

# FONCTIONS DE CÔUT LOSS FUNCTIONS

Jérémie Cabessa  
Laboratoire DAVID, UVSQ

# LEARNING PROBLEM

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

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

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

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

# LEARNING PROBLEM

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

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

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

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

# LEARNING PROBLEM

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

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

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

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

# LEARNING PROBLEM

**Features**

**X**

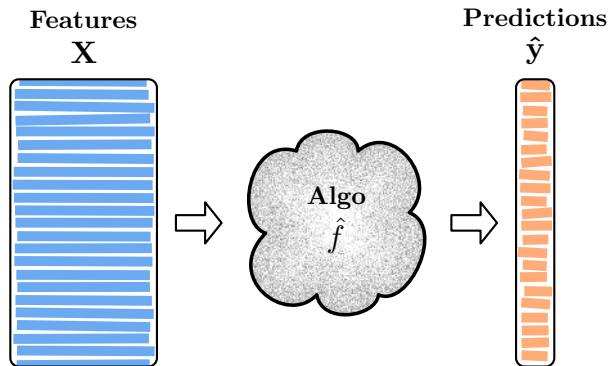


# LEARNING PROBLEM

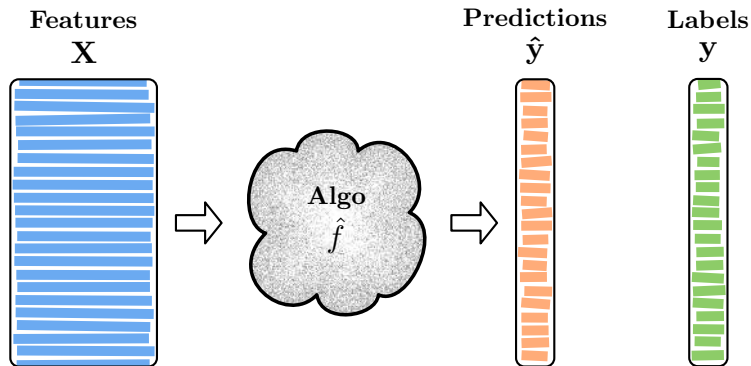
Features  
 $X$



# LEARNING PROBLEM

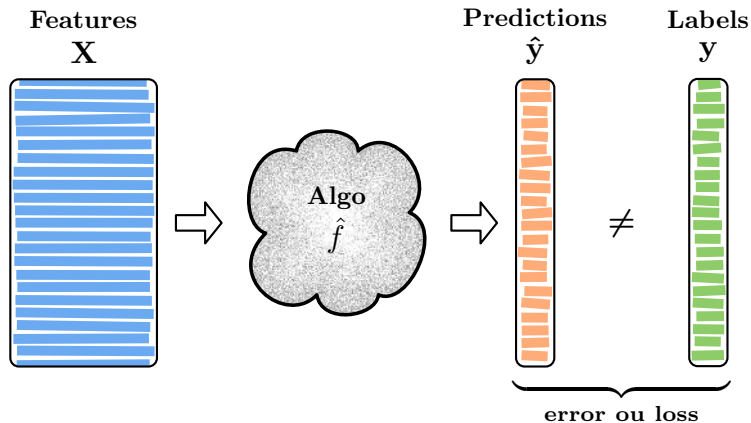


## LEARNING PROBLEM

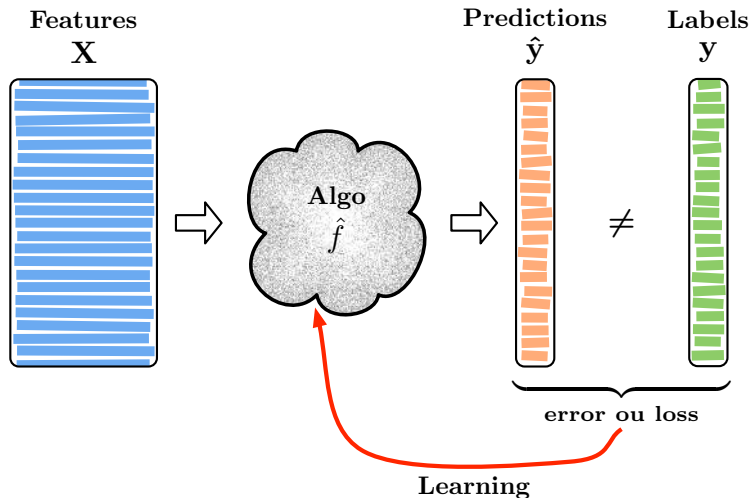




## LEARNING PROBLEM



## LEARNING PROBLEM



# LEARNING PROBLEM

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

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

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

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

# LEARNING PROBLEM

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

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

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

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

# LEARNING PROBLEM

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

$$\ell : \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\Theta \longmapsto \ell(\hat{y}_i, y_i; \Theta)$$

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

$$\Theta \longmapsto \mathcal{L}(\hat{y}_1, \dots, \hat{y}_N, y_1, \dots, y_N; \Theta)$$

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

# LEARNING PROBLEM

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

$$\ell : \mathbb{R}^{|\Theta|} \longrightarrow \mathbb{R}$$

$$\Theta \longmapsto \ell(\hat{y}_i, y_i; \Theta)$$

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

$$\Theta \longmapsto \mathcal{L}(\hat{y}_1, \dots, \hat{y}_N, y_1, \dots, y_N; \Theta)$$

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

# LEARNING PROBLEM

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

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

- ▶ 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  $\hat{f}(\dots; \Theta)$  consiste à déterminer des paramètres  $\Theta$  qui minimisent les fonction de coût

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

- ▶ 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  $\hat{f}(\dots; \Theta)$  consiste à déterminer des paramètres  $\Theta$  qui minimisent les fonction de coût

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

- ▶ 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  $\hat{f}(\dots; \Theta)$  consiste à déterminer des paramètres  $\Theta$  qui minimisent les fonction de coût

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

- ▶ 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  $\hat{f}(\dots; \Theta)$  consiste à déterminer des paramètres  $\Theta$  qui minimisent les fonction de coût

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

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

# LEARNING PROBLEM

$$\ell(\dots; \Theta) \text{ or } \mathcal{L}(\dots; \Theta)$$

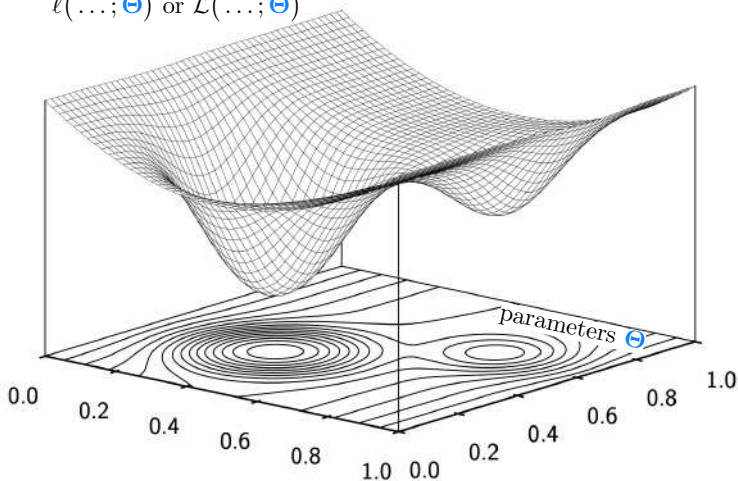


Figure adapted from [Fleuret, 2022]

# LEARNING PROBLEM

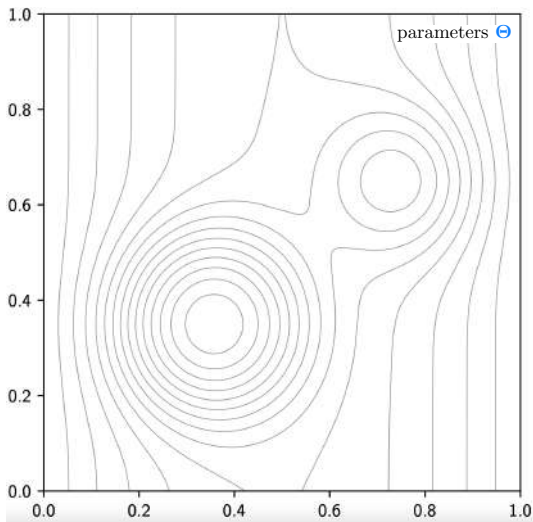


Figure adapted from [Fleuret, 2022]

# MEAN SQUARED ERROR (MSE)

## Problème de régression

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

- 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.

# MEAN SQUARED ERROR (MSE)

## Problème de régression

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

- 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.

# MEAN SQUARED ERROR (MSE)

## Problème de régression

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : i = 1, \dots, N \right\}$$

- 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.



# MEAN SQUARED ERROR (MSE)

► Erreur individuelle:

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

► Erreur collective:

$$\begin{aligned}\mathcal{L} : \mathbb{R}^{|\Theta|} &\longrightarrow \mathbb{R} \\ \Theta &\longmapsto \mathcal{L}(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N, \mathbf{y}_1, \dots, \mathbf{y}_N; \Theta) \\ &= \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \left\| \hat{f}(\mathbf{x}_i; \Theta) - \mathbf{y}_i \right\|_2^2\end{aligned}$$

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

# MEAN SQUARED ERROR (MSE)

► Erreur individuelle:

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

► Erreur collective:

$$\begin{aligned}\mathcal{L} : \mathbb{R}^{|\Theta|} &\longrightarrow \mathbb{R} \\ \Theta &\longmapsto \mathcal{L}(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N, \mathbf{y}_1, \dots, \mathbf{y}_N; \Theta) \\ &= \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \left\| \hat{f}(\mathbf{x}_i; \Theta) - \mathbf{y}_i \right\|_2^2\end{aligned}$$

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

# MEAN SQUARED ERROR (MSE)

- Erreur individuelle:

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

- Erreur collective:

$$\begin{aligned}\mathcal{L} : \mathbb{R}^{|\Theta|} &\longrightarrow \mathbb{R} \\ \Theta &\longmapsto \mathcal{L}(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N, \mathbf{y}_1, \dots, \mathbf{y}_N; \Theta) \\ &= \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \left\| \hat{f}(\mathbf{x}_i; \Theta) - \mathbf{y}_i \right\|_2^2\end{aligned}$$

- 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

$$S = \left\{ (\mathbf{x}_i, y_i) \in \mathbb{R}^{d_1} \times \{1, \dots, C\} : i = 1, \dots, N \right\}$$

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

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

# CATEGORICAL CROSS ENTROPY (CCE)

## Problème de classification

- Soit le training set

$$S = \left\{ (\mathbf{x}_i, y_i) \in \mathbb{R}^{d_1} \times \{1, \dots, C\} : i = 1, \dots, N \right\}$$

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

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

# 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:

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

- ▶ En utilisant l'erreur quadratique, on a:

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

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

# 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:

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

- ▶ En utilisant l'erreur quadratique, on a:

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

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

# 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:

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

- ▶ En utilisant l'erreur quadratique, on a:

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

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



# CATEGORICAL CROSS ENTROPY (CCE)

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

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

- On considère comme fonction de coût à minimiser 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

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

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

# CATEGORICAL CROSS ENTROPY (CCE)

► Erreur individuelle:

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

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

# CATEGORICAL CROSS ENTROPY (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

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

- En utilisant l'entropie croisée, on a:

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

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

# CATEGORICAL CROSS ENTROPY (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

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

- En utilisant l'entropie croisée, on a:

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = \mathbb{H}(\mathbf{y}, \hat{\mathbf{y}}) = -\log(2) = -0.693\dots \\ \ell(\hat{\mathbf{y}}, \mathbf{y}') = \mathbb{H}(\mathbf{y}, \hat{\mathbf{y}}') = -\log(0) = +\infty$$

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

# CATEGORICAL CROSS ENTROPY (CCE)

- Reconsidérons la réalité et les deux prédictions suivantes:

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

- En utilisant l'entropie croisée, on a:

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

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

# CATEGORICAL CROSS ENTROPY (CCE)

- Erreur collective:

$$\begin{aligned}\mathcal{L} : \mathbb{R}^{|\Theta|} &\longrightarrow \mathbb{R} \\ \Theta &\longmapsto \mathcal{L}(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N, \mathbf{y}_1, \dots, \mathbf{y}_N; \Theta) \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{H}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^C \mathbf{y}_{i,k} \log(\hat{\mathbf{y}}_{i,k}) \\ &= -\frac{1}{N} \sum_{i=1}^N \log(\hat{\mathbf{y}}_{i,c_i}) = -\frac{1}{N} \sum_{i=1}^N \log(\hat{f}(\mathbf{x}_i; \Theta)_{c_i})\end{aligned}$$

où  $c_i \in \{1, \dots, C\}$  est la classe de  $\mathbf{y}_i$  (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:

$$\begin{aligned}\mathcal{L} : \mathbb{R}^{|\Theta|} &\longrightarrow \mathbb{R} \\ \Theta &\longmapsto \mathcal{L}(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N, \mathbf{y}_1, \dots, \mathbf{y}_N; \Theta) \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{H}(\mathbf{y}_i, \hat{\mathbf{y}}_i) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^C \mathbf{y}_{i,k} \log(\hat{\mathbf{y}}_{i,k}) \\ &= -\frac{1}{N} \sum_{i=1}^N \log(\hat{\mathbf{y}}_{i,c_i}) = -\frac{1}{N} \sum_{i=1}^N \log(\hat{f}(\mathbf{x}_i; \Theta)_{c_i})\end{aligned}$$

où  $c_i \in \{1, \dots, C\}$  est la classe de  $\mathbf{y}_i$  (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).  
Cross entropy — Wikipedia, the free encyclopedia.