

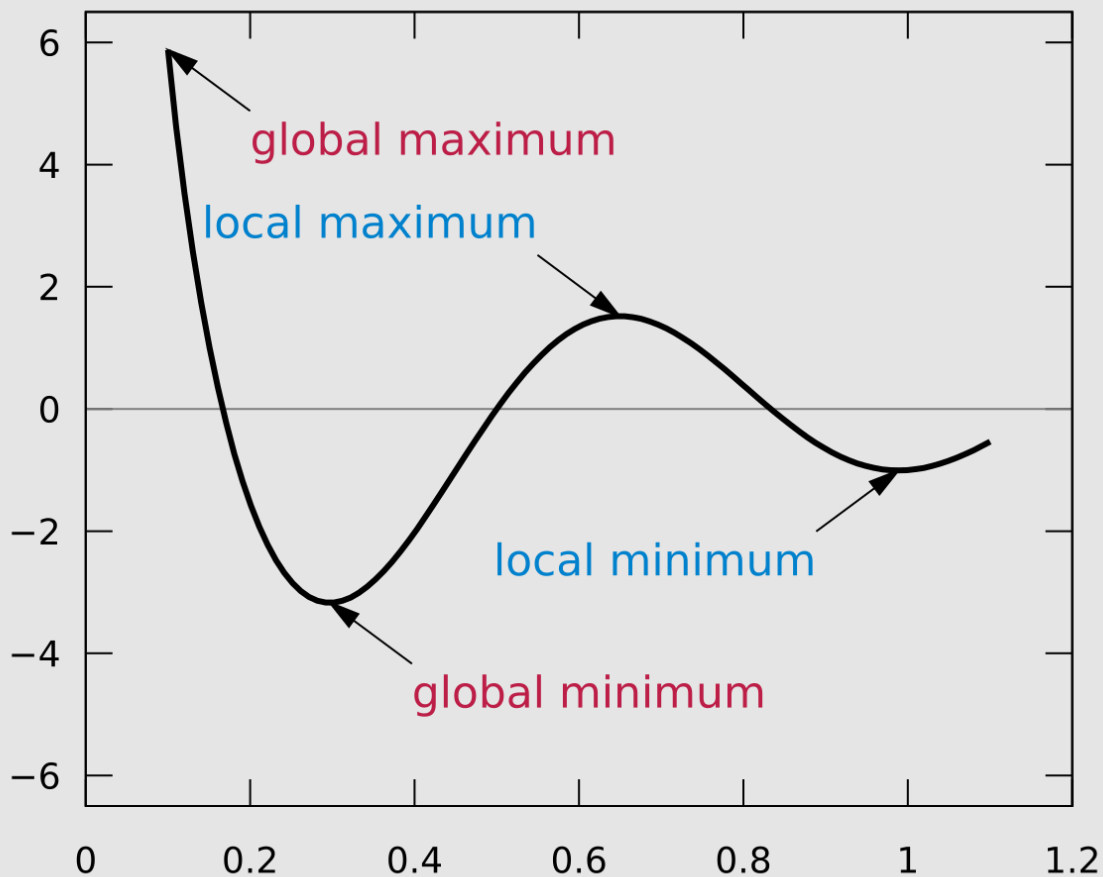
Gradient Descent : 경사 하강법

머신 러닝에서는 오차를 측정할 때 사용되는 **목적 함수**의 출력을 최소화하는 최적의 포인트를 찾기 위해 이 방법을 사용한다.

경사 하강법을 사용하지 않고 위의 그것을 적용해야 한다면 매우 비효율적인 연산 과정을 거쳐야 한다.

Local minimum & Global minimum

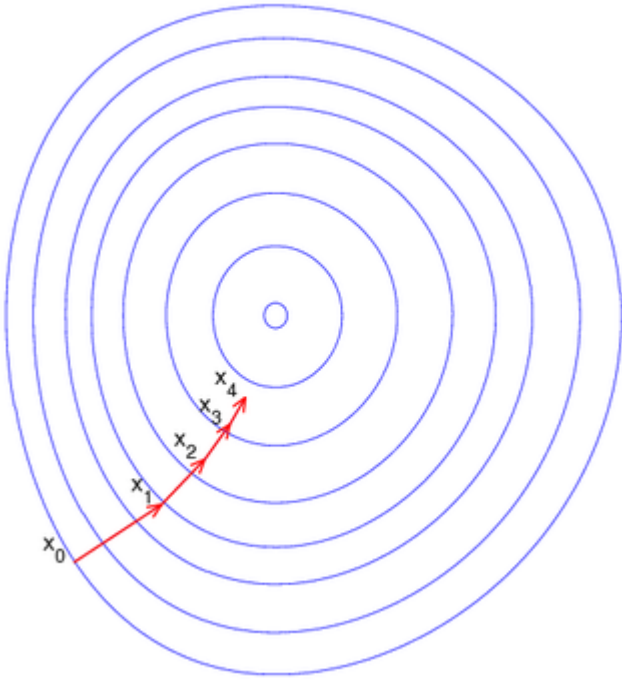
경사 하강법 진행 중 최적의 포인트라고 판단되는 곳에서 도출되는 **minimum** 은 **Local** 과 **Global** 한 두 성질로 구분된다.



성질의 이름으로부터 알 수 있듯 어떤 범위 내에 한정해 **minimum** 을 찾는다고 한다면 그 때 **Local minimum**, 범위가 없이 모든 것들에 한한다면 **Global minimum** 이다.

Minimum 을 찾아가는 경사 하강법의 과정

1. 초기 시작 포인트를 임의로 설정한다.
2. 초기 시작 포인트에서의 **목적 함수** 미분 값을 계산한다.
3. 미분 값을 바탕으로 포인트를 갱신한다.
4. 갱신된 포인트에서의 **목적 함수** 미분 값을 또 계산한다.
5. 3-4 의 과정을, 미분 값이 0\$에 수렴할 때 까지 반복한다.



경사 하강법은 **목적 함수**가 미분 가능한 형태를 띠다면 항상 적용이 가능하며, 또한 **convex; convexity** 성질을 지녔다면 **Local minimum == Global minimum** 이 항상 성립한다.

경사 하강법의 응용

기본적인 경사 하강법 자체는 전체 데이터 내에서 최적의 포인트를 찾아내기 위해 노력한다. 그리고 경사 하강법은 여러 파생 형태가 존재한다. 그 형태를 구분하는 기준은 먼저 전체 데이터를 어떻게 만지며 경사 하강법을 진행하느냐이다.

Stochastic Gradient Descent

전체 데이터를 전부 만지지 않고, 일부 데이터만 골라 그것들에 대한 경사 하강법을 진행하고, 결과를 보며 어떤 것이 최적의 포인트일지 판단한다.

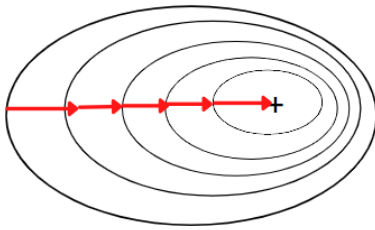
일부 데이터들에 대한 **목적 함수**의 미분 값이 0 에 수렴한다고 하더라도, 경사 하강법을 몇 번 더 진행한다. ... 그래서 기본 경사 하강법보단 많은 연산량이 필요하지만 다루는 데이터 크기 자체가 매우 작기 때문에 속도는 빠른 편이다.

Mini-batch Gradient Descent

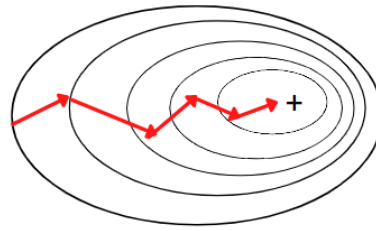
전체 데이터를 몇 조각으로 쪼개, 그 조각 하나하나에 대한 경사 하강법을 진행하는 방법이다. 쪼갠 조각 하나의 단위를 **Mini-batch** 라고 칭하며 이것을 어떻게 설정해 주느냐에 따라 경사 하강법의 성능이 결정된다.

Mini-batch Gradient Descent 는 일반 경사 하강법과 **Stochastic Gradient Descent** 중간에 위치한 방법이다.

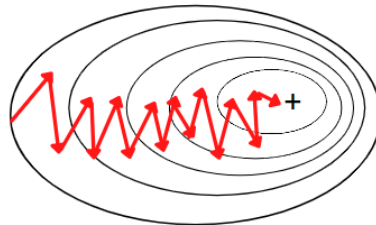
Batch Gradient Descent



Mini-Batch Gradient Descent



Stochastic Gradient Descent



Momentum

물리학에서의 **momentum** 개념을 도입해 경사 하강법 수식에 변형을 주었고, 수식 내 새로운 변수 v 가 **momentum** 의 핵심인데 이는 경사 하강법 알고리즘 진행 시 최적화에 도움을 준다.

기존 경사 하강법이 **Local minimum** 에 계속해서 빠져버리는 문제를 이를 통해 해결할 수 있다.

AdaGrad

Adaptive Gradient Descent 의 약자로, 데이터셋의 각 *feature* 마다 다른 학습률을 부여하는 경사 하강법이다.

각 *feature* 마다 큰 기울기를 가진 것들에는 학습률을 낮게, 작은 기울기를 가진 것들에는 학습률을 높게 부여해 학습을 보다 효율적으로 할 수 있게 한다.

RMSProp

Root Mean Square Propagation 의 약자로, **Adagrad** 의 학습 중단 문제를 해결하고자 지수 이동 평균 개념을 응용해 창안된 경사 하강법이다.

지수 이동 평균은 일반적으로 이산적 순서가 존재하는 시스템 내에서, 순서가 낮은 데이터에는 어떤 작업의 비중을 낮게 두고, 순서가 높은 데이터에는 어떤 작업의 비중을 높게 두는 것이며, 경사 하강법에 이를 적용하면 계속해서 갱신된 기울기 정보를 그렇게 다룬다.

Adam

RMSProp 과 **Momentum** 의 핵심 동작 과정을 합쳐 새롭게 등장한 경사 하강법이며, 지수 이동 평균으로 기울기 정보를 다룸과 동시에 **Momentum** 을 최적화에 반영한다.

자율주행자동차

| @KOCW

| 한국교통대학교 박만복 교수님

| <http://www.kocw.net/home/cview.do?cid=0db40444a9908778>

자율주행자동차 개요

자율주행 기술은 다양한 센서와 기술을 통합하여 구현된다. 이러한 센서와 기술에는 다음과 같은 것들이 존재한다:

- **라이다 센서 (LIDAR)**: 레이저를 사용하여 주변 환경의 거리와 형태를 측정
- **레이더 센서 (Radar)**: 전파를 사용하여 물체의 속도와 거리 정보를 획득
- **카메라**: 시각 정보를 수집하여 물체 인식, 차선 인식 등 다양한 기능을 수행
- **레이저 스캐너**: 주로 근거리의 정밀한 거리 측정에 사용
- **초음파 센서**: 주로 근거리 장애물 감지에 사용
- **통신 장치**: 차량 간 통신(V2V) 및 인프라와의 통신(V2I)을 통해 정보를 교환
- **GPS**: 차량의 위치를 정확하게 파악

이러한 센서들이 수집한 데이터를 실시간으로 처리하기 위해 고성능 컴퓨터가 필요하며, 이 데이터를 기반으로 차량의 주행을 결정하는 다양한 알고리즘이 사용된다.

처리 알고리즘

자율주행 자동차에서는 데이터를 처리하고 노이즈(잡음)를 제거하기 위해 여러 필터링 알고리즘이 사용된다.

Discrete-Time Lowpass Filter

이 필터는 높은 주파수의 노이즈를 제거하여 정보(신호)의 품질을 개선한다. 필터는 다음과 같은 수식으로 표현할 수 있다:

$$H(z) = (1 - z^{-1}) / (1 - \alpha z^{-1})$$

여기서 α 는 0과 1 사이의 상수이며 이 값에 따라 필터의 반응 정도를 조절한다. 이러한 형태의 수식 이외에 차분 방정식 형태로 표현된 수식이 있는데 이전의 출력을 현재의 입력으로 계속해서 사용하는 형태를 띈다.

결국 핵심은 이 필터가 현재 시점과 가까운 정보(신호)에 더 많은 비중을 두어 처리하고, 현재 시점과 먼 정보(신호)는 비중을 적게 두어 처리한다는 것이다.

Kalman Filter

칼만 필터는 어떤 시스템에서 상태를 추정하는데 사용된다. 이는 관찰된 측정값에 존재하는 노이즈를 줄이고, 모르는 정보를 추정한다. 칼만 필터의 업데이트 식은 다음과 같다:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)(y(k) - H \hat{x}(k|k-1))$$

여기서 $K(k)$ 는 칼만 이득, $y(k)$ 는 관측(측정)된 값, H 는 측정 행렬이다.

Bayesian Filter

베이지안(베이즈) 필터는 확률론을 사용하여 시스템의 상태를 추정한다. 관측 데이터와 사전 정보를 결합하여 사후 확률을 계산하는 방법으로, 다음과 같은 베이즈 정리를 기반으로 한다:

$$P(x|z) = P(z|x)P(x) / P(z)$$

여기서 $P(x|z)$ 는 관측(측정)된 확률에 기반한 상태의 확률(사후 확률), $P(z|x)$ 는 조건부 확률(우도), $P(x)$ 는 사전 확률, $P(z)$ 는 관측(측정)된 확률(증거)다.

판단 알고리즘

자율주행 자동차는 다양한 상황에서 최적의 주행 경로와 행동을 결정하기 위해 판단 알고리즘을 사용한다. 대표적으로 다음과 같은 알고리즘들이 사용된다:

- **경로 계획 (Path Planning)**: 목표 지점까지의 최적 경로를 찾는 알고리즘(대표적 방법으로는 A* 알고리즘과 Dijkstra 알고리즘)
- **장애물 회피 (Obstacle Avoidance)**: 주변 환경을 고려하여 안전한 주행 경로를 결정하는 알고리즘

참고로 Dijkstra 알고리즘은 유한한 디지털 시스템에 최적화되어 있으므로, 현실 세계에 조금 더 맞는 결과를 얻으려면 A* 알고리즘을 사용하는 편이 일반적으로 좋다.

제어 알고리즘

제어 알고리즘은 자율주행 자동차가 정확하게 주행할 수 있도록 속도와 방향을 조정한다. 주요 제어 방법은 다음과 같다:

- **PID 제어 (Proportional-Integral-Derivative Control)**: 비례, 적분, 미분 제어를 결합하여 차량의 속도와 방향을 제어
 - **모델 예측 제어 (Model Predictive Control)**: 시스템의 미래 상태를 예측하여 최적의 제어 입력을 결정
-