Dominic MICHAUD  
Dominic JOBIN  
Groupe 0001

**Reconnaissance de caractères à l’aide de réseau de neurones artificiels**  
  
Travail présenté à M. François Bertrand  
Exploration des nouvelles technologies  
420-501-SF

Département de l’informatique  
Cégep de Sainte-Foy  
Vendredi le 1er septembre 2017

Table des matières

[Sujet 2](#_Toc491873755)

[Enjeux 2](#_Toc491873756)

[Objectifs 3](#_Toc491873757)

[Expérimentation 3](#_Toc491873758)

[Limites 3](#_Toc491873759)

[Incertitudes 4](#_Toc491873760)

[Ressources 4](#_Toc491873761)

[Planification 5](#_Toc491873762)

# Sujet

Dans le cours d’exploration des nouvelles technologies, nous rechercherons comment développer un réseau de neurones artificiels (« *artificial neural network* » ou *ANN*) qui sera en mesure de détecter des caractères (chiffres et lettres) dessinées à l’écran. Le but de la recherche est de comprendre le fonctionnement derrière les technologies d’apprentissage automatique (« *machine learning »*) en exploitant un outil de reconnaissance optique de caractères.

Dans le contexte plus général de l’intelligence artificiel, l’enjeu principal est souvent d’automatiser des tâches qui sont faciles pour un humain mais difficiles pour un ordinateur. La reconnaissance optique de caractère (« *optical character recognition* » ou *OCR*) est un bon exemple de tâche qui est difficile à programmer de façon algorithmique.

La reconnaissance de caractères a une application importante au niveau de l’archivage de données, par exemple lorsque les documents sont seulement disponibles sur papier. Il est facile de numériser un document mais difficile d’extraire l’information sous forme de texte pour la traiter par la suite, à moins de transcrire le texte manuellement. Une autre application possible serait de détecter des codes postaux.

Ce projet fera le lien avec des connaissances que nous avons acquises dans nos cours de programmation et d’algèbre linéaire et nous permettra de les mettre en pratique concrètement.

# Enjeux

Pour arriver à la réussite du projet, beaucoup de connaissances seront nécessaires à acquérir. Le plus difficile sera le fonctionnement derrière les réseaux de neurones artificiels et les détails qui les entourent qui relèvent de l’algèbre linéaire et de calcul avancé. Heureusement, il existe plusieurs librairies dans le langage Python qui nous faciliteront la tâche.

L’intelligence artificielle est un sujet qui est de plus en plus d’actualité dans des domaines qui jusqu’à maintenant étaient crus comme étant réservés aux humains, par exemple la conduite d’automobiles. Les réseaux de neurones artificiels sont souvent à la base des systèmes d’intelligence artificielle complexes. La motivation de notre recherche est donc de pouvoir expliquer le lien entre la reconnaissance de caractère et les technologies plus poussées qui ont été développées dans les dernières années par les compagnies comme Google et IBM.

Notre outil de démonstration se limitera à la reconnaissance individuelle de caractères, mais il serait possible de l’utiliser pour transcrire correctement des documents numérisés pour en extraire le texte. Il y a plusieurs domaines où il est utile d’extraire du texte contenu dans une image, par exemple pour archiver les données de façon plus compacte ou pour l’envoyer dans un système de synthèse vocale pour aider les personnes avec des problèmes de vision.

# Objectifs

* Créer un réseau de neurones artificiels
* Utiliser une collection d’images de caractères pour entraîner le réseau pour qu’il apprenne à détecter les caractères avec un taux de réussite d’au moins 95%
* Développer une interface graphique simple qui permettra de dessiner un caractère à l’écran
* À partir du réseau qui aura été entraîné, afficher sous forme de texte le caractère qui a été dessiné à l’écran et le niveau de certitude en pourcentage

# Expérimentation

La recherche sera menée d’abord en lisant les ressources à propos des réseaux de neurones qui sont disponibles sur internet. La recherche sera ensuite menée en grande partie à l’aide d’outils d’expérimentation. L’outil principal sera le langage de programmation Python dans lequel notre outil de démonstration sera programmé. Les librairies Numpy (opérations sur des matrices et tableaux) et Pygame (interfaces graphiques) seront utilisées.

Les étapes suivantes nous permettront d’arriver au but visé :

* Télécharger une collection d’images contenant des caractères dessinés à la main. Au besoin, en créer nous-même.
* Dans un script Python, créer le réseau de neurones artificiels qui sera modélisé sous forme de matrice à l’aide de Numpy. Les valeurs contenues dans le réseau seront initialement aléatoires.
* Développer une fonction qui transforme les images de caractères en tableau de 1 et de 0 (noir et blanc) qui sera lisible par le réseau et qui retournera une prédiction de caractère.
* Implémenter l’algorithme du gradient pour minimiser la fonction d’erreur entre la prédiction du réseau et les résultats attendus. Il faudra laisser le script rouler pendant quelques minutes voire quelques heures pour avoir un taux de réussite d’au moins 95%. Il faudra possiblement ajuster le réseau de neurones afin d’obtenir un résultat optimal.
* Développer une interface simple permettant à l’utilisateur de dessiner un caractère. Les pixels dessinés à l’écran passeront dans le réseau de neurones qui affichera de quel caractère il s’agit.

# Limites

Dans le cadre de la recherche, l’application finale sera limitée à la reconnaissance d’un seul caractère à la fois, et l’ensemble des caractères sera limité aux chiffres et aux lettres majuscules.

Nous prévoyons également que le réseau de neurones artificiels aura de la difficulté à distinguer les caractères similaires. Par exemple, l’application risquera de détecter un « 5 » alors que le caractère était un « s ». Il sera intéressant de voir comment les pourcentages de certitude seront répartis pour des caractères similaires.

# Incertitudes

Voici les éléments qui nous sont inconnus lors du départ de la recherche ainsi que des pistes de solutions ciblées :

* *Backpropagation,* l’algorithme qui entraîne le réseau de neurones. Il existe plusieurs ressources telles que [www.neuralnetworksanddeeplearning.com](http://www.neuralnetworksanddeeplearning.com) qui l’expliquent en détail.
* L’utilisation de la librairie Numpy qui sera nécessaire. La documentation complète de la librairie est disponible à cette adresse : <https://docs.scipy.org/doc/numpy/>

# Ressources

Les ressources suivantes seront utilisées lors de l’expérimentation et de la recherche :

Ressources matérielles :

* Processeur : Intel Core i3-3210M CPU @ 2.50 GHz
* Mémoire vive : 8 GB

Ressources logicielles :

* Système d’exploitation : Windows 10 Professionnel 64 bits
* Langage de programmation : Python 3.6
* Librairies Python : Numpy version 1.13.1 et Pygame version 1.9.3

# Planification

| **Périodes** | **Temps prévu** | **Descriptions des tâches à réaliser** | **Responsable** |
| --- | --- | --- | --- |
| Semaines 1 à 3 | 6 heures | Recherche documentaire | Dominic J. et Dominic M. |
| Semaines 4 à 8 | 20 heures | Construction des outils d’expérimentation : Interface de dessin de caractère, base de données de caractères, réseau de neurones | Dominic J. et Dominic M. |
| Semaine 9 et 10 | 8 heures | Mise au point des outils d’expérimentation | Dominic J. |
| Semaine 11 et 12 | 8 heures | Expérimentation avec les outils, par exemple ajuster le nombre de neurones dans la couche cachée du réseau | Dominic J. et Dominic M. |
| Semaines 13 | 4 heures | Validation des résultats obtenus | Dominic J. et Dominic M. |
| Semaine 14 | 2 heures | Analyse des résultats | Dominic M. |
| Semaine 15 | 2 heures | Publication des résultats | Dominic J. et Dominic M. |