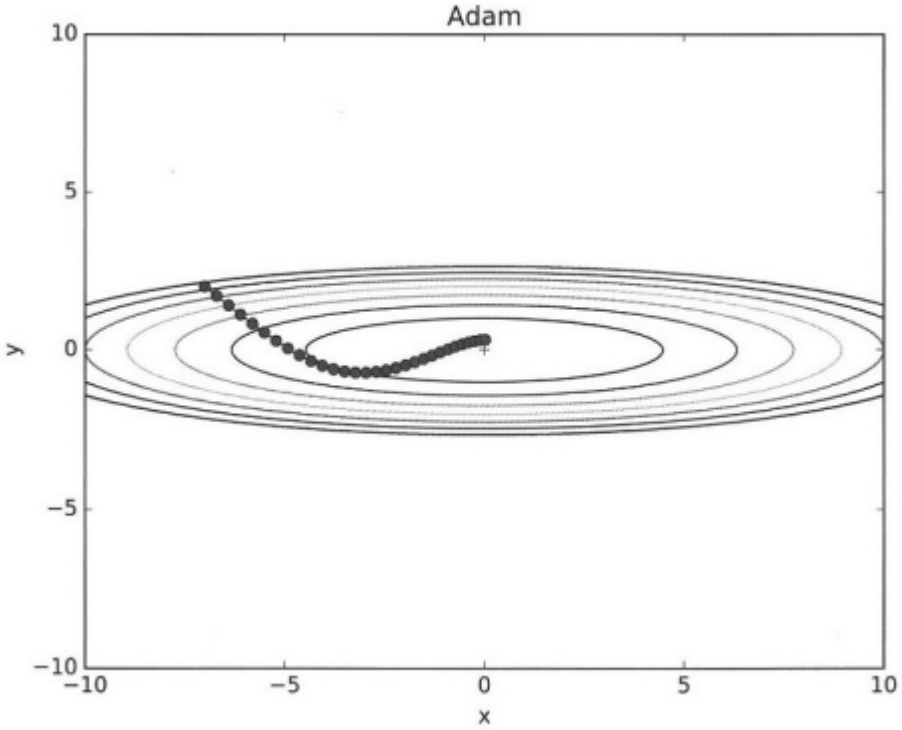


6.1.6 Adam

- 모멘텀과 Adagrad를 융합한 것
- 하이퍼파라미터의 편향보정이 진행됨

- 최적화 갱신 경로

그림 6-7 Adam에 의한 최적화 갱신 경로



- 모멘텀과 비슷한 패턴인데, 공의 좌우 흔들림이 적음

6.1.7 비교

그림 6-8 최적화 기법 비교 : SGD, 모멘텀, AdaGrad, Adam

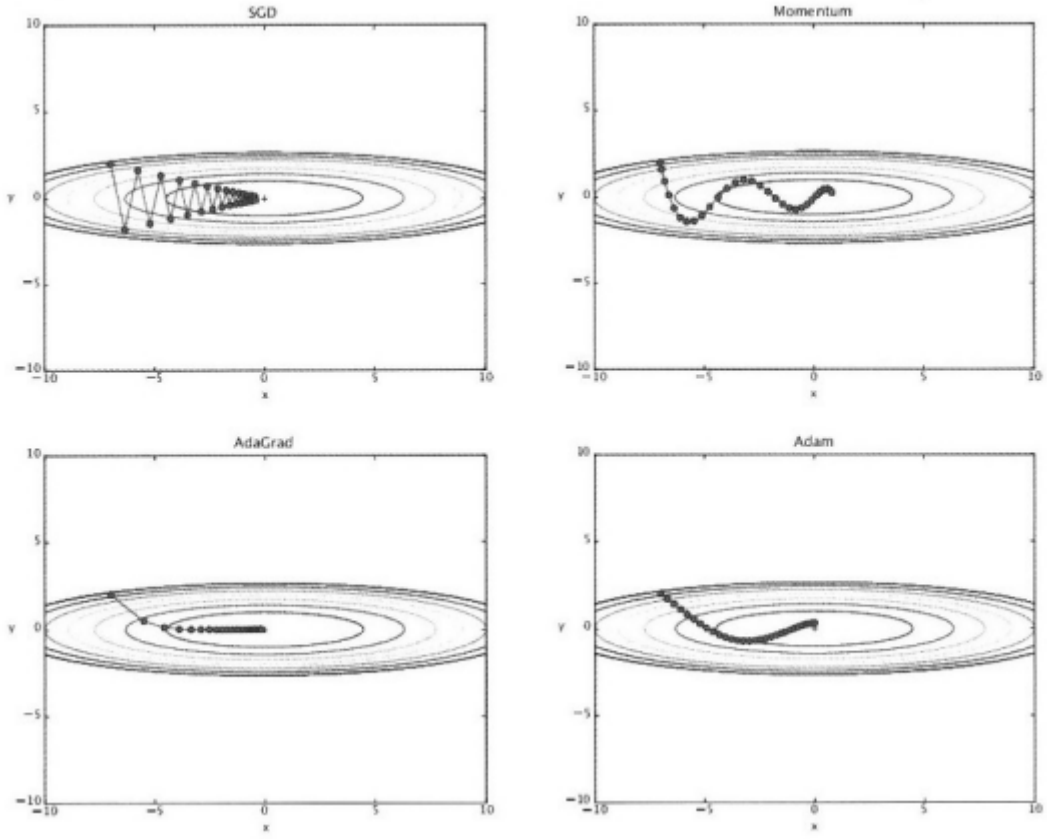
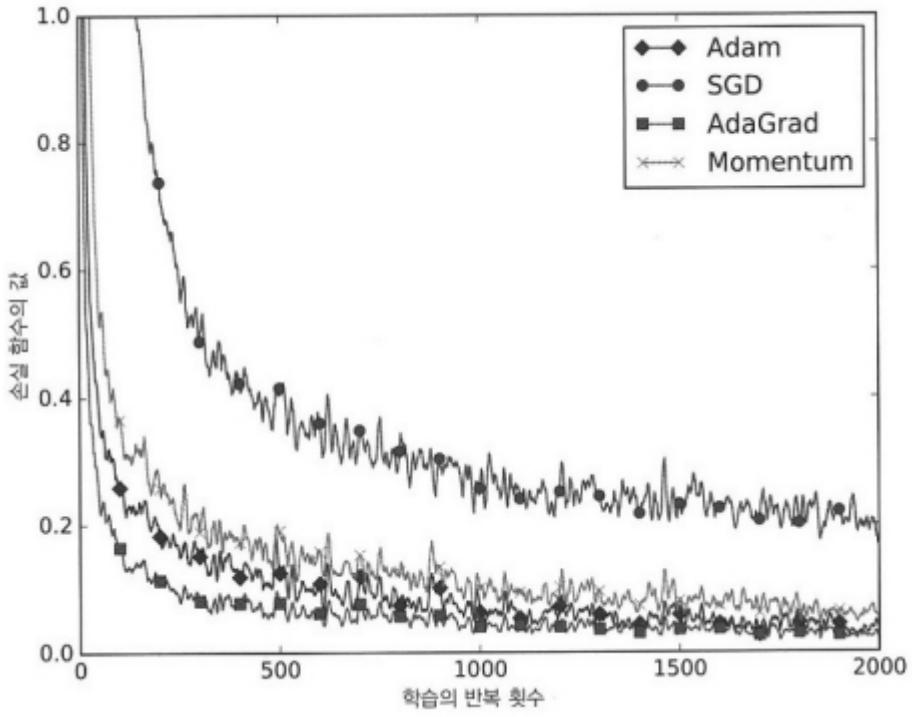


그림 6-9 MNIST 데이터셋에 대한 학습 진도 비교



- 학습진도 : SGD가 가장 느리고, Adagrad가 가장 빠름

6.2 가중치의 초깃값

👾 활성화 함수가 ReLU : He 초깃값

활성화 함수가 선형 sigmoid, tanh : Xavier 초깃값

- 가중치의 초깃값을 적절히 설정하면 각 층의 활성화값 분포가 적당히 퍼지면서 학습이 원활하게 수행됨.

6.2.1 초깃값을 0으로 하면 ?

- 지금까지의 가중치 초깃값 : 0.01 * np.random.randn(10,100)
  - 정규분포에서 생성된 값에 0.01

❌ 학습이 제대로 이루어지지 않음

- WHY? 오차역전파법에서 모든 가중치의 값이 똑같이 갱신되기 때문.

6.2.2 은닉층의 활성화값 분포

💡 가중치의 초깃값에 따라 은닉층 활성화값들이 어떻게 변화하는가

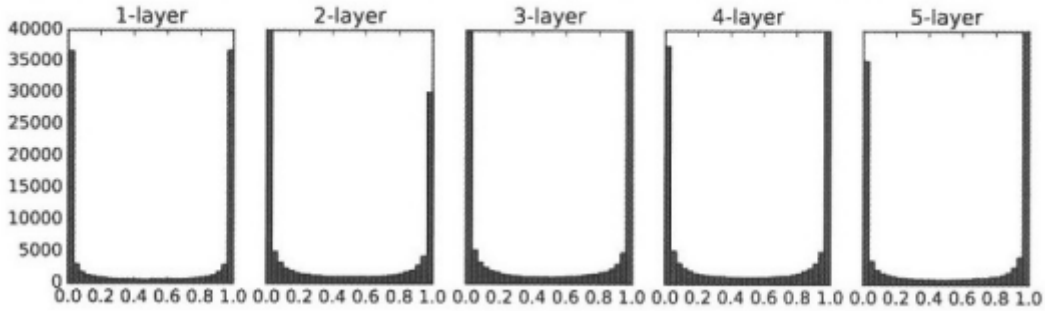
- 활성화 함수로 시그모이드 함수를 사용하는 5층 신경망에 무작위로 입력한 입력 데이터를 흘리며 각 층의 활성화값 분포를 히스토그램으로 그려볼 것.

- w (손실함수의 기울기) 할당값에 따라 은닉층의 활성화값이 어떻게 분포하고 있는가.

1) 표준편차가 1인 정규분포

```
w = np.random.randn(  _ ) * 1 :
```

그림 6-10 가중치를 표준편차가 1인 정규분포로 초기화할 때의 각 층의 활성화값 분포

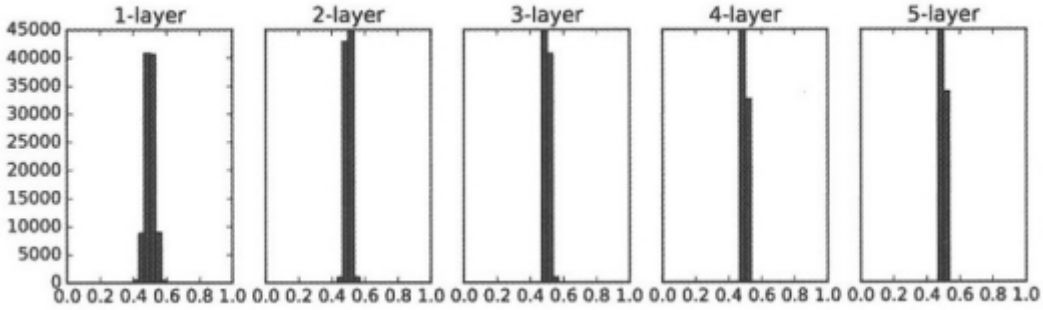


- 기울기 소실

2) 표준편차 0.01

```
w = np.random.randn( _ ) * 0.01
```

그림 6-11 가중치를 표준편차가 0.01인 정규분포로 초기화할 때의 각 층의 활성화값 분포



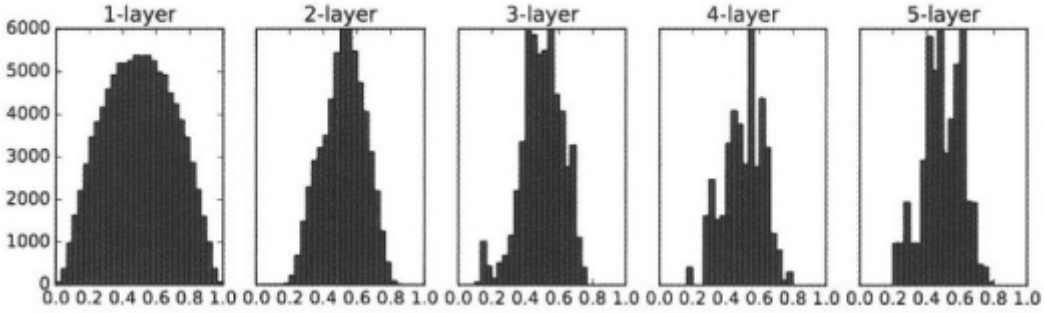
- 표현력 제한

3) Xavier 초깃값

- 노드가 n개 라면 표준편차가 $\frac{1}{n}$인 분포 사용

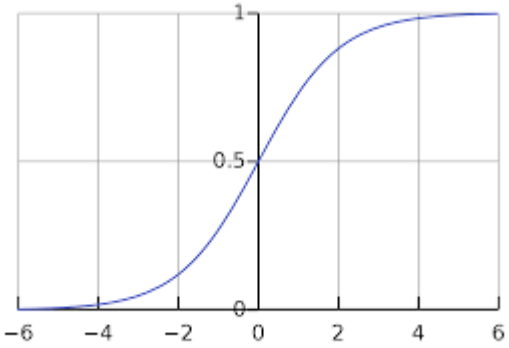
```
# node_num = 100
w = np.random.randn( 100,100 ) * np.sqrt(100)
```

그림 6-13 가중치의 초깃값으로 'Xavier 초깃값'을 이용할 때의 각 층의 활성화값 분포

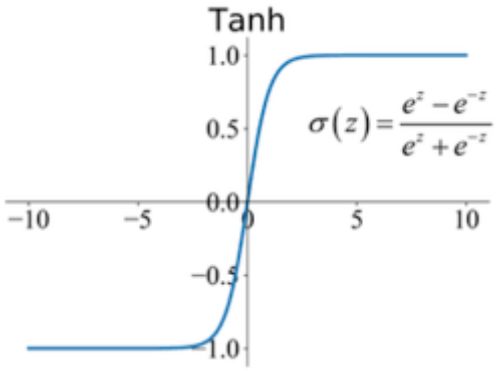


- 전제: 활성화함수가 선형 (sidmoid, tanh)

- sidmoid

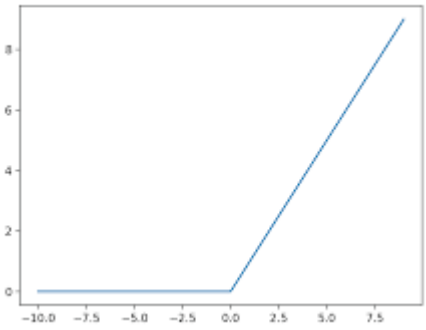


- tanh



6.2.3 ReLU(활성화함수)를 사용할 때의 가중치 초깃값

- He 초깃값 사용
  - WHY? ReLU는 선형이 아니기 때문에 Xavier 초깃값 사용 불가.



ReLU

- 앞 계층의 노드가 n개일 때, 표준편차가 $\sqrt{\frac{2}{n}}$인 정규분포 사용

6.3 배치 정규화

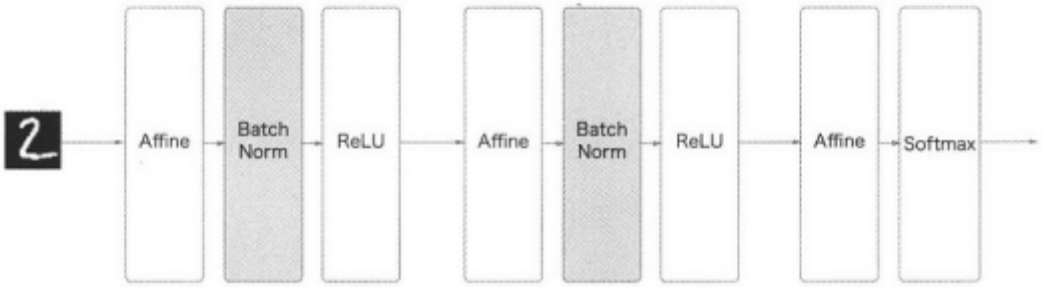
- 각 층이 활성화를 적당히 퍼뜨리도록 '강제' —→ 배치 정규화

6.3.1 배치 정규화 알고리즘

- 학습을 빨리 진행할 수 있다
- 초깃값에 크게 의존하지 않는다
- 오버피팅을 억제한다

1) 데이터 분포를 정규화하는 '배치 정규화 계층' 을 신경망에 삽입

그림 6-16 배치 정규화를 사용한 신경망의 예



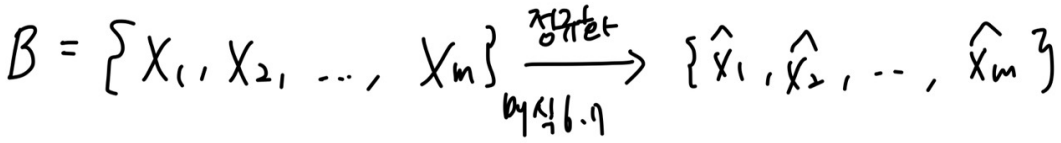
2) 미니배치를 단위로 정규화함 - 평균 0, 분산 1

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

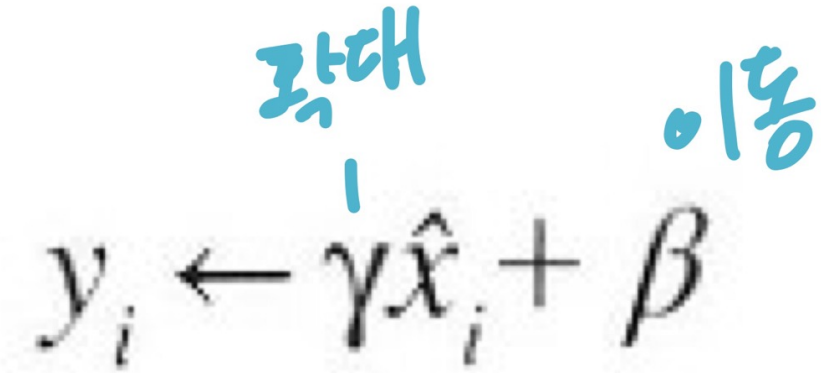
$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

[식 6.7]



- 이 처리를 활성화 함수의 앞 또는 뒤에 삽입 —> 데이터 분포의 치우침 막을 수 있음.

3) 배치 정규화 계층마다 이 정규화된 데이터에 고유한 학대와 이동 변환을 수행함



$\gamma = 1\beta$ =0부터 시작

6.4 바른 학습을 위해

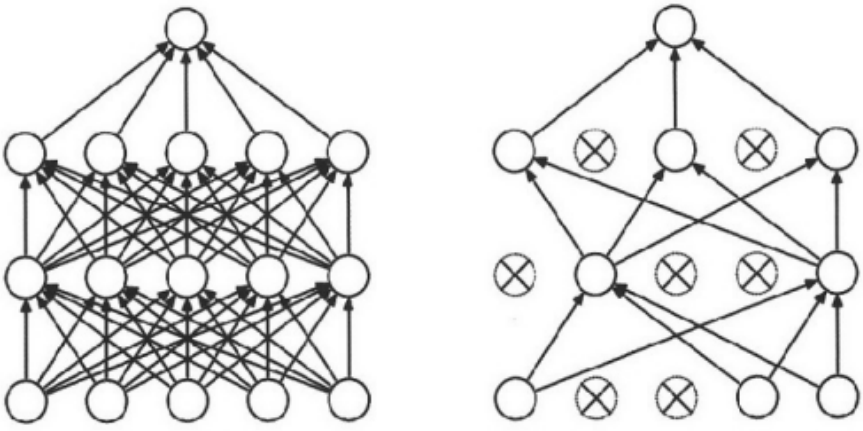
6.4.1 오버피팅

- 발생
  - 매개변수가 많고 표현력이 높은 모델
  - 훈련데이터가 적음
- 억제
  - 가중치 감소 : 큰 가중치에 대해서는 그에 상응하는 큰 페널티 부여.
    - 손실함수에 가중치의 L2노름 더함 ($\frac{1}{2}\lambda W^2$)

6.4.3드롭아웃

- 뉴런을 임의로 삭제하면서 학습하는 방법
- 훈련 때 은닉층이 뉴런을 무작위로 골라 삭제, 삭제된 은닉층은 신호 전달 하지 않음.

그림 6-22 드롭아웃의 개념(문헌[14]에서 인용) : 왼쪽이 일반적인 신경망, 오른쪽이 드롭아웃을 적용한 신경망. 드롭아웃은 뉴런을 무작위로 선택해 삭제하여 신호 전달을 차단한다.



(a) 일반 신경망 (b) 드롭아웃을 적용한 신경망

- 시험때는 모든 뉴런에 신호를 전달하지만, 각 뉴런의 출력에 훈련 때 삭제한 비율을 곱하여 출력함 (flg = True 설정하면 별도로 비율을 곱해줄 필요 없음)

6.5 적절한 하이퍼파라미터 값 찾기

6.5.1 검증 데이터

- 훈련데이터 : 매개변수 학습
- 시험데이터 : 범용 성능 평가 but 하이퍼파라미터의 성능을 평가할 때는 사용하면 안 됨!

  WHY? 시험데이터를 사용하여 하이퍼파라미터를 조정하면 시험데이터에 오버피팅 되기 때문에 —> 하이퍼파라미터 조정용 데이터 : **검증데이터**
- 검증데이터 : 하이퍼파라미터 성능평가

  MNIST : 훈련데이터 중 20%정도를 검증데이터로 먼저 분리.

6.5.2 하이퍼파라미터 최적화

하이퍼파라미터의 ‘최적 값’이 존재하는 범위를 조금씩 줄여감

1) 대략적인 범위 설정

2) 무작위로 샘플링

3) 그 값으로 정확도 평가

4) 2,3번을 반복하여 그 정확도의 결과를 보고 아이퍼파라미터의 범위를 좁힘.