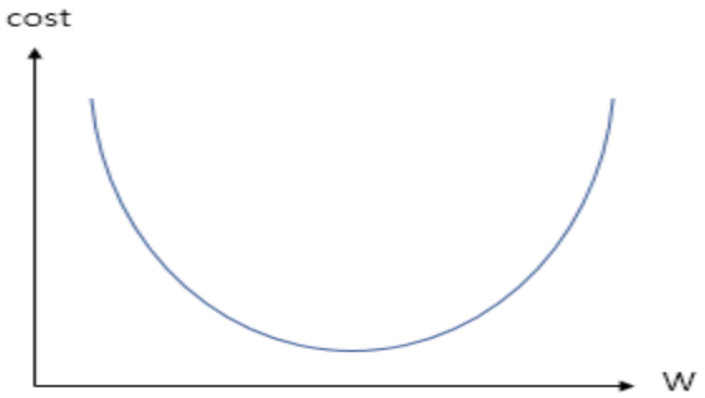
경사 하강법
(Gradient Descent)

경사하강법이란?

비용 함수를 최소화하는 매개 변수를 찾기 위해 사용되는 알고리즘입니다.



경사하강법의 대략적인 순서는 다음과 같습니다.

1.임의의 매개변수를 정해 비용함수의 시작지점(x축) 으로 지정합니다.

2.해당 매개변수로 모델의 오차를 구한 다음, 비용함수의 시작지점(y축) 으로 지정합니다.

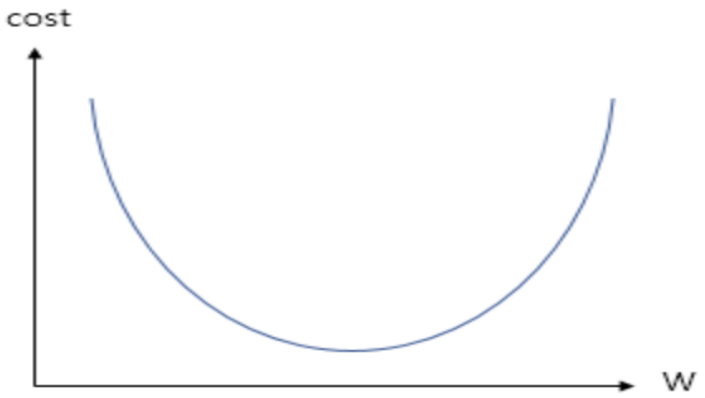
3.시작 지점에서 다음 지점으로 갈 방향을 정하기 위해, 시작 지점의 기울기를 계산합니다.

4.기울기(Gradient)와 보폭(Learning rate)를 사용해 다음 지점으로 이동합니다.

5.위의 과정을 최소값에 도달 할 때 까지 반복합니다.

경사하강법이란?

비용 함수를 최소화하는 매개 변수를 찾기 위해 사용되는 알고리즘입니다.



경사하강법의 대략적인 순서는 다음과 같습니다.

1.임의의 매개변수를 정해 비용함수의 시작지점(x축) 으로 지정합니다.

2.해당 매개변수로 모델의 오차를 구한 다음, 비용함수의 시작지점(y축) 으로 지정합니다.

3.시작 지점에서 다음 지점으로 갈 방향을 정하기 위해, 시작 지점의 기울기를 계산합니다.

4.기울기(Gradient)와 보폭(Learning rate)를 사용해 다음 지점으로 이동합니다.

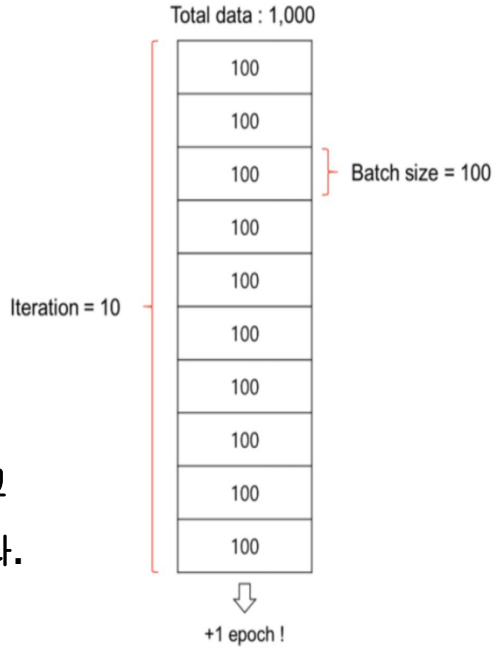
5.위의 과정을 최소값에 도달 할 때 까지 반복합니다.

배치 사이즈란?

기울기 업데이트 시, 훈련세트를 몇 개씩 묶어서 사용할 지에 대한 정보입니다.

- 훈련세트의 사이즈가 크다면, 전체 훈련세트를 한 번에 처리해 기울기를 업데이트 하는 것 보단, 아주 작은 훈련세트로 나눠서 배치 단위마다 기울기를 업데이트 하는게 빠를 것입니다.
- 예를 들어, 훈련세트가 5000000일 때, 배치 사이즈를 5000000개라고 한다면, 1개의 묶음이 나오게 되므로 기울기는 1번 업데이트 됩니다.
- 하지만, 훈련세트가 5000000개일 때, 배치 사이즈를 100000개라고 한다면, 50개의 묶음이 나오게 되므로 기울기는 50번 업데

Epoch, Batch size, Iterations 용어 정리



Epoch : 모든 훈련 데이터셋을 학습하는 횟수입니다.

- 주의 사항 : 너무 많은 **Epoch**는 **overfitting**의 위험이 있습니다.

Batch Size :

- 훈련 데이터셋 중 몇 개의 데이터를 묶어서 가중치 값을 갱신할 것인지에 대한 정보입니다.
- 위 그림처럼, 훈련 데이터셋의 개수는 **1000**개이며, **1 Epoch**를 진행한다고 했을 때 **Batch Size**를 **100**으로 설정하면 총 **10**번의 갱신 과정을 거칩니다.
- 주의 사항
  - **Batch Size**는 메모리에 적재시킬 수 있을 만큼의 **Size**로 정하는게 좋습니다.
  - **Batch Size**를 너무 작게하면 **iteration**이 증가하여 학습시간 **(forward + backward propagation)** 이 오래 걸릴 수 있습니다.

Iterations : 한 **Epoch**를 진행하기 위해, 몇 번의 가중치 갱신이 이루어지는지에 대한 정보입니다.

- 위의 예시와 동일하게 훈련 데이터셋의 개수는 1000개이며, 1 Epoch를 진행한다고 했을 때 Batch Size를 100으로 설정하면 총 10번의 갱신 과정을 거칩니다.
- 즉, iterations은 10번 입니다.

배치 사이즈 정하는 방법

- **Batch Size**는 보통 **2**의 **n**승으로 지정하는데, 메모리에 적재시킬 수 있을 만큼의 **Size**로 정하는게 좋습니다.
  - 전형적인 배치 사이즈는 **64 ~ 512** 입니다.

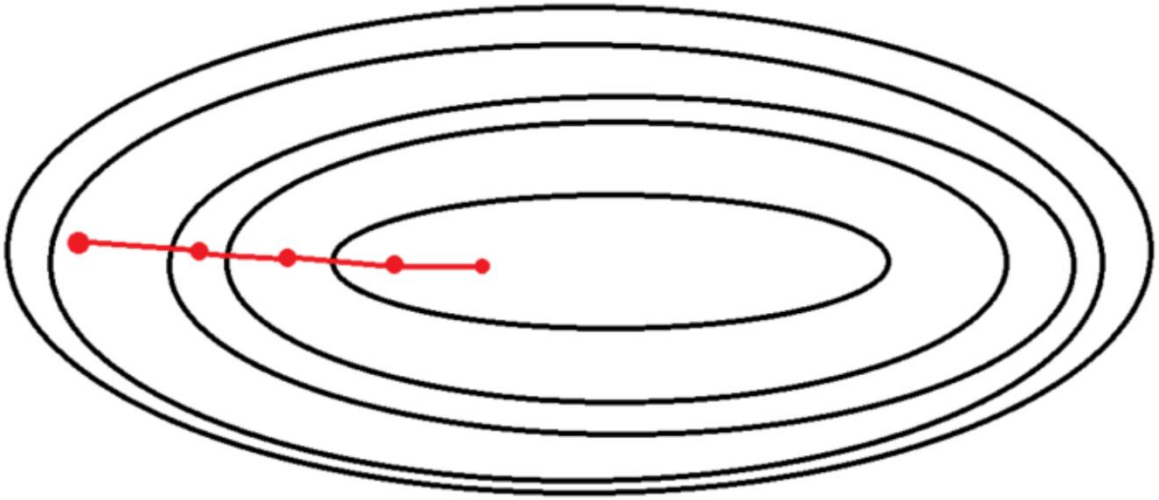
- 또한, **Batch Size**는 가능하면 훈련세트 개수와 딱 맞아 떨어지도록 나누는 것이 좋습니다.
  - 마지막 남은 배치 사이즈가 이전 배치 사이즈와 다르면, 해당 배치의 데이터가 학습에 더 큰 비중을 갖게 됩니다.
  - 예를 들어, **530** 개의 데이터를 **100**개의 배치로 나누면, 각 배치 속 데이터는 **1/100** 만큼의 영향력을 갖게 됩니다.
  - 그러나 마지막 배치**(30**개**)**의 데이터는 **1/30**의 영향력을 갖게 되어 과평가되는 경향이 있습니다.
  - 그렇기 때문에 보통 마지막 배치의 사이즈가 다를 경우는 이를 버리는 방법을 사용합니다.

- **Batch Size**를 너무 작게하면 학습시간 **(forward + backward propagation)**이 오래 걸릴 수 있습니다.

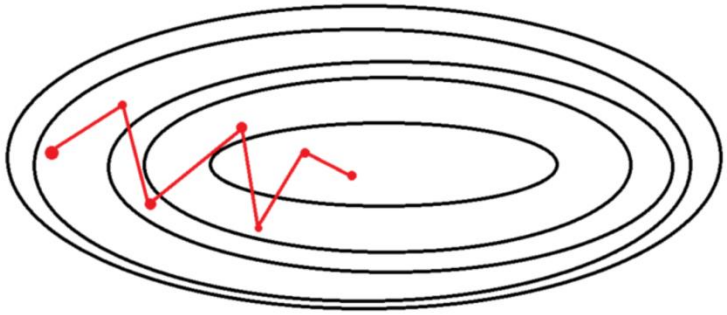
- 보통 훈련세트가 **2000**개보다 작을 때는 나누지 않는게 좋습니다.
  - 작은 훈련세트를 갖고 계시는 경우, 전체 훈련세트를 꽤 빨리 처리할 수 있기 때문입니다.

배치 경사하강법 (Batch Gradient Descent, BGD)

- 배치 사이즈가 훈련세트 사이즈와 동일한 경사하강법입니다.
- 따라서, 전체 훈련세트를 한 번에 처리해 기울기를 업데이트 합니다.
- **특징**
  - 항상 같은 데이터에 대해 경사를 구하기 때문에, 수렴이 안정적입니다. (위 그림 참고)
  - 전체 훈련세트를 한 번에 처리하기 때문에, 메모리가 가장 많이 필요하며
  - 긴 시간이 소요됩니다.

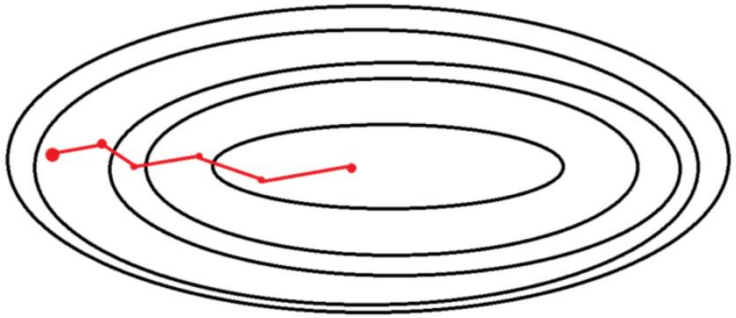


확률적 경사하강법 (Stochastic Gradient Descent, SGD)



- 배치 사이즈가 1개인 경사하강법입니다.
- 전체 훈련세트 중, 랜덤하게 하나의 데이터를 선택해 기울기를 업데이트하기 때문에 확률적이라고 부릅니다.
- 따라서, 1개의 훈련데이터만 처리해 기울기를 업데이트 합니다.
- **특징**
  - ➢ 수렴에 Shooting이 발생합니다.
    - ✓ 1개의 데이터마다 비용함수의 기울기는 약간씩 다르기 때문에, 각각의 데이터에 대해 미분을 수행하면 기울기의 방향이 매번 크게 바뀝니다.
    - ✓ 전역 최저점(Global Minimum)에 수렴하기는 어렵지만, 지역 최저점(Local Minimum)에 빠질 확률을 줄여줍니다.
- 훈련데이터를 1개씩 처리하기 때문에, 벡터화 과정에서 대부분의 속도를 잃으며 GPU의 병렬 처리를 잘 활용하지 못합니다.

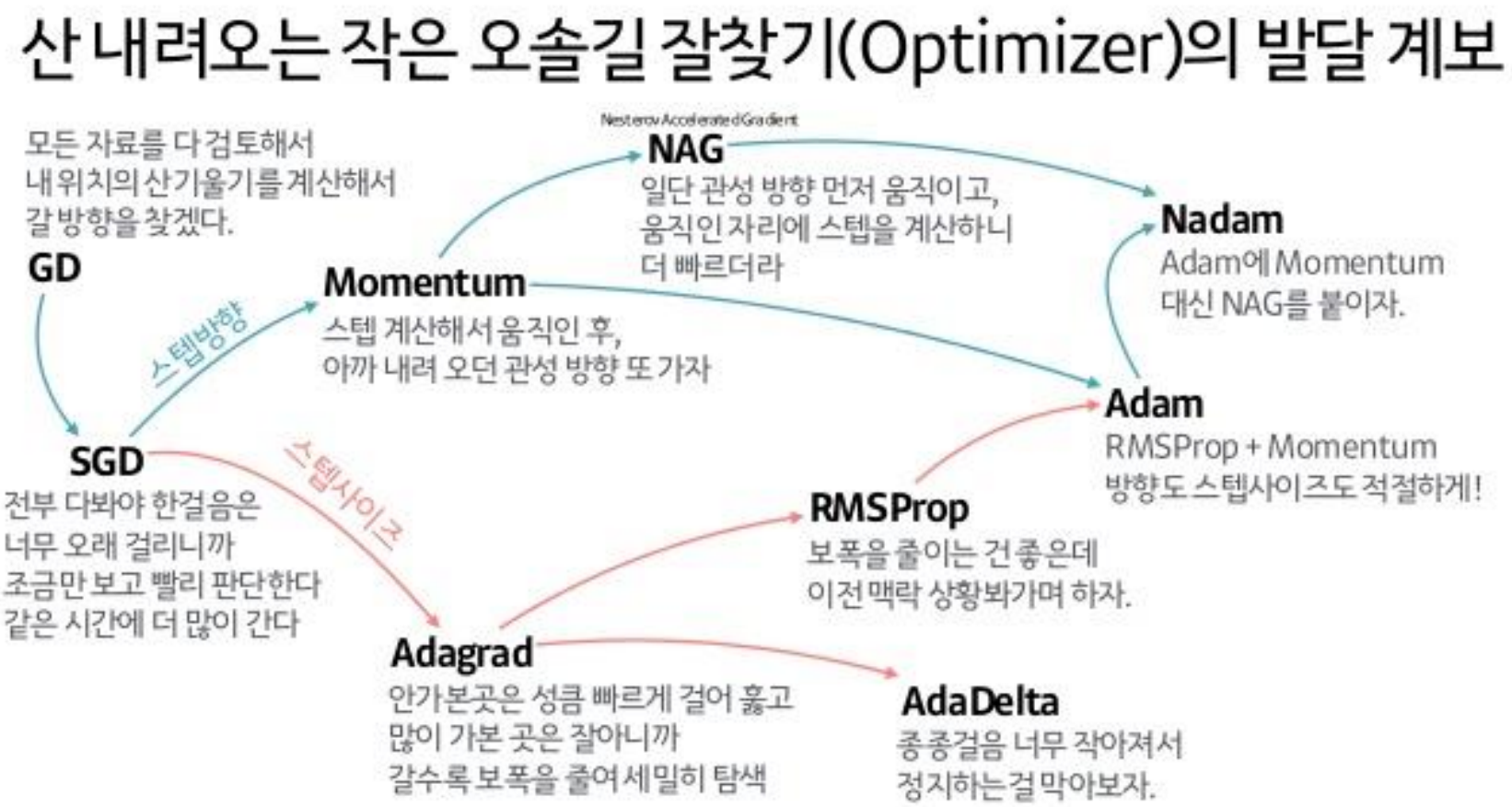
미니 배치 확률적 경사하강법
(Mini-Batch Stochastic Gradient Descent, MSGD)



- **배치 경사하강법과 확률적 경사하강법의 절충안으로, 전체 훈련세트를 1~M 사이의 적절한 batch size로 나누어 학습하는 것입니다.**
  - ➢ 예를 들어, 전체 훈련세트가 5000000개일 때, 배치 사이즈를 100000개라고 한다면, 50개의 묶음이 나오게 되므로 기울기는 1 Epoch 당 50번 업데이트 됩니다.
- 특징
  - ➢ 훈련세트의 사이즈가 클 경우, 배치 경사하강법보다 속도가 빠릅니다.

  Shooting이 적당히 발생합니다. (Local Minimum를 어느정도 회피할 수 있습니다.)

경사하강법 종류


산 내려오는 작은 오솔길 잘찾기(Optimizer)의 발달 계보
모든 자료를 다 검토해서
내 위치의 산기울기를 계산해서
갈 방향을 찾겠다.
GD
스텝방향
SGD
전부 다봐야 한걸음은
너무 오래 걸리니까
조금만 보고 빨리 판단한다
같은 시간에 더 많이 간다
Momentum
스텝 계산해서 움직인 후,
아까 내려 오던 관성 방향 또 가자
Nesterov Accelerated Gradient
NAG
일단 관성 방향 먼저 움직이고,
움직인 자리에 스텝을 계산하니
더 빠르더라
Nadam
Adam에 Momentum
대신 NAG를 붙이자.
Adam
RMSProp + Momentum
방향도 스텝사이즈도 적절하게!
스텝사이즈
RMSProp
보폭을 줄이는 건 좋은데
이전 맥락 상황봐가며 하자.
Adagrad
안가본곳은 성큼 빠르게 걸어 훑고
많이 가본 곳은 잘아니까
갈수록 보폭을 줄여 세밀히 탐색
AdaDelta
종종걸음 너무 작아져서
정지하는걸 막아보자.

정리

- 배치 사이즈는 경사 하강법 1회 업데이트에 사용되는 데이터의 사이즈입니다.
- 배치 사이즈에 따라 아래와 같이 나눌 수 있습니다.
  - 배치 경사 하강법 (BGD)
    - ✓ 배치 크기 == 전체 학습 데이터
  - 확률적 경사 하강법 (SGD)
    - ✓ 배치 크기 == 1
  - 미니 배치 확률적 경사 하강법 (MSGD)
    - ✓ 배치 크기 == 사용자가 지정
- 배치 사이즈는 아래 조건에 따라 정합니다.
  - 2의 n승이 좋습니다.
  - 메모리 크기를 고려해야합니다.
  - 가능하면 전체 데이터 수가 나누어 떨어지도록 정하고, 아니라면 마지막 배치는 버리는게 좋습니다.
  - 훈련 세트의 크기가 2000보다 작다면 배치 경사하강법이 빠릅니다.