FONCTIONS DE CÔUT Loss Functions

Jérémie Cabessa Laboratoire DAVID, UVSQ

Learning Problem

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

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

L'entraînement du modèle $\hat{f}(\cdot; \Theta)$ consiste à déterminer les paramètres Θ qui minimisent l'erreur (ou la loss) entre les *prédictions* $\hat{y}_i = \hat{f}(x_i; \Theta)$ et les *réalités* 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 $\hat{f}(\cdot; \Theta)$ un **modèle** qui dépend de *paramètres* Θ :

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

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

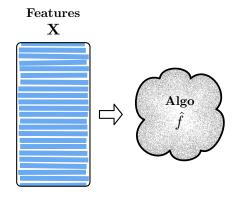
$$egin{aligned} \hat{f}(\cdot;oldsymbol{\Theta}): \mathbb{R}^{d_1} &\longrightarrow & \mathbb{R}^{d_2} \ oldsymbol{x} &\longmapsto & \hat{oldsymbol{y}}:=\hat{f}(oldsymbol{x};oldsymbol{\Theta}) \end{aligned}$$

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

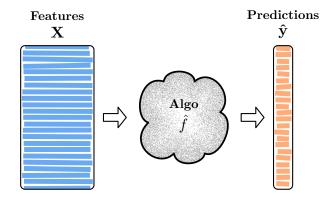
Features

 \mathbf{X}

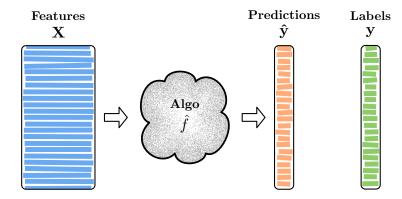




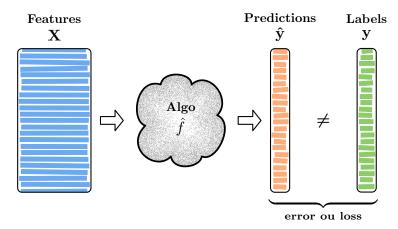
LEARNING PROBLEM



LEARNING PROBLEM

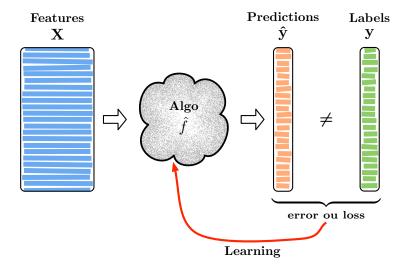


LEARNING PROBLEM



000000

LEARNING PROBLEM



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

$$egin{array}{lll} \ell: \mathbb{R}^{d_2} imes \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ (\hat{m{y_i}}, m{y_i}) & \longmapsto & \ell\left(\hat{m{y_i}}, m{y_i}
ight) \end{array}$$

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

$$egin{array}{lll} \mathcal{L}: \mathbb{R}^{d_2} imes \cdots & \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ \hat{m{y}}_1, \ldots, \hat{m{y}}_{m{N}}, m{y_1} \ldots, m{y_N}) & \longmapsto & \mathcal{L}\left(\hat{m{y}}_1, \ldots, \hat{m{y}}_{m{N}}, m{y_i} \ldots, m{y_N}
ight) \end{array}$$

► En général, la loss globale est la moyenne des loss individuelles

$$\mathcal{L}\left(\hat{\boldsymbol{y}}_{1},\ldots,\hat{\boldsymbol{y}}_{N},\boldsymbol{y}_{1}\ldots,\boldsymbol{y}_{N}\right) = \frac{1}{N}\sum_{i=1}^{N}\ell\left(\hat{\boldsymbol{y}}_{i},\boldsymbol{y}_{i}\right)$$

0000000

Learning Problem

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

$$egin{array}{lll} \ell: \mathbb{R}^{d_2} imes \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ (\hat{m{y_i}}, m{y_i}) & \longmapsto & \ell\left(\hat{m{y_i}}, m{y_i}
ight) \end{array}$$

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

$$egin{array}{lll} \mathcal{L}: \mathbb{R}^{d_2} imes \cdots & \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ (\hat{m{y_1}}, \ldots, \hat{m{y_N}}, m{y_1}, \ldots, m{y_N}) & \longmapsto & \mathcal{L}\left(\hat{m{y_1}}, \ldots, \hat{m{y_N}}, m{y_i}, \ldots, m{y_N}
ight) \end{array}$$

$$\mathcal{L}\left(\hat{\boldsymbol{y}}_{1},\ldots,\hat{\boldsymbol{y}}_{N},\boldsymbol{y}_{1}\ldots,\boldsymbol{y}_{N}\right) = \frac{1}{N}\sum_{i=1}^{N}\ell\left(\hat{\boldsymbol{y}}_{i},\boldsymbol{y}_{i}\right)$$

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

$$egin{array}{lll} \ell: \mathbb{R}^{d_2} imes \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ (\hat{m{y}}_{m{i}}, m{y}_{m{i}}) & \longmapsto & \ell\left(\hat{m{y}}_{m{i}}, m{y}_{m{i}}
ight) \end{array}$$

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

$$egin{array}{lll} \mathcal{L}: \mathbb{R}^{d_2} imes \cdots & \mathbb{R}^{d_2} & \longrightarrow & \mathbb{R} \ (\hat{m{y_1}}, \ldots, \hat{m{y_N}}, m{y_1} \ldots, m{y_N}) & \longmapsto & \mathcal{L}\left(\hat{m{y_1}}, \ldots, \hat{m{y_N}}, m{y_i} \ldots, m{y_N}
ight) \end{array}$$

► En général, la loss globale est la moyenne des loss individuelles

$$\mathcal{L}\left(\hat{\boldsymbol{y}}_{1},\ldots,\hat{\boldsymbol{y}}_{N},\boldsymbol{y}_{1}\ldots,\boldsymbol{y}_{N}\right) = \frac{1}{N}\sum_{i=1}^{N}\ell\left(\hat{\boldsymbol{y}}_{i},\boldsymbol{y}_{i}\right)$$

LEARNING PROBLEM

- Pour différents paramètres Θ , on aura différentes prédictions $\hat{y}_i = \hat{f}(x_i; \Theta)$, et donc différentes erreurs $\ell(\ldots)$ et $\mathcal{L}(\ldots)$.
- Ainsi, ℓ et L sont aussi des fonctions des paramètres Θ:

$$egin{array}{lll} \ell: \mathbb{R}^{|\Theta|} & \longrightarrow & \mathbb{R} \ & oldsymbol{\Theta} & \longmapsto & \ellig(\hat{oldsymbol{y}}_i, oldsymbol{y}_i; oldsymbol{\Theta}ig) \ & \mathcal{L}: \mathbb{R}^{|\Theta|} & \longrightarrow & \mathbb{R} \ & oldsymbol{\Theta} & \longmapsto & \mathcal{L}ig(\hat{oldsymbol{y}}_1, \ldots, \hat{oldsymbol{y}}_N, oldsymbol{y}_1, \ldots, oldsymbol{y}_N; oldsymbol{\Theta}ig) \end{array}$$

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

LEARNING PROBLEM

- Pour différents paramètres Θ , on aura différentes prédictions $\hat{y}_i = \hat{f}(x_i; \Theta)$, et donc différentes erreurs $\ell(\ldots)$ et $\mathcal{L}(\ldots)$.
- ▶ Ainsi, ℓ et \mathcal{L} sont aussi des fonctions des paramètres Θ :

$$egin{array}{lll} \ell: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \ellig(\hat{y}_i, y_i; oldsymbol{\Theta}ig) \ \mathcal{L}: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \mathcal{L}ig(\hat{y}_1, \ldots, \hat{y}_N, y_1, \ldots, y_N; oldsymbol{\Theta}ig) \end{array}$$

où $|\Theta|$ est le nombre de paramètres Θ .

CCE

LEARNING PROBLEM

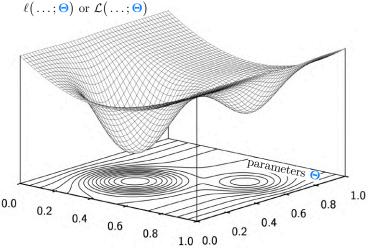


Figure adapted from [Fleuret, 2022]

CCE

LEARNING PROBLEM

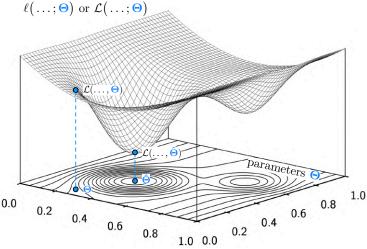


Figure adapted from [Fleuret, 2022]

LEARNING PROBLEM

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

$$\ell(\ldots; \boldsymbol{\Theta})$$
 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 0000000

> $ightharpoonup L'entraînement du modèle <math>\hat{f}(\cdots; \Theta)$ consiste à déterminer des paramètres Θ qui minimisent les fonction de coût

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

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

LEARNING PROBLEM 0000000

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

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

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

LEARNING PROBLEM

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

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

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

LEARNING PROBLEM

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

$$\ell(\ldots; \boldsymbol{\Theta})$$
 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

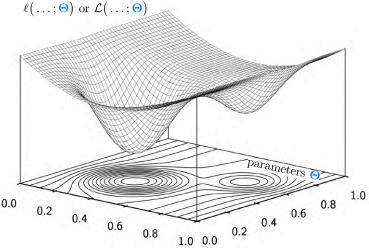


Figure adapted from [Fleuret, 2022]

LEARNING PROBLEM

000000

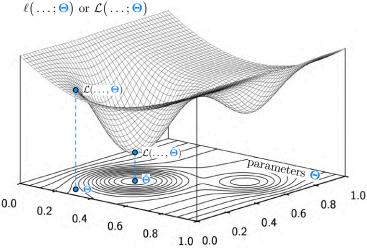
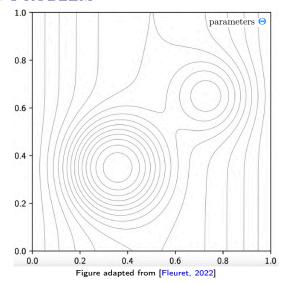
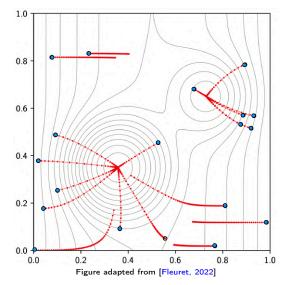


Figure adapted from [Fleuret, 2022]



LEARNING PROBLEM



Problème de régression

Soit le training set

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

LEARNING PROBLEM

Problème de régression

► Soit le training set

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

- 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.

Problème de régression

► Soit le training set

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

- 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.

Erreur individuelle:

$$egin{array}{lll} \ell: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \ellig(\hat{oldsymbol{y_i}}, oldsymbol{y_i}; oldsymbol{\Theta}ig) \ & = \|\hat{oldsymbol{y_i}} - oldsymbol{y_i}\|_2^2 = \left\|\hat{f}(oldsymbol{x_i}; oldsymbol{\Theta}) - oldsymbol{y_i}
ight\|_2^2 \end{array}$$

► Erreur collective:

$$egin{array}{lll} \mathcal{L}: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \mathcal{L}ig(\hat{oldsymbol{y}}_{1}, \ldots, \hat{oldsymbol{y}}_{N}, oldsymbol{y}_{1}, \ldots, oldsymbol{y}_{N}; oldsymbol{\Theta}ig) \ & = rac{1}{N} \sum_{i=1}^{N} \|\hat{oldsymbol{y}}_{i} - oldsymbol{y}_{i}\|_{2}^{2} = rac{1}{N} \sum_{i=1}^{N} \left\|\hat{f}(oldsymbol{x}_{i}; oldsymbol{\Theta}) - oldsymbol{y}_{i}
ight\|_{2}^{2} \end{array}$$

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

Erreur individuelle:

$$egin{array}{lll} \ell: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \ellig(\hat{oldsymbol{y_i}}, oldsymbol{y_i}; oldsymbol{\Theta}ig) \ & = \|\hat{oldsymbol{y_i}} - oldsymbol{y_i}\|_2^2 = \left\|\hat{f}(oldsymbol{x_i}; oldsymbol{\Theta}) - oldsymbol{y_i}
ight\|_2^2 \end{array}$$

Erreur collective:

$$\mathcal{L}: \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\boldsymbol{\Theta} \longmapsto \mathcal{L}(\hat{\boldsymbol{y}}_{1}, \dots, \hat{\boldsymbol{y}}_{N}, \boldsymbol{y}_{1}, \dots, \boldsymbol{y}_{N}; \boldsymbol{\Theta})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_{i} - \boldsymbol{y}_{i}\|_{2}^{2} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{f}(\boldsymbol{x}_{i}; \boldsymbol{\Theta}) - \boldsymbol{y}_{i}\|_{2}^{2}$$

Learning Problem

Erreur individuelle:

Learning Problem

$$egin{array}{lll} \ell: \mathbb{R}^{|oldsymbol{\Theta}|} & \longrightarrow & \mathbb{R} \ oldsymbol{\Theta} & \longmapsto & \ellig(\hat{oldsymbol{y_i}}, oldsymbol{y_i}; oldsymbol{\Theta}ig) \ & = \|\hat{oldsymbol{y_i}} - oldsymbol{y_i}\|_2^2 = \left\|\hat{f}(oldsymbol{x_i}; oldsymbol{\Theta}) - oldsymbol{y_i}
ight\|_2^2 \end{array}$$

Erreur collective:

$$\mathcal{L}: \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\boldsymbol{\Theta} \longmapsto \mathcal{L}(\hat{\boldsymbol{y}}_{1}, \dots, \hat{\boldsymbol{y}}_{N}, \boldsymbol{y}_{1}, \dots, \boldsymbol{y}_{N}; \boldsymbol{\Theta})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{y}}_{i} - \boldsymbol{y}_{i}\|_{2}^{2} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{f}(\boldsymbol{x}_{i}; \boldsymbol{\Theta}) - \boldsymbol{y}_{i}\|_{2}^{2}$$

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

Problème de classification

► Soit le training set

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

CCE ●00000

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

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

CCE

CATEGORICAL CROSS ENTROPY (CCE)

Problème de classification

Soit le training set

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

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

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

- Dans un contexte de classification, l'erreur quadratique n'est pas appropriée.

$$y = 3 \stackrel{\text{1-hot}}{\Leftrightarrow} y = (0\ 0\ 1) \text{ et } \hat{y} = (-1\ -1\ 2)$$

 $\hat{y}' = (1\ 1\ 0)$

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

LEARNING PROBLEM

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

$$y = 3 \stackrel{\text{1-hot}}{\Leftrightarrow} y = (0\ 0\ 1) \text{ et } \hat{y} = (-1\ -1\ 2)$$
 $\hat{y}' = (1\ 1\ 0)$

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

LEARNING PROBLEM

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

$$y = 3 \stackrel{\text{1-hot}}{\Leftrightarrow} y = (0\ 0\ 1) \text{ et } \hat{y} = (-1\ -1\ 2)$$
 $\hat{y}' = (1\ 1\ 0)$

En utilisant l'erreur quadratique, on a:

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

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

CCE 00•000

CATEGORICAL CROSS ENTROPY (CCE)

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

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

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

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

CCE 00•000

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

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

Erreur individuelle:

$$egin{aligned} \ell: \mathbb{R}^{|m{\Theta}|} & \longrightarrow & \mathbb{R} \ m{\Theta} & \longmapsto & \ellig(\hat{m{y}}_{m{i}}, m{y}_{m{i}}; m{\Theta}ig) \ & = \mathbb{H}\left(m{y}_{m{i}}, \hat{m{y}}_{m{i}}
ight) = -\sum_{k=1}^C m{y}_{m{i}_k} \log\left(\hat{m{y}}_{m{i}_k}
ight) \ & = -\log\left(\hat{m{y}}_{m{i}_{c_i}}
ight) = -\log\left(\hat{m{f}}(m{x}_{m{i}}; m{\Theta})_{c_i}
ight) \end{aligned}$$

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

Learning Problem

► Reconsidérons la réalité et les deux prédictions suivantes:

$$y=3 \stackrel{\text{1-hot}}{\Leftrightarrow} y=(0\ 0\ 1) \quad \text{et} \quad \hat{y}=(-1\ -1\ 2)$$
 $\hat{y}'=(1\ 1\ 0)$

En utilisant l'entropie croisée, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\log(2) = -0.693\cdots$$

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y'}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}') = -\log(0) = +\infty$$

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

► Reconsidérons la réalité et les deux prédictions suivantes:

$$y = 3 \stackrel{\text{1-hot}}{\Leftrightarrow} y = (0 \ 0 \ 1) \text{ et } \hat{y} = (-1 \ -1 \ 2)$$

 $\hat{y}' = (1 \ 1 \ 0)$

CCE

En utilisant l'entropie croisée, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\log(2) = -0.693\cdots$$

 $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y'}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}'}) = -\log(0) = +\infty$

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

▶ Reconsidérons la réalité et les deux prédictions suivantes:

$$y = 3 \stackrel{\text{1-hot}}{\Leftrightarrow} y = (0 \ 0 \ 1) \text{ et } \hat{y} = (-1 \ -1 \ 2)$$

 $\hat{y}' = (1 \ 1 \ 0)$

En utilisant l'entropie croisée, on a:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\log(2) = -0.693\cdots$$

 $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y'}) = \mathbb{H}(\boldsymbol{y}, \hat{\boldsymbol{y}'}) = -\log(0) = +\infty$

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

Erreur collective:

$$\mathcal{L}: \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\mathbf{\Theta} \longmapsto \mathcal{L}(\hat{\mathbf{y}}_{1}, \dots, \hat{\mathbf{y}}_{N}, \mathbf{y}_{1}, \dots, \mathbf{y}_{N}; \mathbf{\Theta})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{H}(\mathbf{y}_{i}, \hat{\mathbf{y}}_{i}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \mathbf{y}_{i_{k}} \log(\hat{\mathbf{y}}_{i_{k}})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \log(\hat{\mathbf{y}}_{i_{c_{i}}}) = -\frac{1}{N} \sum_{i=1}^{N} \log(\hat{\mathbf{f}}(\mathbf{x}_{i}; \mathbf{\Theta})_{c_{i}})$$

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

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

Erreur collective:

$$\mathcal{L}: \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\mathbf{\Theta} \longmapsto \mathcal{L}(\hat{\mathbf{y}}_{1}, \dots, \hat{\mathbf{y}}_{N}, \mathbf{y}_{1}, \dots, \mathbf{y}_{N}; \mathbf{\Theta})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{H}(\mathbf{y}_{i}, \hat{\mathbf{y}}_{i}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \mathbf{y}_{i_{k}} \log(\hat{\mathbf{y}}_{i_{k}})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \log(\hat{\mathbf{y}}_{i_{c_{i}}}) = -\frac{1}{N} \sum_{i=1}^{N} \log(\hat{\mathbf{f}}(\mathbf{x}_{i}; \mathbf{\Theta})_{c_{i}})$$

où $c_i \in \{1, \ldots, C\}$ est la classe de y_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).

 ${\sf Cross\ entropy--Wikipedia},\ {\sf the\ free\ encyclopedia}.$