INF8225 – Intelligence Artificielle: Techniques Probabilistes et d'Apprentissage

TP3

Éric Morissette

1631103

15 Mars 2016

École Polytechnique de Montréal

Pour le TP3, il nous fallait implémenter un réseau de neurones afin de résoudre un problème classique: la reconnaissance de chiffres écris à la main.

Avec les données du MNIST, qui est un ensemble de 60 000 fichiers 28x28 pixels représentant différentes caligraphies, nous devions utiliser deux réseaux différents, l'un comportant une seule couche cachée de neurones tandis que le second devait en comprendre deux. Il fallait principalement implémenter un réseau pouvant prendre une multitude de paramètres en entrée, y compris le nombre de couches ainsi que leur taille, afin d'avoir un système très souple et variable.

Partie I – Pseudocode d'un réseau de neurones

Répéter tant que l'on veut ou tant qu'on n'ait pas atteint un seul de précision:

Pour chaque exemple de test:

Pour chaque noeud de la couche d'entrée:

Lire les pixels en entrée

Pour chaque couche autre que la couche d'entrée:

Calculer “in_i” = sum($W_{j,i} * a_j$) pour tout nœud I de la couche courante avec les nœuds j de la couche inférieur

Calculer “a_i” = activation(in_i)

Pour chaque noeud i de la couche de sortie:

Calculer Δ_i = activation'(in_i)* (y_i - a_i)

Pour chaque couche autre que la couche de sortie, en allant vers la couche d'entrée:

Pour chaque noeud j de la couche k:

Δ_j = activation'(in_j) * sum($W_{j,i} * \Delta_i$)

Pour chaque noeud i de la couche k+1:

$W_{j,i} = W_{j,i}$ + (alpha * a_j * Δ_j)

Partie II – Implémentation et comparaison de deux réseaux

Pour ce qui est de l'implémentation, j'ai ajouté une étape de régularisation L1 et L2 et, après quelques tests, j'ai choisi les valeurs de 0.0 et 1.5 respectivement.

Lambda1, Lambda2	Précision moyenne après 100 itérations
0.0, 1.5	0.920600
0.015, 1.485	0.920400
0.05, 1.45	0.915850
0.1, 1.4	0.910200

Selon les données du tableau, il est facile de voir que les valeurs de [0.0, 1.5] donnent les meilleurs résultats.

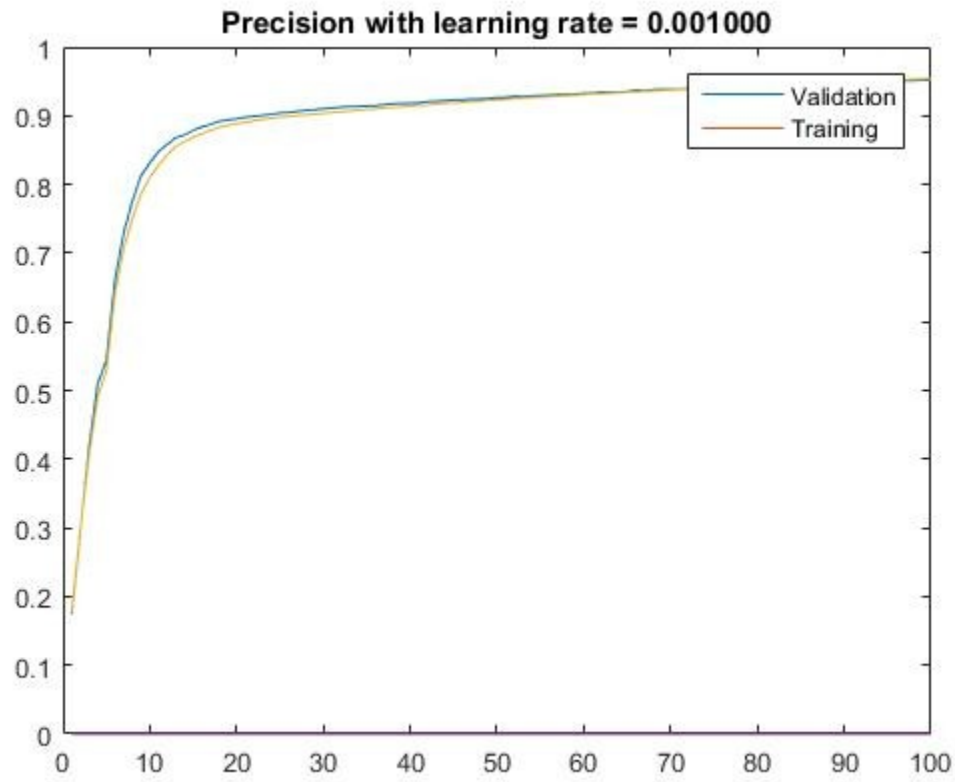
Par la suite, il fallait déterminer le nombre d'exemplaires dans l'algorithme de mini-batch, ainsi que le taux d'apprentissage, qui est intimement lié à la taille des mini-batchs.

Ici, on utilise les lambdas [0.0, 1.5]

Taille des mini-batchs et taux d'apprentissage	10 itérations	50 itérations	100 itérations
[25, 0.001]	0.812400	0.923300	0.947600
[50, 0.002]	0.818500	0.919300	0.945900
[100, 0.004]	0.807400	0.918200	0.944500

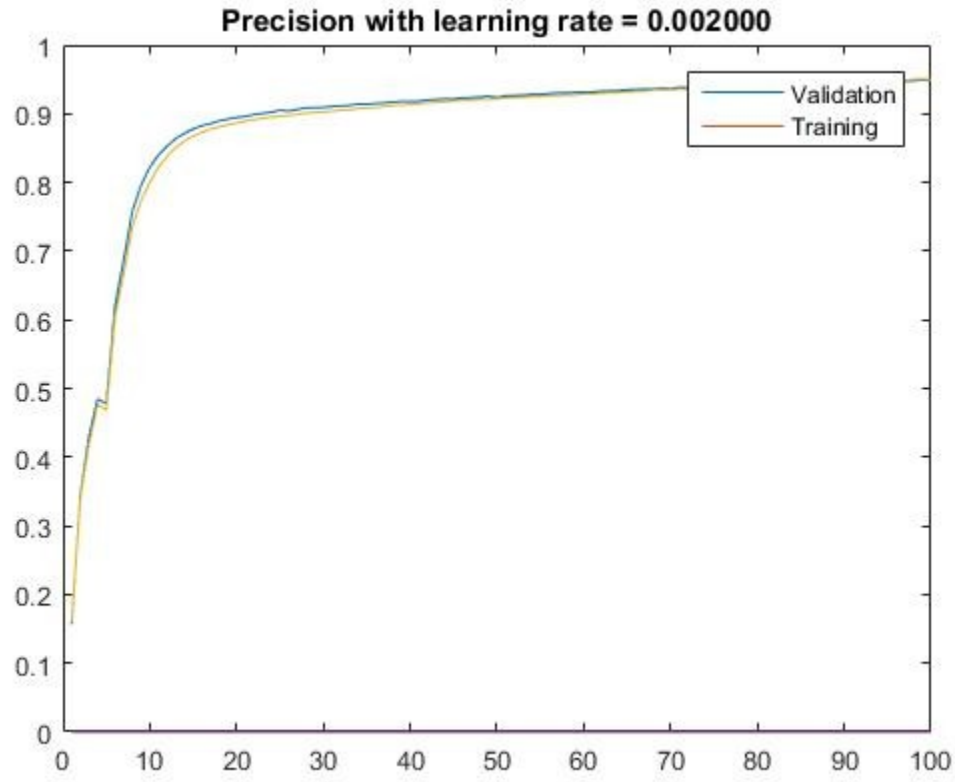
Les valeurs portent a tendre vers les mini-batchs de 25 ainsi qu'un taux d'apprentissage de 0.001.

Par contre, étant donné le temps de calcul nécessaire pour rouler le réseau avec plusieurs couches ainsi que des mini-batchs de taille 25 était relativement élevé, j'ai décidé de couper un peu de précision et d'utiliser une méthode plus rapide.

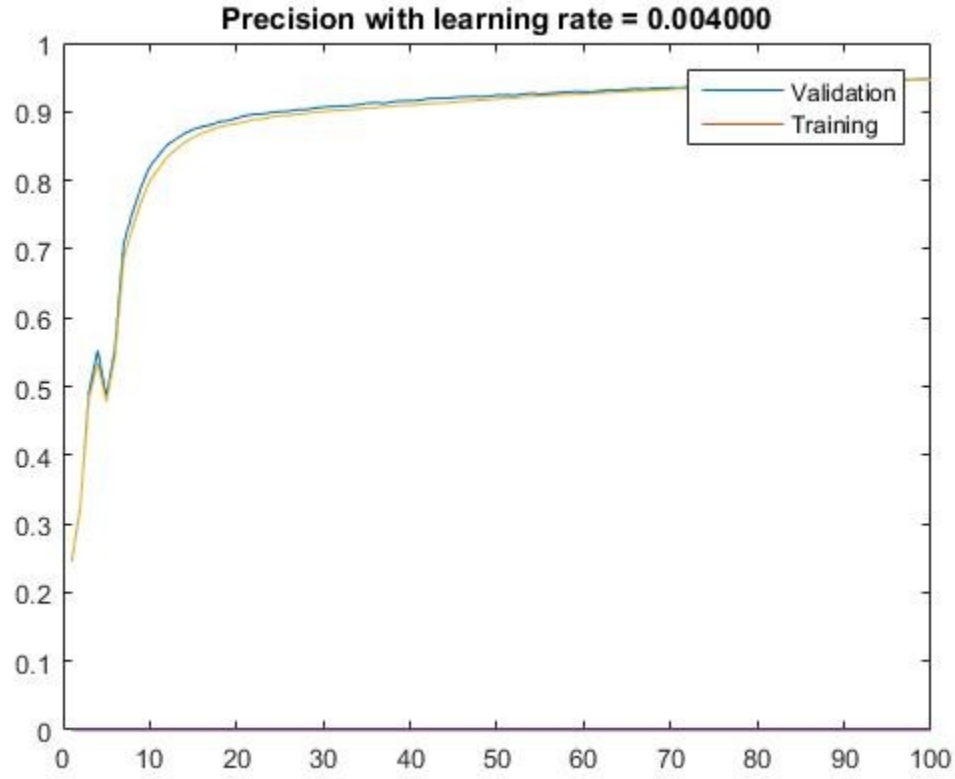


^ Mini-Batch de 25, apprentissage de 0.001

v Mini-Batch de 50, apprentissage de 0.002



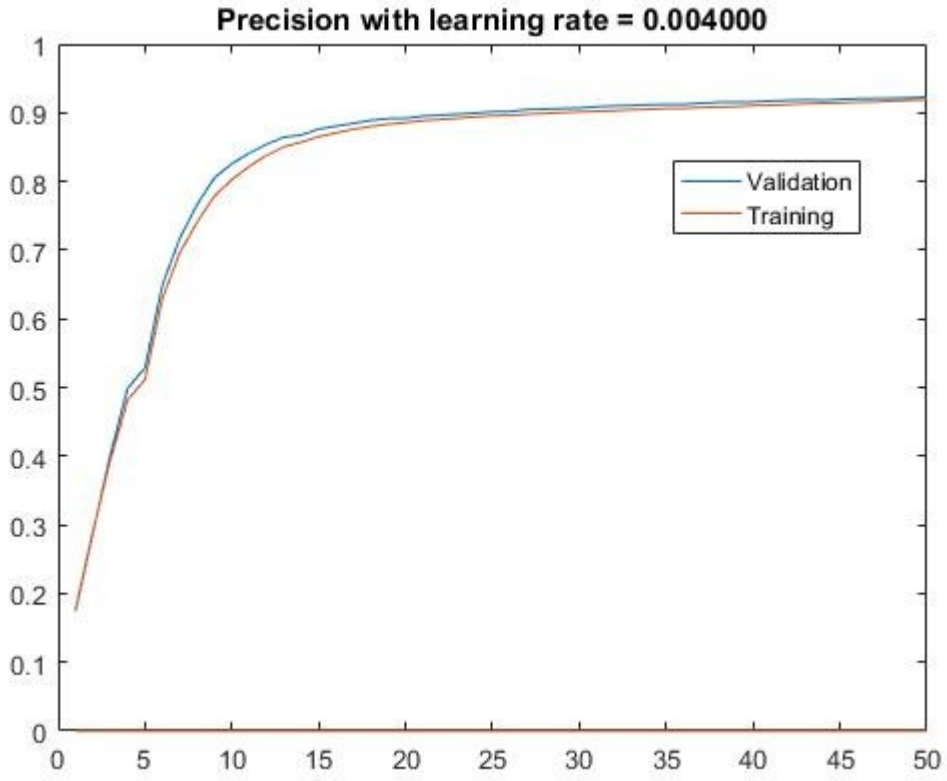
∨ Mini-Batch de 100, apprentissage de 0.004



2 couches cachées de taille 526 et 268

mini-batch de taille 100

50 itérations



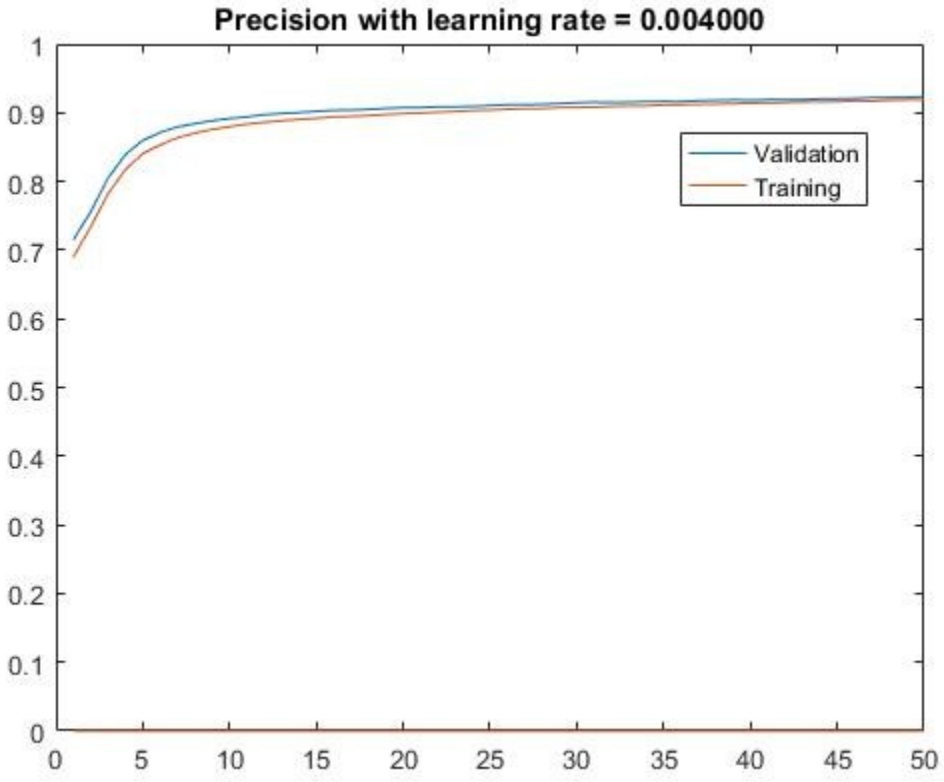
Final Test : 0.917200

Elapsed time is 307.739898 seconds.

1 couche cachée de taille 387

mini-batch de taille 100

50 itérations




```
Final Test : 0.920200

Elapsed time is 172.663776 seconds.
```

On peut voir que le réseau avec une seule couche arrive plus rapidement à une réponse, qui est aussi supérieure. Par contre, si on analyse les graphes, on peut voir que le réseau à deux couches ne semble pas avoir atteint un plateau, comme le réseau à une couche. On pourrait donc imaginer que le réseau à deux couches, malgré sa lenteur, permet d'avoir des meilleurs valeurs après plusieurs cycles d'apprentissage.

En guise de conclusion, il est très facile de voir la quantitée d'hyperparamètres pouvant affecter les résultats d'un réseau de neurones est immense. Aussi, je n'ai pas ajouté d'autres fonctions d'optimisations ou de méthodes pour contrer le sur-apprentissage des données telles que le dropout. La plus grande question lorsqu'on veut utiliser un réseau de neurones serait de comparer le temps pris pour l'apprentissage versus la précision requise pour le travail en question.