

UNIVERSITÉ DE MONTRÉAL

Rapport : Devoir3

Pierre Gérard
Mathieu Bouchard

IFT3395-6390 Fondements de l'apprentissage machine
Pascal Vincent, Alexandre de Brébisson et César Laurent

Année académique 2015 - 2016

1 Partie pratique : Implémentation du réseau de neurones

1.1 Utilisation

Pour tester notre programme, exécutez simplement la commande dans le dossier *src* :

```
python main.py
```

1.2 Notes relative à l'implémentation

1.3 Exercices 1 et 2

Nous avons implémentée la vérification de gradient par différence finie. Pour chaque scalaire "perturbé", on conserve le ratio dans une liste. Une boucle vérifie ensuite que les ratios sont bien entre 0.99 et 1.01. Les résultats semblent indiquer que notre implémentation est correcte.

Pour un réseau $2 \times 2 \times 2$, on obtient :

```
>>EXERCICE 1 et 2
```

```
Liste des ratio W1, b1, W2, b2
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0.99997500062302613, 1.0000250006249738]
```

```
>Tout les ratio sont bien entre 0.99 et 1.01
```

1.4 Exercices 3 et 4

On procède comme à l'exercice 1 et 2 sauf qu'on somme l'ensemble des gradients pour K exemples avant de calculer le ratio. Les résultats semblent eux-aussi indiquer que notre implémentation est correcte.

Pour un réseau $2 \times 5 \times 2$, on obtient :

```
>>EXERCICE 3 et 4
```

```
Liste des ratio W1, b1, W2, b2
```

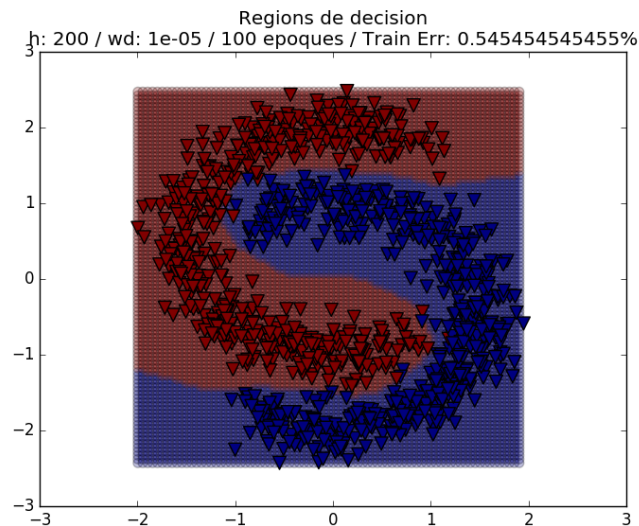
```
[1, 1, 1.000002072572213, 1.000008244958942, 1, 1, 0.99999469791365814, 0.99997890803949541,
```

```
>Tout les ratio sont bien entre 0.99 et 1.01
```

1.5 Exercices 5

On a ensuite entraîné notre réseau par descente de gradient mini-batch. L'entraînement semble un succès.

Les résultats semblent plus on augmente le nombre d'époques et de neurones cachés, plus l'erreur sera petite.



1.6 Exercices 6 et 7

1.6.1 Implémentation

1.6.2 Verification

Pour vérifier que les gradients soient équivalents, on commence par instancier les deux classes correspondant aux deux implémentations *NeuralNetwork* et *NeuralNetworkEfficient*. On initialise ensuite $W1$ et $W2$ des deux classes aux mêmes valeurs. Ensuite on exécute l'entraînement pour $K=1$. On utilise ensuite une méthode qui pour deux réseaux de neurones vérifie que les gradients soient équivalents. On recommence le procédé pour $K=10$.

Les tests de comparaison nous donnent (ok signifie que les deux gradients sont égaux) :

```

--- K=1 ---
Test Ok : gradient b2
Test Ok : gradient w2
Test Ok : gradient b1
Test Ok : gradient w1
--- K=10 ---
Test Ok : gradient b2
Test Ok : gradient w2
Test Ok : gradient b1
Test Ok : gradient w1

```

1.7 Exercices 8

Les résultats des mesures de temps sont les suivants :

```

>>EXERCICE 8 MNIST
--- Reseau de depart ---
Cela a mis : 0.419163 secondes

```

Explication
des
modifs

--- Reseau optimise ---

Cela a mis : 0.123445 secondes

Sur cette exécution, on remarque que le temps d'exécution sur le réseau avec matrice est entre 3 et 4 fois plus rapide que sur le réseau avec boucles. Le gain de performance n'est donc pas négligeable.

1.8 Exercices 9 et 10