

INFO-F-311: Intelligence Artificielle

Projet 5: Réseaux de neurones

Pascal Tribel

Yannick Molinghen

Tom Lenaerts

1 Préambule

Ce projet vise à vous familiariser avec la notion de **réseaux de neurones** et leur utilisation. Il s'agit d'un *cas d'école*: la reconnaissance de chiffres écrits à la main, issus du jeu de données MNIST.

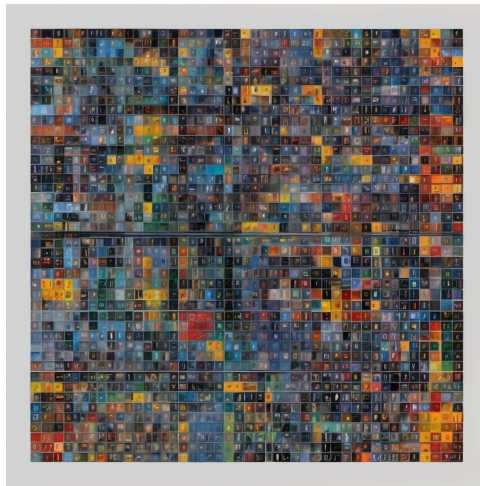


Figure 1: Un réseau de neurones reconnaissant les chiffres du MNIST comme une photo hyperréaliste.

Source: <https://stablediffusion.fr/xl>

2 Introduction

Le jeu de données <https://www.kaggle.com/datasets/oddrational/mnist-in-csv> contient deux fichiers *csv*:

- Un fichier d'entraînement, de 60000 éléments,
- Un fichier de test, de 10000 éléments.

Chaque ligne commence par le chiffre représenté, suivi de 28×28 nombres représentant chacun un pixel de l'image. Chaque pixel correspond à un niveau de gris représenté par un entier entre 0 et 255. Vous pouvez partir du principe que ces fichiers sont corrects et bien encodés.

3 Réseaux de neurones

Les réseaux de neurones composent une grande famille d'algorithmes de Machine Learning. Dans ce projet, vous vous concentrerez sur une classe élémentaire: les Multi-layer Perceptrons (MLP). Vous allez implémenter un réseau doté d'une seule couche intermédiaire. Ce réseau doit apprendre le lien existant entre les pixels d'une image et le nombre que cette image représente.

3.1 Entrées et sorties

L'entrée du réseau de neurones est un tableau numpy de dimension 784 (sous forme d'un vecteur, pas d'une matrice 28×28). La sortie est un autre tableau, de taille 10, où le i^e élément représente la probabilité que l'entrée représente le chiffre i . La méthode `load_mnist_data` doit permettre de récupérer les données d'entrée et de sortie stockées dans un fichier *csv* spécifié en paramètre. Veillez à normaliser

les pixels dans l'intervalle $[0, 1]$. Par exemple, la figure suivante représente le chiffre 7, et un réseau entraîné a effectivement reconnu ce chiffre.

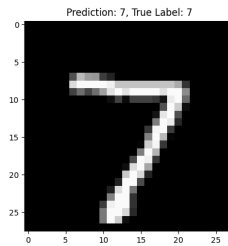


Figure 2: Un chiffre 7 du jeu MNIST, reconnu comme tel par un MLP entraîné

3.2 Algorithme d'apprentissage

L'apprentissage du réseau se fait en plusieurs étapes:

- Présenter une image au réseau, en calculer la sortie (forward)
- Calculer l'erreur de prédiction pour évaluer les performances du modèle
- Rétropropager l'erreur de prédiction dans le réseau, pour mieux approcher la sortie espérée (backward)

Les calculs correspondants à ces étapes sont spécifiés dans les sections suivantes. Vous devez les implémenter. Veillez à utiliser les fonctions de numpy pour permettre une exécution suffisamment rapide de votre code.

La fonction d'entraînement vous est fournie. Elle présente un exemple au réseau, calcule les sorties de chaque couche, et rétropropage l'erreur dans le réseau. Ce processus est répété pour toutes les images du jeu d'entraînement, e fois, pour e le nombre d'*epochs*. A chaque *epoch*, la MSE (Mean Squared Error) est calculée pour donner une estimation de l'apprentissage du réseau.

Passe avant (forward)

Comptez que le réseau est composé d'une matrice de poids W_h entre la couche d'entrée et la couche intermédiaire, et d'une matrice W_o entre la couche intermédiaire et la couche de sortie.

- Pour obtenir l'entrée de la couche intermédiaire du réseau en fonction d'un vecteur d'entrée x , il faut d'abord obtenir le produit $x_h = x \times W_h$. Notez que si x a pour dimension m et W a pour dimension de sortie n , la matrice W doit avoir comme dimensions $m \times n$.
- Ensuite, il faut appliquer une fonction d'activation sur chaque composante du vecteur sortant de la couche intermédiaire. Dans ce projet, vous utiliserez la fonction $y_h = \tanh(-x_h)$.
- Obtenez l'entrée de la couche de sortie $x_o = y_h \times W_o$.
- Appliquez la fonction d'activation sur la couche de sortie $y_o = \text{softmax}(x_o)$.
- Pour l'étape de rétropropagation, vous devrez stocker la sortie de la couche intermédiaire y_h et de la couche de sortie y_o .

Erreur de prédiction (mse_loss)

L'erreur de prédiction entre le vecteur de probabilité y_t attendu et le vecteur y_o prédit utilisée dans ce projet est la Mean Squared Error: $\text{MSE}(y_t, y_o)$. La MSE est définie comme $\text{MSE}(y_t, y_o) = \frac{\sum (y_t - y_o)^2}{|y_t|}$.

Rétropropagation (backward)

La rétropropagation de l'erreur permet d'ajuster la valeur des poids du réseau pour s'approcher de la sortie espérée. Elle se déroule en plusieurs étapes:

- Calcul de l'erreur de prédiction (voir section précédente)

- Rétropropagation de l'erreur:
 - L'erreur de sortie est définie comme $e_o = \frac{y_t - y_o}{|y_t|}$.
 - L'erreur de la couche intermédiaire est donnée par $e_h = (e_o \times W_o^T) * y_h * (1 - y_h)$
- Correction des poids:
 - $W_o = W_o - \mu(y_o \times e_o)$
 - $W_h = W_h - \mu(x^T \times e_h)$

où l'opérateur $|a|$ désigne le nombre de composantes de a , μ désigne le *learning rate* de l'algorithme (fixé initialement à 0.01), l'opérateur \times désigne le produit d'un vecteur avec un vecteur ou avec une matrice (np.dot), l'opérateur $*$ désigne le produit composante par composante (*), et a^T signifie la transposée de a .

Fonctions supplémentaires

On vous fournit les fonctions `predict`, `visualize_prediction` ainsi que `confusion_matrix`. Veuillez à vous renseigner sur leur utilisation pour produire des figures dans votre rapport.

4 Consignes

On vous fournit deux fichiers Python:

- `main.py`, qui contient le code à compléter. Vous aurez besoin des librairies `numpy` et `matplotlib`.
- `test_main.py`, qui contient les différents tests à passer. Vous aurez besoin de la librairie `pytest`.

4.1 Rapport

Produisez un rapport *scientifique*, qui suit la structure suivante :

1. Introduction
2. Cadre expérimental
3. Résultats
4. Analyse
5. Conclusion
6. Bibliographie (éventuelle)

et qui ne doit pas excéder **3 pages A4**, en comptant les figures et la bibliographie. Veuillez à obtenir un rendu similaire à Latex (mais vous pouvez utiliser un outil similaire, comme <http://typst.app>). **N'incluez ni page de garde, ni table des matières.** Veuillez à être *concis-e*: c'est une qualité.

4.2 Questions

On vous demande:

1. De compléter le fichier `main.py` (5 points)
2. D'analyser l'impact de la taille de la couche intermédiaire W_h (2 point)
3. D'analyser l'impact de la valeur du *learning rate* μ (2 point)
4. D'analyser l'impact du nombre d'*epochs* e (1 point)
5. De proposer une raison de l'analyse des performances du réseau sur un fichier `test.csv` plutôt que sur le fichier `train.csv` (1 point)
6. De présenter et d'analyser la matrice de confusion (2 points)
7. De suggérer deux modifications à faire au réseau pour améliorer les performances (2 points)

La qualité générale du rapport (structure, introduction, conclusion, orthographe, sources, pertinence des questions et des remarques) compte pour 5 points.

4.3 Utilisation de Chat-GPT, Copilot, ... et autres assistants

Dans la conclusion de votre rapport, détaillez votre utilisation éventuelle d'outils d'assistanat, que ce soit pour l'écriture du code ou du rapport. Toutes vos sources doivent être pertinentes et exister. L'utilisation de tels outils est autorisée **mais déconseillée pour le code**: en effet, l'intérêt de l'écriture du code est la familiarisation avec les opérations qui composent l'apprentissage des réseaux de neurones. Ce travail est *individuel*.

Remise

- Un fichier .zip portant votre nom_prénom (exemple: tribel_pascal.zip), contenant deux fichiers :
 - Un exécutable main.py. Vous pouvez considérer qu'il sera exécuté dans un répertoire contenant aussi les fichiers train.csv et test.csv, mais n'incluez pas ces fichiers .csv dans votre remise. Incluez le fichier test test_main.py.
 - Votre rapport, au format nom_prenom.pdf
- Tout manquement à une consigne de remise entraînera 0.5 points en moins.

Le travail est **individuel** et doit être rendu sur l'Université Virtuelle pour le vendredi 05/01/2024 à 23:59. Vous pouvez envoyer vos questions à l'adresse **pascal.tribel@ulb.be**. Nous répondrons à celles-ci lors de la séance questions-réponses du **lundi 18 décembre**. Mentionnez l'intitulé du cours (F311) dans l'en-tête de votre mail.

Bon travail !