Chapitre 7 : Convexification

25 février 2026

Plan du chapitre

Le problème du Risque Naturel (Coût 0-1)

En classification binaire ($\mathscr{Y} = \{-1, +1\}$), le but est d'apprendre une fonction $h : \mathscr{X} \to \mathscr{Y}$ minimisant le pourcentage d'erreurs.

Traduction mathématique d'une erreur

Si $y_i \in \{-1, 1\}$ et $h(\mathbf{x}_i) \in \{-1, 1\}$, une erreur se produit si

$$y_i \neq h(\mathbf{x}_i).$$

Cela revient exactement à dire que

$$y_i h(\mathbf{x}_i) < 0$$

Probabilité d'erreur et Optimisation

La probabilité de fausse décision (risque $0/1$) s'écrit formellement comme l'espérance de la fonction indicatrice d'erreur :

$$L_{\mathbb{P}}(h) = \mathbb{P}(Yh(X) < 0) = \mathbb{E}_{\mathbb{P}}\left[\mathbb{1}_{Yh(X)<0}\right]$$

Le mur de l'optimisation

Bien que cette mesure (le coût 0-1) soit très naturelle, sa minimisation directe est un cauchemar mathématique.
La fonction indicatrice n'est pas continue, sa dérivée est nulle presque partout (donc pas de descente de gradient) et le problème d'optimisation est NP-difficile (non-convexe).

Convexification

Puisqu'on ne peut pas minimiser $\mathbb{1}_{x<0}$, on va la remplacer par une fonction ϕ plus commode qui sert de majorant :

$$\phi(x) \geq \mathbb{1}_{x<0}$$

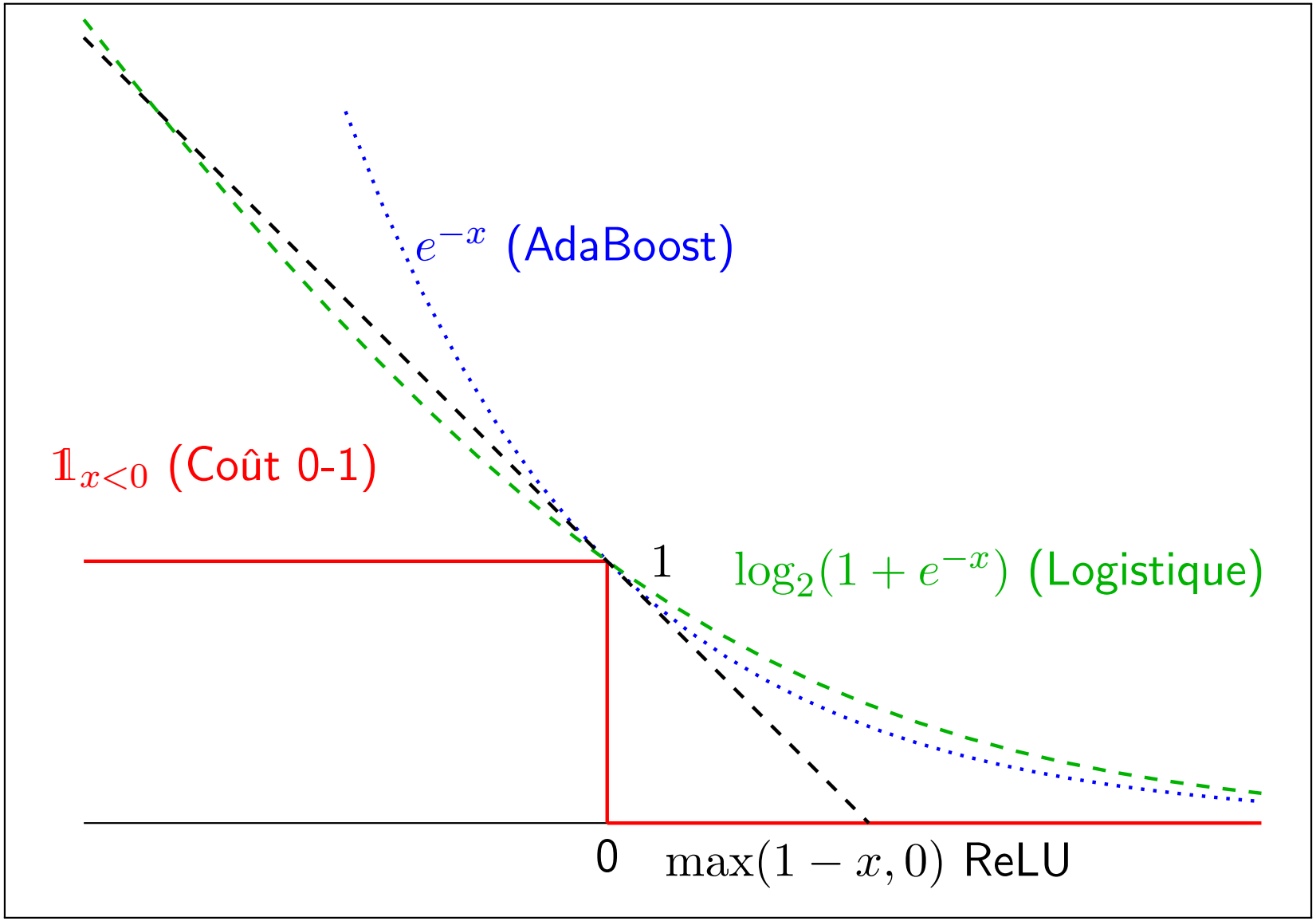
Propriétés requises pour ϕ (calibrée)

- Strictement **convexe** (garantit un minimum global).
- **Décroissante** (pénalise d'autant plus que l'erreur est grande).
- $\phi(0) = 1$ et $\phi'(0) < 0$.

Le risque naturel est alors remplacé par un risque convexe (plus grand, mais optimisable) :

$$\mathbb{E}_{\mathbb{P}}\left[\phi(Yh(X))\right]$$

Les grandes familles de fonctions de coût



Zoom sur ReLU (Rectified Linear Unit)

Le coût Hinge ($\max(1-x,0)$) est très similaire à la fonction d'activation ReLU utilisée massivement en Deep Learning.

Pourquoi ReLU a révolutionné les Réseaux de Neurones ?

1. **Fini l'évanouissement du gradient :** La fonction logistique (Sigmoïde) "sature" vite (dérivée proche de 0), empêchant l'apprentissage des couches profondes. La partie linéaire de ReLU a une dérivée constante (1), le gradient circule librement !
2. **Efficacité de calcul :** C'est la fonction la plus simple à calculer par une machine (un simple $\max(0,x)$). Pas d'exponentielle coûteuse.

Choix de l'ensemble des classifieurs ($\mathscr{H}$)

Grâce à la fonction ϕ, l'optimisation (Minimisation du Risque Empirique) est maintenant un problème convexe facile à résoudre numériquement.

Il ne reste qu'à choisir l'espace des fonctions $\mathscr{H}$ dans lequel on va chercher la solution.
Deux choix fructueux en Machine Learning classique sont :

- **Les combinaisons linéaires de classifieurs faibles :** C'est l'approche des algorithmes d'ensemble (comme le **Boosting** / AdaBoost).
- **Les espaces de Hilbert à noyau reproduisant (RKHS) :** C'est l'approche des **Machines à Vecteurs de Support (SVM)** et des méthodes à noyau (Kernel Trick), qui permettent de créer des frontières non linéaires extrêmement complexes tout en conservant la convexité du problème !