Seminar on Machine Learning for Optimization

Introductory Talk

Camille Castera & Sheheryar Mehmood

Mathematical Optimization Group

Saarland University, April 2024

UNIVERSITÄT DES SAARLANDES

Learning to Optimize

Optimization

- The science of **minimizing** and **maximizing** quantities.
- Optimization is ubiquitous: Physics, resource management, machine learning, economics,...
- "Everything can be formulated as an optimization problem" **but** almost none of these problems are solvable.

Machine Learning

- Using **models** to learn/extract information "automatically" **from data**.
- Becomes the "state-of-the-art" approach in many applications.
- Computer assisted: Often requires large amounts of data and important computational resources.

Learning to optimize: Can we use learning as a tool for optimization?

Optimization

For solving

$$\min_{x \in \mathbb{R}^n} f(x)\,.$$

Iterative Algorithm

$$x^{(k+1)} = x^{(k)} + d^{(k)}\,, \quad \text{for } k \geq 0\,.$$

- $x^{(0)}$ is the initial point.
- Run only for $k = 0, \ldots, K-1$.
- Use stopping criterion, e.g., stop when $\left\|\nabla f(x^{(k)})\right\| \leq \varepsilon$.

Examples and Computation Cost

$$x^{(k+1)} = x^{(k)} + d^{(k)}$$

Gradient Descent with step size $\alpha > 0$

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)}), \quad \text{for } k \geq 0.$$

Newton's Method

$$x^{(k+1)} = x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}), \quad \text{for } k \geq 0.$$

Computation Cost

- Per-iteration cost. How expensive is $x^{(k)} \longrightarrow x^{(k+1)}$?
- Convergence speed. How big is K?

Supervised Learning

Machine learning (ML)

Concept

Learn a relation between some **input** variable x and **output** variable y.

Such relations are too complex, we shall **approximate** them.

In ML we use models

Function $\mathcal{M}$ **parameterized** by $\theta \in \mathbb{R}^P$. Given input x, yields

$$\hat{y} = \mathcal{M}(x, \theta).$$

We want θ such that $\hat{y}$ is "close" to y.

Example: Image classification

Given an image x, the output y equals 1 if x contains a dog, and y equals 0 otherwise.

$x =$  $y = 0$

$x =$  $y = 1$

A specific class of ML models

Compositional structure in *layers* $(\mathcal{M}_\ell)_{\ell \in \{1,\ldots,L\}}$:

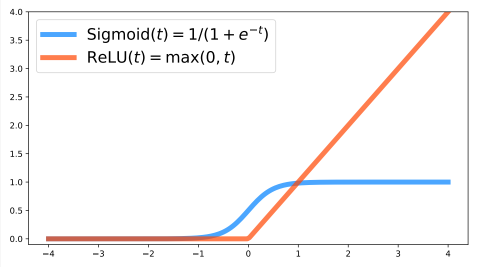
$$\mathcal{M} = \mathcal{M}_L \circ \mathcal{M}_{L-1} \circ \ldots \circ \mathcal{M}_1.$$

Typical layer: $\mathcal{M}_1(x, \theta_1) = g_1\left(W_1 x + b_1\right)$, where,

– W_1 is a matrix, b_1 a vector,

– g_1 is an **activation function** (non-linear).

Common activation functions



– **The parameter** $\theta \in \mathbb{R}^P$ of $\mathcal{M}$ contains the coefficients of the matrices and vectors of the layers.

– **Deep learning:** ML with neural networks.

Central question: How to select the parameter θ?

Loss function

– **Training dataset:** a collection of N examples

$$(x_n, y_n)_{n \in \{1,\dots,N\}}.$$

– **Loss function:** sum of the errors made by $\mathcal{M}$ on the training set, e.g.,

$$\mathcal{L}(\theta) \overset{\text{e.g.}}{=} \frac{1}{N} \sum_{n=1}^{N} \|\mathcal{M}(x_n, \theta) - y_n\|_2^2.$$

Training is an optimization problem

We seek $\theta \in \mathbb{R}^P$ which minimizes $\mathcal{L}$:

$$\min_{\theta \in \mathbb{R}^P} \mathcal{L}(\theta) \overset{\text{def}}{=} \min_{\theta \in \mathbb{R}^P} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\theta).$$

Learning to Optimize

What is Learning to Optimize?

Optimizing

Solving

$$\min_{x \in \mathbb{R}^n} f(x)$$

using algorithms:

$$x^{(k+1)} = x^{(k)} + d^{(k)}.$$

Warning: Training is NOT learning to optimize.

Training

We have seen that to learn (or train) a model $\mathcal{M}$, we must solve an optimization problem:

$$\min_{\theta \in \mathbb{R}^P} \mathcal{L}(\theta)$$

Learning to Optimize

Solving

$$\min_{x \in \mathbb{R}^n} f(x)$$

using a model $\mathcal{M}$:

$$\boxed{x^{(k+1)} = x^{(k)} + \mathcal{M}(x^{(k)}, \theta).}$$

We want to find a model $\mathcal{M}(\cdot, \theta)$, that is "good" at solving optimization problems.

To select the parameter θ we will additionally need to solve the training problem.

Chain rule and Automatic Differentiation

Backpropagation

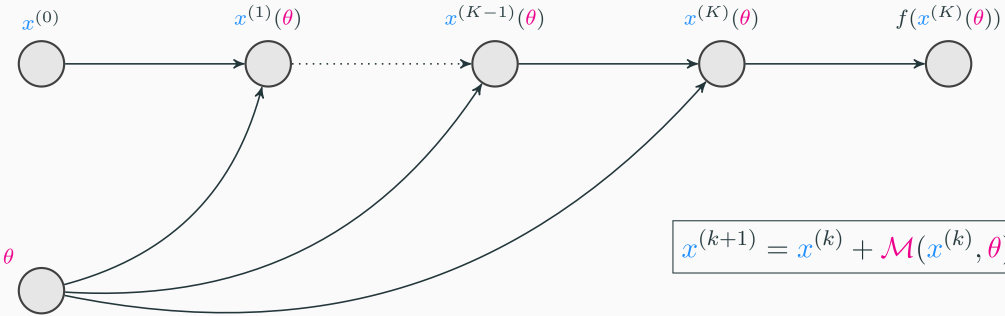
How does a change in θ affects $f(x^{(K)}(\theta))$?

We need to compute its derivative w.r.t θ!

Chain rule:

$$\nabla_\theta f(x^{(K)}(\theta)) = \left[\nabla_\theta\, x^{(K)}(\theta)\right]\left[\nabla f\left(x^{(K)}(\theta)\right)\right]$$

Unrolling:



$$\boxed{x^{(k+1)} = x^{(k)} + \mathcal{M}(x^{(k)}, \theta)}$$

To find a good θ, we will need to train $\mathcal{M}(\cdot, \theta)$

- **Training dataset:** a set of functions and initializations:

$$\{(f_i, x_i^{(0)}) : 1 \leq i \leq I\}.$$

- For each pair $(f_i, x_i^{(0)})$, run the algorithm to obtain $x_i^{(K)}(\theta)$:

for $k = 0, \ldots, K-1$:

$$x_i^{(k+1)}(\theta) = x_i^{(k)}(\theta) + \mathcal{M}(x_i^{(k)}(\theta), \theta).$$

- **Loss Function:** sum of the values of all function f_i

$$\mathcal{L}(\theta) = \frac{1}{I}\sum_{i=1}^{I} f_i\left(x_i^{(K)}(\theta)\right).$$

- Compute $\nabla_\theta \mathcal{L}(\theta) = \frac{1}{I}\sum_{i=1}^{I} \nabla_\theta f_i\left(x_i^{(K)}(\theta)\right)$ and use it to update θ:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$$

Seminar on Machine Learning for Optimization

Introductory Talk

Camille Castera & Sheheryar Mehmood

Mathematical Optimization Group

Saarland University, April 2024

UNIVERSITÄT DES SAARLANDES