Fonctions de côut
Loss Functions

Jérémie Cabessa

Laboratoire DAVID, UVSQ

Learning Problem

- Soit le training set

$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \ldots, N \right\}$$

- Soit $\hat{f}(\cdot; \boldsymbol{\Theta})$ un **modèle** qui dépend de *paramètres* $\boldsymbol{\Theta}$:

$$\begin{aligned} \hat{f}(\cdot; \boldsymbol{\Theta}) : \mathbb{R}^{d_1} &\longrightarrow \mathbb{R}^{d_2} \\ \boldsymbol{x} &\longmapsto \hat{\boldsymbol{y}} := \hat{f}(\boldsymbol{x}; \boldsymbol{\Theta}) \end{aligned}$$

- L'entraînement du modèle $\hat{f}(\cdot; \boldsymbol{\Theta})$ consiste à déterminer les paramètres $\boldsymbol{\Theta}$ qui minimisent l'erreur (ou la loss) entre les *prédictions* $\hat{\boldsymbol{y}}_i = \hat{f}(\boldsymbol{x}_i; \boldsymbol{\Theta})$ et les *réalités* $\boldsymbol{y}_i$, pour $i = 1, \ldots, N$.

Learning Problem

- Soit le training set

$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \ldots, N \right\}$$

- Soit $\hat{f}(\cdot; \boldsymbol{\Theta})$ un **modèle** qui dépend de *paramètres* $\boldsymbol{\Theta}$:

$$\begin{aligned} \hat{f}(\cdot; \boldsymbol{\Theta}) : \mathbb{R}^{d_1} &\longrightarrow \mathbb{R}^{d_2} \\ \boldsymbol{x} &\longmapsto \hat{\boldsymbol{y}} := \hat{f}(\boldsymbol{x}; \boldsymbol{\Theta}) \end{aligned}$$

- L'entraînement du modèle $\hat{f}(\cdot; \boldsymbol{\Theta})$ consiste à déterminer les paramètres $\boldsymbol{\Theta}$ qui minimisent l'erreur (ou la loss) entre les *prédictions* $\hat{\boldsymbol{y}}_i = \hat{f}(\boldsymbol{x}_i; \boldsymbol{\Theta})$ et les *réalités* $\boldsymbol{y}_i$, pour $i = 1, \ldots, N$.

Learning Problem

- Soit le training set

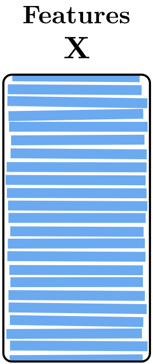
$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

- Soit $\hat{f}(\cdot; \boldsymbol{\Theta})$ un **modèle** qui dépend de *paramètres* $\boldsymbol{\Theta}$:

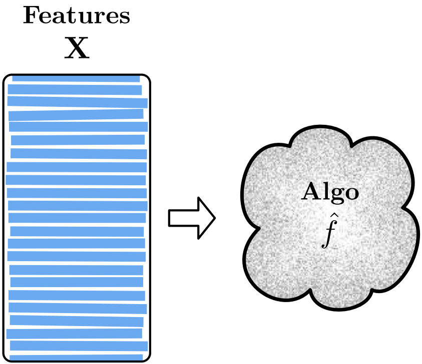
$$\begin{aligned} \hat{f}(\cdot; \boldsymbol{\Theta}) : \mathbb{R}^{d_1} & \longrightarrow \mathbb{R}^{d_2} \\ \boldsymbol{x} & \longmapsto \hat{\boldsymbol{y}} := \hat{f}(\boldsymbol{x}; \boldsymbol{\Theta}) \end{aligned}$$

- L'entraînement du modèle $\hat{f}(\cdot; \boldsymbol{\Theta})$ consiste à déterminer les paramètres $\boldsymbol{\Theta}$ qui minimisent l'erreur (ou la loss) entre les *prédictions* $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \hat{f}(\boldsymbol{x_i}; \boldsymbol{\Theta})$ et les *réalités* $\boldsymbol{y_i}$, pour $i = 1, \dots, N$.

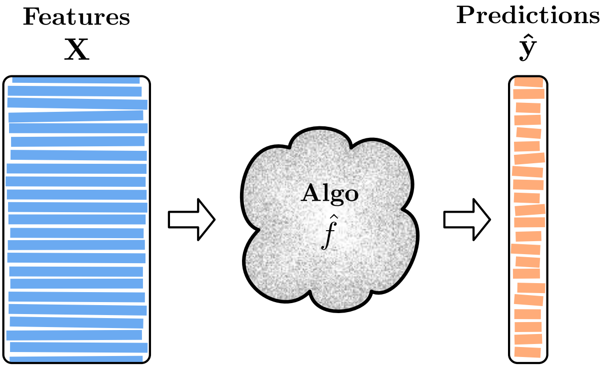
Learning Problem



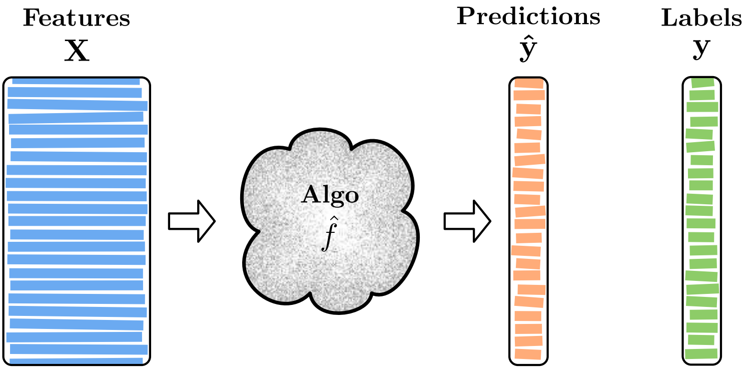
Learning Problem



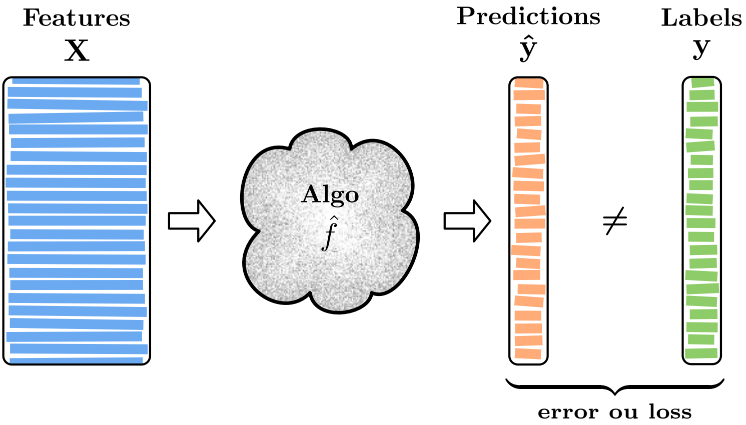
Learning Problem



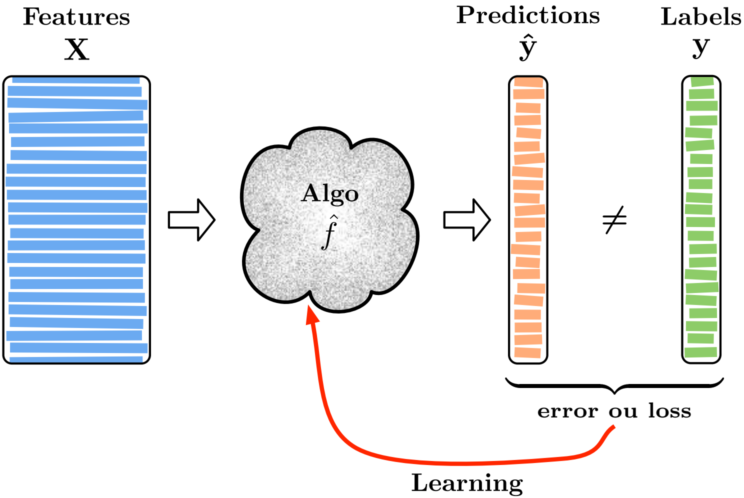
Learning Problem



Learning Problem



Learning Problem



Learning Problem

- Une **fonction de coût (cost or loss function)** mesure l'erreur entre une *prédiction* $\hat{\boldsymbol{y}}_i$ et une *réalité* $\boldsymbol{y}_i$:

$$\begin{aligned} \ell : \mathbb{R}^{d_2} \times \mathbb{R}^{d_2} &\longrightarrow \mathbb{R} \\ (\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i) &\longmapsto \ell(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i) \end{aligned}$$

- La **fonction de coût (loss function)** peut être généralisée à un ensemble de *prédictions* et de *réalités*:

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{d_2} \times \cdots\cdots \times \mathbb{R}^{d_2} &\longrightarrow \mathbb{R} \\ (\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N, \boldsymbol{y}_i \ldots, \boldsymbol{y}_N) &\longmapsto \mathcal{L}(\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N, \boldsymbol{y}_i \ldots, \boldsymbol{y}_N) \end{aligned}$$

Learning Problem

- Une **fonction de coût (cost or loss function)** mesure l'erreur entre une *prédiction* $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ et une *réalité* $\boldsymbol{y}_{\boldsymbol{i}}$:

$$\begin{aligned} \ell : \mathbb{R}^{d_2} \times \mathbb{R}^{d_2} & \longrightarrow \mathbb{R} \\ (\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}) & \longmapsto \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}\right) \end{aligned}$$

- La **fonction de coût (loss function)** peut être généralisée à un ensemble de *prédictions* et de *réalités*:

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{d_2} \times \cdots\cdots \times \mathbb{R}^{d_2} & \longrightarrow \mathbb{R} \\ (\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{i}} \ldots, \boldsymbol{y}_{\boldsymbol{N}}) & \longmapsto \mathcal{L}\left(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{i}} \ldots, \boldsymbol{y}_{\boldsymbol{N}}\right) \end{aligned}$$

Learning Problem

- Pour différents paramètres $\boldsymbol{\Theta}$, on aura différentes prédictions $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \hat{f}\left(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}\right)$, et donc différentes erreurs $\ell\left(\dots\right)$ et $\mathcal{L}\left(\dots\right)$.
- Ainsi, ℓ et $\mathcal{L}$ sont aussi des fonctions des paramètres $\boldsymbol{\Theta}$:

$$
\begin{aligned}
\ell : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} &\longmapsto \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\right) \\
\mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} &\longmapsto \mathcal{L}\left(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \dots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{1}}, \dots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\right)
\end{aligned}
$$

où $|\boldsymbol{\Theta}|$ est le nombre de paramètres $\boldsymbol{\Theta}$.

Learning Problem

- Pour différents paramètres $\boldsymbol{\Theta}$, on aura différentes prédictions $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \hat{f}\left(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}\right)$, et donc différentes erreurs $\ell\left(\dots\right)$ et $\mathcal{L}\left(\dots\right)$.
- **Ainsi, ℓ et $\mathcal{L}$ sont aussi des fonctions des paramètres $\boldsymbol{\Theta}$:**

$$
\begin{aligned}
\ell : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} &\longmapsto \ell\big(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\big) \\
\mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} &\longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \dots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{1}}, \dots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big)
\end{aligned}
$$

où $|\boldsymbol{\Theta}|$ est le nombre de paramètres $\boldsymbol{\Theta}$.

Learning Problem

- *L'entraînement* du modèle $\hat{f}(\cdots;\boldsymbol{\Theta})$ consiste à déterminer des paramètres $\boldsymbol{\Theta}$ qui minimisent les fonction de coût

$$\ell(\ldots;\boldsymbol{\Theta}) \quad \text{ou} \quad \mathcal{L}(\ldots;\boldsymbol{\Theta}).$$

- Pour minimiser la fonction de coût, on utilise des descentes de gradient:
  - **gradient descent**
  - stochastic gradient descent
  - mini-batch stochastic gradient descent

Learning Problem

- *L'entraînement* du modèle $\hat{f}(\cdots;\boldsymbol{\Theta})$ consiste à déterminer des paramètres $\boldsymbol{\Theta}$ qui minimisent les fonction de coût

$$\ell(\ldots;\boldsymbol{\Theta}) \text{ ou } \mathcal{L}(\ldots;\boldsymbol{\Theta}).$$

- Pour minimiser la fonction de coût, on utilise des descentes de gradient:
  - gradient descent
  - stochastic gradient descent
  - mini-batch stochastic gradient descent

Learning Problem

- *L'entraînement* du modèle $\hat{f}(\cdots;\mathbf{\Theta})$ consiste à déterminer des paramètres $\mathbf{\Theta}$ qui minimisent les fonction de coût

$$\ell(\ldots;\mathbf{\Theta}) \text{ ou } \mathcal{L}(\ldots;\mathbf{\Theta}).$$

- Pour minimiser la fonction de coût, on utilise des descentes de gradient:
  - **gradient descent**
  - stochastic gradient descent
  - mini-batch stochastic gradient descent

Learning Problem

- *L'entraînement* du modèle $\hat{f}(\cdots;\boldsymbol{\Theta})$ consiste à déterminer des paramètres $\boldsymbol{\Theta}$ qui minimisent les fonction de coût

$$\ell(\ldots;\boldsymbol{\Theta}) \text{ ou } \mathcal{L}(\ldots;\boldsymbol{\Theta}).$$

- Pour minimiser la fonction de coût, on utilise des descentes de gradient:

  - **gradient descent**
  - **stochastic gradient descent**
  - **mini-batch stochastic gradient descent**

Learning Problem

- *L'entraînement* du modèle $\hat{f}(\cdots;\mathbf{\Theta})$ consiste à déterminer des paramètres $\mathbf{\Theta}$ qui minimisent les fonction de coût

$$\ell(\ldots;\mathbf{\Theta}) \text{ ou } \mathcal{L}(\ldots;\mathbf{\Theta}).$$

- Pour minimiser la fonction de coût, on utilise des descentes de gradient:

- **gradient descent**
- **stochastic gradient descent**
- **mini-batch stochastic gradient descent**

Learning Problem

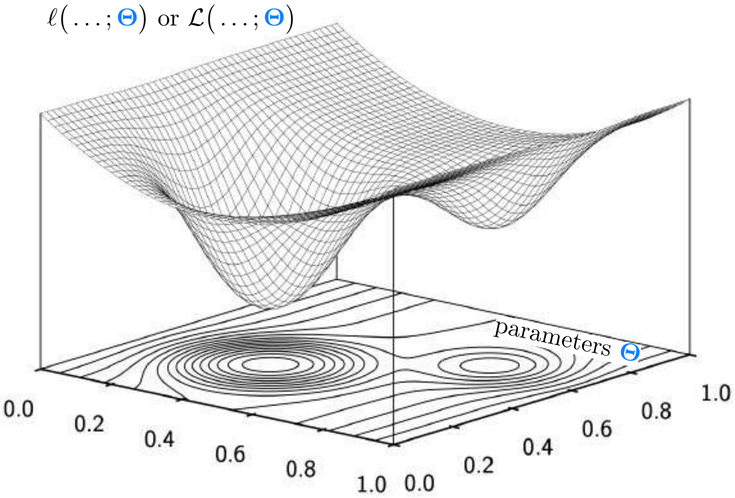


Figure adapted from [Fleuret, 2022]

Learning Problem

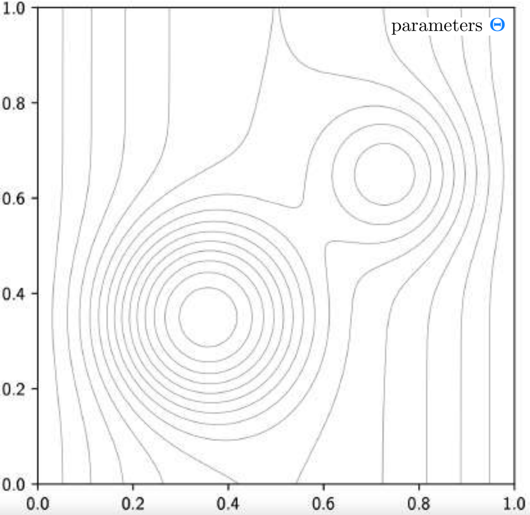


Figure adapted from [Fleuret, 2022]

Mean Squared Error (MSE)

Problème de régression

- Soit le training set

$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \ldots, N \right\}$$

- On considère comme fonction de coût à minimiser l'**erreur quadratique moyenne (mean squared error, MSE)**.
- C'est la moyenne des distances au carrés entre prédictions et réalités.

Mean Squared Error (MSE)

Problème de régression

- Soit le training set

$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \ldots, N \right\}$$

- On considère comme fonction de coût à minimiser l'**erreur quadratique moyenne (mean squared error, MSE)**.
- C'est la moyenne des distances au carrés entre prédictions et réalités.

Mean Squared Error (MSE)

Problème de régression

- Soit le training set

$$S = \left\{ (\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \ldots, N \right\}$$

- On considère comme fonction de coût à minimiser l'**erreur quadratique moyenne (mean squared error, MSE)**.
- C'est la moyenne des distances au carrés entre prédictions et réalités.

Mean Squared Error (MSE)

- Erreur individuelle:

$$\begin{aligned} \ell : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{\Theta} &\longmapsto \ell\big(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\big) \\ &= \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \left\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\right\|_2^2 \end{aligned}$$

- Erreur collective:

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{\Theta} &\longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{1}}, \ldots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big) \\ &= \frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \left\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\right\|_2^2 \end{aligned}$$

- La minimisation de cette loss s'effectue généralement par une descente de gradient (cf. chapitre suivant).

Mean Squared Error (MSE)

- Erreur individuelle:

$$
\begin{aligned}
\ell : \mathbb{R}^{|\boldsymbol{\Theta}|} & \longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} & \longmapsto \ell\big(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\big) \\
& \qquad = \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \Big\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\Big\|_2^2
\end{aligned}
$$

- Erreur collective:

$$
\begin{aligned}
\mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} & \longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} & \longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \dots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{1}}, \dots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big) \\
& \qquad = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \Big\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\Big\|_2^2
\end{aligned}
$$

- La minimisation de cette loss s'effectue généralement par une descente de gradient (cf. chapitre suivant).

Mean Squared Error (MSE)

- Erreur individuelle:

$$\begin{aligned} \ell : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{\Theta} &\longmapsto \ell\big(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\big) \\ &= \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \left\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\right\|_2^2 \end{aligned}$$

- Erreur collective:

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{\Theta} &\longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\boldsymbol{1}}, \ldots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big) \\ &= \frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \left\|\hat{f}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta}) - \boldsymbol{y}_{\boldsymbol{i}}\right\|_2^2 \end{aligned}$$

- La minimisation de cette loss s'effectue généralement par une descente de gradient (cf. chapitre suivant).

Categorical Cross Entropy (CCE)

Problème de classification

- Soit le training set

$$S = \Big\{ (\boldsymbol{x_i}, y_i) \in \mathbb{R}^{d_1} \times \{1, \ldots, C\} : i = 1, \ldots, N \Big\}$$

- Pour les y_i, on utilise le 1-hot encoding:

$$y_i = k \quad \longmapsto \quad \boldsymbol{y_i} = \underbrace{(0 \cdots 1 \cdots 0)}_{k\text{-th comp} = 1}$$

Categorical Cross Entropy (CCE)

Problème de classification

- Soit le training set

$$S = \Big\{ (\boldsymbol{x_i}, y_i) \in \mathbb{R}^{d_1} \times \{1, \ldots, C\} : i = 1, \ldots, N \Big\}$$

- Pour les y_i, on utilise le 1-hot encoding:

$$y_i = k \quad \longmapsto \quad \boldsymbol{y_i} = \underbrace{(0 \cdots 1 \cdots 0)}_{k\text{-th comp } = 1}$$

Categorical Cross Entropy (CCE)

- Dans un contexte de classification, l'erreur quadratique n'est pas appropriée.
- En effet, considérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{array}{rcl} \hat{\boldsymbol{y}} & = & (-1\ \ -1\ \ 2) \\ \hat{\boldsymbol{y}}' & = & (1\ 1\ 0) \end{array}$$

- En utilisant l'erreur quadratique, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 = 3 = \|\hat{\boldsymbol{y}}' - \boldsymbol{y}\|^2 = \ell(\hat{\boldsymbol{y}}', \boldsymbol{y})$$

alors que $\hat{\boldsymbol{y}}$ est une prédiction juste et $\hat{\boldsymbol{y}}'$ une prédiction fausse de $\boldsymbol{y}$.

Categorical Cross Entropy (CCE)

- Dans un contexte de classification, l'erreur quadratique n'est pas appropriée.
- En effet, considérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{aligned} \hat{\boldsymbol{y}} &= (-1\ \ -1\ \ 2) \\ \hat{\boldsymbol{y}}' &= (1\ 1\ 0) \end{aligned}$$

- En utilisant l'erreur quadratique, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 = 3 = \|\hat{\boldsymbol{y}}' - \boldsymbol{y}\|^2 = \ell(\hat{\boldsymbol{y}}', \boldsymbol{y})$$

alors que $\hat{\boldsymbol{y}}$ est une prédiction juste et $\hat{\boldsymbol{y}}'$ une prédiction fausse de $\boldsymbol{y}$.

Categorical Cross Entropy (CCE)

- Dans un contexte de classification, l'erreur quadratique n'est pas appropriée.
- En effet, considérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{array}{rcl} \hat{\boldsymbol{y}} & = & (-1\ -1\ \ 2) \\ \hat{\boldsymbol{y}}' & = & (1\ 1\ 0) \end{array}$$

- En utilisant l'erreur quadratique, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|^2 = 3 = \|\hat{\boldsymbol{y}}' - \boldsymbol{y}\|^2 = \ell(\hat{\boldsymbol{y}}', \boldsymbol{y})$$

alors que $\hat{\boldsymbol{y}}$ est une prédiction juste et $\hat{\boldsymbol{y}}'$ une prédiction fausse de $\boldsymbol{y}$.

Categorical Cross Entropy (CCE)

- Soient p, q deux distributions. L'**entropie croisée (categorical cross entropy)** de p et q est donnée par

$$\mathbb{H}(p,q) = -\mathrm{E}_p\left[\log(q)\right] = -\sum_k p(k)\log(q(k))$$

- On considère comme fonction de coût à minimiser l'**l'entropie croisée (categorical cross entropy)** entre réalité(s) et prédiction(s) (cf. régression logistique).

Categorical Cross Entropy (CCE)

- Soient p, q deux distributions. L'**entropie croisée (categorical cross entropy)** de p et q est donnée par

$$\mathbb{H}(p,q) = -\mathrm{E}_p\left[\log(q)\right] = -\sum_k p(k)\log(q(k))$$

- On considère comme fonction de coût à minimiser l'**l'entropie croisée (categorical cross entropy)** entre réalité(s) et prédiction(s) (cf. régression logistique).

Categorical Cross Entropy (CCE)

- Erreur individuelle:

$$
\begin{aligned}
\ell : \mathbb{R}^{|\boldsymbol{\Theta}|} \quad &\longrightarrow \quad \mathbb{R} \\
\boldsymbol{\Theta} \quad &\longmapsto \quad \ell\big(\hat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{i}}; \boldsymbol{\Theta}\big) \\
&= \mathbb{H}\left(\boldsymbol{y}_{\boldsymbol{i}}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}\right) = -\sum_{k=1}^{C} \boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}} \log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}\right) \\
&= -\log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{c_i}}\right) = -\log\left(\hat{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta})_{\boldsymbol{c_i}}\right)
\end{aligned}
$$

où $c_i \in \{1, \ldots, C\}$ est la classe de $\boldsymbol{y_i}$ (en fait $c_i = y_i$).

Categorical Cross Entropy (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{aligned} \hat{\boldsymbol{y}} &= (-1\ \ -1\ \ 2) \\ \hat{\boldsymbol{y}}' &= (1\ 1\ 0) \end{aligned}$$

- En utilisant l'entropie croisée, on a:

$$\begin{aligned} \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) &= \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\log(2) = -0.693\cdots \\ \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}') &= \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}') = -\log(0) = +\infty \end{aligned}$$

- Avec cette loss, $\hat{y}$ est une bien meilleure prédiction que $\hat{y}'$...

Categorical Cross Entropy (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{array}{rcl} \hat{\boldsymbol{y}} & = & (-1\ \ -1\ \ 2) \\ \hat{\boldsymbol{y}}' & = & (1\ 1\ 0) \end{array}$$

- En utilisant l'entropie croisée, on a:

$$\begin{array}{rcl} \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) & = & \mathbb{H}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = -\log(2) = -0.693\cdots \\ \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}') & = & \mathbb{H}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}'\right) = -\log(0) = +\infty \end{array}$$

- Avec cette loss, $\hat{y}$ est une bien meilleure prédiction que $\hat{y}'$...

Categorical Cross Entropy (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

$$y = 3 \overset{\text{1-hot}}{\Leftrightarrow} \boldsymbol{y} = (0\ 0\ 1) \quad \text{et} \quad \begin{aligned} \hat{\boldsymbol{y}} &= (-1\ -1\ \ 2) \\ \hat{\boldsymbol{y}}' &= (1\ 1\ 0) \end{aligned}$$

- En utilisant l'entropie croisée, on a:

$$\begin{aligned} \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) &= \mathbb{H}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = -\log(2) = -0.693\cdots \\ \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}') &= \mathbb{H}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}'\right) = -\log(0) = +\infty \end{aligned}$$

- Avec cette loss, $\hat{\boldsymbol{y}}$ est une bien meilleure prédiction que $\hat{\boldsymbol{y}}'$...

Categorical Cross Entropy (CCE)

- Erreur collective:

$$
\begin{aligned}
\mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\
\boldsymbol{\Theta} &\longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\mathbf{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\mathbf{1}}, \ldots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big) \\
&= \frac{1}{N} \sum_{i=1}^{N} \mathbb{H}\left(\boldsymbol{y}_{\boldsymbol{i}}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}\right) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}} \log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}\right) \\
&= -\frac{1}{N} \sum_{i=1}^{N} \log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{c_i}}\right) = -\frac{1}{N} \sum_{i=1}^{N} \log\left(\hat{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta})_{\boldsymbol{c_i}}\right)
\end{aligned}
$$

où $c_i \in \{1, \ldots, C\}$ est la classe de $\boldsymbol{y}_{\boldsymbol{i}}$ (en fait $c_i = y_i$).

- La minimisation de cette loss s'effectue généralement par une descente de gradient (cf. chapitre suivant).

Categorical Cross Entropy (CCE)

- Erreur collective:

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{|\boldsymbol{\Theta}|} &\longrightarrow \mathbb{R} \\ \boldsymbol{\Theta} &\longmapsto \mathcal{L}\big(\hat{\boldsymbol{y}}_{\mathbf{1}}, \dots, \hat{\boldsymbol{y}}_{\boldsymbol{N}}, \boldsymbol{y}_{\mathbf{1}}, \dots, \boldsymbol{y}_{\boldsymbol{N}}; \boldsymbol{\Theta}\big) \\ &= \frac{1}{N} \sum_{i=1}^{N} \mathbb{H}\left(\boldsymbol{y}_{\boldsymbol{i}}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}\right) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \boldsymbol{y}_{\boldsymbol{i},\boldsymbol{k}} \log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{k}}\right) \\ &= -\frac{1}{N} \sum_{i=1}^{N} \log\left(\hat{\boldsymbol{y}}_{\boldsymbol{i},\boldsymbol{c_i}}\right) = -\frac{1}{N} \sum_{i=1}^{N} \log\left(\hat{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{i}}; \boldsymbol{\Theta})_{\boldsymbol{c_i}}\right) \end{aligned}$$

  où $c_i \in \{1, \dots, C\}$ est la classe de $\boldsymbol{y}_{\boldsymbol{i}}$ (en fait $c_i = y_i$).

- La minimisation de cette loss s'effectue généralement par une descente de gradient (cf. chapitre suivant).

Bibliographie

Fleuret, F. (2022).
Deep Learning Course.

Wikipedia contributors (2022).
Cross entropy — Wikipedia, the free encyclopedia.