

Réseau de neurones

CPE — ETI5-IMI

30 novembre 2016
durée : 2 heures

1 Introduction

Si les réseaux de neurones ne sont pas vraiment adaptés à certains problèmes nécessitant des calculs exacts, ils le sont bien plus pour tout ce qui est reconnaissance d'images (de visages par exemple) ou la classification de données.

Lors de ce TP, nous allons étudier un réseau de neurones de type "rétine". C'est-à-dire que son étage d'entrée sera une image. En particulier, il sera spécialisé dans la reconnaissance de lettres majuscules provenant de diverses polices de caractères.

Question 1 : Récupérez l'archive `neurones.tar.gz`, compilez et exécutez sans paramètres pour vérifier que tout fonctionne bien.

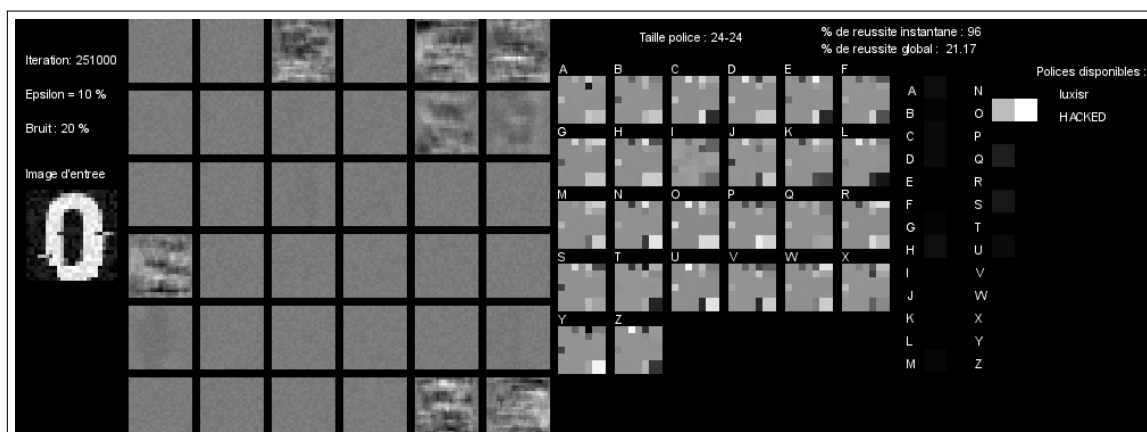


FIGURE 1 – Une vue du réseau de neurones en action

Vous devriez obtenir une fenêtre semblable à la figure 1.

Les sources de ce programme sont organisées ainsi :

- Le répertoire `lib` contient les fichiers que vous connaissez déjà permettant d'utiliser OpenGL plus simplement, et le `font-manager` qui utilise la bibliothèque `freetype` pour créer des images de chaque lettre de chaque police de caractères.
- Le répertoire `freetype` et ses sous-répertoires contiennent la bibliothèque du même nom. Cette bibliothèque n'est malheureusement pas installée sur vos postes, c'est pourquoi ses sources sont incluses ici.
- Dans le répertoire `src`, se trouvent les trois fichiers principaux de l'application. `reseau.cpp` et `reseau.h` contiennent le code implémentant le réseau de neurones proprement dit, avec les deux étapes de propagation des données. `main.cpp` contient le code utilisant le réseau de neurones pour l'entraîner, et réalise l'affichage des statistiques que l'on peut voir sur la figure 1. Vous pouvez étudier ces trois fichiers afin de comprendre le fonctionnement interne du réseau de neurones.

2 Utilisation du programme neurones

2.1 Description

Une fois lancé, le programme affiche une fenêtre telle que celle de la figure 1 et met en œuvre (entraîne) un réseau de neurones à rétro-propagation simple possédant 36 neurones dans sa couche cachée et 26 neurones dans sa couche de sortie. La couche d'entrée est représentée par une image (de taille configurable) en niveau de gris comportant une lettre d'une police. Cette lettre a une taille inférieure ou égale à la taille de l'image. Du bruit est ajouté à l'image d'une part pour s'approcher de conditions réelles, et d'autre part pour aider le réseau à sortir de certains minimums locaux lors de l'apprentissage.

À chaque itération, une nouvelle image est créée et présentée au réseau, qui essaie de reconnaître la lettre, en transmettant les informations de couches en couches, d'abord à la couche intermédiaire de 36 neurones puis à la couche de sortie qui comporte 26 neurones. On considérera que la lettre a été correctement déterminée si le neurone correspondant de la couche de sortie est celui le plus activé. Dans tous les cas, la réponse correcte est rétro-propagée afin d'adapter un peu plus le réseau. L'amplitude de cette rétro-propagation est dirigée par un paramètre nommé *epsilon*.

La première colonne de la figure 1 indique le numéro d'itération courant, les valeurs d'epsilon et de bruit courants ainsi que l'image présentée au réseau. Chaque pixel de cette image (la couche d'entrée) est lié à chaque neurones de la couche intermédiaire via des poids.

La seconde partie montre la valeur des poids des neurones de la couche intermédiaire. On peut donc voir 36 "images" de même taille que l'image de la couche d'entrée.

Puis viennent les poids des neurones de la couche de sortie : 26 neurones comportant 36 poids chacun. L'état d'activation de ces neurones est indiqué sur les deux colonnes A–M et N–Z en face de la réponse attendue. Dans l'exemple de la figure 1, la lettre O devait être reconnue, et l'on peut remarquer que c'est bien en face de la lettre O que l'on reconnaît le niveau de gris le plus élevé.

Diverses autres informations sont affichées :

- L'étendue des tailles de polices actuellement testées. La taille minimale est 8 et la taille maximale est la taille du côté de l'image d'entrée. Il est possible de changer ces tailles lors du lancement du programme et même après en utilisant les touches fléchées.
- Le taux de réussite instantané indique combien de lettres ont été correctement reconnues lors des 100 dernières itérations.
- Le taux de réussite global indique le pourcentage de lettres correctement reconnues depuis le lancement du programme. Notez qu'en répondant au hasard (comme c'est le cas au début), le réseau n'aurait que 3,84 % de chance de répondre correctement.
- La liste des polices de caractères disponibles lors de cette exécution du programme. Les polices actives (celles parmi lesquelles les lettres sont actuellement choisies) sont affichées en blanc, les autres en gris.

2.2 Ligne de commande

Le programme `neurones` accepte un certain nombre d'arguments en ligne de commande afin de paramétrer le réseau. Chaque paramètre est de la forme `param=valeur`.

- `bruit` : ce paramètre permet de choisir le niveau de bruit initial. Par exemple, pour choisir un niveau de bruit de 20 %, on pourra appeler le programme ainsi : `./neurones bruit=20`
- `epsilon` : ce paramètre permet de choisir le coefficient de rétro-propagation initial. On choisira un niveau de 15 % ainsi : `./neurones epsilon=15`
- `taille_image` : ce paramètre permet de choisir la taille de l'image d'entrée, il doit être entre 16 et 64. Notez que la vitesse d'exécution du programme est grandement affecté par ce paramètre, et que la vitesse d'apprentissage aussi.
- `polices` : ce paramètre permet de choisir quelles polices de caractères seront disponibles pendant l'exécution. Il s'agit d'une liste de noms sans extension séparés par des virgules. Les polices doivent être au format *truetype* (.ttf) et être placées dans le répertoire `fonts`. Pour choisir les polices `CutiveMono` et `HACKED`, on peut appeler le programme ainsi : `./neurones polices=CutiveMono,HACKED`
- `taille_police` : ce paramètre permet de choisir l'étendue des tailles de polices par défaut. Pour tester des tailles de polices allant de 10 à 24, on pourra appeler le programme de cette manière : `./neurones taille_police=10-24`

Vous pouvez évidemment utiliser plusieurs de ces paramètres à la fois, par exemple :

```
./neurones bruit=25 epsilon=15 polices=helvetica,times taille_image=25 taille_police=18-25
```

2.3 Modification des paramètres pendant l'exécution

Un certain nombre de paramètres peut être changé lors de l'exécution du programme afin de modifier l'apprentissage :

- Les touches fléchées permettent de changer l'étendue des tailles de polices proposées dans les images d'entrée.
- Les touches de fonction permettent d'activer ou de désactiver temporairement certaines polices.
- 'b' et 'B' permettent de diminuer ou d'augmenter la quantité de bruit ajoutée à l'image d'entrée.
- 'e' et 'E' permettent de diminuer ou d'augmenter la valeur de epsilon.
- La touche "échappe" met fin au programme.

3 Travail demandé

Question 2 : *Exécutez le programme plusieurs fois, avec des paramètres différents, et regardez le évoluer.*

Question 3 : *Essayez de comprendre le rôle de chaque paramètre. Dans quels cas le réseau arrive à un taux de reconnaissance satisfaisant ? (plus de 90 %*

Question 4 : *Essayez d'utiliser d'autres polices de caractères. Vous en trouverez notamment sur le site <http://www.dafont.com/>. Que se passe-t-il si le réseau est entraîné à reconnaître une certaine police et qu'on lui en donne une autre à reconnaître ?*