# Stochastic gradient descent

# Gradient descent

Optimization problem:

$$L(w) = \sum_{i=1}^{\ell} L(w; x_i, y_i) \to \min_w$$

$w^0$ — initialization

while True:

$$w^t = w^{t-1} - \eta_t \nabla L(w^{t-1})$$

if $\|w^t - w^{t-1}\| < \epsilon$ then break

# Gradient descent

Mean squared error:

$$\nabla L(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} \nabla(w^T x_i - y_i)^2$$

- $\ell$ gradients should be computed on each step
- If the dataset doesn't fit in memory, it should be read from the disk on every GD step

# Stochastic gradient descent

Optimization problem:

$$L(w) = \sum_{i=1}^{\ell} L(w; x_i, y_i) \to \min_w$$

$w^0$ — initialization

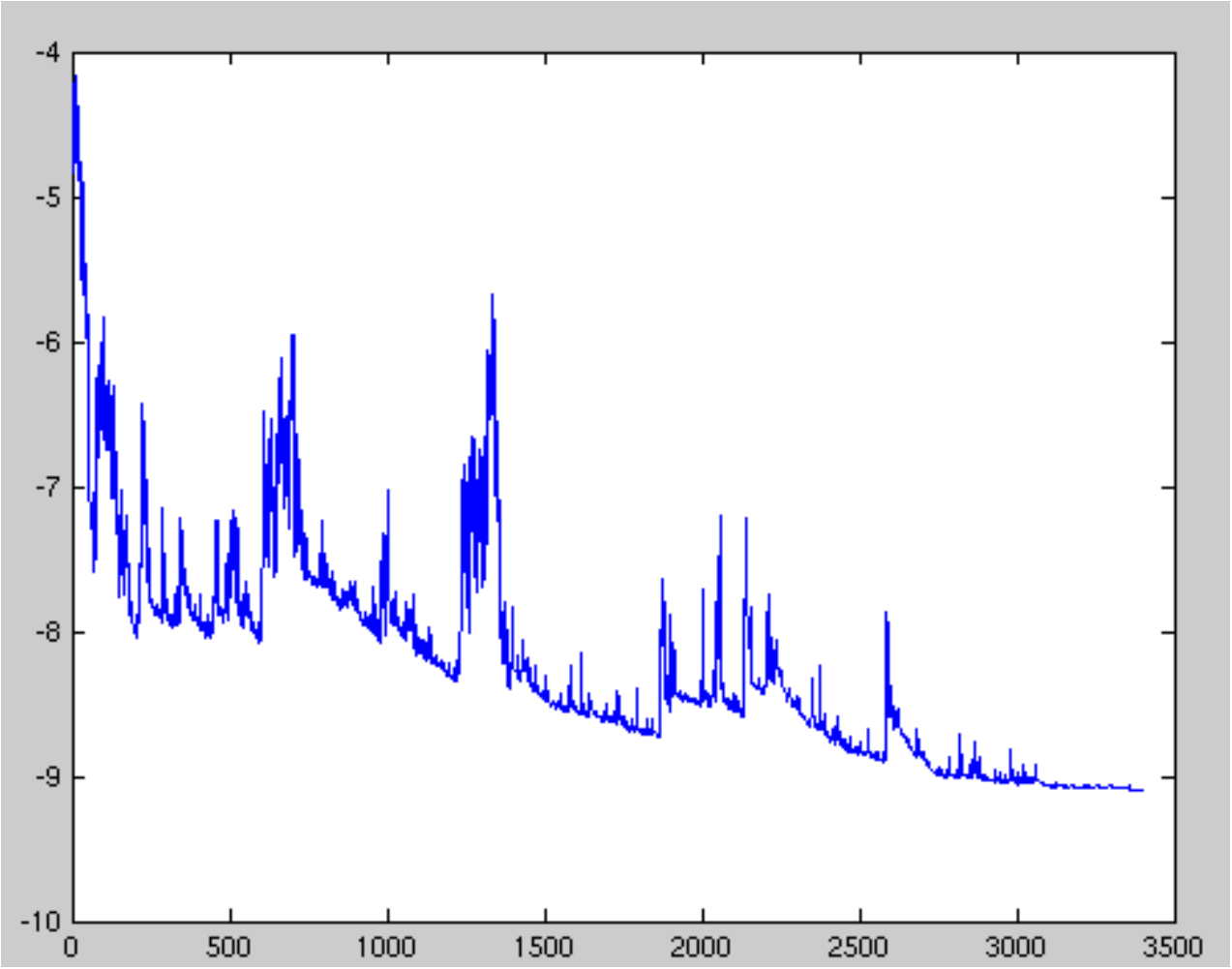
while True:

$i$ = random index between 1 and $\ell$

$w^t = w^{t-1} - \eta_t \nabla L(w^{t-1}; x_i; y_i)$

if $\|w^t - w^{t-1}\| < \epsilon$ then break

# Stochastic gradient descent




Joe pharos, https://en.wikipedia.org/wiki/Stochastic_gradient_descent

# Stochastic gradient descent

- Noisy updates lead to fluctuations
- Needs only one example on each step
- Can be used in online setting
- Learning rate $\eta_t$ should be chosen very carefully

# Mini-batch gradient descent

Optimization problem:

$$L(w) = \sum_{i=1}^{\ell} L(w; x_i, y_i) \to \min_w$$

$w^0$ — initialization

while True:

$i_1, \dots, i_m$ = random indices between 1 and $\ell$

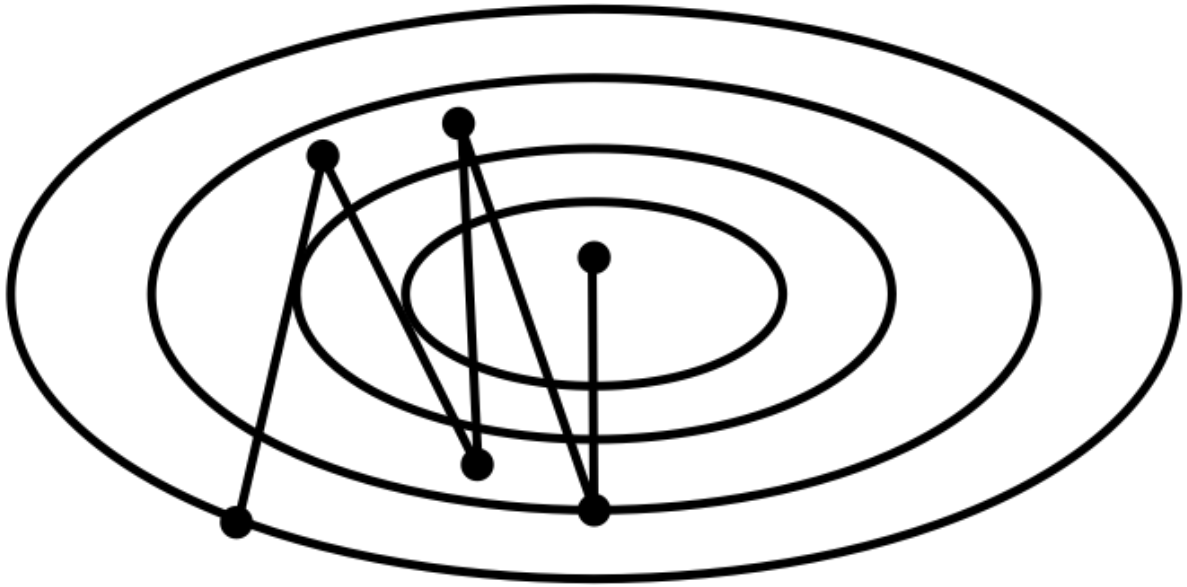
$w^t = w^{t-1} - \eta_t \frac{1}{m} \sum_{j=1}^{m} \nabla L\left(w^{t-1}; x_{i_j}; y_{i_j}\right)$

if $\|w^t - w^{t-1}\| < \epsilon$ then break

# Mini-batch gradient descent

- Still can be used in online setting
- Reduces the variance of gradient approximations
- Learning rate $\eta_t$ should be chosen very carefully

# Difficult function



# Summary

- Gradient descent is infeasible for large training sets
- Stochastic and mini-batch descents use gradient approximations to speed up computations
- Learning rate is quite hard to select
- Methods can be optimized for difficult functions