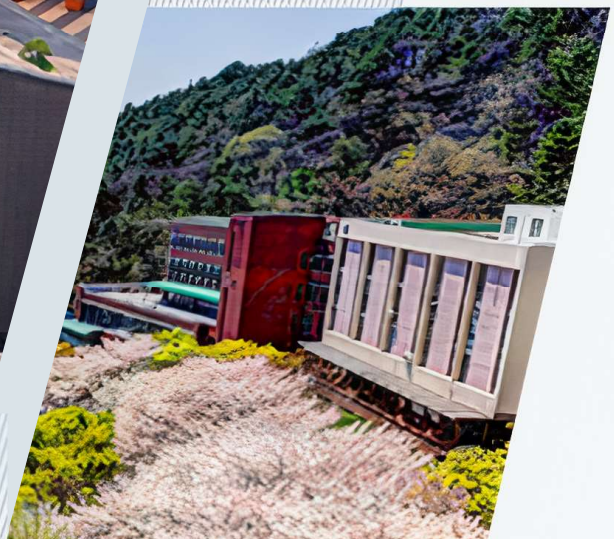
Dong-A Univ. (ISPL)

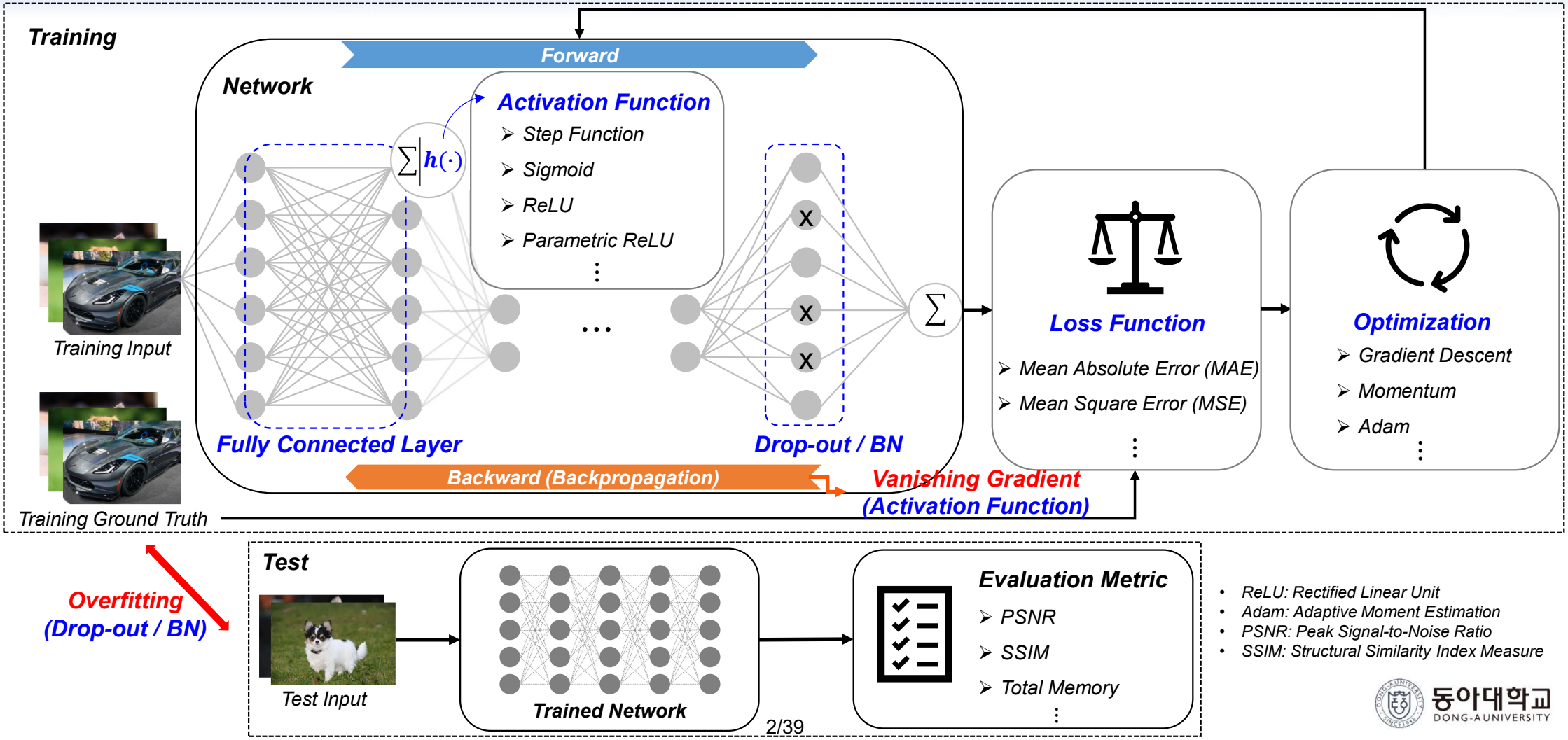
동아대학교
DONG-A UNIVERSITY

Optimization

컴퓨터AI공학부 AI학과
2024년 1학기 인공지능



Overall Architecture of Deep Learning



- *ReLU: Rectified Linear Unit*
- *Adam: Adaptive Moment Estimation*
- *PSNR: Peak Signal-to-Noise Ratio*
- *SSIM: Structural Similarity Index Measure*

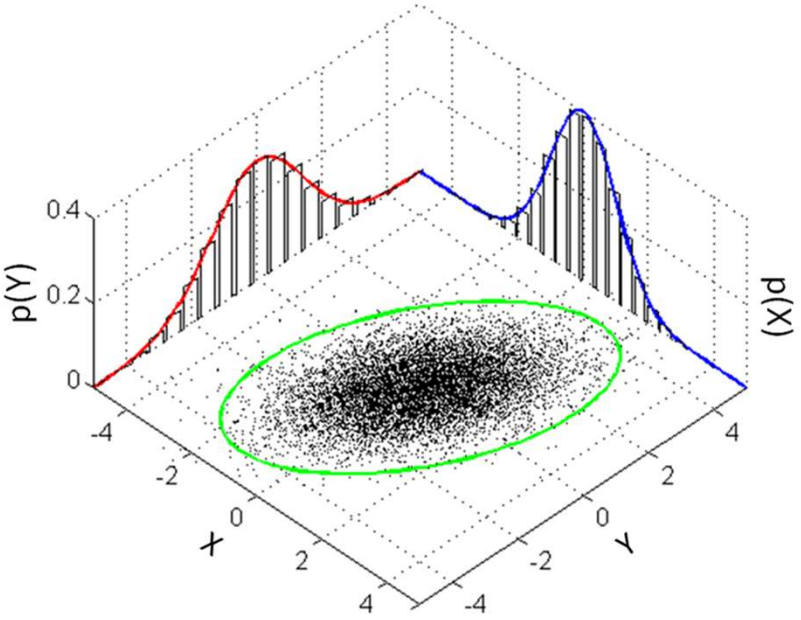
Normalization (정규화)

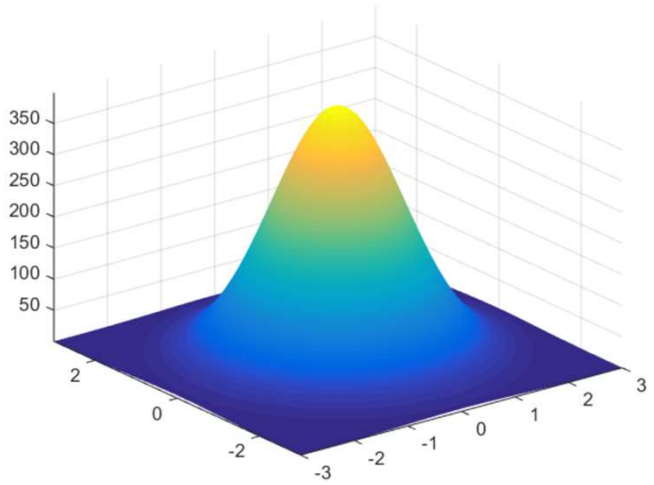
Normalization (정규화)

(1) Min-Max 정규화

(2) 표준 정규화

기기명	온도	진동	전류값	사용시간
x1	a1			
x2	a2			





다변량 정규 분포(Multivariate Normal Distribution)

Min-Max 정규화

① 정규화(Normalization) : 변수 X를 정규화하면 값 X'

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

⇒ i) $X = X_{min}$: $X' = 0$

ii) $X = X_{max}$: $X' = 1$

0과 1 사이의 값으로 변함

- 0~100사이라면, 1-0 / 100-0 = 0.01

	feature_1	feature_2	feature_3
0	1.280187	-1.156924	-81.977837
1	0.519024	0.277231	-78.493732
2	-1.340744	0.564647	51.682415
3	0.880929	1.037069	45.883654
4	-1.260126	1.257954	15.080874
5	0.401379	-1.310234	90.150390
6	-1.142048	0.243710	57.606259
7	0.566775	-0.396015	64.846291
8	-0.724533	-0.510327	-5.383149
9	-1.615751	-0.056775	130.638733
10	-0.721374	-0.627100	108.228715

변환 전

	feature_1	feature_2	feature_3
0	1.280187	-1.156924	0.000000
1	0.519024	0.277231	0.016387
2	-1.340744	0.564647	0.628645
3	0.880929	1.037069	0.601371
4	-1.260126	1.257954	0.456496
5	0.401379	-1.310234	0.809571
6	-1.142048	0.243710	0.656506
7	0.566775	-0.396015	0.690558
8	-0.724533	-0.510327	0.360248
9	-1.615751	-0.056775	1.000000
10	-0.721374	-0.627100	0.894599

정규화 변환 후

표준 정규화

② 표준화(Standardization) : 변수 X를 표준화한 값 X'

$$X' = \frac{X - \mu}{\sigma}$$

평균u = 0 표준편차σ = 1

	feature_1	feature_2	feature_3
0	1.280187	-1.156924	-81.977837
1	0.519024	0.277231	-78.493732
2	-1.340744	0.564647	51.682415
3	0.880929	1.037069	45.883654
4	-1.260126	1.257954	15.080874
5	0.401379	-1.310234	90.150390
6	-1.142048	0.243710	57.606259
7	0.566775	-0.396015	64.846291
8	-0.724533	-0.510327	-5.383149
9	-1.615751	-0.056775	130.638733
10	-0.721374	-0.627100	108.228715

변환 전

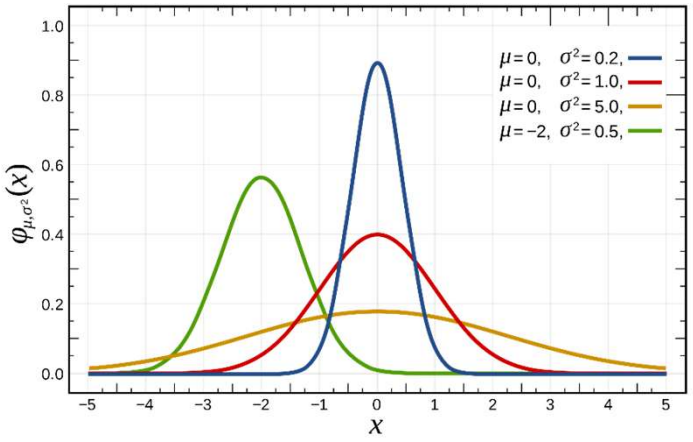
	feature_1	feature_2	feature_3
0	1.280187	-1.156924	-1.707156
1	0.519024	0.277231	-1.656828
2	-1.340744	0.564647	0.223561
3	0.880929	1.037069	0.139798
4	-1.260126	1.257954	-0.305147
5	0.401379	-1.310234	0.779229
6	-1.142048	0.243710	0.309130
7	0.566775	-0.396015	0.413712
8	-0.724533	-0.510327	-0.600749
9	-1.615751	-0.056775	1.364081
10	-0.721374	-0.627100	1.040369

표준화 변환 후

약 -2~ 2사이로 변환된 걸 볼 수 있다.

정규분포 (표준정규분포)

$$N(x|\mu,\,\sigma^2) \equiv \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

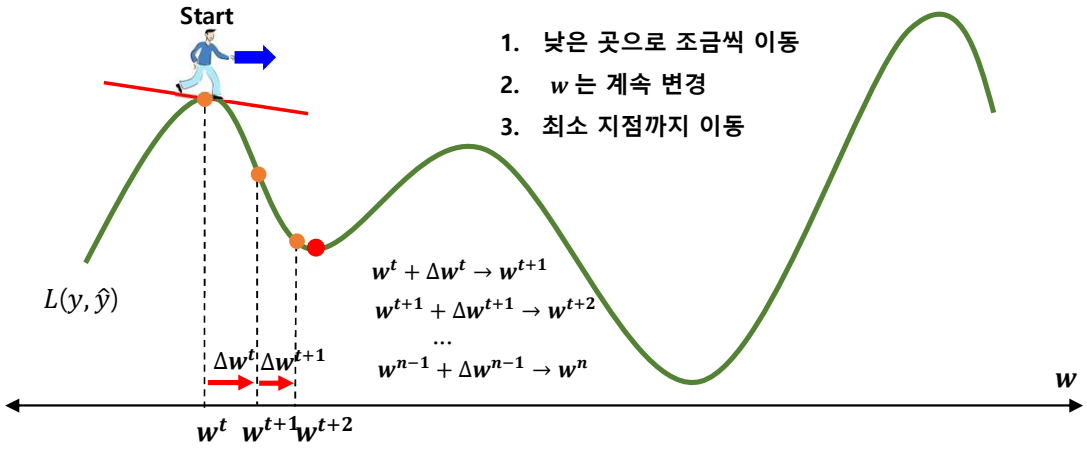


Contents

1. Gradient Descent
2. Stochastic Gradient Descent
3. Momentum
4. AdaGrad
5. RMSProp
6. Adam

[Review] Descent Method (하강법)

- 주어진 어떤 지점에서부터 오차가 더 작은 곳으로 이동하려는 방법
- 하강법을 위해 Backpropagation으로 **가중치**와 **편향**을 수정하는 과정을 확인

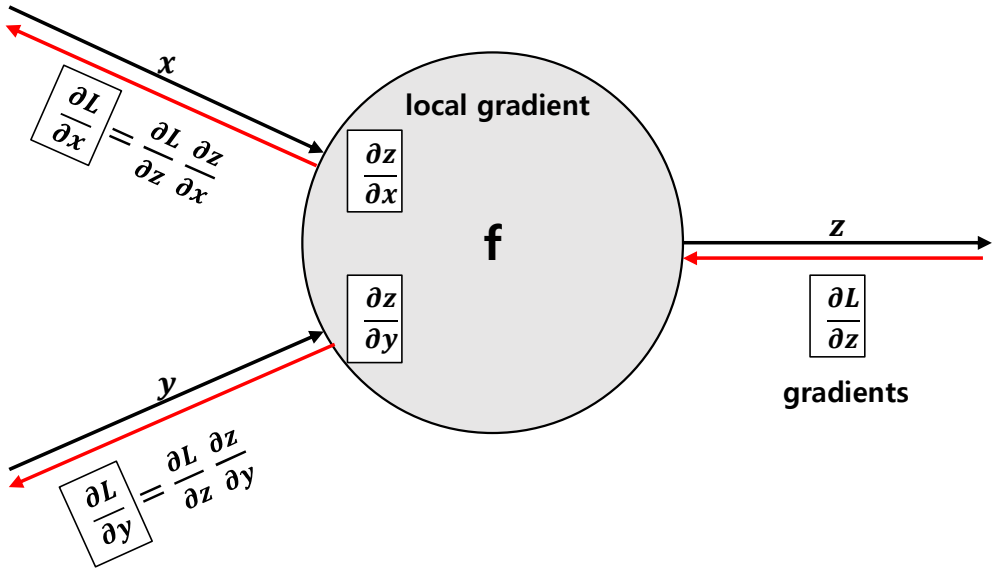


학습률(learning rate, step size)

$$w^{t+1} \leftarrow w^t - \mu \nabla f(w^t), \quad where \ \nabla f(w^t)\Delta w < 0$$

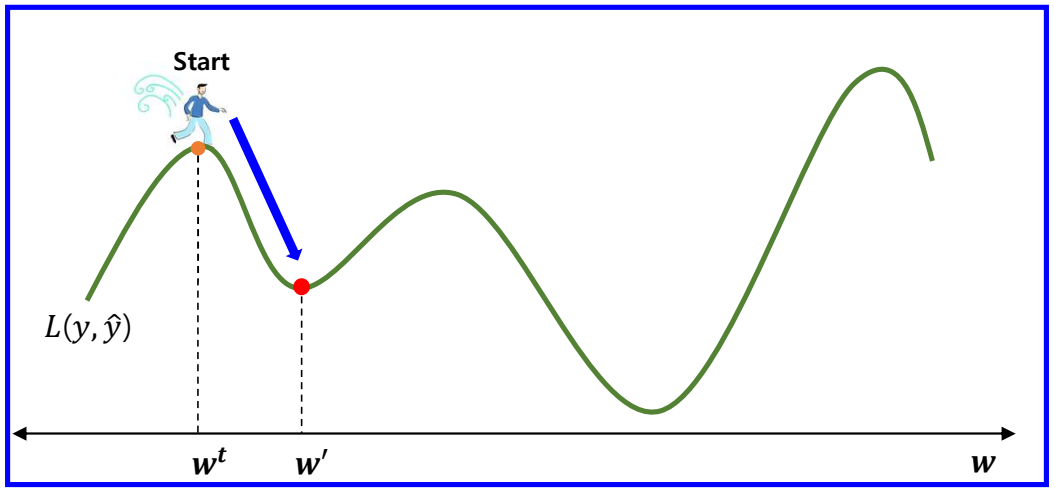
탐색 방향

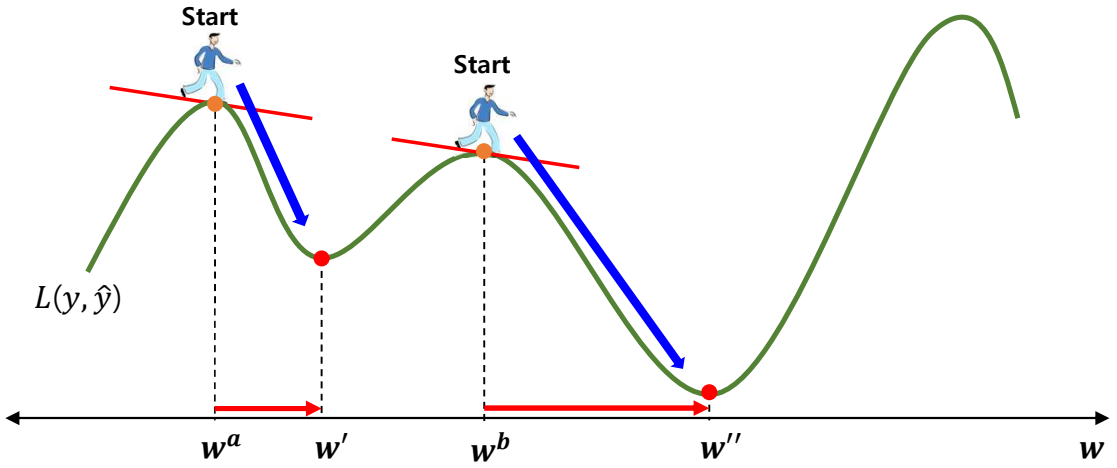
< Descent Method >



[Review] Descent Method (하강법)

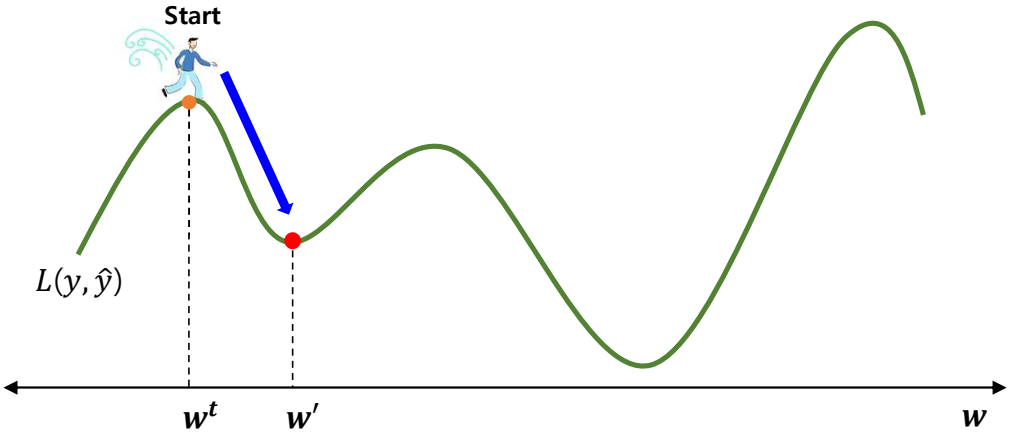
- **하강법보다 빠르게 내려가려면 어떻게 할까?**
- 초기 위치에 따라 도착지점이 정해지는 단점을 어떻게 해결할까?

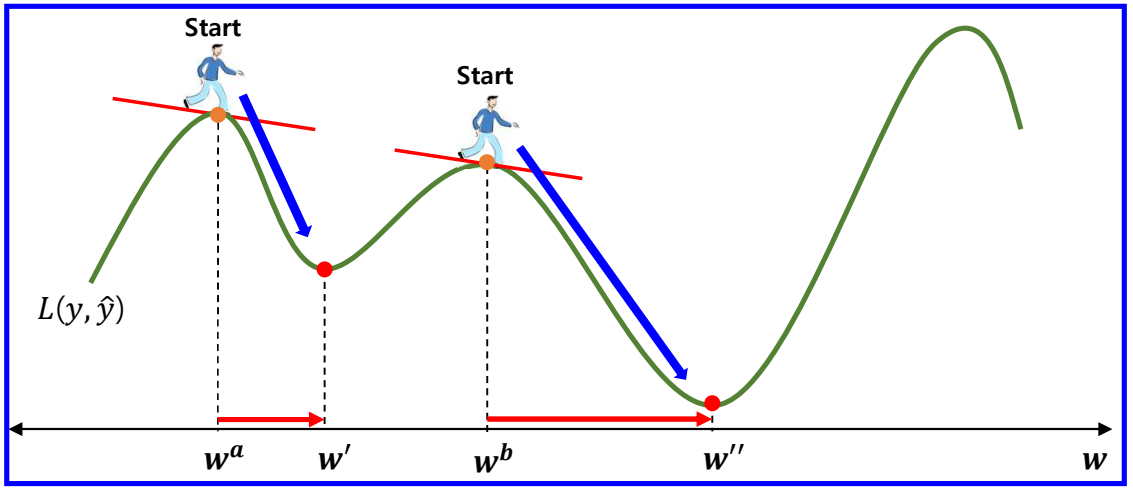




[Review] Descent Method (하강법)

- 하강법보다 빠르게 내려가려면 어떻게 할까?
- **초기 위치에 따라 도착지점이 정해지는 단점을 어떻게 해결할까?**

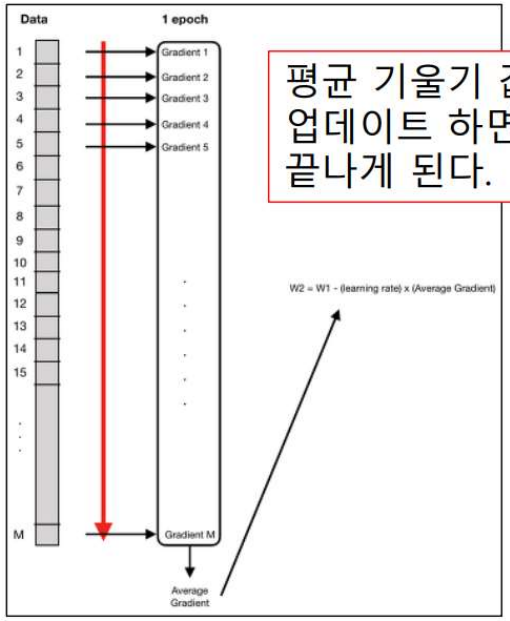




Gradient Descent (GD)

- **전체 학습데이터를 하나의 Batch로 묶어서 Gradient 값을 한번만 계산하여 가중치를 갱신함**

  ※ 일반적인 Batch 단위가 아닌 전체 데이터 셋이라는 점을 유의할 것



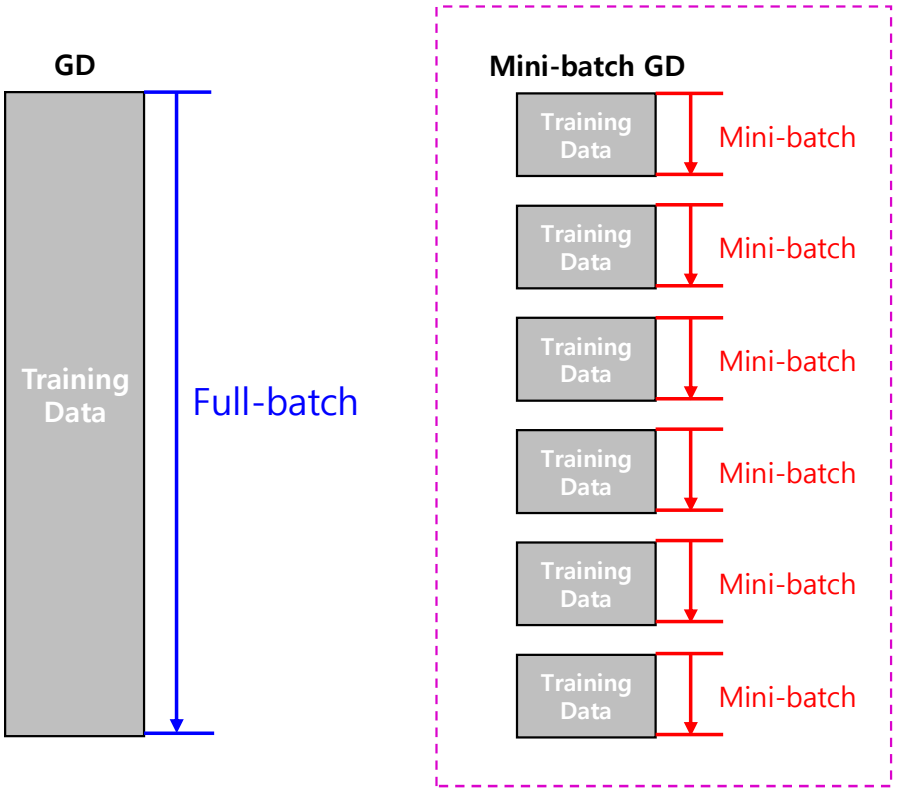
평균 기울기 값을 구해 매개변수를 업데이트 하면 한번의 epoch 가 끝나게 된다.

- **Pros.**
  - ✓ 전체 학습데이터에 대해 한번의 업데이트가 이루어지므로 전체적인 연산횟수가 적음
  - ✓ 전체 학습데이터에 대해 Gradient를 계산하므로, 수렴이 안정적으로 진행

- **Cons.**
  - ✓ 한 Step에 모든 학습데이터를 사용하기 때문에 학습이 오래 걸림
  - ✓ Local optimal 상태가 되면 빠져나오기 어려움
  - ✓ 모델 파라미터의 업데이터가 이루어지기 전까지 모든 학습데이터에 대해 저장하므로 많은 메모리가 필요

Mini-batch Gradient Descent (Mini-batch GD)

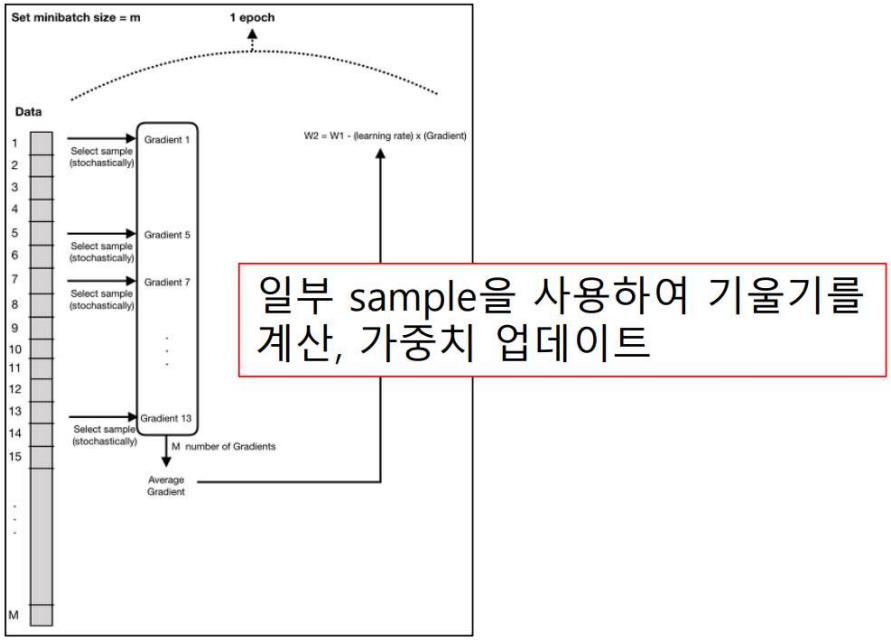
- 전체 학습데이터를 여러 개의 Batch 단위로 나누어서 Gradient 값을 계산하여 가중치를 갱신함



- **Pros.**
  - ✓ GD보다 local optimal에 빠질 위험이 적음
  - ✓ 병렬처리에 유리
  - ✓ 전체 학습데이터가 아닌 일부분만 사용하므로 메모리에 부하가 낮음
- **Cons.**
  - ✓ 적절한 크기의 Batch size를 설정하여야 함

Stochastic Gradient Descent (SGD)

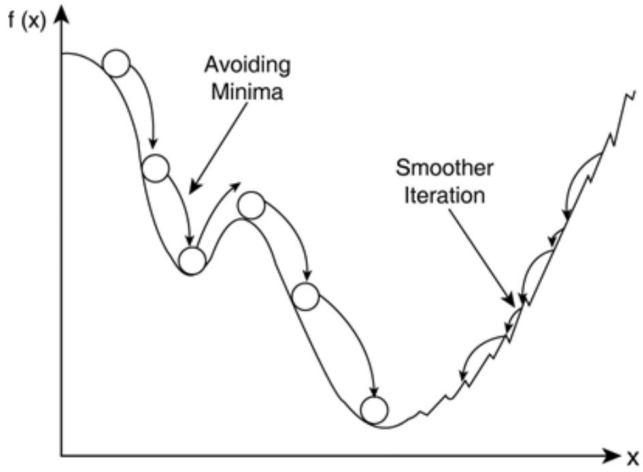
- **학습데이터 세트들 중 하나의 데이터를 Random하게 선택하여 Gradient를 갱신함**



- **Pros.**
  - ✓ 하나의 랜덤한 데이터를 고려하여 가중치를 갱신하므로 메모리 요구량이 낮음
  - ✓ GD에 비해 학습속도가 빠름
- **Cons.**
  - ✓ 가중치의 학습이 불안정할 수 있음
  - ✓ GD에 비해 정확도가 낮을 수 있음

Momentum

- 기울기 방향으로 힘을 받아 물체가 가속된다는 물리법칙을 적용
- Momentum을 적용하여 학습 방향이 바로 변하지 않고, 일정한 방향을 유지하며 움직임



$$v_t = \alpha v_{t-1} - \eta \frac{\partial L}{\partial W}$$

$$W = W + v_t$$

W : 갱신할 가중치 매개변수

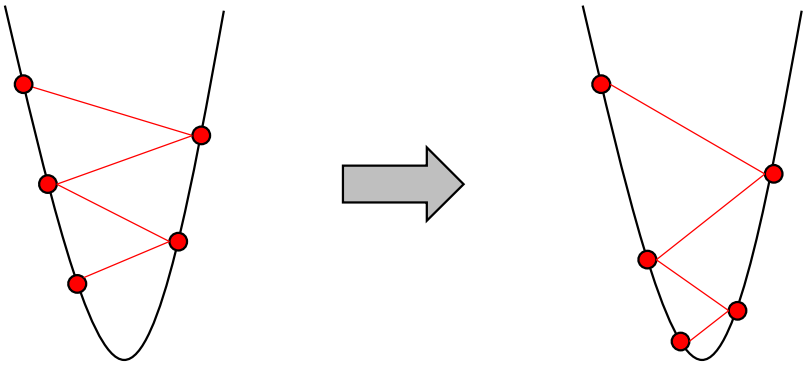
$\frac{\partial L}{\partial W}$: W에 대한 손실 함수의 기울기

* η : 학습률

- **Pros.**
  - ✓ 기존에 이동하는 방향에 대한 관성을 이용하여 Local minimum을 빠져나올 수 있음
- **Cons.**
  - ✓ 기존의 변수들 외에도 과거에 이동하는 양을 별도로 저장하므로 메모리 요구량이 증가

Adaptive Gradient (AdaGrad)

- **AdaGrad는 개별 매개변수에 적응적으로 Learning rate를 조정하면서 학습을 진행**
  - ✓ h 는 기존 기울기 값을 제곱하여 계속 더함
  - ✓ 매개변수를 갱신할 때 $1/h$ 를 곱해 학습률 조정



동일한 Learning rate을 사용하여 학습

Learning rate를 서서히 감소하며 학습

$$h \leftarrow h + \frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$$

$$W \leftarrow W - \eta \frac{1}{\sqrt{h}} \frac{\partial L}{\partial W}$$

W : 갱신할 가중치 매개변수

$\frac{\partial L}{\partial W}$: W에 대한 손실 함수의 기울기

* η : 학습률

h : 기존 기울기 값

- **Pros.**
  - ✓ 랜덤으로 들어오는 변수들에 대해 효율적으로 학습시켜 최적점을 빨리 찾을 수 있음
    - ➢ 학습을 진행하며 학습이 많이 된 변수라면 최적점 가까이 갔다고 판단
    - ➢ 학습이 아직 덜 된 변수라면 학습을 더 빨리해야 한다고 판단

- **Cons.**
  - ✓ 기존 기울기 값이 계속 누적되어 값이 커지게 되며, 전체 값이 작아지게 되고 학습을 할 수 없게 되는 문제발생

Ref) https://morioh.com/p/0631c4986e98

Root Mean Square Propagation (RMSprop)

- AdaGrad를 개선한 기법으로 갱신된 기울기에 weight를 달리 주는 기법
- α값은 사용자가 별도로 지정
- 과거의 기울기를 균일하게 더하지 않고 새로운 기울기 정보를 크게 반영

$$h \leftarrow \alpha h + (1-\alpha)\frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$$

$$W \leftarrow W - \eta \frac{1}{\sqrt{h}} \frac{\partial L}{\partial W}$$

W : 갱신할 가중치 매개변수

$\frac{\partial L}{\partial W}$: W에 대한 손실 함수의 기울기

* η : 학습률

h : 기존 기울기 값

Adaptive Moment Estimation (Adam)

- **Momentum + RMSProp**
- **Momentum의 지난 gradient의 지수 감소 평균을 사용**
  - 물체가 가속된다는 물리법칙을 적용
- **RMSProp의 지난 gradient의 제곱 지수 감소를 사용**
  - 갱신된 기울기에 weight를 달리 적용

Momentum ➡ $v_t = \alpha v_{t-1} - \eta \frac{\partial L}{\partial W}$

$W = W + v_t$

RMSProp ➡ $h \leftarrow \alpha h + (1-\alpha)\frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$

$W \leftarrow W - \eta \frac{1}{\sqrt{h}} \frac{\partial L}{\partial W}$

$$m_t = \beta_1 m_{t-1} - (1-\beta_1)\frac{\partial L}{\partial W}$$

$$v_t = \beta_2 v_{t-1} - (1-\beta_2)\frac{\partial L}{\partial W} \odot \frac{\partial L}{\partial W}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$W = W - \eta \frac{1}{\sqrt{\hat{v}_t}} \hat{m}_t$$

W : 갱신할 가중치 매개변수

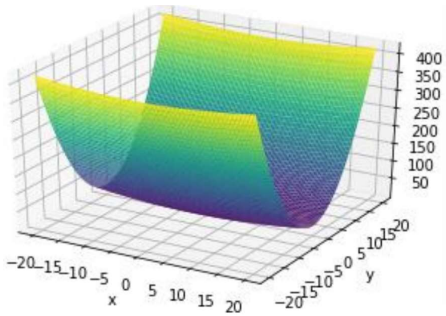
$\frac{\partial L}{\partial W}$: W에 대한 손실 함수의 기울기

* η : 학습률

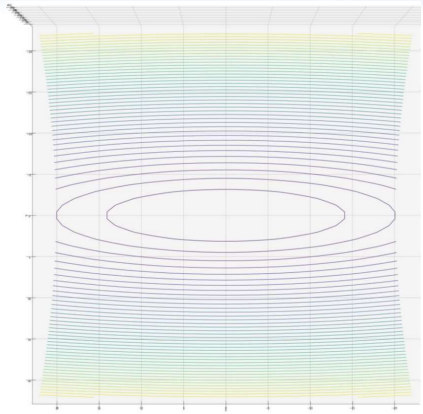
h : 기존 기울기 값

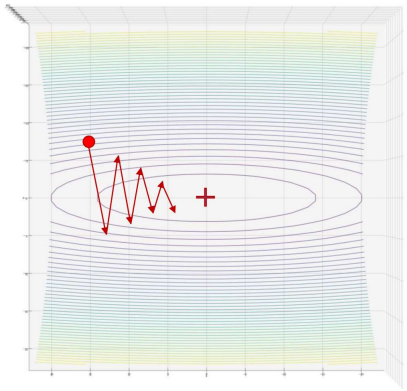
보통 설정은...
β_1 = 0.9
β_2 = 0.999

대표기법 비교

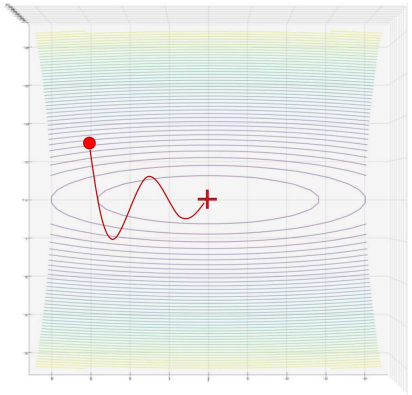


$$f(x, y) = \frac{1}{20}x^2 + y^2$$

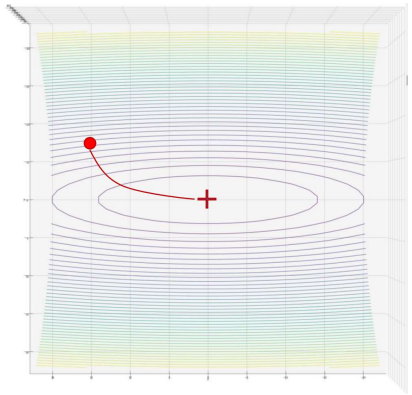




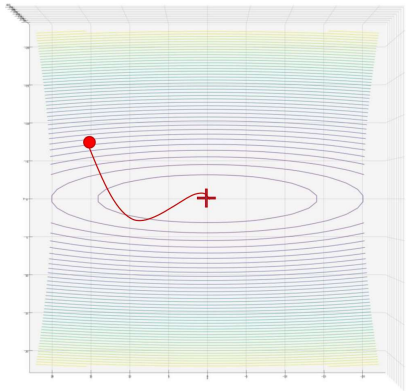
SGD



Momentum



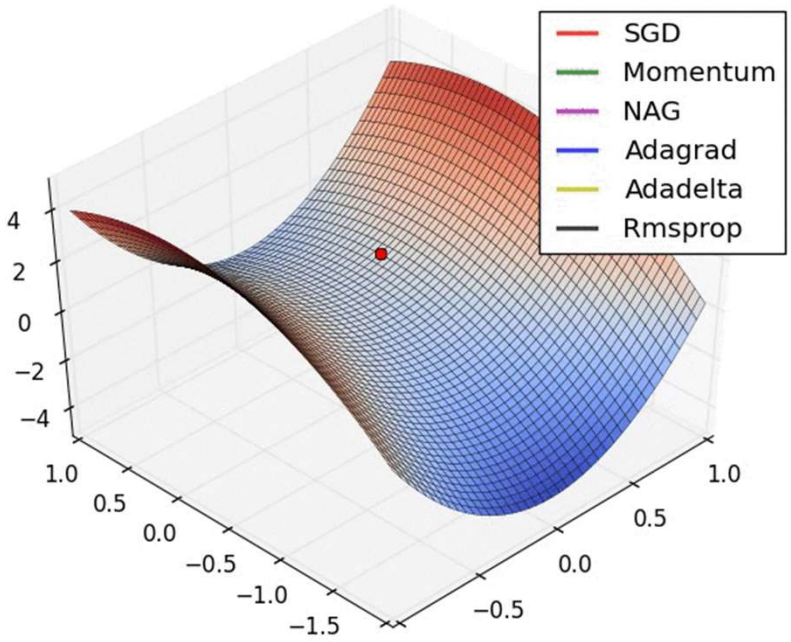
AdaGrad

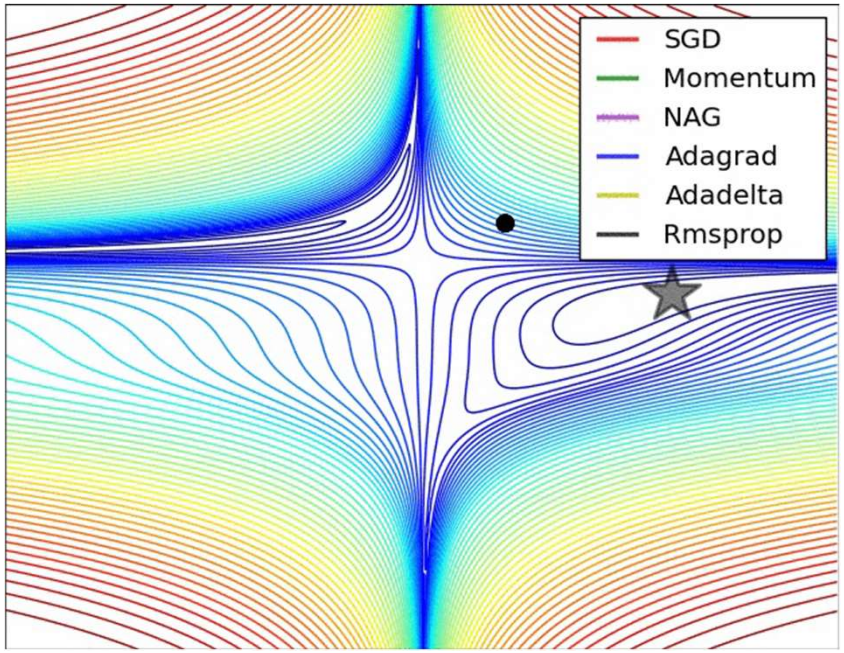


Adam

대표기법 비교

- GD
- SGD
- Momentum
- NAG
- Adagrad
- RMSProp
- Adam





Ref) https://truman.tistory.com/164

Questions & Answers

Dongsan Jun (dsjun@dau.ac.kr)
Image Signal Processing Laboratory (www.donga-ispl.kr)
Division of Computer·AI Engineering
Dong-A University, Busan, Rep. of Korea