Machine Learning II

Jannik Irmai, Jannik Presberger, David Stein, Bjoern Andres

Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/25-summer/ml2/

Summer Term 2025

Definition. For any finite, non-empty set S, called a set of **samples**, any $X \neq \emptyset$, called an **feature space** and any $x : S \to X$, the tuple (S, X, x) is called **unlabeled data**.

For any $y : S \to \{0, 1\}$, given in addition and called a **labeling**, the tuple (S, X, x, y) is called **labeled data**.

The **supervised learning problem** is an optimization problem. It consists in finding, in a family of functions, one function that minimizes a weighted sum of two objectives:

1. It deviates little from given labeled data, as quantified by a loss function
2. It has low complexity, as quantified by a regularizer.

Definition. For any labeled data $T = (S, X, x, y)$, any $\Theta \neq \varnothing$ and $f : \Theta \to \mathbb{R}^X$, any $R : \Theta \to \mathbb{R}_0^+$, called a **regularizer**, any $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, called a **loss function**, and any $\lambda \in \mathbb{R}_0^+$, the instance of the **supervised learning problem** has the form

$$\inf_{\theta \in \Theta} \quad \underbrace{\lambda R(\theta) + \sum_{s \in S} L(f_\theta(x_s), y_s)}_{=:\varphi(\theta)} \tag{1}$$

Example. l_2-regularized logistic regression: Given a finite index set $J \neq \varnothing$ and $\Theta := \mathbb{R}^J$, let

$$\begin{aligned} R(\theta) &:= \|\theta\|_2^2 \\ L(r, y') &:= -y' r + \log_2(1 + 2^r) \end{aligned}$$

Algorithm. Steepest descent with parameters $\eta, \epsilon \in \mathbb{R}_0^+$ and initialization $\theta \in \mathbb{R}^J$:

```
repeat
    d := (∇_θ φ)(θ)
    θ := θ − ηd
    if ‖d‖ < ε
        return θ
```

Definition. For any unlabeled data $T = (S, X, x)$, any $\hat{f} : X \to \mathbb{R}$ and any $L : \mathbb{R} \times \{0, 1\} \to \mathbb{R}_0^+$, the instance of the **inference problem** wrt. $T, \hat{f}$ and L is defined as

$$\min_{y \in \{0,1\}^S} \sum_{s \in S} L(\hat{f}(x_s), y_s) \tag{2}$$

Lemma. The solutions to the inference problem are the $y : S \to \{0, 1\}$ such that

$$\forall s \in S: \quad y_s \in \operatorname*{argmin}_{\hat{y} \in \{0,1\}} L(\hat{f}(x_s), \hat{y}) \ . \tag{3}$$

Consider a real feature space $X := \mathbb{R}^K$ with finite $K \neq \varnothing$.

What functions $f_\theta \colon X \to \mathbb{R}$ do we wish to learn?

- Linear functions f_θ, i.e. $\Theta := \mathbb{R}^K$ and $\forall x' \in X \colon f_\theta(x') = \langle \theta, x' \rangle$
- Functions f_θ defined by a **compute graph**, i.e. a deep (artificial netural) network.

Notation. Let $G = (V, E)$ a digraph.

- For any $v \in V$, let

$$P_v = \{u \in V \mid (u, v) \in E\} \quad \text{the set of \textbf{parents} of } v \tag{4}$$
$$C_v = \{w \in V \mid (v, w) \in E\} \quad \text{the set of \textbf{children} of } v \ . \tag{5}$$

- For any $u, v \in V$, let $\mathcal{P}(u, v)$ denote the set of all uv-paths of G. (Any path is a subgraph. For any node u, the uu-path $(\{u\}, \emptyset)$ exists.)

Let G be **acyclic**.

- For any $v \in V$, let

$$A_v = \{u \in V \mid \mathcal{P}(u, v) \neq \emptyset\} \setminus \{v\} \quad \text{the set of \textbf{ancestors} of } v \tag{6}$$
$$D_v = \{w \in V \mid \mathcal{P}(v, w) \neq \emptyset\} \setminus \{v\} \quad \text{the set of \textbf{descendants} of } v \ . \tag{7}$$

Definition. A tuple $(V, D, D', E, \Theta, \{g_{v\theta} \colon \mathbb{R}^{P_v} \to \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$ is called a **compute graph**, iff the following conditions hold:

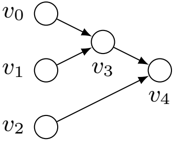
- $G = (V \cup D \cup D', E)$ is an acyclic digraph.
- For any $v \in V$, called an **input node**, $P_v = \emptyset$.
- For any $v \in D'$, called an **output node**, $C_v = \emptyset$.
- For any $v \in D$, called a **hidden node**, $P_v \neq \emptyset$ and $C_v \neq \emptyset$.

Definition. For any compute graph $(V, D, D', E, \Theta, \{g_{v\theta} \colon \mathbb{R}^{P_v} \to \mathbb{R}\}_{v \in (D \cup D') \setminus V, \theta \in \Theta})$, any $v \in V \cup D \cup D'$ and any $\theta \in \Theta$, let $\alpha_{v\theta} \colon \mathbb{R}^V \to \mathbb{R}$ such that for all $\hat{x} \in \mathbb{R}^V$:

$$\alpha_{v\theta}(\hat{x}) = \begin{cases} \hat{x}_v & \text{if } v \in V \\ g_{v\theta}(\alpha_{P_v\theta}(\hat{x})) & \text{otherwise} \end{cases} \quad . \tag{8}$$

For any $\theta \in \Theta$ let $f_\theta \colon \mathbb{R}^V \to \mathbb{R}^{D'}$ such that $f_\theta = \alpha_{D'\theta}$.
We call $\alpha_{v\theta}(\hat{x})$ the **activation** of v for **input** $\hat{x}$ and **parameters** θ.
We call $f_\theta(\hat{x})$ the **output** of the compute graph for input $\hat{x}$ and parameters θ.

Example. Consider $V = \{v_0, v_1, v_2\}$, $D = \{v_3\}$, $D' = \{v_4\}$ and the edge set E of the digraph depicted below.



Consider, in addition, $\Theta = \{\theta_0, \theta_1\}$ and

$$g_{v_3\theta}\colon \quad \mathbb{R}^{\{v_0,v_1\}} \to \mathbb{R}\colon \quad x \mapsto x_{v_0} + \theta_0 x_{v_1} \tag{9}$$

$$g_{v_4\theta}\colon \quad \mathbb{R}^{\{v_2,v_3\}} \to \mathbb{R}\colon \quad x \mapsto x_{v_2} + x_{v_3}^{\theta_1} \ . \tag{10}$$

The compute graph $(V, D, D', E, \Theta, \{g_{v_3\theta}, g_{v_4\theta}\})$ defines the function

$$f_\theta\colon \quad \mathbb{R}^V \to \mathbb{R}^{D'}\colon \quad x \mapsto x_{v_2} + (x_{v_0} + \theta_0\, x_{v_1})^{\theta_1} \ . \tag{11}$$

Summary.

- The supervised deep learning problem is a supervised learning problem (e.g. the l_2-regularized logistic regression problem) with respect to a family of functions f_θ defined by a compute graph (i.e. a deep network).
- In order to apply the steepest descent algorithm to this problem, we need to repeatedly calculate $\nabla_\theta \varphi$ and thus $\nabla_\theta f$. This can be done, e.g., by the forward propagation algorithm or the backward propagation algorithm, which are discussed in the course Machine Learning I.