Chapter 6. 학습 관련 기술들
Chapter 7. 합성곱 신경망 (CNN)

밑바닥부터 시작하는 딥러닝

210108 Jisu Han

Chapter 6. 학습 관련 기술들

설명 순서: 딥러닝 모델 학습 -> 매개변수 초깃값 설정 -> 매개변수 갱신 -> 하이퍼파라미터 조정 -> Regularization

1. 매개변수 갱신 -3
2. 가중치의 초깃값 -2
3. 배치 정규화 -4
4. 바른 학습을 위해 -1
5. 적절한 하이퍼파라미터 값 찾기

딥러닝 모델 학습시

- 학습해야 할 대상: 모델 파라미터/매개변수/가중치, 학습자가 조정할 대상: 하이퍼파라미터
- 오버피팅: 신경망이 훈련 데이터에만 지나치게 적응되어 그 외의 데이터에는 제대로 대응하지 못하는 상태
  - 매개변수가 많고 표현력이 높은 모델: 가중치 매개변수의 값이 큰 상황
  - 훈련 데이터가 적음
  - 과적합을 막기 위한 Regularization 기법을 사용한다.
- Dataset
  - 주의점: 하이퍼 파라미터의 성능을 평가할 때는 시험 데이터를 사용해서 안된다.
  - 훈련 데이터: 매개변수 학습
  - 검증 데이터Validation dataset: 하이퍼파라미터 조정용 데이터 (성능 평가)
  - 시험 데이터: 신경망의 범용 성능 평가

딥러닝 모델 학습 전 Dataset

- Dataset
  - 주의점: 하이퍼 파라미터의 성능을 평가할 때는 시험 데이터를 사용해서 안된다.
  - 훈련 데이터: 매개변수 학습
  - 검증 데이터Validation dataset: 하이퍼파라미터 조정용 데이터 (성능 평가)
  - 시험 데이터: 신경망의 범용 성능 평가
- 데이터셋 조정: Batch Normalization 배치 정규화
  - 각 층에서의 활성화 값이 적당히 분포되도록 조정하는 것
  - 장점: 학습 속도 개선, 초깃값 설정에 도움, 오버피팅 억제
  - $\{x_1, x_2, \ldots, x_m\} \sim N(\mu_B, \sigma_B^2) \longrightarrow \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\} \sim N(0,1)$

매개변수 초깃값 설정

- 가중치의 대칭적인 구조를 무너뜨리려면 초깃값을 무작위로 설정해야 한다.
- Xavier 초깃값: 각 층의 활성화값들을 광범위하게 분포시킬 목적으로 가중치의 적절한 분포 착지, Sigmoid나 tanh 등의 S자 모양 곡선
  - 앞 계층의 노드가 n개라면 표준편차가 $\frac{1}{\sqrt{n}}$인 분포 사용
- He 초깃값: ReLU에 특화된 초깃값
  - 앞 계층의 노드가 n개라면 표준편차가 $\sqrt{\frac{2}{n}}$인 분포 사용

매개변수 갱신

W: Parameter, $\frac{\partial L}{\partial W}$: derivation of loss function, η: learning rate(hyper-parameter)

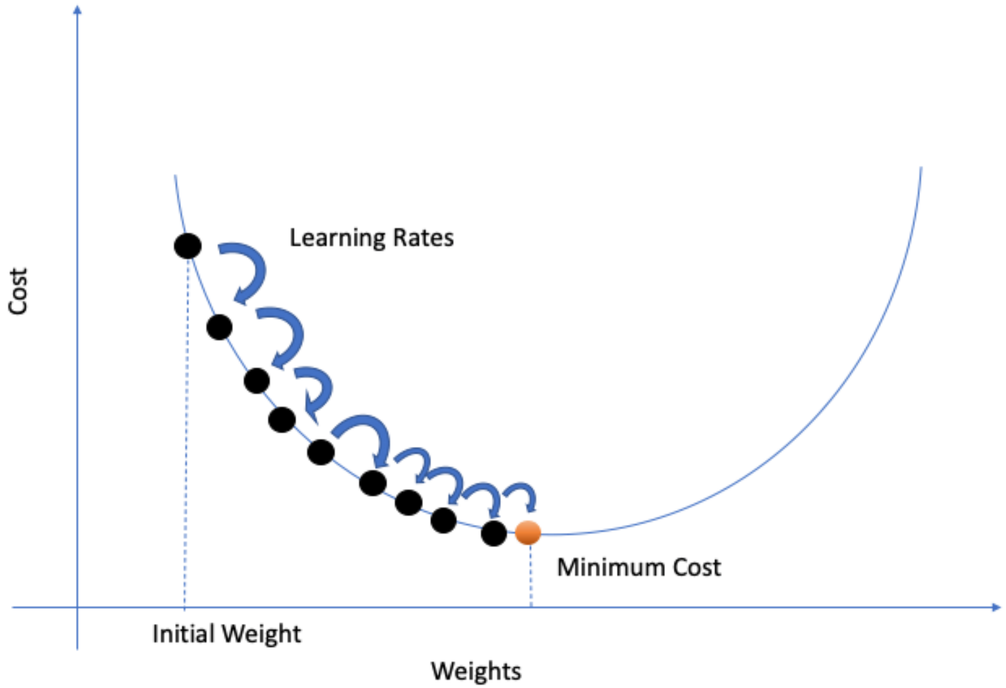
1. Stochastic Gradient Descent(SGD)
2. Momentum
3. AdaGrad
4. Adam

Stochastic Gradient Descent(SGD)

W: Parameter, $\frac{\partial L}{\partial W}$: derivation of loss function, η: learning rate(hyper-parameter)

$$W \longleftarrow W - \eta \frac{\partial L}{\partial W}$$

- 기울어진 방향으로 일정한 거리만 가겠다는 단순한 방법

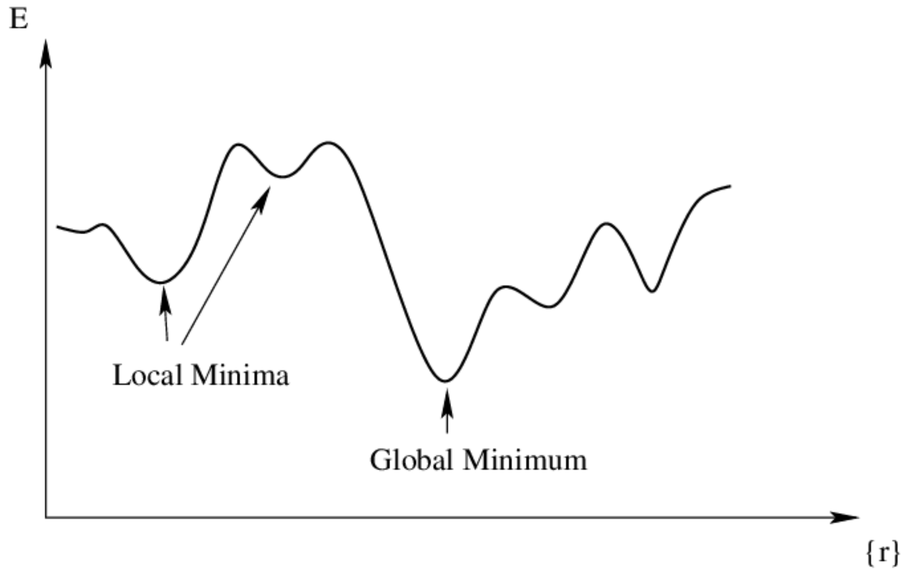


Stochastic Gradient Descent(SGD)

W: Parameter, $\frac{\partial L}{\partial W}$: derivation of loss function, η: learning rate(hyper-parameter)

$$W \longleftarrow W - \eta \frac{\partial L}{\partial W}$$

- 기울어진 방향으로 일정한 거리만 가겠다는 단순한 방법
- Cons
  - 지그재그로 이동하여 비효율적이다.
  - 비등방성 함수에서는 탐색 경로가 비효율적: Local, Global Minimum



Momentum

W: Parameter, $\frac{\partial L}{\partial W}$: derivation of loss function, η: learning rate(hyper-parameter)

$$W \longleftarrow W + v$$

$$v \longleftarrow \alpha v - \eta \frac{\partial L}{\partial W}$$

- 기울어진 방향으로 ~~일정한 거리만 가겠다는 방법~~ 기울기에 따라 다른 이동 거리를 이야기
  - 모멘텀의 원리: 공이 그릇의 곡면(기울기)에 따라 구르듯 움직인다.
- Pros: 지그재그 정도가 '덜'하다.

AdaGrad

$$W \longleftarrow W - \eta \frac{1}{\sqrt{h}} \frac{\partial L}{\partial W}, h \longleftarrow h + \frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$$

- 학습률 감소 관점) 매개변수 '전체'의 학습률 값을 일괄적으로 낮추는 방법, '각각의' 매개변수에 '맞춤형 값'
  - h: 과거의 기울기 값을 제곱하여 계속 더하는 방식, 매개변수의 원소 중 많이 움직인 원소는 학습률이 낮아진다.
- Pros: 최솟값을 향해 효율적으로 움직임

Adam

- 이동 관점) 그릇 바닥을 구르는 듯한 움직임 + 학습률 감소 관점) 매개변수의 원소마다 적응적으로 갱신 정도를 조정
- Pros: 하이퍼파라미터의 '편향 보정'이 진행된다.

적절한 하이퍼파라미터 찾기

- 하이퍼파라미터 최적화
  - 최적값이 존재하는 범위를 줄여 ‘대략적’으로 값 정하기
    - 1. 대략적인 범위를 설정하고
    - 2. 그 범위에서 무작위로 하이퍼파라미터 값을 고르고,
    - 3. 그 값으로 정확도를 평가
    - 4. 특정횟수 반복하여, 그 정확도와 결과를 보고 하이퍼파라미터의 범위를 좁힌다

Regularization일반화 strategy

- 모델 복잡도에 대한 패널티로 Overfitting을 예방하고 일반화 성능을 높인다.
- 기법
  - Early stopping: 과적합이 되기 전에 학습을 멈추는 방식
  - 가중치 감소: 학습과정에 큰 가중치에 대해서는 그에 상응하는 큰 페널티(L1, L2 Regularization)를 부과하여 오버 피팅을 억제하는 방법

$$L(W) \longleftarrow L(W) + \frac{1}{2}\lambda W^2, \frac{\partial L}{\partial W} \longleftarrow \frac{\partial L}{\partial W} + \lambda W$$

$$W \longleftarrow W - \eta \frac{\partial L}{\partial W}$$

  - 드롭아웃(Dropout): 뉴런을 임의로 삭제하면서 학습하는 방법
  - 앙상블 학습(Ensemble): 개별적으로 학습시킨 여러 모델의 출력을 평균 내어 추론하는 방법

Chapter 7. 합성곱 신경망(CNN)

1. 합성곱 계층
2. 풀링 계층
3. 합성곱/풀링 계층 구현하기
4. CNN 구현하기
5. CNN 시각화하기
6. 대표적인 CNN

합성곱 신경망(CNN) 구조

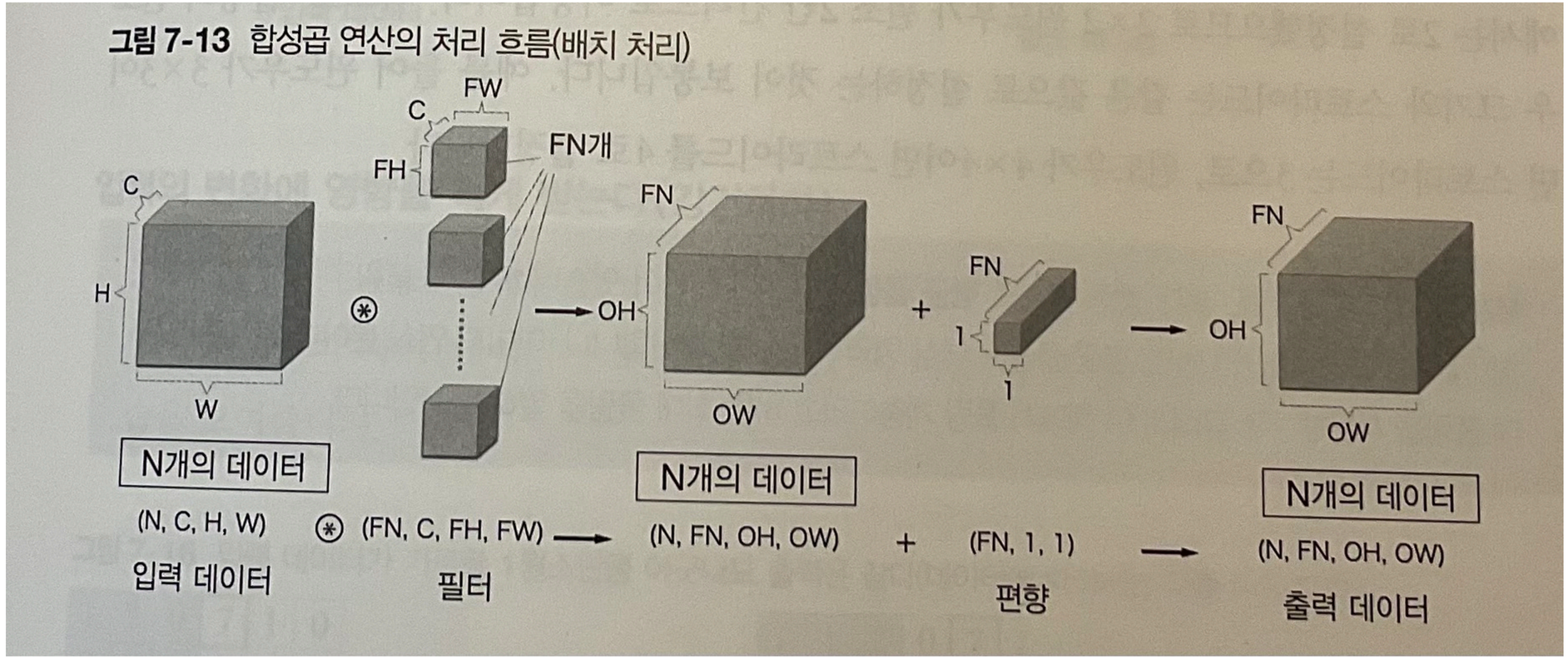
- 기존: Affine-ReLU
- CNN: Conv-ReLU-(Pooling)
  - 단일 곱셈-누산: [Input feature map] ⊙ [kernel] ⟶ [Output feature map]
  - Convolutional Layer
  - Pooling Layer
- 고유 용어
  - Padding: 입력데이터에 주변을 특정 값으로 채우기
  - Stride: 필터를 적용하는 위치의 간격

Convolution Layer

- 2차원 데이터: $(H, W) \odot (FH, FW) \longrightarrow (OH, OW)$
  - 입력 크기 (H, W), 필터 크기 (FH, FW), 출력 크기 (OH, OW), padding P, stride S
    - $OH = \frac{H + 2P - FH}{S} + 1$
    - $OW = \frac{W + 2P - FW}{S} + 1$
- 3차원 데이터: $(C, H, W) \odot (C, FH, FW) \longrightarrow (1, OH, OW)$
  - 블록 사용: $(C, H, W) \odot (FN, C, FH, FW) \longrightarrow (FN, OH, OW)$
  - 편향 사용: $(C, H, W) \odot (FN, C, FH, FW) \longrightarrow (FN, OH, OW) + (FN, 1, 1) \longrightarrow (FN, OH, OW)$
  - 배치 사용: $(N, C, H, W) \odot (FN, C, FH, FW) \longrightarrow (N, FN, OH, OW) + (FN, 1, 1) \longrightarrow (N, FN, OH, OW)$

CNN 연산 처리

- $(N, C, H, W) \odot (FN, C, FH, FW) \longrightarrow (N, FN, OH, OW) + (FN,1,1) \longrightarrow (N, FN, OH, OW)$



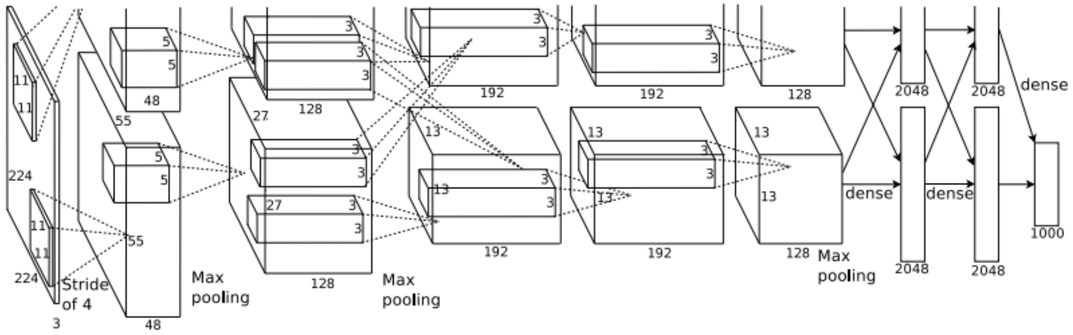
Pooling Layer

- 학습해야 할 매개변수가 없다.
- 채널 수가 변하지 않는다.
- 입력의 변화에 영향을 적게 받는다(강건하다)
- 종류
  - Max pooling
  - Average pooling

대표적인 CNN

- 합성곱 계층을 여러 겹 쌓으면, 층이 깊어지면서 더 복잡하고 추상화된 정보가 추출된다.

- AlexNet



- LeNet