

# TP2A : Rudiments d'image, labélisation et entropie croisée

## 1 Introduction et représentation des images

Nous travaillerons avec des images extrêmement simples ( $2 \times 2$  pixels) pour introduire les concepts de base d'image, les sorties probabilistes (softmax) d'un réseau, et la fonction d'erreur d'entropie croisée catégorielle nécessaire à la classification.

Une image en nuances de gris peut être représentée comme une matrice de pixels où chaque pixel a une intensité comprise entre 0 (noir) et 255 (blanc). Par exemple :

$$\text{Image brute} = \begin{bmatrix} 0 & 255 \\ 127 & 127 \end{bmatrix}.$$

**Normalisation :** Pour faciliter le traitement par un réseau de neurones, les pixels sont normalisés entre 0 et 1 :

$$\text{Image normalisée} = \begin{bmatrix} 0 & 1 \\ 0.5 & 0.5 \end{bmatrix}.$$

**Aplatissement :** Avant d'envoyer l'image dans un réseau dense (fully connected), on la transforme en un vecteur unidimensionnel :

$$\text{Vecteur aplati} = [0, 1, 0.5, 0.5].$$

## 2 Objectif

Notre tâche est de reconnaître si une image contient une diagonale ou pas. Par exemple : - Image avec une diagonale sombre ( $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ ) : codage du label [1 0]. - Image sans diagonale ( $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ ) : codage du label [0 1].

**Architecture :** Le réseau dense ici aura nécessairement : - Une couche d'entrée de 4 neurones (pour les 4 pixels de l'image aplatie). - Une couche de sortie de 2 neurones, correspondant aux deux classes possibles (présence ou absence de diagonale). Mais aussi un certain nombre de couches intermédiaires.

**Scores et sortie :** - Le réseau génère des scores (c'est à dire les valeurs de sorties de chaque neurone de la dernière couche avant la fonction d'activation)  $z_1$  et  $z_2$  pour les deux classes. - Ces scores sont transformés en probabilités via la fonction softmax :

$$q_i = \frac{e^{z_i}}{\sum_j e^{z_j}}, \quad i \in \{1, 2\}.$$

**Pourquoi des probabilités ?** Les probabilités permettent de quantifier l'incertitude du réseau. Par exemple, si  $q = [0.7, 0.3]$ , le réseau "pense" que l'image contient une diagonale avec une confiance de 70%.

**Fonction d'erreur : Entropie croisée catégorielle** Pour évaluer la performance du réseau, on compare les sorties attendues  $p$  (par exemple,  $p = [1, 0]$ ) aux probabilités prédites  $q$ . L'entropie croisée catégorielle est définie comme :

$$\text{ECC}(p, q) = - \sum_i p_i \ln(q_i),$$

où  $p_i$  est la probabilité attendue et  $q_i$  la probabilité prédite.

**Pourquoi l'ECC ?** Cette fonction pénalise fortement les erreurs importantes. Par exemple : - Si  $q = [0.8, 0.2]$  et  $p = [1, 0]$ , l'erreur est faible ( $ECC = 0.22$ ). - Si  $q = [0.01, 0.99]$  et  $p = [1, 0]$ , l'erreur est beaucoup plus grande ( $ECC = 4.6$ ).

## 3 Génération et affichage d'images

### 3.1 Génération et affichage d'images $2 \times 2$ avec Python

Nous créons et manipulons des images très simples ( $2 \times 2$ ) en Python pour comprendre les concepts fondamentaux. Voici quelques exemples.

**Création et affichage d'une image simple :**

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import get_cmap

# Création d'une image 2x2
image = np.array([[0, 1],
                  [1, 0]])

# Affichage
cmp = get_cmap('gray')
plt.imshow(image, cmap=cmp, vmin=0, vmax=1)
plt.show()
```

**Transformation d'une image (aplatissement) :** Pour utiliser cette image dans un réseau dense, nous devons l'aplatir en un vecteur unidimensionnel.

```
# Aplatissement
image_flattened = image.flatten()
print(image_flattened) # Résultat : [0, 1, 1, 0]
```

**Ajout de bruit à une image :** Voici comment ajouter du bruit à une image pour tester la robustesse des modèles :

```
# Ajout de bruit
noise = np.random.normal(0, 0.1, image.shape)
image_noisy = np.clip(image + noise, 0, 1)

# Affichage côte à côte
plt.imshow(image_noisy, cmap=cmp, vmin=0, vmax=1)
plt.show()
```

## 4 Exercices

### 4.1 Exercice 1

**Rappel sur les scores :** les scores sont les valeurs produites par le réseau de neurones avant d'être transformées en probabilités avec la fonction softmax. Ces scores permettent au réseau de prioriser certaines classes en fonction de leur pertinence.

1. Crée une image en niveau de gris  $2 \times 2$  (8 bits par pixel) avec la matrice suivante :

$$\text{Image} = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}.$$

Aplatir la matrice en un vecteur :

Vecteur aplati =

2. Supposons que le réseau associe les scores suivants à cette image :

$$\hat{z} = [1.2, 0.5],$$

où  $\hat{x} = 1.2$  correspond à la classe "diagonale présente" et  $\hat{y} = 0.5$  à "diagonale absente". Calculer les probabilités associées avec la fonction softmax :

$$q_1 = \frac{e^{\hat{x}}}{e^{\hat{x}} + e^{\hat{y}}}, \quad q_2 = \frac{e^{\hat{y}}}{e^{\hat{x}} + e^{\hat{y}}}.$$

En déduire la classe prédite par le réseau.

—

#### 4.2 Exercice 2 : Entropie croisée catégorielle (ECC)

1. Reprendre les probabilités  $q = [q_1, q_2]$  calculées précédemment. Calculer l'erreur d'entropie croisée catégorielle :

$$\text{ECC}(p, q) = -(p_1 \ln q_1 + p_2 \ln q_2).$$

2. Interprète cette valeur d'erreur : est-elle faible ou élevée ?
- 

#### 4.3 Exercice 3 : Distinction de motifs similaires

Soit les motifs simples  $2 \times 2$  et ses labels :

- Image 1 :  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ , label  $p = [1, 0]$ .
- Image 2 :  $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ , label  $p = [0, 1]$ .

1. Suppose que les scores produits par le réseau sont :

$$\hat{z}_{\text{Image 1}} = [1.5, 0.2], \quad \hat{z}_{\text{Image 2}} = [0.3, 1.1].$$

2. Calcule les probabilités  $q$  pour chaque cas.
3. Utilise ECC pour comparer les probabilités aux labels attendus  $p$ .
4. Conclue sur les performances du réseau dans la distinction entre ces motifs.
- 

#### 4.4 Exercice 4 : Diagonale ou pas ?

Utiliser vos connaissances pour :

- Générer des données artificielles sous forme d'images  $2 \times 2$  avec des niveaux de gris (utiliser la section 3, ajouter par exemple du bruit à une première image purement diagonale par exemple).
- Trouver un moyen pour labelliser ces images en fonction de la présence ou non d'une diagonale distinguable.
- Fabriquer 100 images d'entraînement labélisées, puis 20 images de tests labélisées.
- Entraîner un réseau de neurones dense pour classer ces images (aidez vous du code de la partie 3 du TP2).