INF8225 – Intelligence artificielle  
Techniques probabilistes et d’apprentissage

Travail pratique 2  
Réseau de neurones

Carl-Vincent Landry-Duval - 1748935  
Luc Coubariaux -

Soumis à : Alexandre Piché

18 février 2018

# Partie I

1. Pseudocode pour l’algorithme de rétropropagation :

Quelques définitions de constantes nécessaires :

* Posons : Largeur du réseau de neurone. Prédéfini comme étant 100 unités;
* Posons L : Nombre de couches cachées du réseau de neurone;
* Posons : Matrice de poids de dimension pour la couche ;
* Posons : Matrice de poids de dimension pour la couche ;
* Posons : Vecteur de valeurs d’entrée de dimension pour la couche ;
* Posons : Vecteur avec 1 ajouté à la fin de dimension pour la couche ;
* Posons : Vecteur de valeurs d’entrée initial de dimension . Synonyme de ;
* Posons : Valeur attendue en sortie pour le vecteur d’entrée ;
* Posons : Taux d’apprentissage du réseau de neurone;
* Posons : Rapport entre la taille de l’ensemble d’entraînement et de la taille l’ensemble destiné à l’apprentissage.

Quelques définitions de fonctions :

Pour des fins de simplicité, nous considérons que la fonction exponentielle appliquée à un vecteur applique la fonction exponentielle sur chaque élément du vecteur. Les opérations arithmétiques élémentaires tel que l’addition, la soustraction, la multiplication et la division s’appliqueront de façon similaire. Pour distinguer la multiplication du produit matriciel, l’opérateur sera utilisé pour indiquer le produit matriciel.

* Posons : Fonction de calcul de la perte globale;
* Posons : Fonction pour obtenir la réponse;
* Posons : Fonction pour obtenir le vecteur ;

Calcul du gradient de la perte :

Pseudocode rétropropagation :

* Poser  : Ensemble d’entraînement aléatoirement ordonné formé par les paires
* Poser  : Les premiers éléments de ;
* Poser  : Ensemble des matrices crées avec des poids aléatoires, incluant le vecteur à sa fin;
* Pour chaque dans faire :
  + ;
  + Pour allant de 1 à :
    - ;
  + ;
  + ;
  + ;
  + Pour j allant de à 1 :
    - ;
    - ;
  + ;

1. Optimisations possibles :

Afin d’optimiser le traitement d’un grand nombre de données d’entraînement, il est possible d’utiliser plusieurs techniques. Une première serait d’utiliser du parallélisme afin de réaliser les multiplications des matrices et des vecteurs de façon accélérée. Une seconde serait d’utiliser des minibatchs ce qui réduirait le nombre de cycles de modifications des poids du réseau de neurones. Il serait possible aussi de réduire le nombre de couches cachées avec, en contrepartie, un apprentissage moins profond.

Afin d’optimiser la précision du résultat, il est possible de réduire le taux d’apprentissage pour plus lentement mener vers le résultat désiré, mais en augmentant la certitude du résultat.