

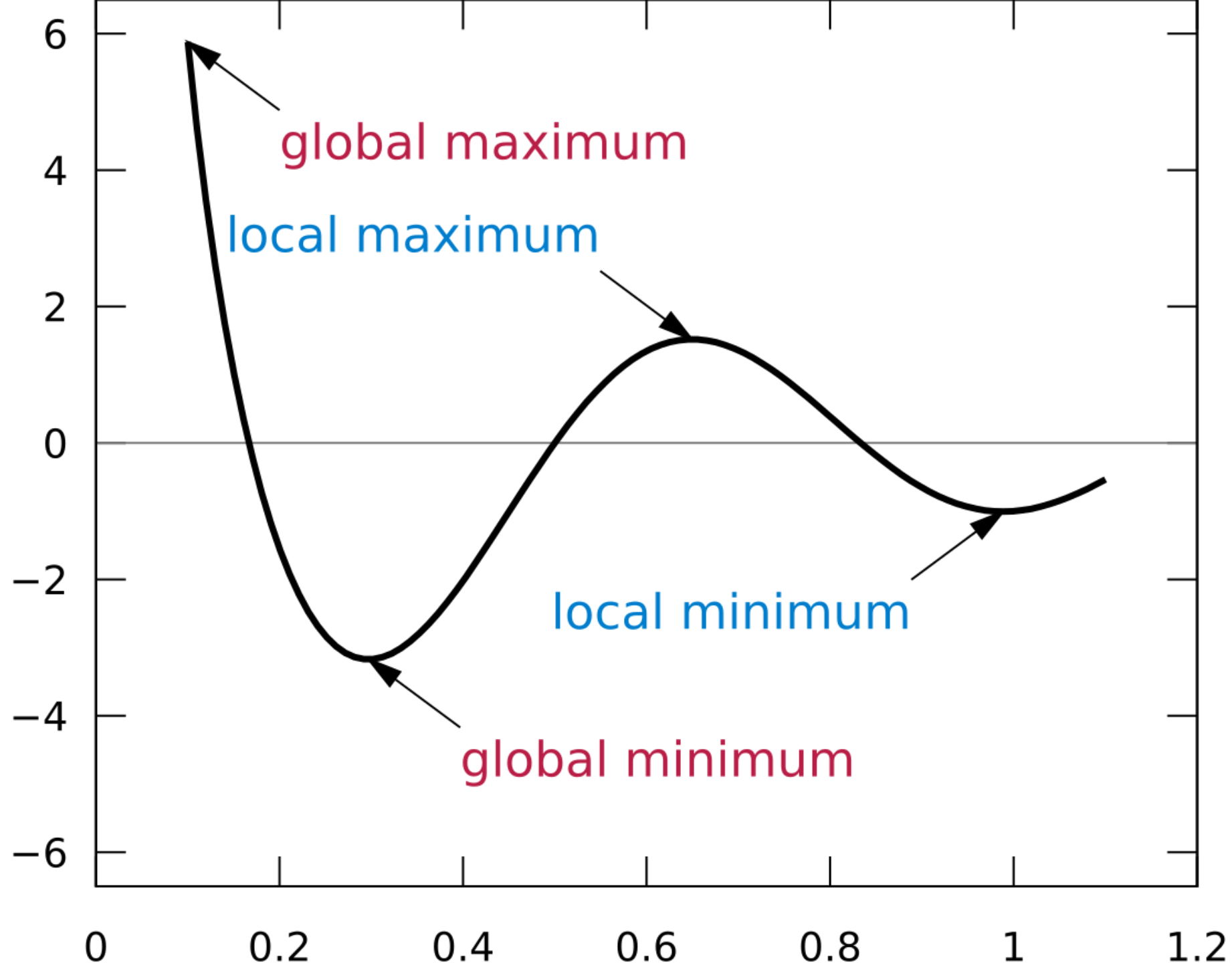
Gradient Descent : 경사 하강법

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

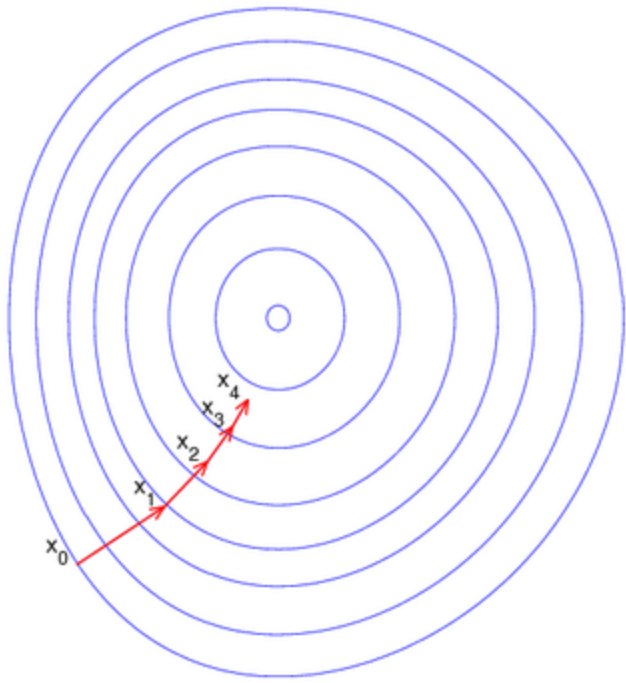
경사 하강법 진행 중 최적의 포인트라고 판단되는 곳에서 도출되는 minimum은 Local 과 Global 한 두 성질로 구분된다.



성질의 이름으로부터 알 수 있듯 어떤 범위 내에 한정해 **minimum** 을 찾고 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 0에 수렴할 때 까지 반복한다.



경사 하강법은 **목적 함수**가 미분 가능한 형태를 띤다면 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.

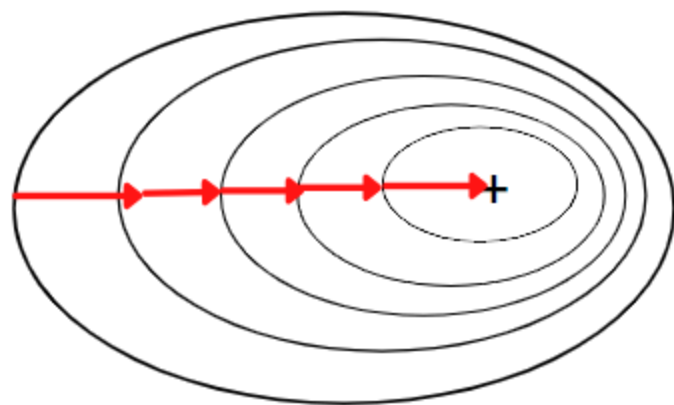
일부 데이터들에 대한 **목적 함수**의 미분 값이 0에 수렴한다고 하더라도, 경사 하강법을 몇 번 더 진행한다. ... 그래서 기본 경사 하강법보단 많은 연산량이 필요하지만 다루는 데이터 크기 자체가 매우 작기 때문에 속도는 빠른 편이다.

Mini-batch Gradient Descent

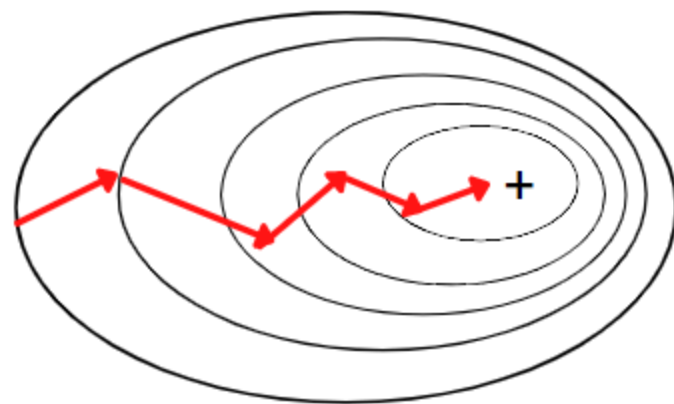
전체 데이터를 몇 조각으로 쪼개, 그 조각 하나하나에 대한 경사 하강법을 진행하는 방법이다. 쪼갠 조각 하나의 단위를 **Mini-batch** 라고 칭하며 이것을 어떻게 설정해 주느냐에 따라 경사 하강법의 성능이 결정된다.

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

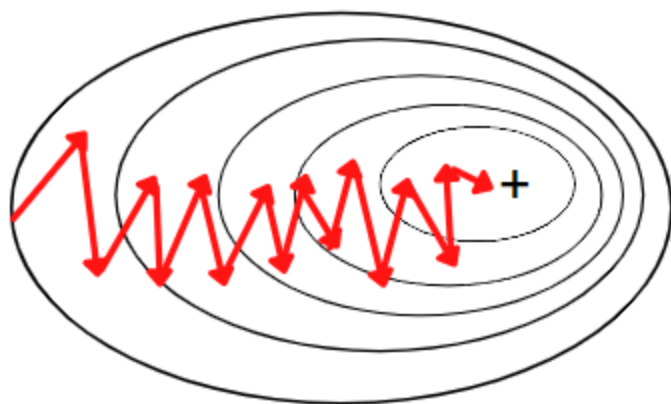
Batch Gradient Descent



Mini-Batch Gradient Descent



Stochastic Gradient Descent



Deep Neural Network

일반적인 머신 러닝에서 발생하는 두 가지 근본적 문제는, *Non-Linear Data handling, Overfitting* 이다.

Non-Linear Data handling

비선형성을 선형성으로 표현할 수 있는 방법은, 수 많은 선형성의 무엇이 모여 전체적으로 봤을 때 그것이 비선형성의 무언가가 되도록 하는 것이다.

Overfitting

준비된 데이터가 충분히 많다면 이 문제를 해결할 수 있다. 다만, 준비된 데이터는 iid condition : 학습용 데이터와 검증용 데이터의 상태 분포가 서로 동일해야 한다. 을 만족해야 한다.

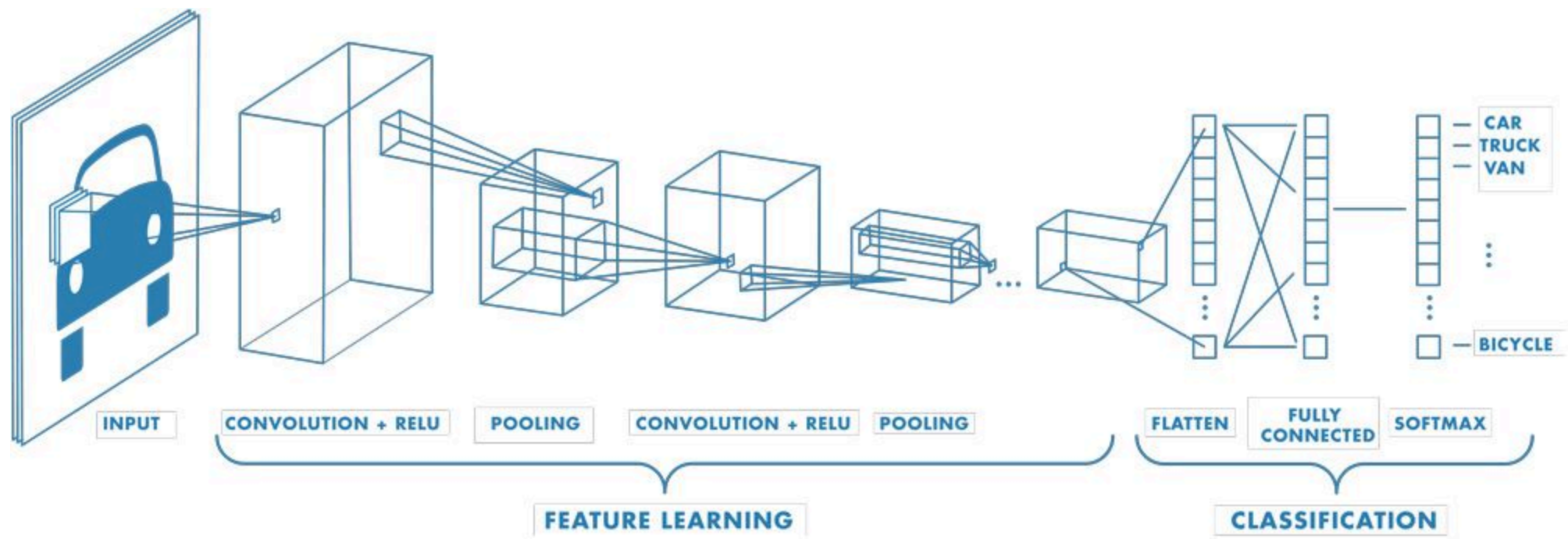
Convolution Neural Netwrok

Deep Neural Network 의 가장 기본적인 형태를 하고 있고, 일반적으로 이미지 데이터 분류에 사용된다.

Convolution Neural Network 의 기본 흐름은 : 이미지 데이터 입력 - Convolution Layer - Fully Connected Layer - result 이다.

Convolution Layer 내부엔 Pooling Layer 라는 것이 또 존재하는데, 이것은 각 Layer 의 결과를 다음 Layer 로 원활히 보내주기 위해 사용된다.

또한 각 Layer 끝에는 Activation Function 이 위치해 있고 이는 비선형성을 표현하기 위해, 또는 다음 Layer 로 어떤 정보를 보낼 때 알맞게 변환하기 위해 사용된다.



요약

- **Pooling Layer** : 입력받은 데이터의 복잡도를 줄여준다.
- **Convolution Layer** : 여러 **Convolution filter** 를 사용해 입력받은 데이터의 변환을 수행한다.
- **Fully Connect Layer** : **Convolution Layer** 에서 얻은 여러 결과를 바탕으로 새로운 결과를 출력한다.
- **Activation Function** : 입력받은 데이터들의 비선형성을 선형성을 가지도록 변환해 주는 역할을 한다. *Network* 의 제일 마지막 **Layer** 에 위치한다.

Gradient Vanishing

Deep Neural Network 의 Layer 층이 많아질수록, 즉 Deep Neural Network 의 깊이가 깊어질수록 잘 발생하는 문제이다.

각 Layer 의 출력은 안정화된 값과 분포를 따르도록 설정되는데, 이 안정적 출력은 *Gaussian Distribution* 을 따른다. 이 때문에 각 출력이 0 근처 값을 가질 것이고, 이 때문에 각 파라미터 또한 0 에 근접한 값을 가진다. 이러한 파라미터의 연쇄로 말미암아 연산되는 전체 Gradient 는 0에 가까워져 Deep Neural Network 의 학습을 원활히 진행할 수 없게 된다.

