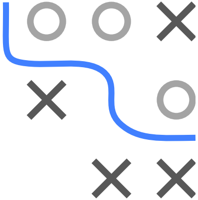
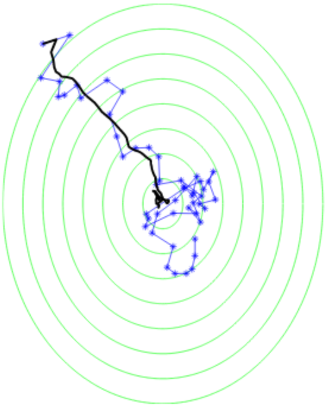
Optimization in Machine Learning



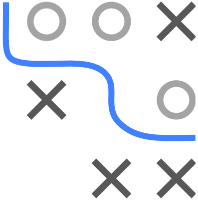
First order methods
SGD



Learning goals

- SGD
- Stochasticity
- Convergence
- Batch size

STOCHASTIC GRADIENT DESCENT



NB: We use g instead of f as objective, bc. f is used as model in ML.

$g : \mathbb{R}^d \rightarrow \mathbb{R}$ objective, g **average over functions**:

$$g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} g_i(\mathbf{x}), \qquad g \text{ and } g_i \text{ smooth}$$

Stochastic gradient descent (SGD) approximates the gradient

$$\nabla_{\mathbf{x}}\, g(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{x}}\, g_i(\mathbf{x}) \quad := \quad \mathbf{d} \quad \text{by}$$

$$\frac{1}{|J|}\sum_{i \in J} \nabla_{\mathbf{x}}\, g_i(\mathbf{x}) \quad := \quad \hat{\mathbf{d}},$$

with random subset $J \subset \{1, 2, ..., n\}$ of gradients called **mini-batch**.
This is done e.g. when computing the true gradient is **expensive**.

SGD IN ML

In ML, we perform ERM:

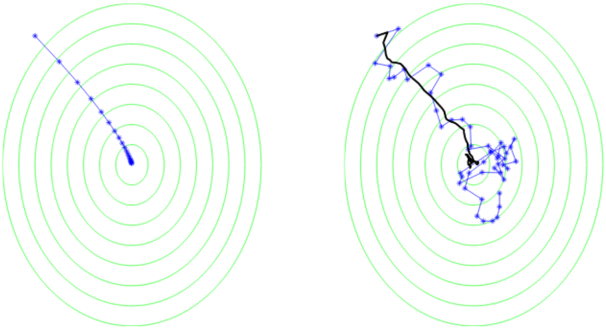
$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)}_{g_i(\boldsymbol{\theta})}$$

- for a data set

$$\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$$

- a loss function $L\left(y, f(\mathbf{x})\right)$, e.g., L2 loss $L\left(y, f(\mathbf{x})\right) = \left(y - f(\mathbf{x})\right)^2$,
- and a model class f, e.g., the linear model $f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right) = \boldsymbol{\theta}^\top \mathbf{x}$.

STOCHASTICITY OF SGD



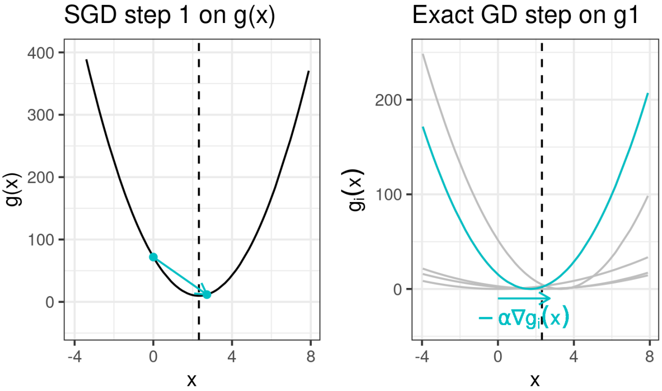
Minimize $g(x_1, x_2) = 1.25(x_1 + 6)^2 + (x_2 - 8)^2$.

Left: GD. **Right:** SGD. Black line shows average value across multiple runs.

(Source: Shalev-Shwartz et al., Understanding Machine Learning, 2014.)

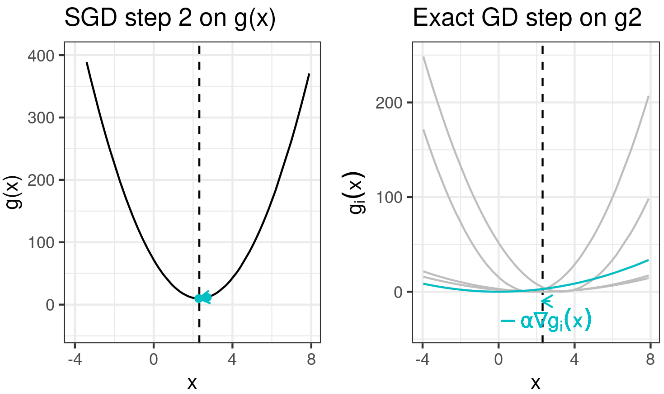
ERRATIC BEHAVIOR OF SGD

Example: $g(\mathbf{x}) = \sum_{i=1}^{5} g_i(\mathbf{x})$, g_i quadratic. Batch size $m = 1$.



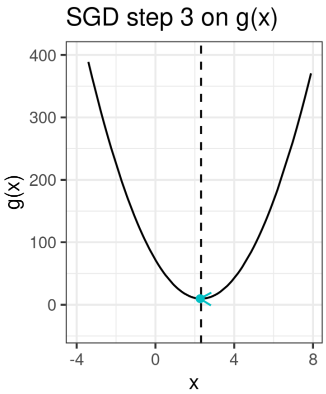
ERRATIC BEHAVIOR OF SGD

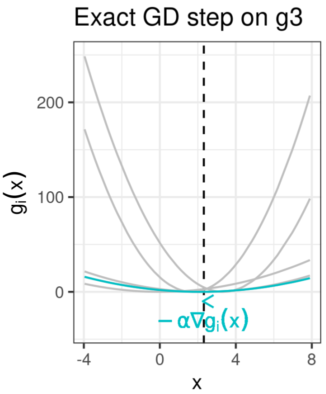
Example: $g(\mathbf{x}) = \sum_{i=1}^{5} g_i(\mathbf{x})$, g_i quadratic. Batch size $m = 1$.

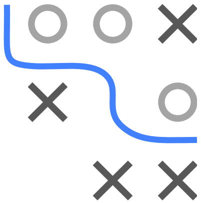


ERRATIC BEHAVIOR OF SGD

Example: $g(\mathbf{x}) = \sum_{i=1}^{5} g_i(\mathbf{x})$, g_i quadratic. Batch size $m = 1$.

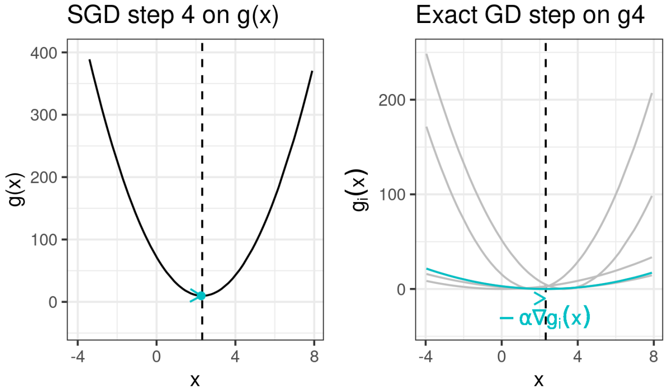




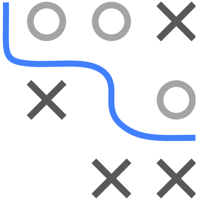


ERRATIC BEHAVIOR OF SGD

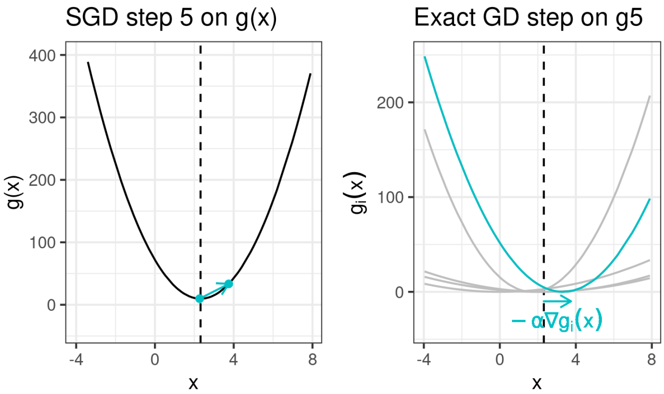
Example: $g(\mathbf{x}) = \sum_{i=1}^{5} g_i(\mathbf{x})$, g_i quadratic. Batch size $m = 1$.



ERRATIC BEHAVIOR OF SGD

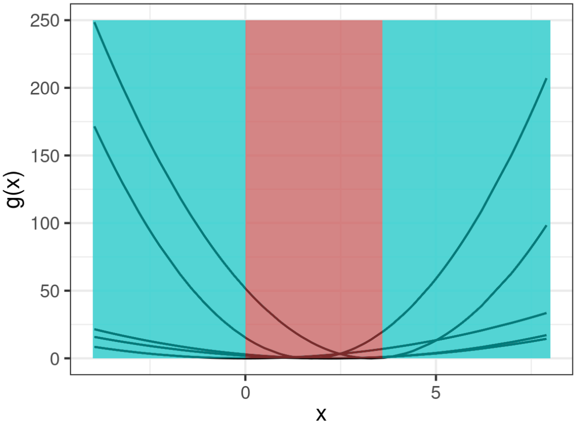


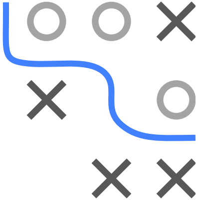
Example: $g(\mathbf{x}) = \sum_{i=1}^{5} g_i(\mathbf{x})$, g_i quadratic. Batch size $m = 1$.



In iteration 5, SGD performs a suboptimal move away from the minimum.

ERRATIC BEHAVIOR OF SGD





Blue area: Each $-\nabla g_i(\mathbf{x})$ points towards minimum.
Red area ("confusion area"): $-\nabla g_i(\mathbf{x})$ might point away from minimum and perform a suboptimal move.

CONVERGENCE OF SGD

As a consequence, SGD has worse convergence properties than GD.

But: Can be controlled via **increasing batches** or **reducing step size**.

The larger the batch size m

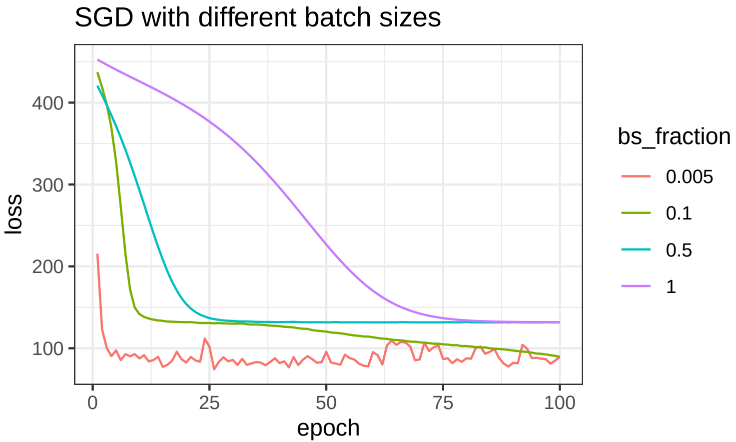
- the better the approximation to $\nabla_{\mathbf{x}} g(\mathbf{x})$
- the lower the variance
- the lower the risk of performing steps in the wrong direction

The smaller the step size α

- the smaller a step in a potentially wrong direction
- the lower the effect of high variance

As maximum batch size is usually limited by computational resources (memory), choosing the step size is crucial.

EFFECT OF BATCH SIZE



SGD for a NN with batch size $\in \{0.5\%, 10\%, 50\%\}$ of the training data.
The higher the batch size, the lower the variance.