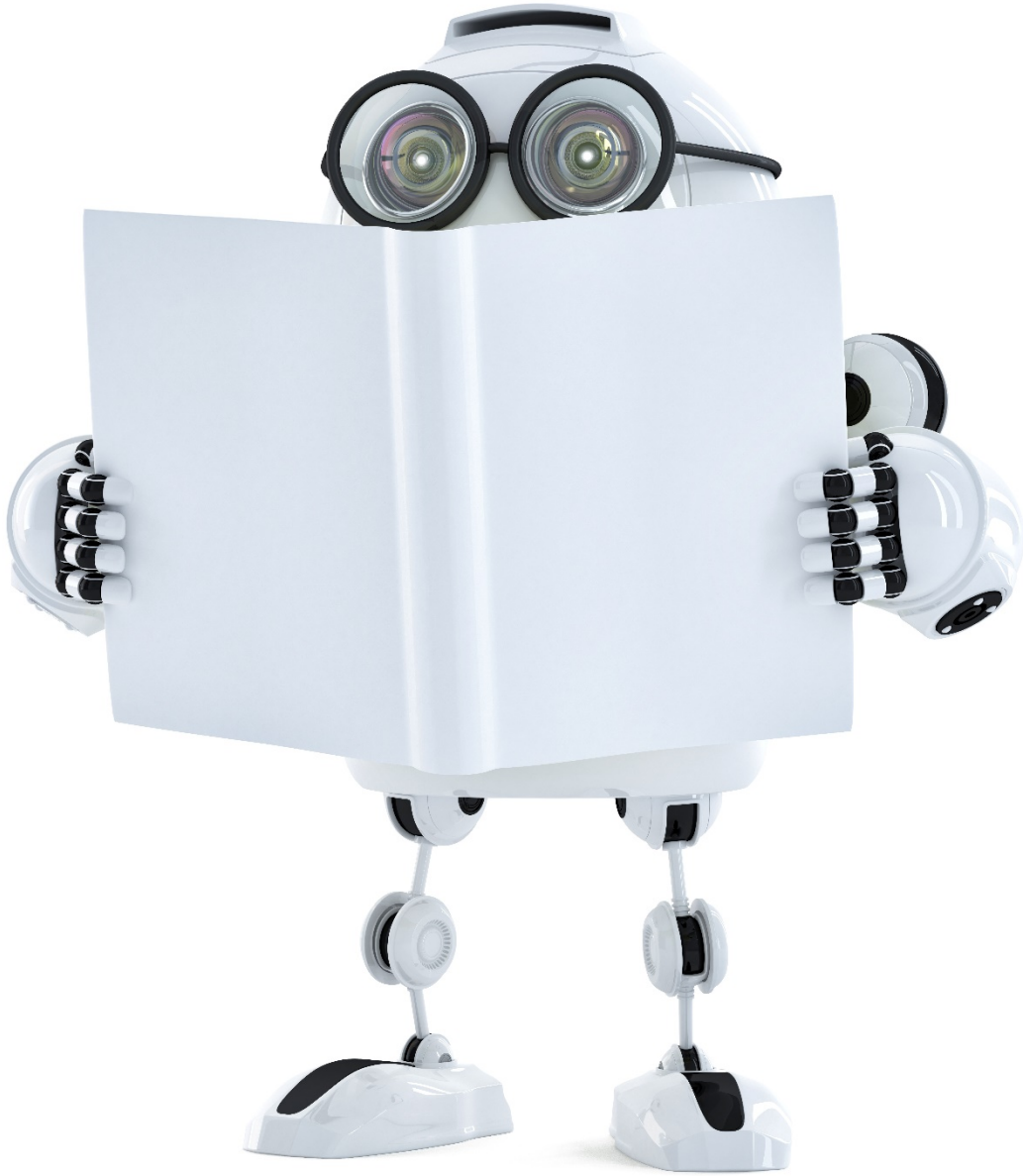
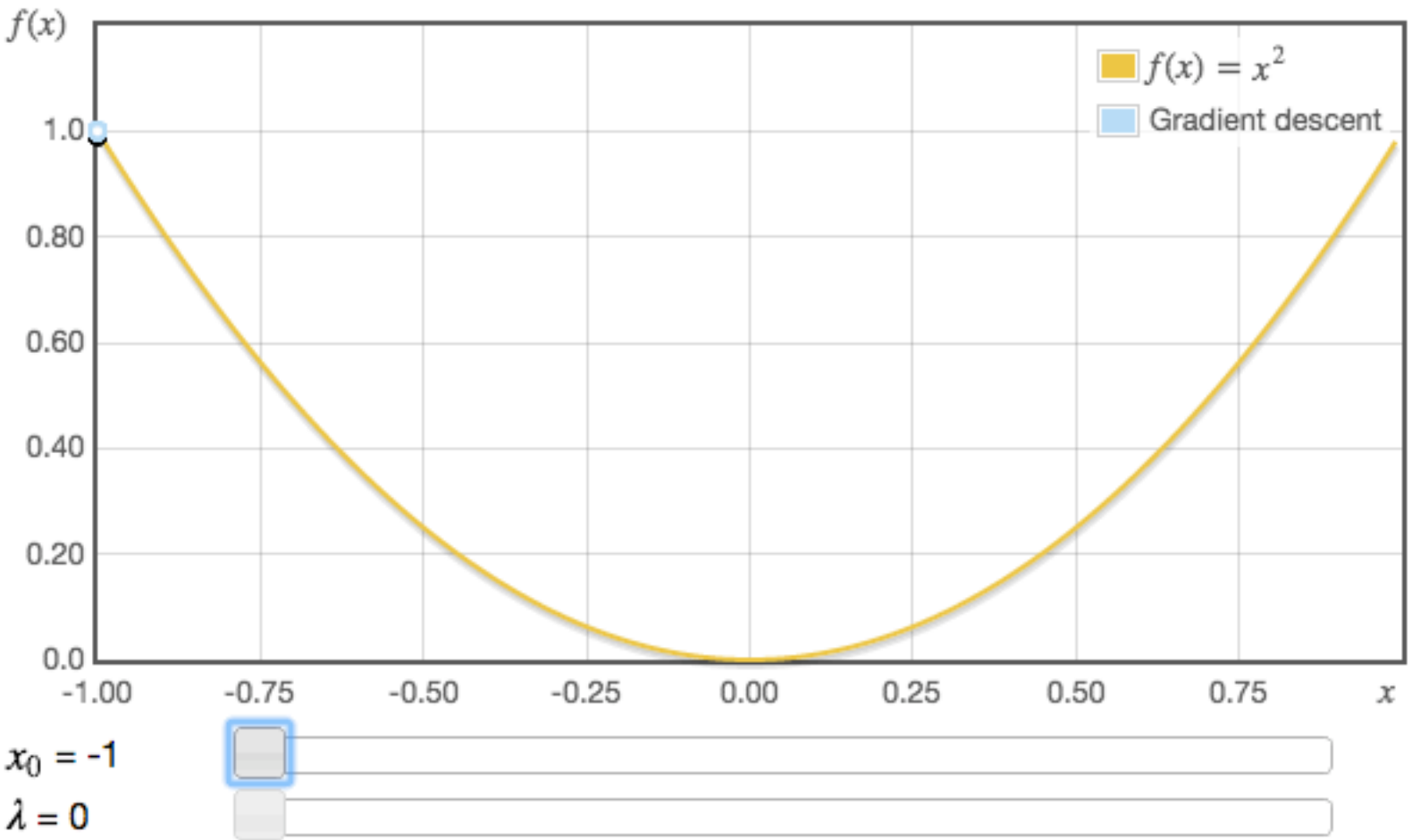
Stochastic Gradient Descent

Linear Regression



Director of TEAMLAB
Sungchul Choi

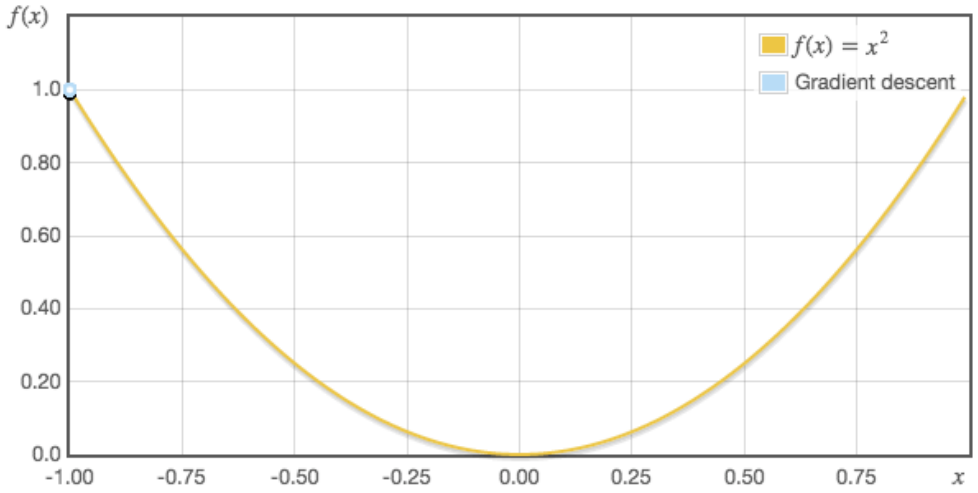
Gradient descent

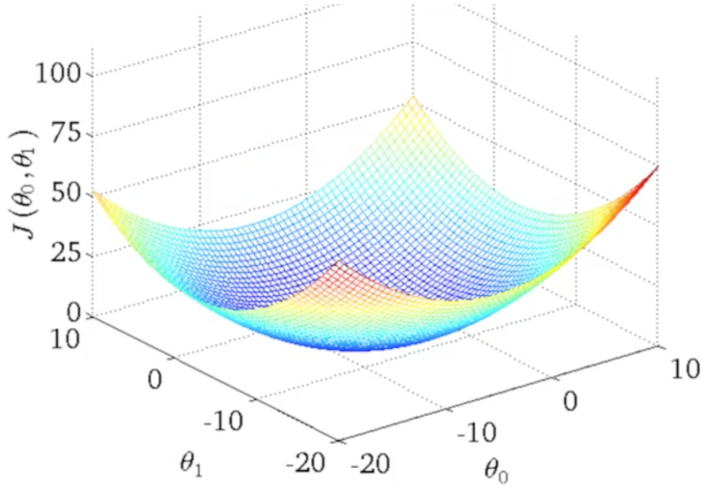


Gradient descent

http://news.jtbc.joins.com/article/article.aspx?news_id=NB10867287

Gradient descent





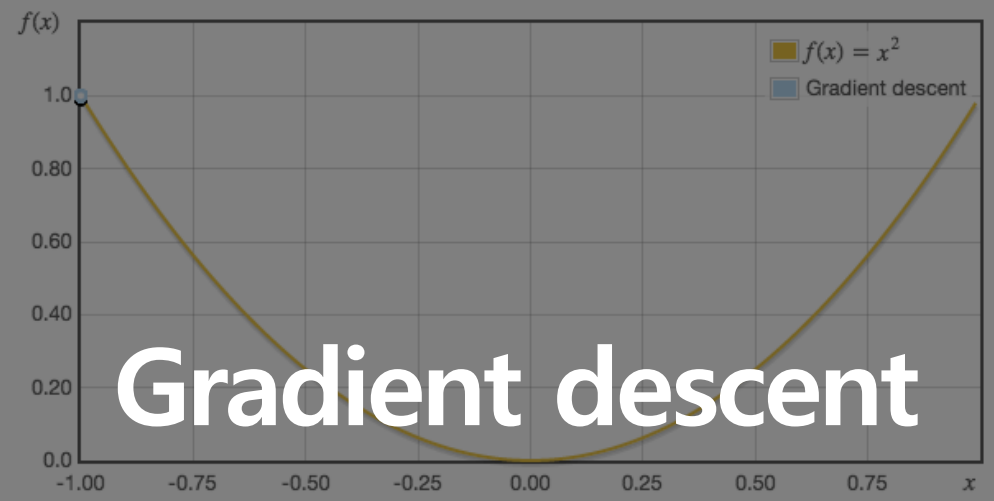
$$x_{new} = x_{old} - \alpha \times (2x_{old})$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\frac{\partial J}{\partial w_0} = \frac{1}{m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)})$$

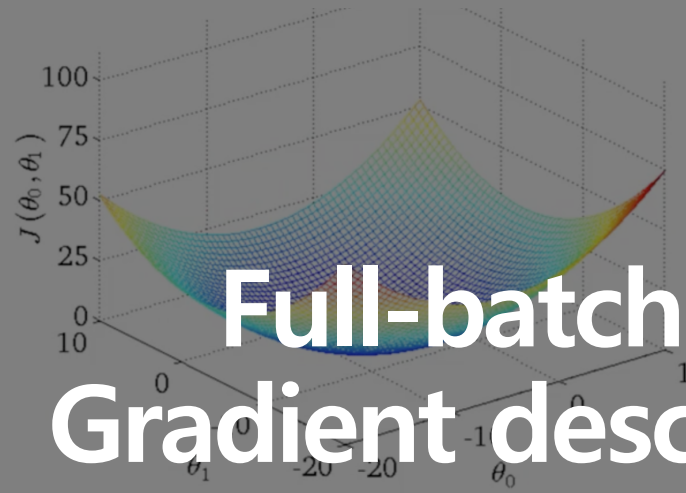
$$\frac{\partial J}{\partial w_1} = \frac{1}{m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)}) x^{(i)}$$

Gradient descent



Gradient descent

$$x_{new} = x_{old} - \alpha \times (2x_{old})$$



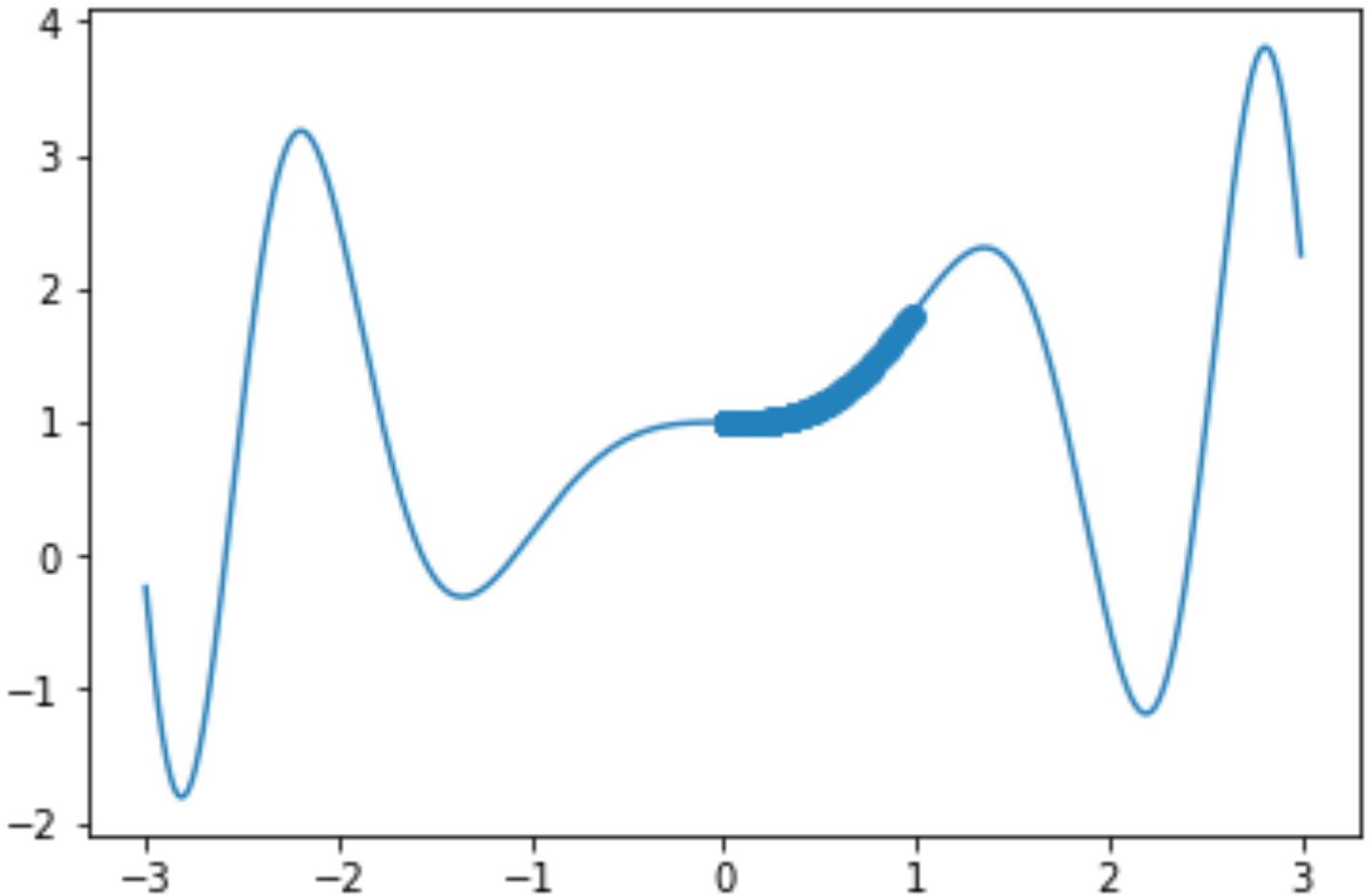
Full-batch Gradient descent

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\frac{\partial J}{\partial w_0} = \frac{1}{m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)})$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)}) x^{(i)}$$

Gradient descent



Full-batch gradient descent

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\frac{\partial J}{\partial w_n} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)}) \cdot x_n$$

Full-batch gradient descent

- GD는 1개의 데이터를 기준으로 미분
- 그러나 일반적으로 GD = (full) batch GD라고 가정
- 모든 데이터 셋으로 학습 $\frac{\partial J}{\partial w_n} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)}) \cdot x_n$

Full-batch gradient descent

- 업데이트 감소 → 계산상 효율적(속도) 가능
- 안정적인 Cost 함수 수렴

- 지역 최적화 가능
- 메모리 문제 (ex – 30억개의 데이터를 한번에?)
- 대규모 dataset → 모델/파라메터 업데이트가 느려짐

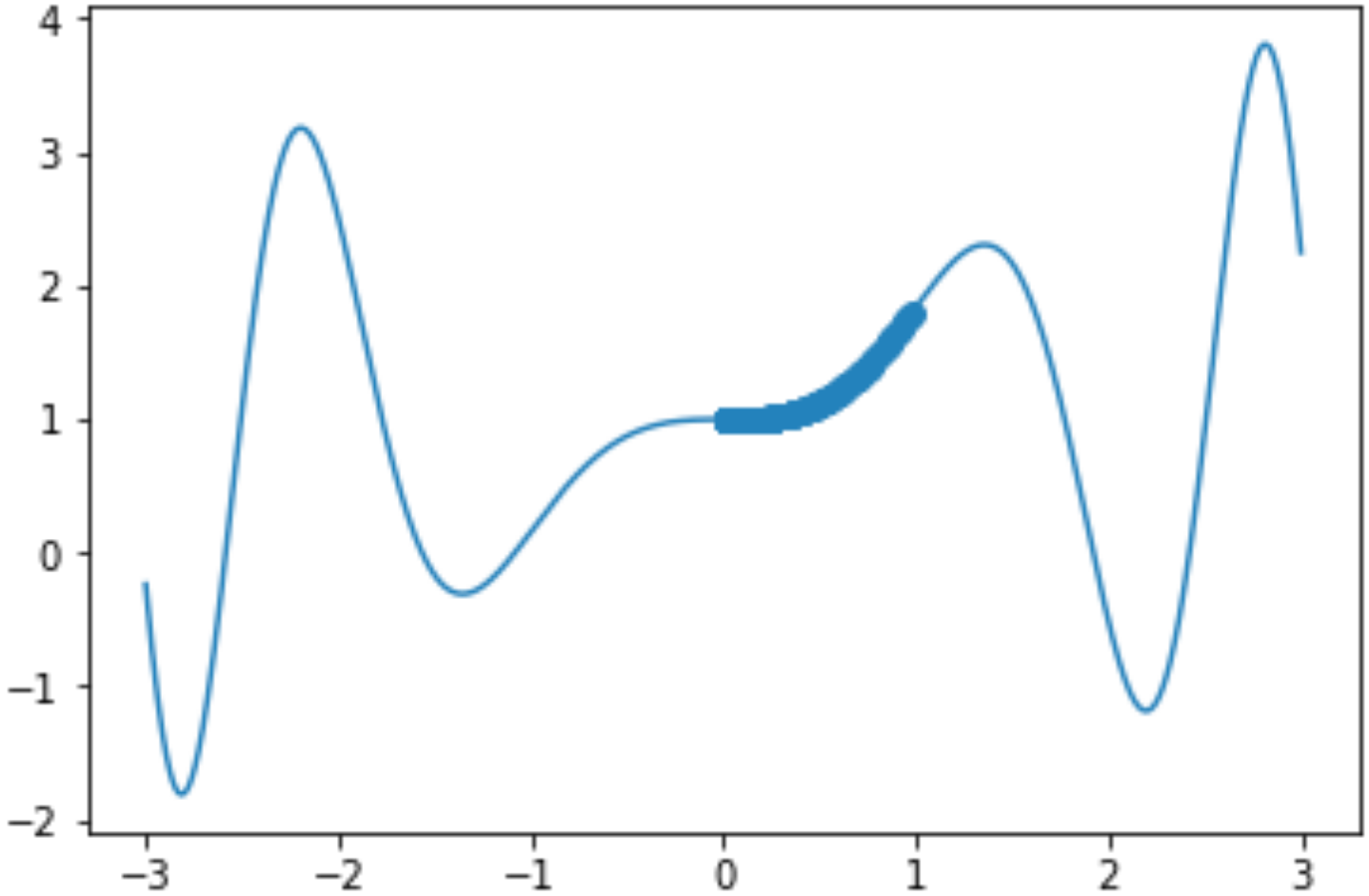
Full-batch gradient descent

- 업데이트 감소 → 계산상 효율적(속도) 가능
- 안정적인 Cost 함수 수렴

- 지역 최적화 가능
- 메모리 문제 (ex – 30억개의 데이터를 한번에?)
- 대규모 dataset → 모델/파라메터 업데이트가 느려짐

Stochastic gradient descent

Stochastic gradient descent



Stochastic gradient descent

- 원래 의미는 dataset에서 random하게 training sample을 뽑은 후 학습할 때 사용함
- Data를 넣기 전에 Shuffle

1: **procedure** SGD
2: `shuffle`(X) ▷ Randomly shuffle data
3: **for** i `in number of` X **do**
4: $\theta_j := \theta_j - \alpha(\hat{y}^{(i)} - y^{(i)})x_j^{(i)}$ ▷ Only one example
5: **end for**
6: **end procedure**

Stochastic gradient descent

- 빈번한 업데이트 모델 성능 및 개선 속도 확인 가능
- 일부 문제에 대해 더 빨리 수렴
- 지역 최적화 회피
- 대용량 데이터시 시간이 오래걸림
- 더 이상 cost가 줄어들지 않는 시점의 발견이 어려움

Mini-batch (stochastic) gradient descent

Mini-batch SGD

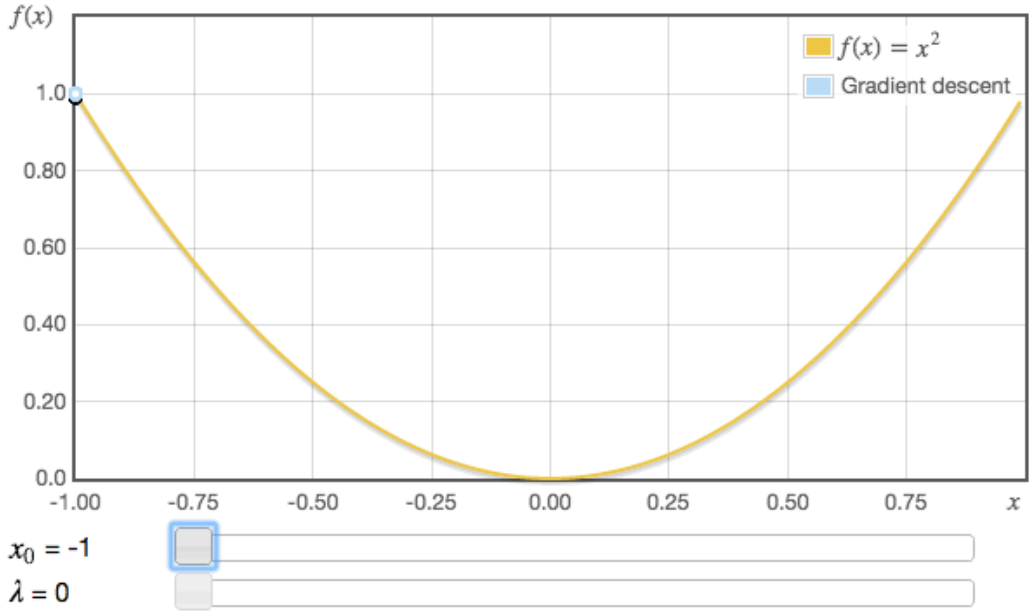
- 한번의 일정량의 데이터를 램덤하게 뽑아서 학습
- SGD와 Batch GD를 혼합한 기법
- 가장 일반적으로 많이 쓰이는 기법

Epoch & Batch-size

- 전체 데이터가 Training 데이터에 들어갈 때 카운팅
- Full-batch를 n번 실행하면 n epoch
- Batch-size 한번에 학습되는 데이터의 개수
- 총 5,120개의 Training data에 512 batch-size면 몇 번 학습을 해야 1 epoch이 되는가?

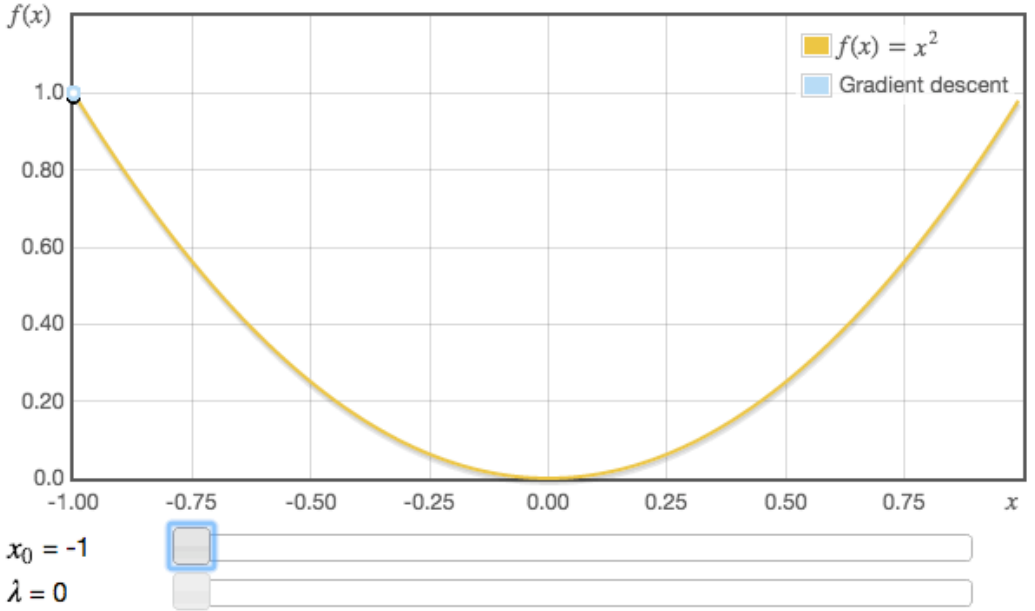
Mini-batch SGD

1: **procedure** MINI-BATCH SGD
2: `shuffle`(X) ▷ Randomly shuffle data
3: `BS` $\leftarrow$ `BATCH SIZE`
4: `NB` $\leftarrow$ `Number of Batches`
5: `NB` $\leftarrow$ `len`(X)`//BS`
6: **for** i `in NB` **do**
7: $\theta_j := \theta_j - \alpha \sum_{k=i\times BS}^{(i+1)*BS} (\hat{y}^{(k)} - y^{(k)}) x_j^{(k)}$ ▷ Batch-sized examples



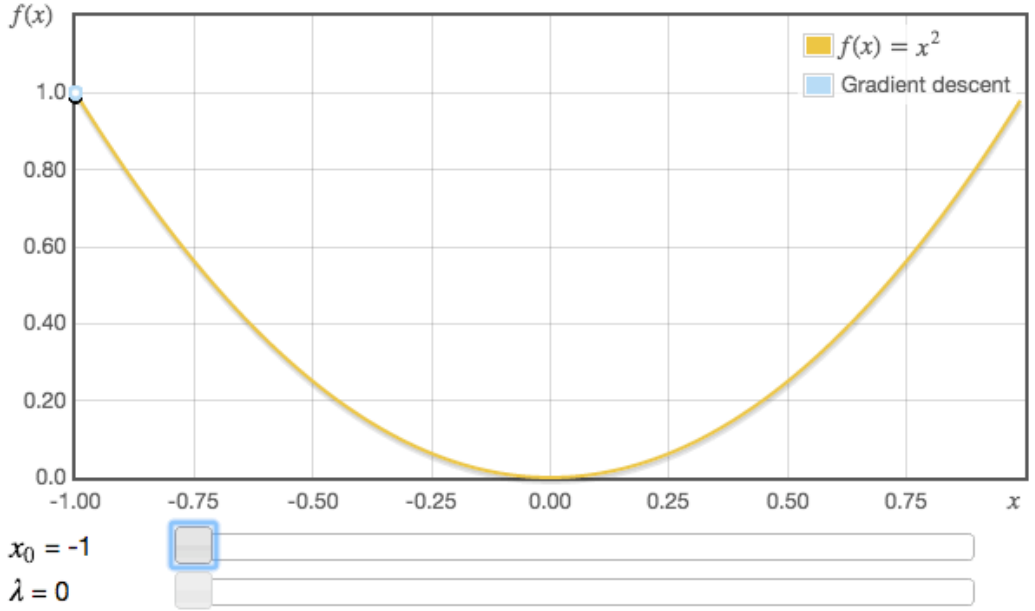
x_0 = -1

λ = 0



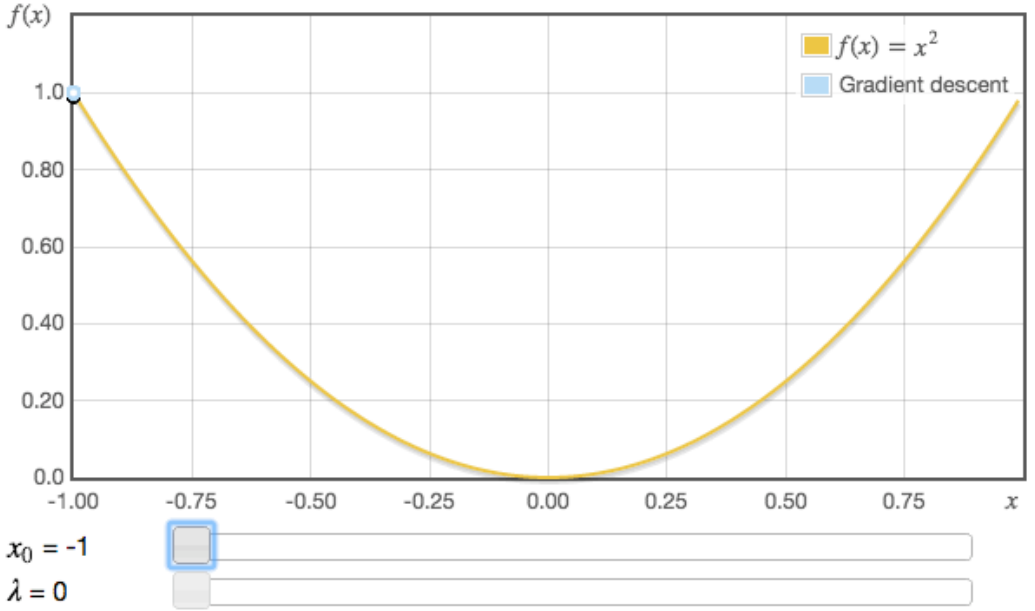
x_0 = -1

λ = 0



x_0 = -1

λ = 0



x_0 = -1

λ = 0

Human knowledge belongs to the world.