

Optimization in Machine Learning

First order methods Gradient descent



Learning goals

- Iterative Descent / Line Search
- Descent directions
- GD
- ERM with GD
- Pseudoresiduals



ITERATIVE DESCENT

- Let f be the height of a mountain depending on the geographic coordinates (x_1, x_2)

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(x_1, x_2) = y.$$

- Goal:** Reach the valley

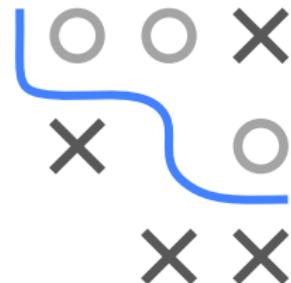
$$\arg \min_{\mathbf{x}} f(\mathbf{x})$$

- Central idea:** Repeat

- At current location $\mathbf{x} \in \mathbb{R}^d$ search for **descent direction** $\mathbf{d} \in \mathbb{R}^d$
- Move along \mathbf{d} until f „sufficiently“ reduces (**step size control**) and update location



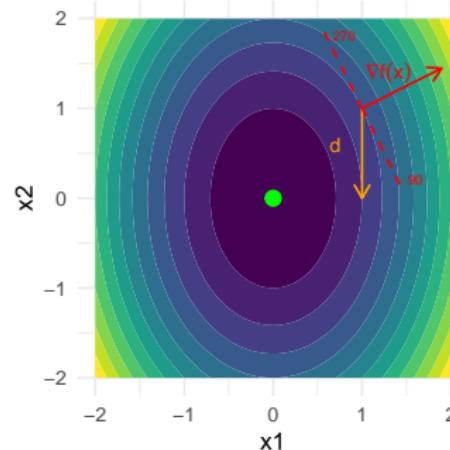
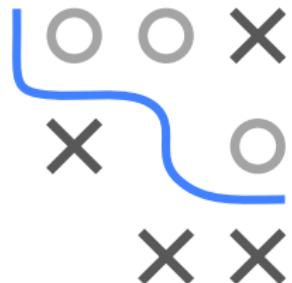
"Walking down the hill, towards the valley."



ITERATIVE DESCENT

- Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ continuously differentiable.
- Definition:** $\mathbf{d} \in \mathbb{R}^d$ is a **descent direction** in \mathbf{x} if

$$D_{\mathbf{d}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T \mathbf{d} < 0 \text{ (neg. directional derivative)}$$



Angle between $\nabla f(\mathbf{x})$ and \mathbf{d} must be $\in (90^\circ, 270^\circ)$.

ITERATIVE DESCEND

Algorithm Iterative Descent / Line search

- 1: Starting point $\mathbf{x}^{[0]} \in \mathbb{R}^d$
 - 2: **while** Stopping criterion not met **do**
 - 3: Calculate a descent direction $\mathbf{d}^{[t]}$ for current $\mathbf{x}^{[t]}$
 - 4: Find $\alpha^{[t]}$ s.t. $f(\mathbf{x}^{[t+1]}) < f(\mathbf{x}^{[t]})$ for $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + \alpha^{[t]} \mathbf{d}^{[t]}$.
 - 5: Update $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + \alpha^{[t]} \mathbf{d}^{[t]}$
 - 6: **end while**
-

NB: Terminology is sometimes ambiguous: “line search” can refer to Step 4 (selecting the step size that decreases $f(\mathbf{x})$) and can mean umbrella term for iterative descent algorithms.

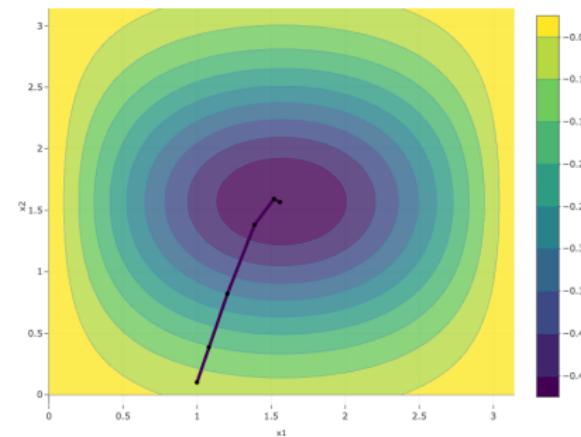
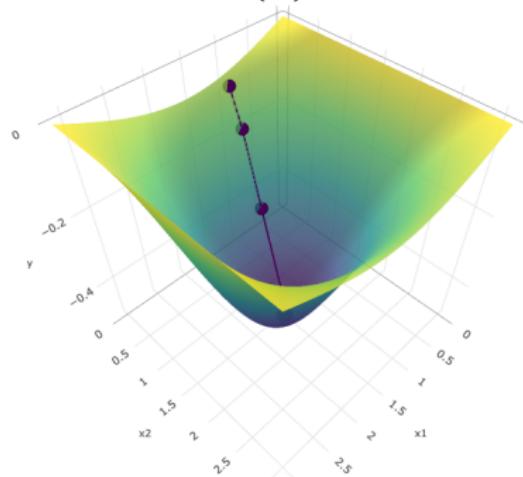
Key questions:

- How to choose $\mathbf{d}^{[t]}$ (now)
- How to choose $\alpha^{[t]}$ (later)



GRADIENT DESCENT

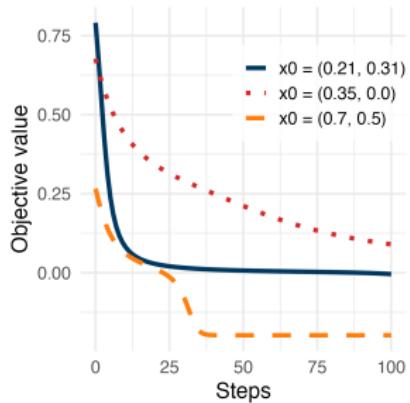
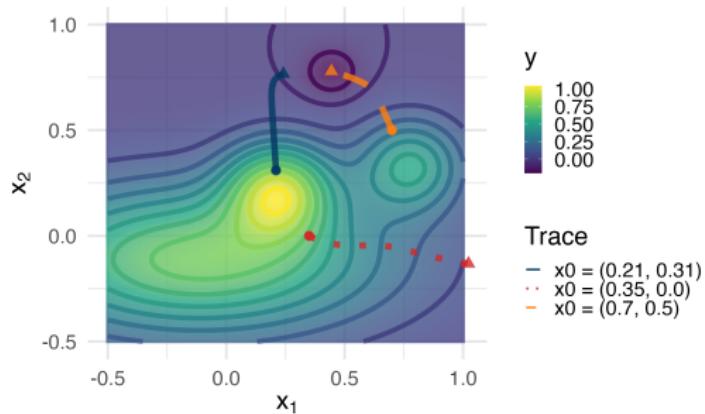
- Properties of gradient:
 - $\nabla f(\mathbf{x})$: direction of greatest increase
 - $-\nabla f(\mathbf{x})$: direction of greatest decrease
- Using $\mathbf{d} = -\nabla f(\mathbf{x})$ is called **gradient descent**.



GD for $f(x_1, x_2) = -\sin(x_1) \cdot \frac{1}{2\pi} \exp((x_2 - \pi/2)^2)$ with sensibly chosen step size $\alpha^{[t]}$.

GD AND MULTIMODAL FUNCTIONS

Outcome will depend on start point.



100 iters of GD with const $\alpha = 10^{-4}$.



OPTIMIZE LS LINEAR REGRESSION WITH GD

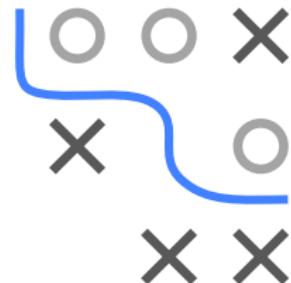
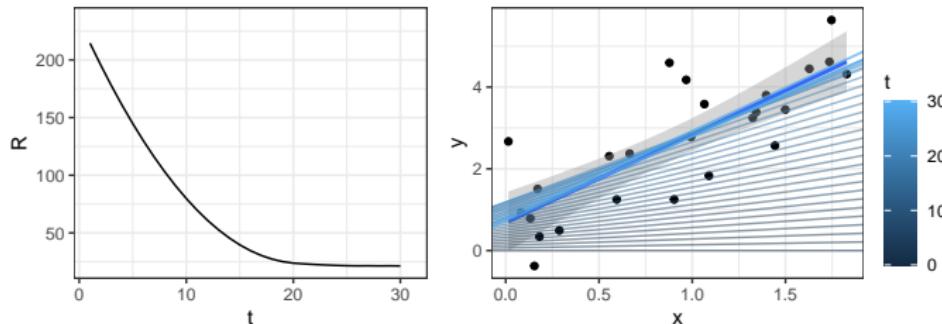
- Let $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)} \right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)} \right) \right)$ and minimize

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^n (\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)})^2$$

NB: For illustration, we use GD even though closed-form solution exists. GD-like (more adv.) approaches like this MIGHT make sense for large data, though.

- Gradient:

$$\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})(\boldsymbol{\theta}) = \frac{\partial \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2 \sum_{i=1}^n (y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}) \mathbf{x}^{(i)}$$

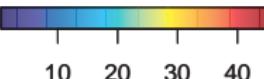
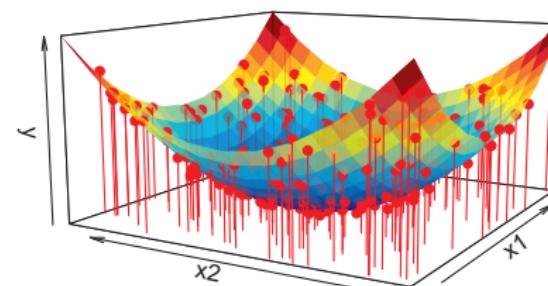


ERM FOR NN WITH GD

- Let $\mathcal{D} = ((\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)}))$, with $y = x_1^2 + x_2^2$ and minimize

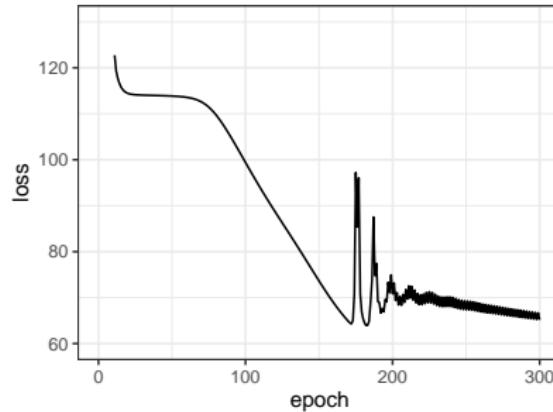
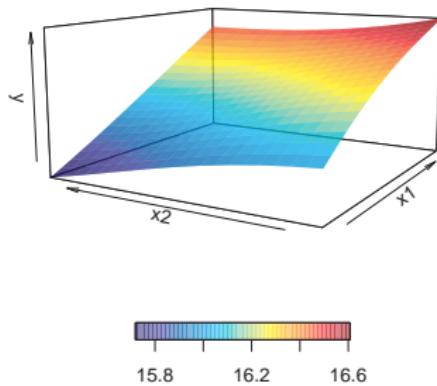
$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^n \left(f(\mathbf{x} | \boldsymbol{\theta}) - y^{(i)} \right)^2$$

where $f(\mathbf{x} | \boldsymbol{\theta})$ is a neural network with 2 hidden layers (2 units each).



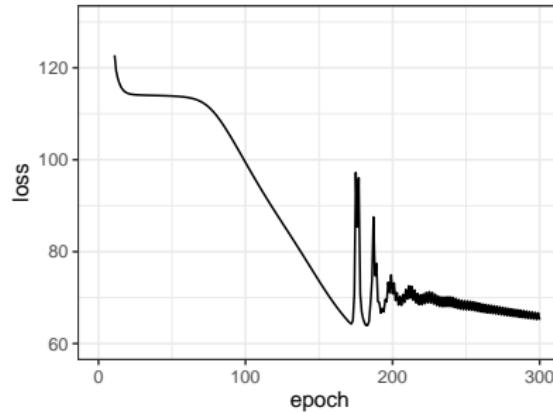
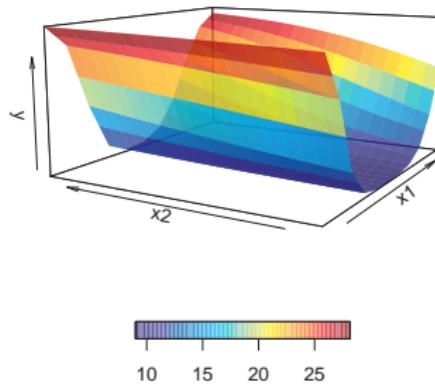
ERM FOR NN WITH GD

After 10 iters of GD:



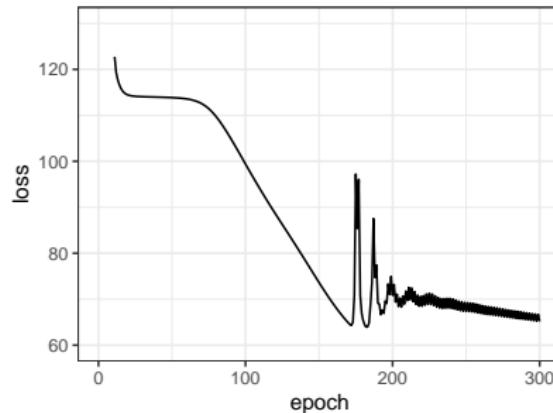
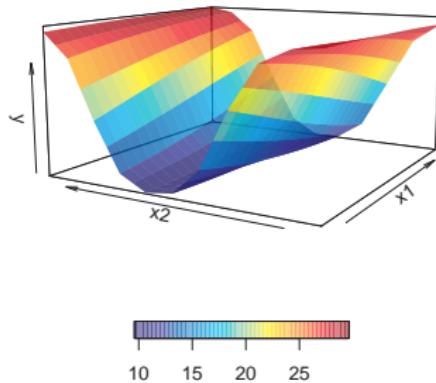
ERM FOR NN WITH GD

After 100 iters of GD:



ERM FOR NN WITH GD

After 300 iters of GD (note the zig-zag-behavior after iter. 200):



GD FOR ERM: PSEUDORESIDUALS

- Gradient for ERM problem:

$$-\frac{\partial \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{\partial \sum_{i=1}^n L(y^{(i)}, f(\mathbf{x}^{(i)} | \boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^n \underbrace{\frac{\partial L(y^{(i)}, f(\mathbf{x}^{(i)}))}{\partial f}}_{\text{pseudo residual } \tilde{r}^{(i)}(f)} \frac{\partial f(\mathbf{x}^{(i)} | \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

- pseudo residuals** tell us how to distort $f(\mathbf{x}^{(i)})$ to achieve greatest decrease of $L(y^{(i)}, f(\mathbf{x}^{(i)}))$ (best pointwise update)
- $\frac{\partial f(\mathbf{x}^{(i)} | \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ tells us how to modify $\boldsymbol{\theta}$ accordingly and wiggle model output
- GD step sums up these modifications across all observations i

NB: Pseudo-residuals $\tilde{r}(f)$ match usual residuals for L2 loss:

$$\begin{aligned}\frac{\partial L(y, f(\mathbf{x}))}{\partial f} &= \frac{1}{2} \left. \frac{\partial (y - f)^2}{\partial f} \right|_{f=f(\mathbf{x})} \\ &= y - f(\mathbf{x})\end{aligned}$$

