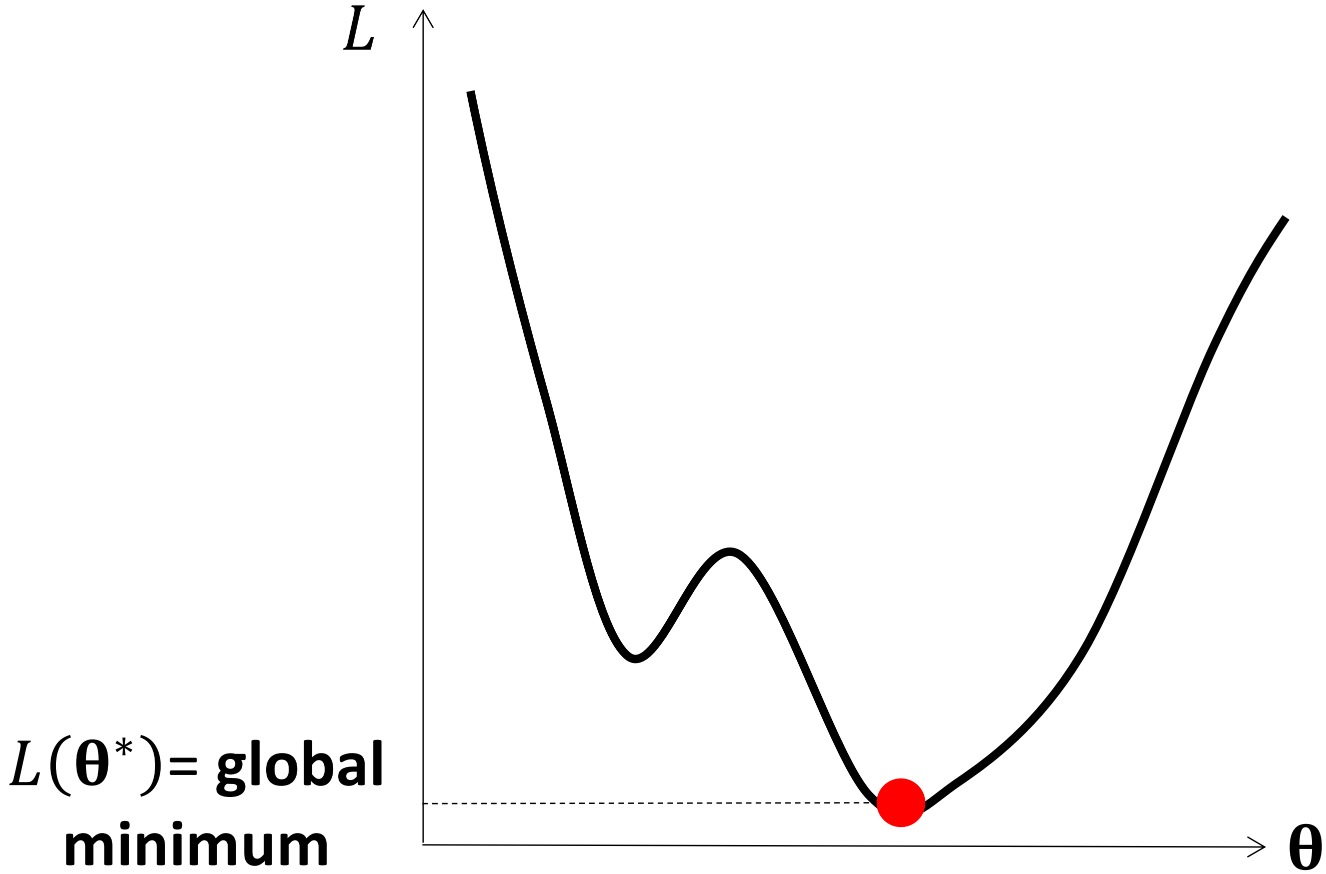
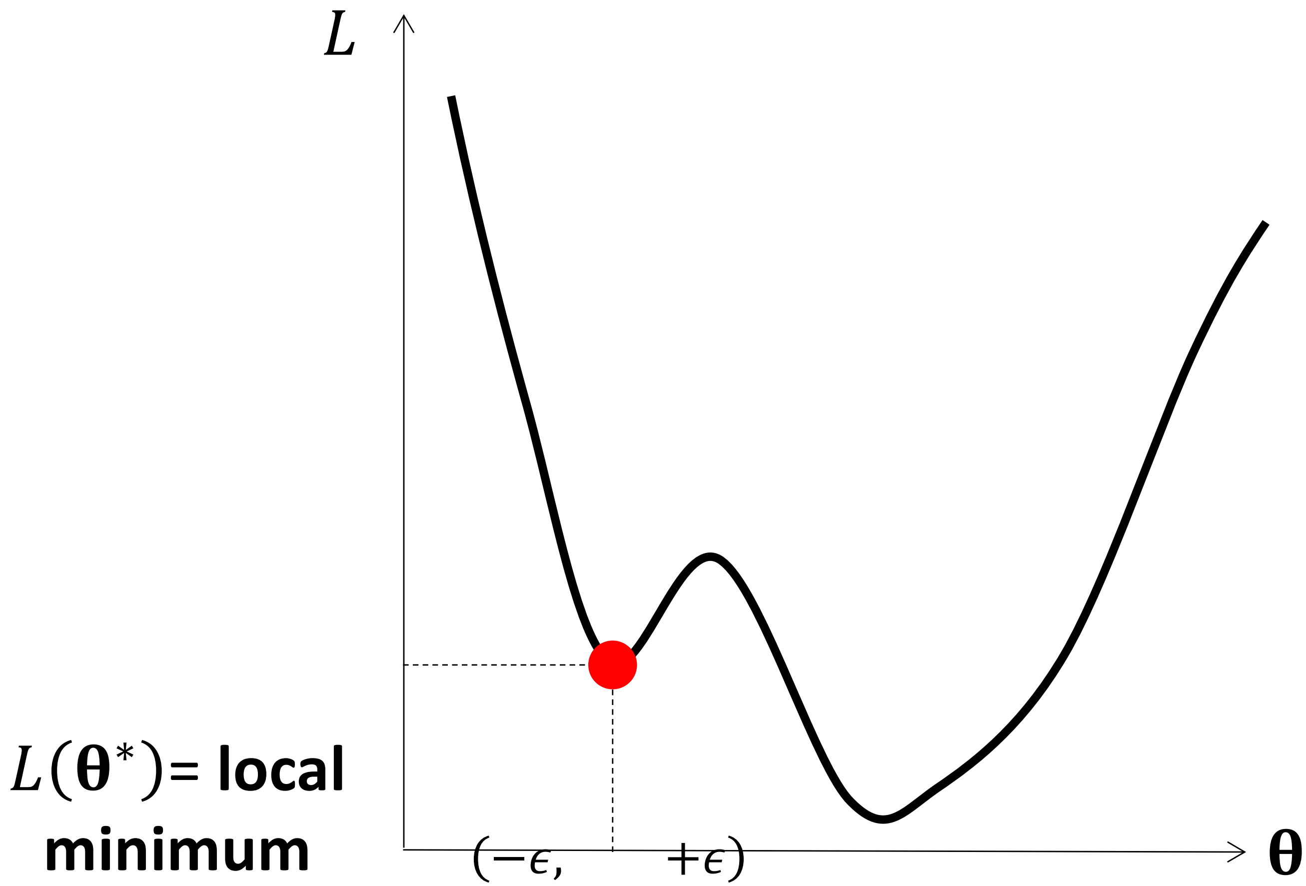
Optimisation

Local vs global minimum



$\boldsymbol{\theta}^*$ = **global minimiser** if
$L(\boldsymbol{\theta}^*) \leq L(\boldsymbol{\theta})$ for every $\boldsymbol{\theta}$

Local vs global minimum



$\boldsymbol{\theta}^*$ = **local minimiser** if $\exists\epsilon > 0$ such that $\boldsymbol{\theta}^*$ is a global minimizer of L in the ball $B_\epsilon(\boldsymbol{\theta}^*)$

Local characterization

First-order Taylor expansion for $L \in \mathcal{C}^1$

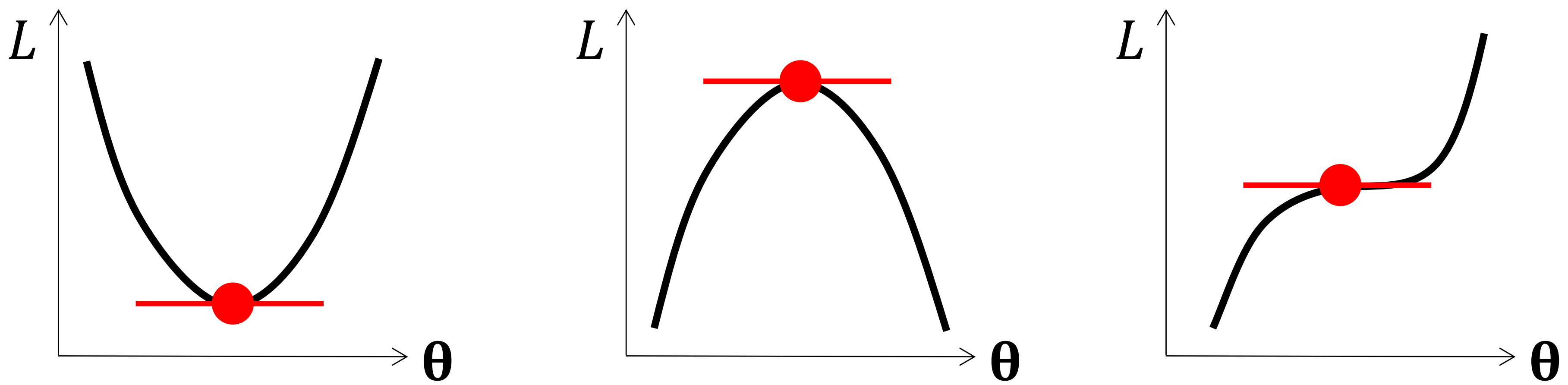
$$L(\boldsymbol{\theta} + \mathbf{d}) = L(\boldsymbol{\theta}) + \nabla L(\boldsymbol{\theta})^T \mathbf{d} + \mathcal{O}(\|\mathbf{d}\|^2)$$

Let $\boldsymbol{\theta}^*$ **local minimiser** of L and $\boldsymbol{\theta} = \boldsymbol{\theta}^* - \alpha \nabla L(\boldsymbol{\theta}^*)$ small $\alpha > 0$

$$\begin{aligned}
0 \leq \frac{1}{\alpha}\Big(L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^*)\Big) &= \frac{1}{\alpha}\Big(L\big(\boldsymbol{\theta}^* - \alpha \nabla L(\boldsymbol{\theta}^*)\big) - L(\boldsymbol{\theta}^*)\Big) \\
&= \frac{1}{\alpha}\Big(-\alpha \nabla L(\boldsymbol{\theta})^{\mathrm{T}} \nabla L(\boldsymbol{\theta}^*) + \mathcal{O}(\|\alpha \nabla L(\boldsymbol{\theta}^*)\|^2)\Big) \\
&= -\|\nabla L(\boldsymbol{\theta}^*)\|^2 + \alpha^2 \mathcal{O}(\|\nabla L(\boldsymbol{\theta}^*)\|^2) \leq 0 \quad \alpha \downarrow 0
\end{aligned}$$

Necessary condition

$\boldsymbol{\theta}^*$ **local minimizer** of $L \;\; \Rightarrow \nabla L(\boldsymbol{\theta}^*) = 0$



Second-order characterization

Taylor expansion for $L \in \mathcal{C}^2$

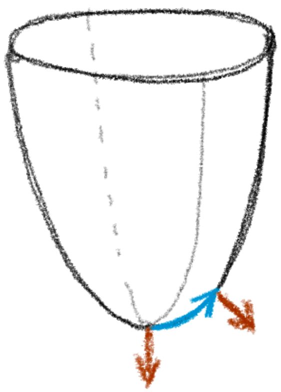
$$L(\boldsymbol{\theta} + \mathbf{d}) = L(\boldsymbol{\theta}) + \nabla L(\boldsymbol{\theta})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 L(\boldsymbol{\theta}) \mathbf{d} + \mathcal{O}(\|\mathbf{d}\|^3)$$

Hessian matrix $\mathbf{H} = \nabla^2 L = \left(\frac{\partial^2 L}{\partial \theta_i \partial \theta_j} \right)$

Curvature in direction $\mathbf{d}$: $\kappa_{\mathbf{d}} \propto \mathbf{d}^T \mathbf{H} \mathbf{d}$

Second-order characterization

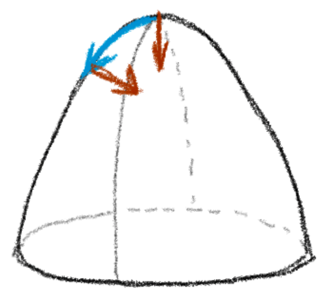
Local minimum



$\kappa_{\mathbf{d}} \geq 0$ for every $\mathbf{d}$

$\mathbf{H} \succcurlyeq 0$ positive semidefinite (non-negative eigenvalues)

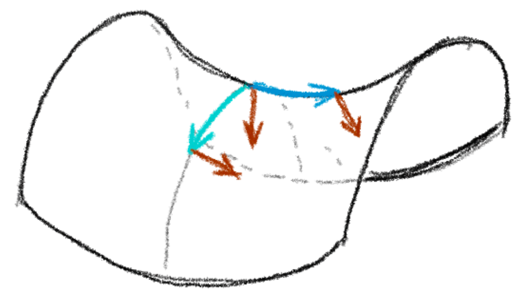
Local maximum



$\kappa_{\mathbf{d}} \leq 0$ for every $\mathbf{d}$

$\mathbf{H} \preccurlyeq 0$ negative semidefinite (non-positive eigenvalues)

Saddle point



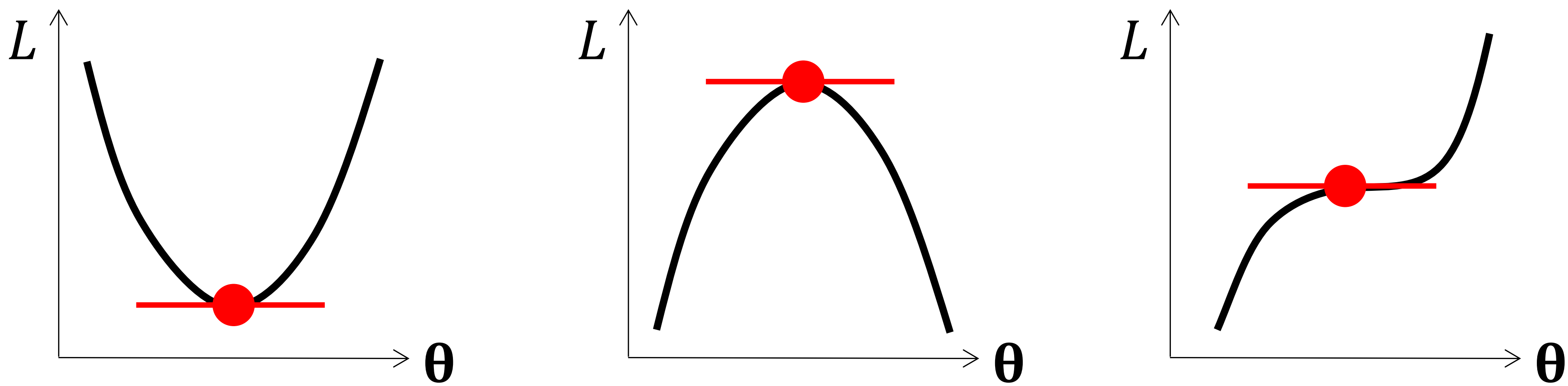
$\kappa_{\mathbf{d}} > 0$ for some $\mathbf{d}$
$\kappa_{\mathbf{d}} < 0$ for some other

$\mathbf{H}$ has both positive and negative eigenvalues

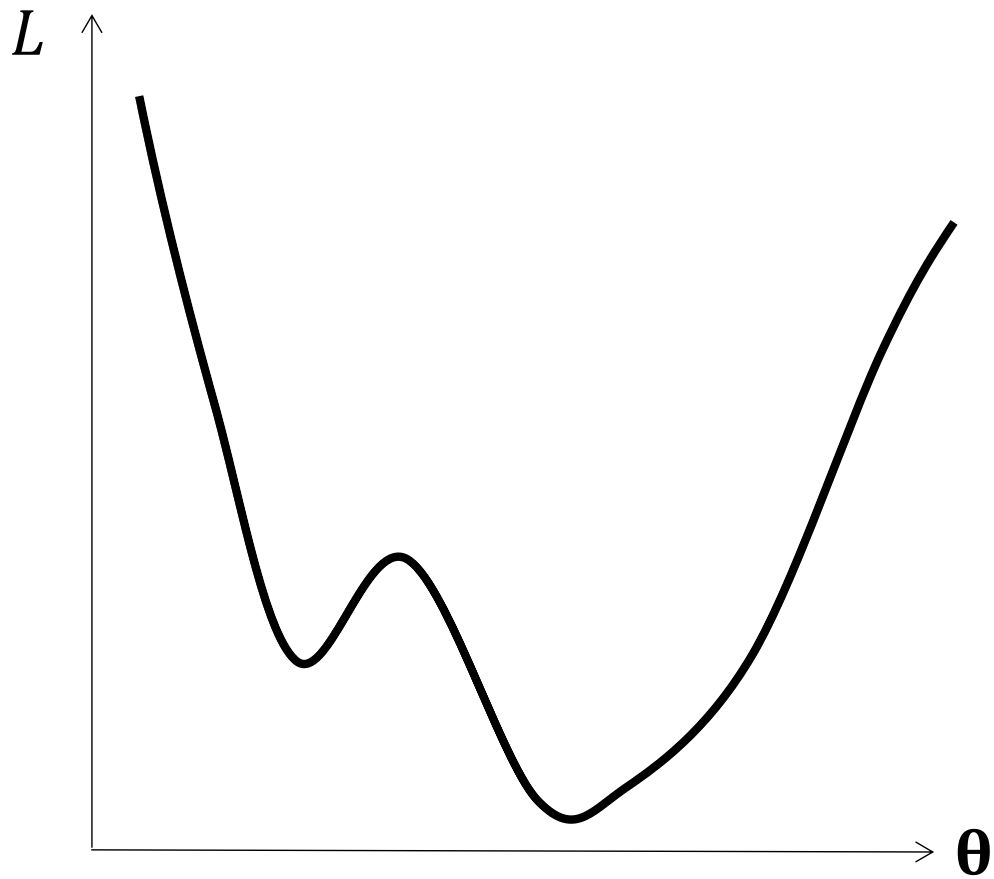
Sufficient condition

$\boldsymbol{\theta}^*$ **local minimizer** of L iff

$\nabla L(\boldsymbol{\theta}^*) = 0$ and $\nabla^2 L(\boldsymbol{\theta}^*) \succcurlyeq 0$



Local vs global minimum

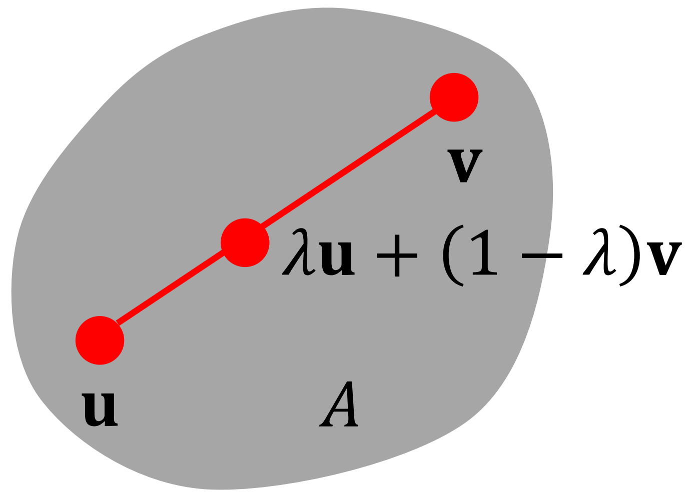


Convexity

Convex combination of $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$

$$\lambda\mathbf{u} + (1-\lambda)\mathbf{v} \text{ with } \lambda \in [0,1]$$

= **line segment** connecting $\mathbf{u}$ and $\mathbf{v}$



$A \subseteq \mathbb{R}^n$ is **convex set** if closed under convex combinations

$$\lambda\mathbf{u} + (1-\lambda)\mathbf{v} \in A$$

$$\forall \mathbf{u}, \mathbf{v} \in A \text{ and } \lambda \in [0,1]$$

Convexity

Convex combination of $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$

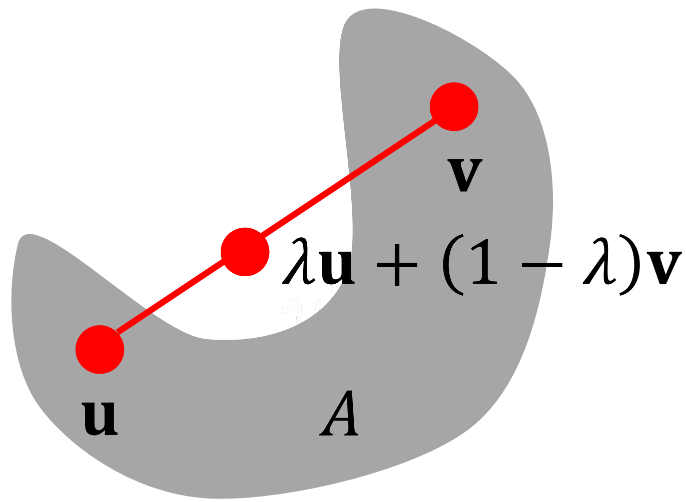
$$\lambda \mathbf{u} + (1-\lambda)\mathbf{v} \text{ with } \lambda \in [0,1]$$

= **line segment** connecting $\mathbf{u}$ and $\mathbf{v}$

$A \subseteq \mathbb{R}^n$ is **convex set** if closed under convex combinations

$$\lambda \mathbf{u} + (1-\lambda)\mathbf{v} \in A$$

$$\forall \mathbf{u}, \mathbf{v} \in A \text{ and } \lambda \in [0,1]$$



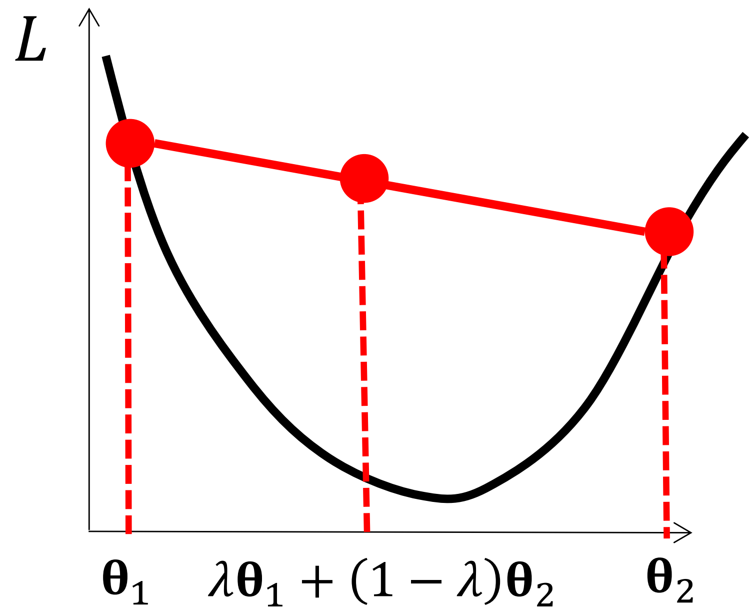
Non-convex

Convex functions

L is a **convex function** iff

$$L(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2) \\ \leq \lambda L(\boldsymbol{\theta}_1) + (1-\lambda)L(\boldsymbol{\theta}_2)$$

for every $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ and $\lambda \in [0,1]$



- graph always below a chord

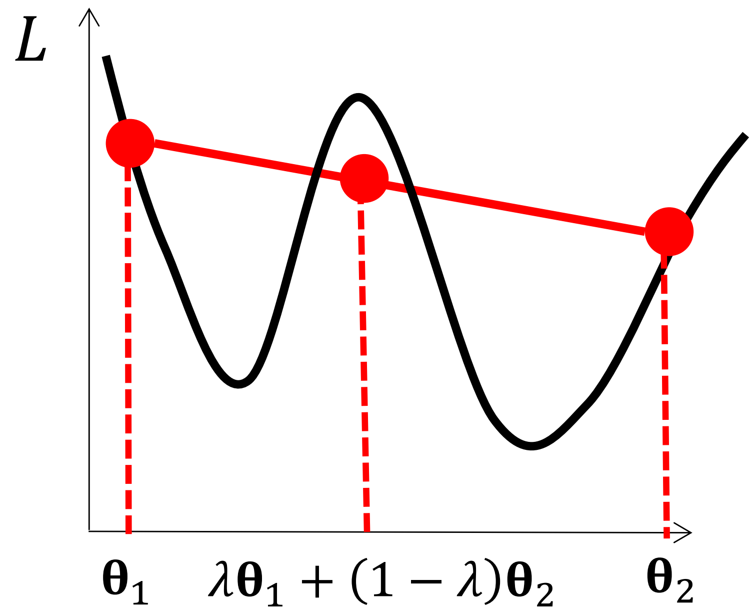
Convex functions

L is a **convex function** iff

$$L(\lambda \boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2) \leq \lambda L(\boldsymbol{\theta}_1) + (1-\lambda) L(\boldsymbol{\theta}_2)$$

for every $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ and $\lambda \in [0,1]$

- graph always below a chord



Non-convex

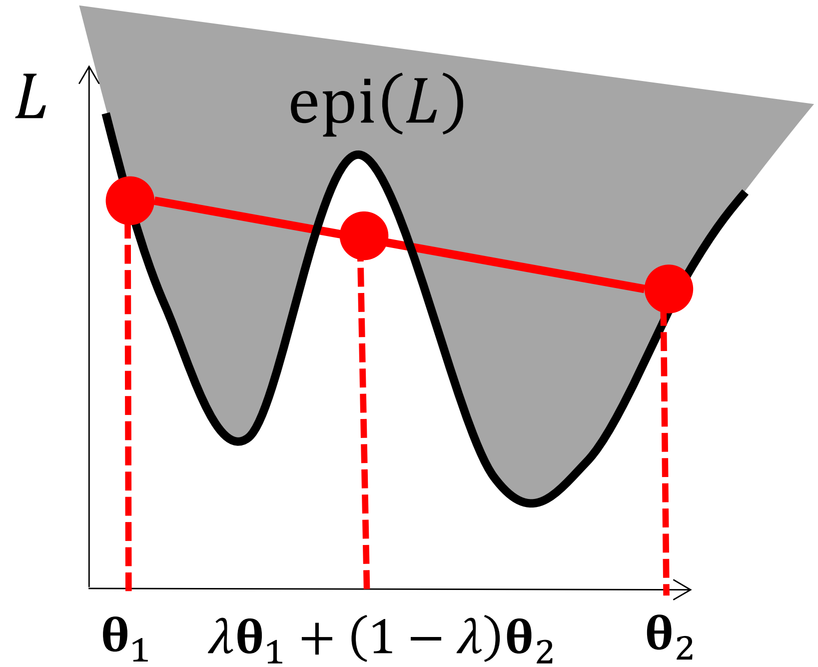
Convex functions

L is a **convex function** iff

$$L(\lambda\boldsymbol{\theta}_1 + (1-\lambda)\boldsymbol{\theta}_2) \\ \leq \lambda L(\boldsymbol{\theta}_1) + (1-\lambda)L(\boldsymbol{\theta}_2)$$

for every $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ and $\lambda \in [0,1]$

- graph always below a chord
- **epigraph** $\mathrm{epi}(L)$ is a convex set



Non-convex

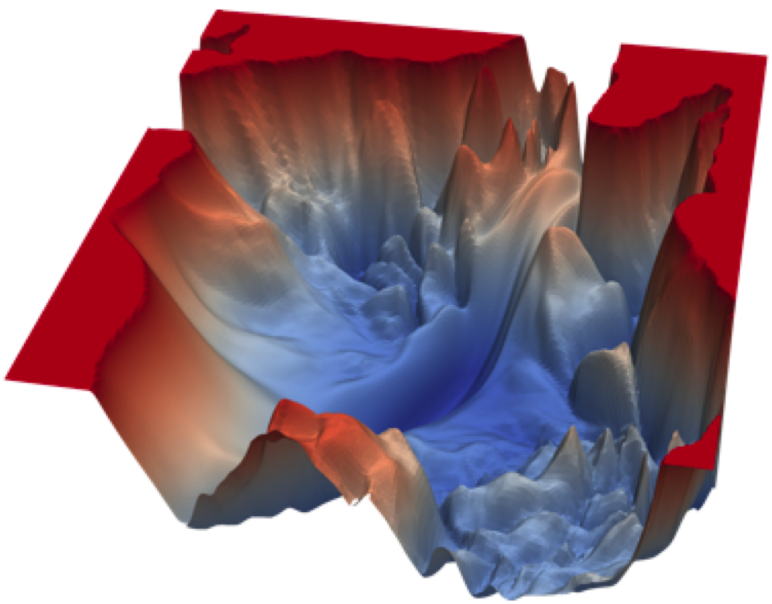
Global optimality

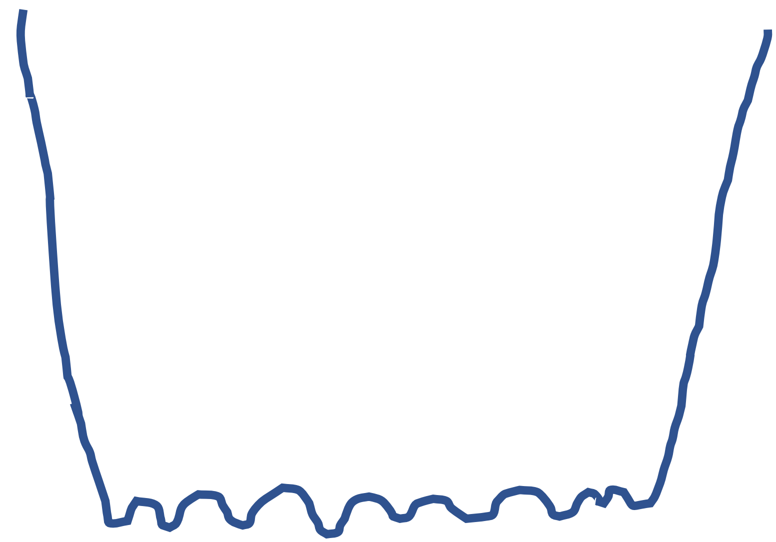
Let $\boldsymbol{\theta}^*$ be a **local minimizer** of a convex function L.
Then $\boldsymbol{\theta}^*$ is also the **global minimizer** of L

Convex vs Non-convex functions

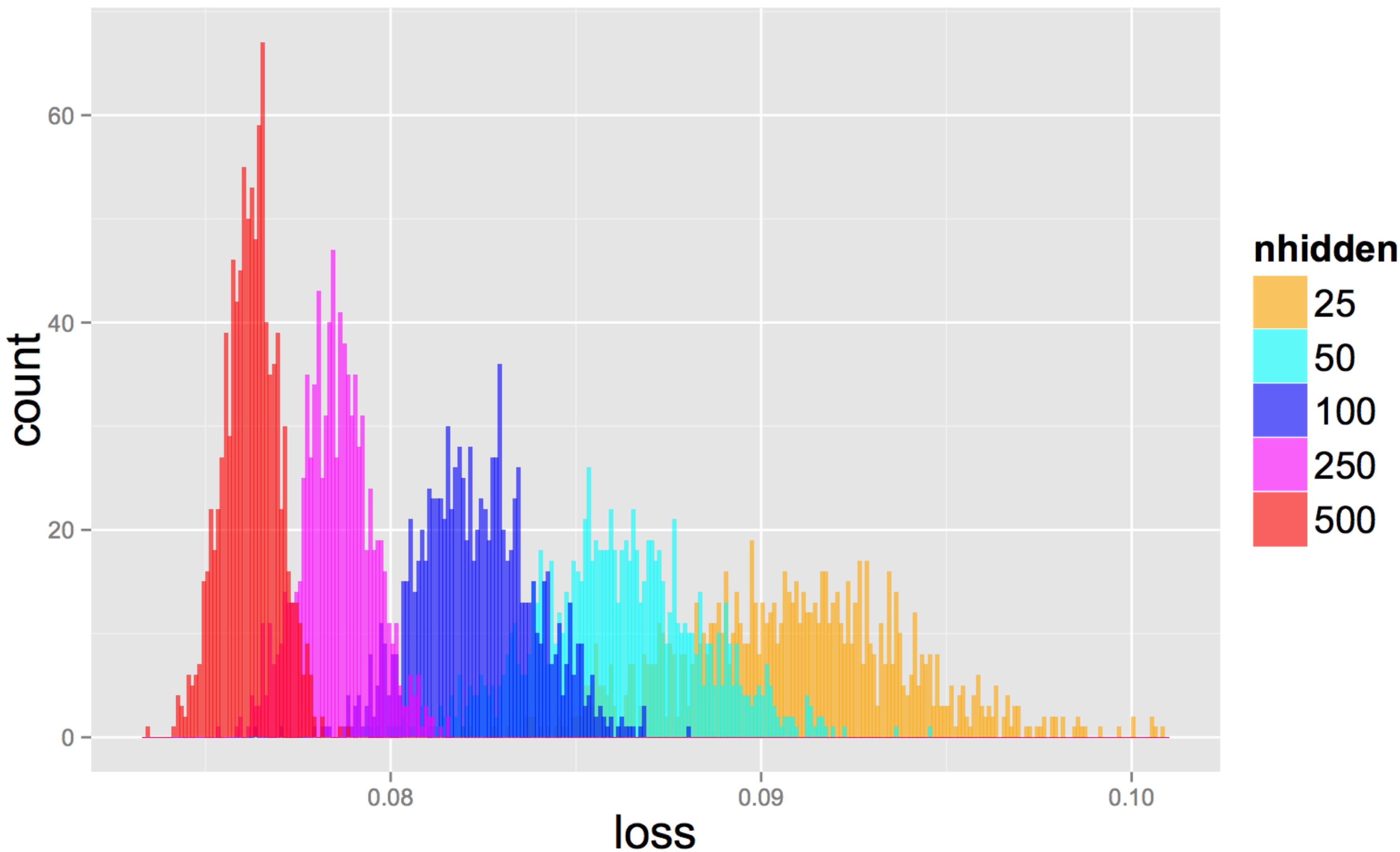
Convex	Non-convex functions
• No negative curvature	• Possibly negative curvature
• Local min = global min	• Possibly local minima that are not global
• Global minimizer found by **descent algorithms**	• Nearly impossibly to guarantee global optimality

Deep learning is non-convex





Deep learning is non-convex



Choromanska et al. 2015

Descent method: general recipe

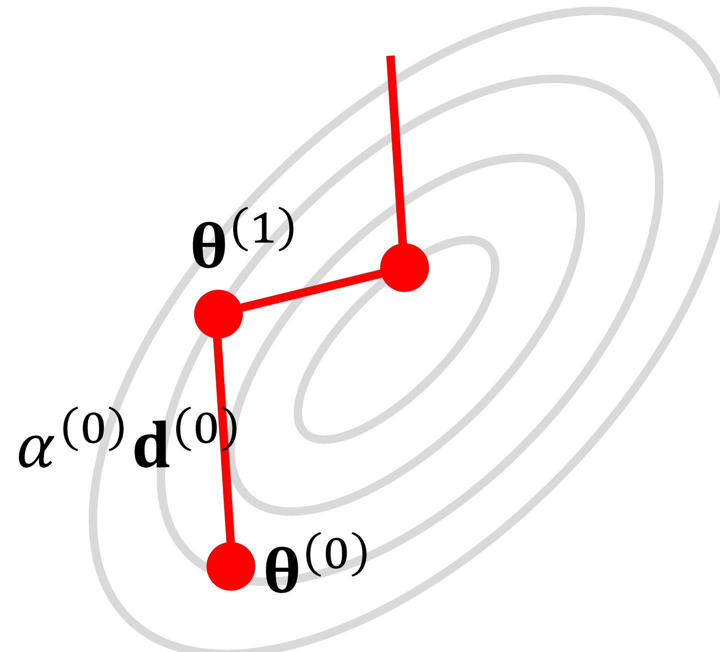
Initialization: start with some $\boldsymbol{\theta}^{(0)}$

For $k = 0$, … **until convergence**

Choose **descent direction** $\mathbf{d}^{(k)}$

Choose **step size** $\alpha^{(k)}$

Update $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} + \alpha^{(k)}\mathbf{d}^{(k)}$



Gradient descent

Select step $\mathbf{d}$ producing **biggest decrease** in the value of the loss function $\hat{L}$

$$\mathbf{d} = \arg\min_{\mathbf{d}} \hat{L}(\boldsymbol{\theta} + \boldsymbol{d}) - \hat{L}(\boldsymbol{\theta}) \quad \text{such that } \|\mathbf{d}\| = 1$$

$$\approx \arg\min_{\mathbf{d}} \; \nabla\hat{L}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{d} \quad \text{such that } \|\mathbf{d}\| = 1$$

Choice of L_2 metric ball:

$$\mathbf{d} = \arg\min_{\mathbf{d}} \nabla\hat{L}(\boldsymbol{\theta})^{T}\mathbf{d} \quad \text{such that } \|\mathbf{d}\|_2 = 1$$

$$= -\nabla\hat{L}(\boldsymbol{\theta})$$ **Gradient descent**

Gradient descent convergence rate

Strong convexity $\nabla^2 \hat{L}(\boldsymbol{\theta}) \succcurlyeq m\mathbf{I} \qquad m > 0$

Lipschitz gradient $\nabla^2 \hat{L}(\boldsymbol{\theta}) \preccurlyeq M\mathbf{I}$

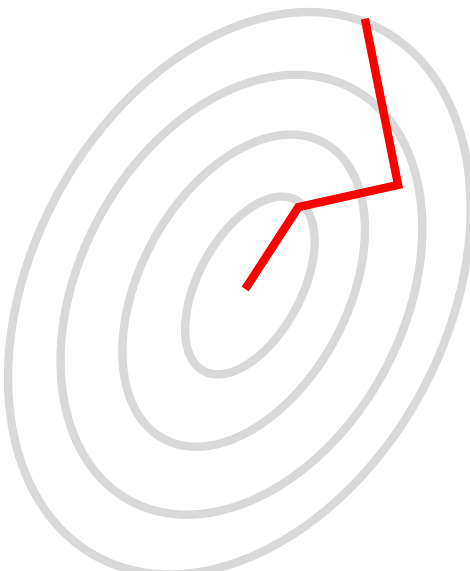
Constant step size $\alpha \leq \dfrac{2}{m + N}$

$$\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^{*}) \leq c^k \cdot \frac{M}{2}\left\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^{(k)}\right\|$$

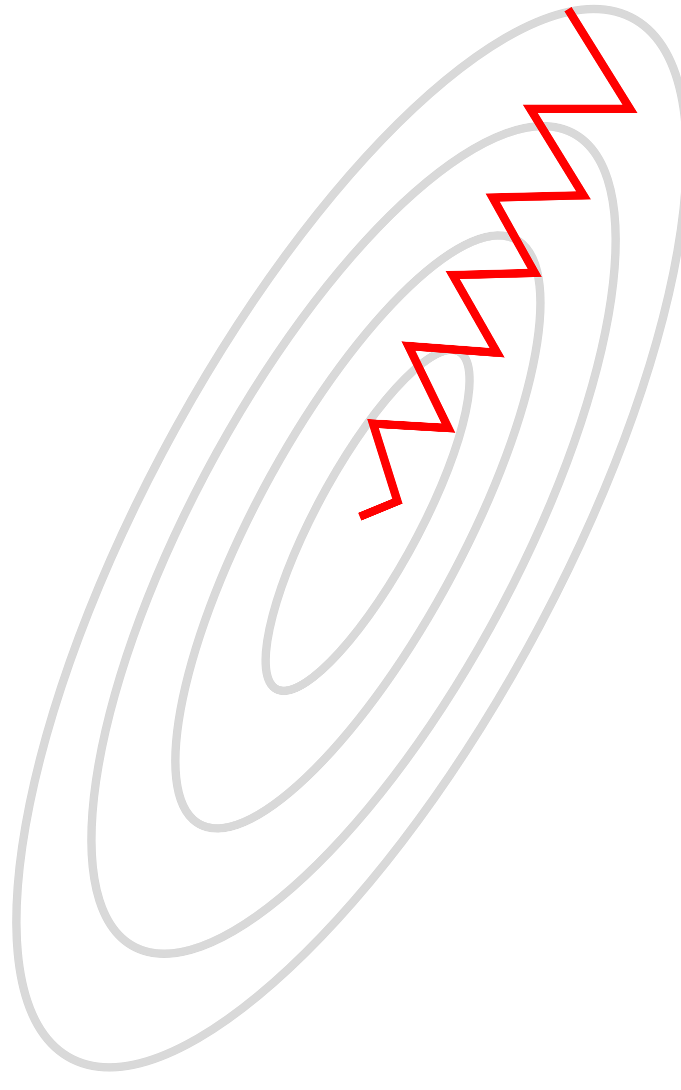
"Linear" convergence

To get $\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^{*}) \leq \epsilon$ one needs $\mathcal{O}\left(\log \dfrac{1}{\epsilon}\right)$ iterations

Gradient descent convergence rate



$$c = 1 - \frac{m}{M}$$



Computational complexity

$$\hat{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$$

single iteration complexity: $\mathcal{O}(n)$

Gradient descent convergence rate:

$$\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^*) = \mathcal{O}\left(c^k\right)$$

- ϵ-optimality requires $\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$ iterations
- overall complexity: $\mathcal{O}\left(n\log\frac{1}{\epsilon}\right)$

Stochastic gradient ~~descent~~

Regular ("batch") optimization

$$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} - \alpha^{(k)} \nabla \hat{L}(\boldsymbol{\theta}^{(k)})$$

- deterministic trajectory
- $-\nabla \hat{L}(\boldsymbol{\theta}^{(k)})$ always descent direction
- iteration cost $\mathcal{O}(n)$

Stochastic optimization

$$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} - \alpha^{(k)} \nabla \ell_k(\boldsymbol{\theta}^{(k)})$$

sample picked at random

- stochastic process
- $-\nabla \ell_k(\boldsymbol{\theta}^{(k)})$ not always a descent direction

  $-\mathbb{E}\nabla \ell_k(\boldsymbol{\theta}^{(k)})$ is a descent direction
- Iteration cost $\mathcal{O}(1)$

Stochastic gradient convergence rate

Stochastic gradient descent:

$$\mathbb{E}\left(\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^*)\right) = \mathcal{O}\left(\frac{1}{k}\right)$$

"sub-linear" convergence

To get $\mathbb{E}\left(\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^*)\right) \leq \epsilon$ requires $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ complexity

Big advantage for large n

Compare to **gradient descent:**

$$\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^*) = \mathcal{O}\left(c^k\right)$$

"linear" convergence

To get $\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^*) \leq \epsilon$ requires $\mathcal{O}\left(n \log \frac{1}{\epsilon}\right)$ complexity

(Batch) stochastic gradient convergence

For m-strongly convex $\hat{L}$ with M-Lipschitz gradient and fixed step size $\alpha \leq \frac{1}{M}$

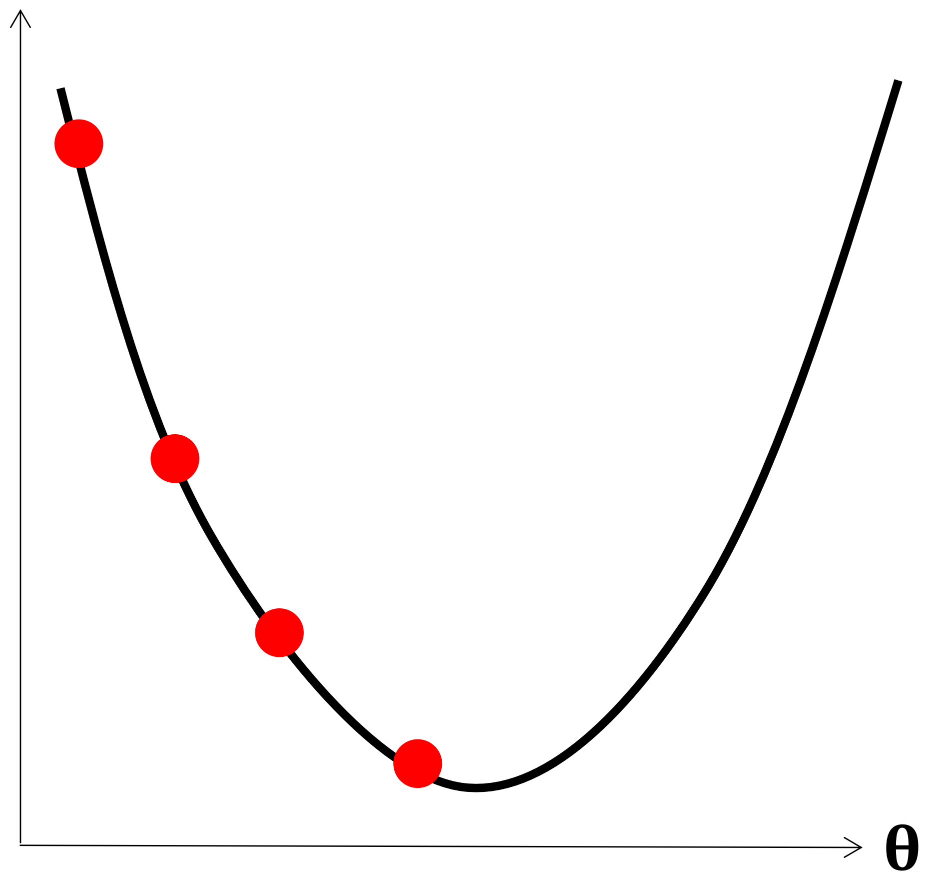
$$\mathbb{E}\left(\hat{L}(\boldsymbol{\theta}^{(k)}) - \hat{L}(\boldsymbol{\theta}^*)\right) \leq \frac{\alpha\sigma^2}{2m} + (1 - \alpha m)^k \left(\hat{L}(\boldsymbol{\theta}^{(0)}) - \hat{L}(\boldsymbol{\theta}^*)\right)$$

where $\sigma^2 = \mathcal{O}\left(\frac{1}{b}\right)$ is a bound on the gradient estimator variance and b is batch size

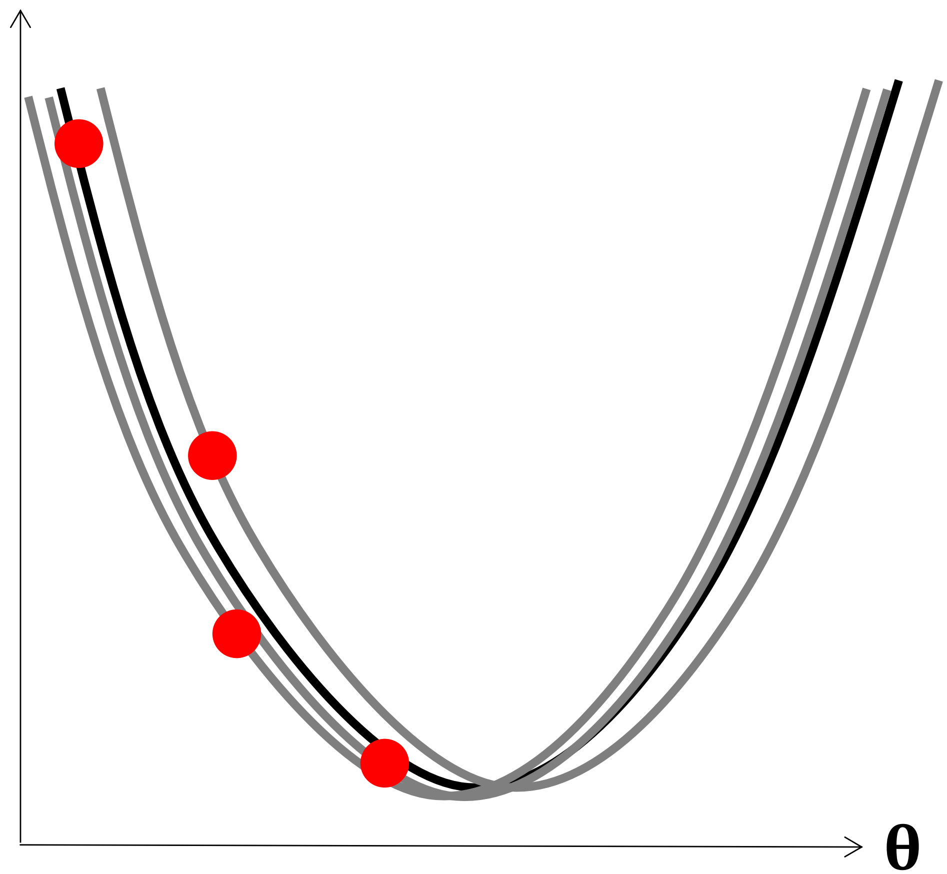
Linear (fast) convergence in the beginning

Gradient noise σ prevents further progress

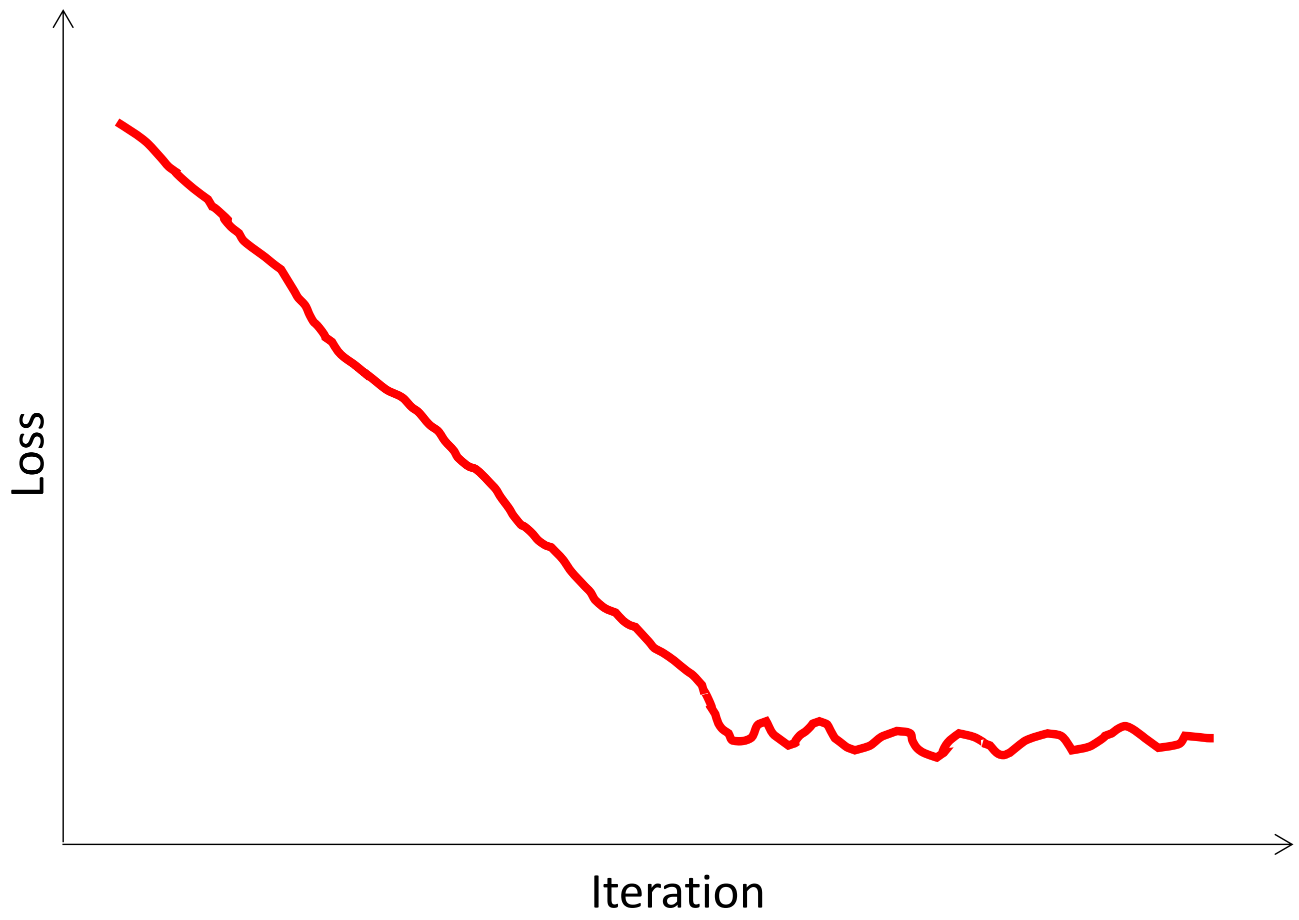
Convergence



Convergence



Convergence



Stochastic gradient convergence

$$\mathbb{E}\left(\hat{L}\left(\boldsymbol{\theta}^{(k)}\right) - \hat{L}(\boldsymbol{\theta}^{*})\right) \leq \frac{\alpha\sigma^2}{2m} + (1 - \alpha m)^k \left(\hat{L}\left(\boldsymbol{\theta}^{(0)}\right) - \hat{L}(\boldsymbol{\theta}^{*})\right)$$

Small step size ⟷ **Large step size**

Small step size	Large step size
Slower initial convergence	Faster initial convergence
Stalls at more accurate result	Stalls at less accurate result

Small batch size ⟷ **Large batch size**

Small batch size	Large batch size
Stalls at less accurate result	Stalls at more accurate result
Lower iteration cost	Higher iteration cost