BOUBEKEUR Tahar
BOURRASS Shems
DAVID Timothée
DUMEUR Merlin

16/2/2017

Reconnaissance de chiffres

Professeur encadrant : Alain Pétrowski

Version 2.0

Sommaire

Projet

Présentation

Nous allons réaliser une application qui a pour but de reconnaître des chiffres manuscrits. Pour cela, nous avons choisi d'utiliser le modèle du réseau de neurones. Il existe plusieurs algorithmes d'apprentissage différents pour les réseaux de neurones, afin de garder un algorithme réalisable à notre niveau nous avons choisi de nous concentrer sur l'implémentation de l'apprentissage par backpropagation qui est simple mais reste efficace dans le cadre de notre sujet. De plus, au delà de la simple réalisation du réseau de neurones nous allons également nous efforcer de prendre en compte des heuristiques connues sur les réseaux de neurones améliorant les performances de notre application.

I. Cahier des charges

L'application devra :

- Être capable de reconnaître des caractères après s'être entraîné sur une base de données contenant des images de chiffres manuscrits
- Avoir un taux d'erreur inférieur à 5%
- Avoir une interface qui soit *user-friendly* qui permet de tester ainsi que d'entraîner l'application

II. Spécifications fonctionnelles de l'application

L'application comportera les fonctions suivantes :

- **Reconnaissance de caractères** : à partir d'une image passée sous forme de matrice de pixels, l'application renvoie pour chaque chiffre un nombre qui tend vers 1 pour le chiffre reconnu, et vers 0 pour les autres.
- **Calcul du résultat** : Déterminer le résultat à renvoyer à l'utilisateur ainsi que calculer le taux de confiance à partir des résultats de la dernière étape.
- **Interface avec la base de données** : cette fonctionnalité permet de prendre des images provenant d'une base de données et de les soumettre à l'application ainsi que de l'entraîner.
- **Apprentissage** : cette fonctionnalité permet de, à partir d'une image et du chiffre qui lui correspond, de modifier le fonctionnement de l'application afin d'améliorer sa précision.
- **Interface utilisateur** : cette fonctionnalité est ce qui permet à l'utilisateur d'interagir avec l'application en lui soumettant des images et en obtenant le résultat donné par l'application ou bien de l'entraîner en lui donnant en même temps le chiffre attendu en sortie.
- **Automatisation de l'entraînement** : à partir d'une base de données cette fonctionnalité permet de faire en sorte que l'application apprenne en choisissant de manière stochastique des cas à soumettre à l'apprentissage, puis effectue une validation en se

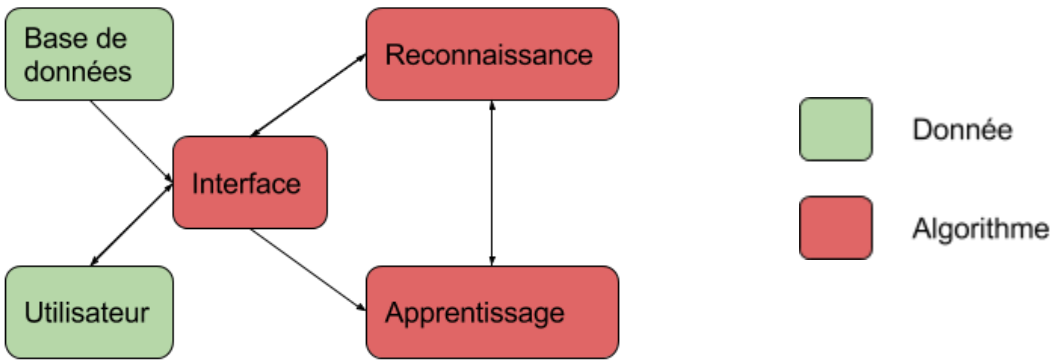
basant sur des cas non déjà traités par l'application

III. Regroupement modulaire des fonctionnalités

On peut regrouper les différentes fonctionnalités en 3 modules :

- **La reconnaissance** : prenant en entrée une image et sortant un chiffre assorti d'un taux de confiance, et qui regroupe la reconnaissance de caractères ainsi que le calcul du résultat
- **L'apprentissage** : prenant en entrée une image ainsi que le chiffre auquel l'image correspond et qui modifie le fonctionnement futur de l'application pour prendre en compte les données passées en entrée, et qui est constitué de la fonctionnalité d'apprentissage accompagnée de son automatisation
- **L'interface** : que ce soit avec l'utilisateur ou avec une base de données, l'application doit pouvoir traiter les données de manière à ce qu'elles puissent être exploitées

IV. Flux de données entre les différents modules



V. Répartition des tâches

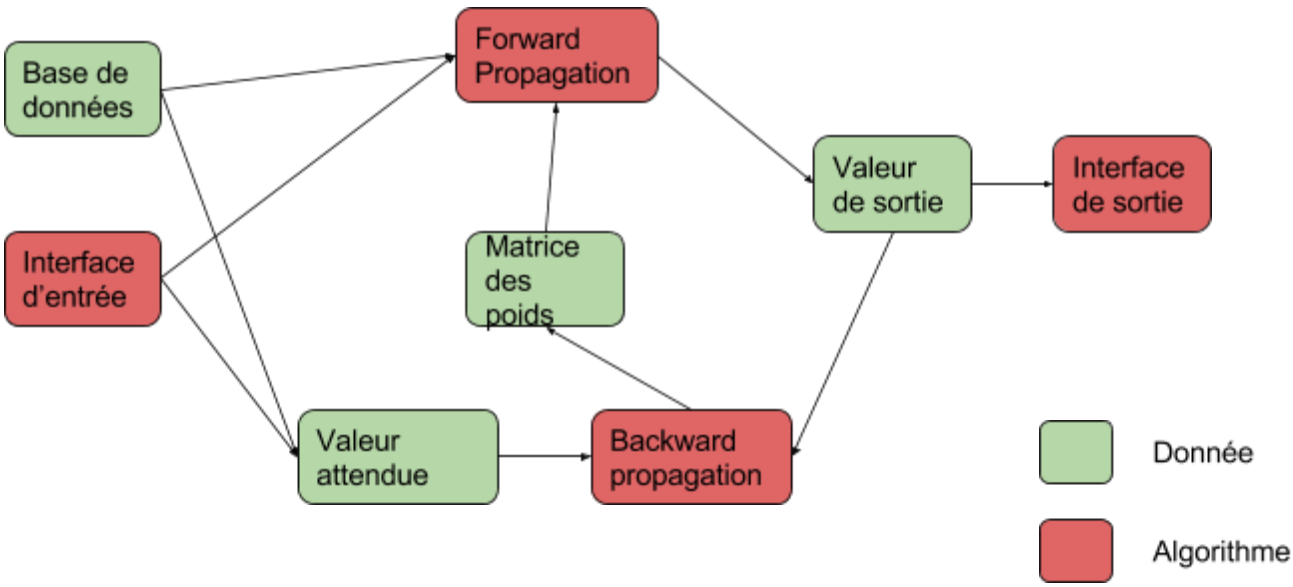
- **Tahar BOUBEKEUR** prend en charge la reconnaissance
- **Merlin DUMEUR** implante les algorithmes d'apprentissage
- **Timothée DAVID** code l'interface
- **Shems BOURRASS** s'occupe de l'automatisation de l'entraînement ainsi que de la validation

VI. Précisions liées à l'utilisation des réseaux de neurones

Dans cette partie nous explicitons la présence d'un réseau de neurones dans notre application.

La fonctionnalité de reconnaissance de caractères de notre application utilise l'algorithme de *forward propagation* qui permet, en passant en entrée notre image, avec un neurone par pixel de l'image, de retourner sur la couche de sortie du réseau de neurones une valeur pour chaque chiffre. Le cas idéal étant celui où la valeur retournée par le neurone associé au chiffre à reconnaître vaut 1 et chaque autre vaut 0. Ce cas n'étant pas atteint dans la pratique on utilise la fonctionnalité de calcul du résultat pour déterminer quel est le caractère reconnu ainsi que donner une estimation de la confiance que l'on peut avoir dans ce résultat.

La fonctionnalité d'apprentissage de notre réseau de neurones emploie l'algorithme de *back propagation*, en calculant l'erreur de l'algorithme par rapport au résultat attendu, et en modifiant les poids des liaisons de manière à corriger l'erreur. L'algorithme de descente de gradient qui sera utilisé reste à déterminer, il sera choisi selon un compromis entre le temps d'exécution de la backprop et la performance de l'algorithme.



VII. Bibliographie

D. E. Rumelhart, G. E. Hinton, R. J. Williams *Learning internal representations by error propagation*, 1988

Yann LeCun, Leon Bottou, Genevieve B. Orr, Klaus-Robert Müller *Efficient BackProp*, 1998