Neural Networks and Deep Learning: Supervised Machine Learning

Nicolas Thome

Conservatoire National des Arts et Métiers (Cnam)
Département Informatique

Supervised Learning

- Input $\mathbf{x}$, output $\mathbf{y}$
- A parametrized model $\mathbf{x} \Rightarrow \mathbf{y}$: $f_{\mathbf{w}}(\mathbf{x}_i) = \hat{\mathbf{y}}_i$
- Supervised context: training set $\mathcal{A} = \{(\mathbf{x}_i, \mathbf{y}_i^*)\}_{i \in \{1,2,\ldots,N\}}$
  - A loss function $\ell(\hat{\mathbf{y}}_i, \mathbf{y}_i^*)$ for each annotated pair $(\mathbf{x}_i, \mathbf{y}_i^*)$
- **Goal of supervised learning:**
  minimizing average loss $\mathcal{L}(\mathbf{w})$ over training set:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) \qquad (1)$$

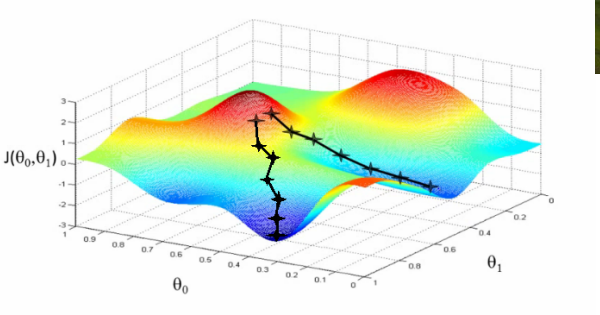
- Eq (1): very general formulation, a specific task requires:
  1. Defining output space for $\mathbf{y}$
  2. Defining how predicted output $\hat{\mathbf{y}}_i$ is computed, *i.e.* $f_{\mathbf{w}}$: $f_{\mathbf{w}}(\mathbf{x}_i) = \hat{\mathbf{y}}_i$
  3. Defining $\ell(\hat{\mathbf{y}}_i, \mathbf{y}_i^*)$: classification, regression, structured loss, *etc*

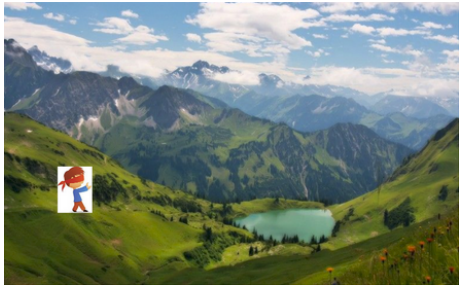
Supervised Machine Learning: Image Classification Example

- CIFAR Images, several parameter initializations, show prediction and compute accuracy
- Message : random sampling the most naive optimization approach to supervised machine learning

Supervised Learning & Gradient Descent

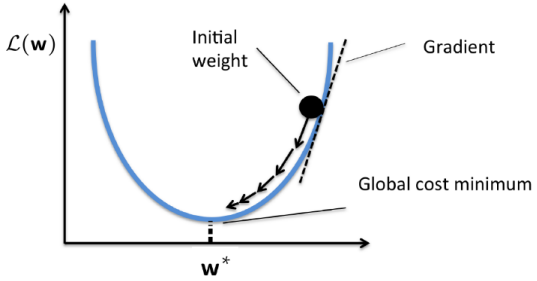
- Assumptions: parameters $\mathbf{w} \in \mathbb{R}^d$ continuous, $\mathcal{L}$ differentiable
- Gradient $\nabla_{\mathbf{w}} = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$: steepest direction to decrease loss $\mathcal{L}$





Supervised Learning & Gradient Descent

- Gradient $\nabla_{\mathbf{w}} = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$: steepest direction to decrease loss $\mathcal{L}$
- Gradient descent algorithm:
  - Initialyze parameters $\mathbf{w}$
  - Update: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$
  - Until convergence, *e.g.* $||\nabla_{\mathbf{w}}||^2 \approx 0$



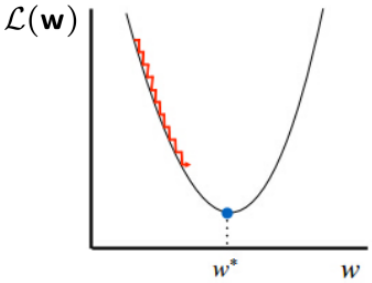
$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i^*)$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$$

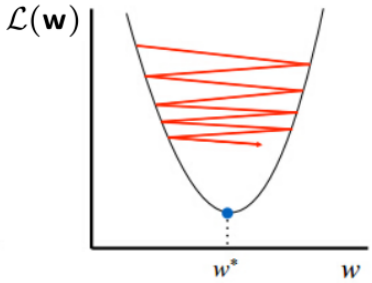
Gradient Descent

Update rule: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$ η **learning rate**

- **Convergence ensured ?** $\Rightarrow$ provided a "well chosen" learning rate η



Too small: converge very slowly

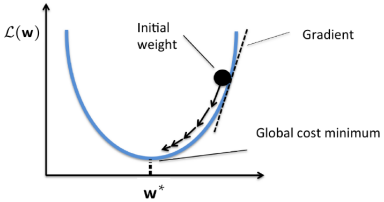


Too big: overshoot and even diverge

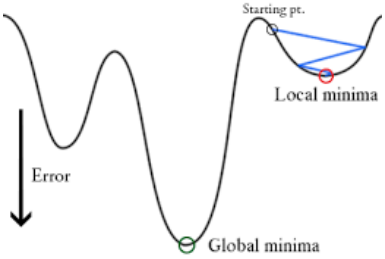
Gradient Descent

Update rule: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$

- **Global minimum ?** ⇒ **convex** a) *vs* **non convex** b) loss functions $\mathcal{L}(\mathbf{w})$



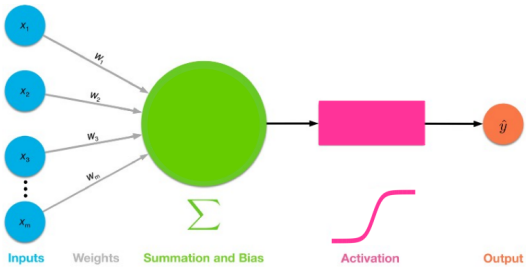
a) Convex function



a) Non convex function

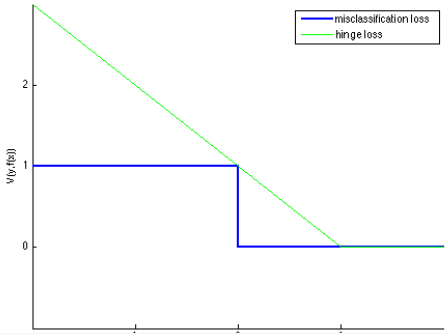
Supervised Learning & Binary Classification

- Supervised loss function $\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i^*)$ very general
- Application for binary classification:
  1. $\mathbf{y} \in \{-1; 1\}$
  2. $\hat{\mathbf{y}}_i = sign\left(\mathbf{w}^\top \mathbf{x} + b\right)$ (Heaviside)
  3. $\ell_{0/1}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) = \begin{cases} 1 & \text{if } \hat{\mathbf{y}}_i \neq \mathbf{y}_i^* \\ 0 & \text{otherwise} \end{cases} = 1_{\hat{y}_i y_i^* < 0}$: **0/1 loss**



Binary Classification & Gradient Descent

1. $\ell_{0/1}$ non differentiable !
2. Common solution: design surrogate function
   - Upper bound: surrogate function=0 $\Rightarrow$ original function =0
   - Smooth $\Rightarrow$ gradient descent
   - Convex $\Rightarrow$ global minimization easier
3. Ex: hinge loss $\ell_{hinge}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) = \max\left[0, 1 - (\mathbf{w}^\top \mathbf{x} + b)\, y_i^*\right]$ (SVM)



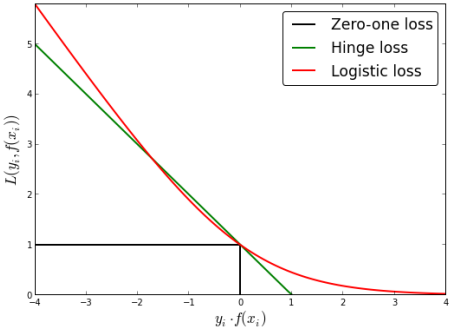
Binary cross entropy

- Binary cross-entropy ($\mathbf{y} \in \{0; 1\}$):

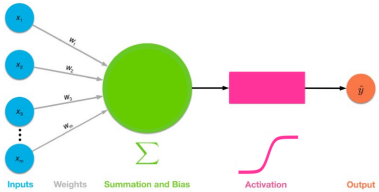
$$\ell_{CEb}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) = -\mathbf{y}_i^* \, log\,(f(\mathbf{x}_i)) - (1 - \mathbf{y}_i^*)\, log\,(1 - f(\mathbf{x}_i))$$

- $\sim$ Logistic loss ($\mathbf{y} \in \{-1; 1\}$):

$$\ell_{log}(\hat{\mathbf{y}}_i, \mathbf{y}_i^*) = log\left[1 + e^{-\mathbf{y}_i^*(\mathbf{w}^\top \mathbf{x} + b)}\right]$$



$$f(\mathbf{x}_i) = \frac{1}{1 + e^{-\boldsymbol{a}(\mathbf{w}^\top \mathbf{x} + \boldsymbol{b})}}$$



Supervised Machine Learning & Neural Networks

- Supervised Training: smooth optimization with gradient descent
- Application to binary classification with convex cross entropy loss
- Supervised training of deep neural networks $\Rightarrow$ following!