

TP4: Régularisation

Gaétan Marceau Caron

18 décembre 2015

1 Introduction

Lors du tp3, vous avez observé le phénomène du sur-apprentissage. Comme vous avez pu le constater, le sur-apprentissage survient lorsque la complexité du modèle est trop grande par rapport aux données. Ceci se produit en augmentant le nombre de couches cachées ou indirectement en augmentant le nombre de paramètres. Il existe de nombreuses solutions à ce problème, mais toutes tentent de restreindre cette complexité. En effet, dans ce TP4, nous présentons la régularisation comme une solution au problème de sur-apprentissage, bien que l'approche Bayésienne intègre naturellement la régularisation dans sa formulation. Par conséquent, les méthodes que nous allons voir sont en fait des cas particuliers d'une théorie plus générale de l'apprentissage statistique.

Nous rappelons que le but de l'apprentissage artificiel est de minimiser l'erreur de prédiction lors de la généralisation. En utilisant un ensemble de validation, nous pouvons détecter de façon en-ligne le moment où le modèle commence à sur-apprendre. Par conséquent, la façon la plus naturelle est d'arrêter l'optimiseur à ce moment précis. Cette technique, appelée dans la littérature «early stopping», est la plus simple et est largement utilisée.

Cependant, cette technique ne prend pas en compte les valeurs des paramètres du modèle courant. De façon générale, si nous avons plusieurs modèles avec les mêmes performances, nous voudrions privilégier les modèles les plus simples. Ce principe du rasoir d'Occam s'implémente facilement en ajoutant un terme de régularisation dans la fonction de coût :

$$\min_{\theta} \quad \frac{1}{N} \sum_i [l_{d_i}(\theta)] + \lambda L(\theta) \text{ où } d_i \sim \mathcal{D} \quad (1)$$

où $\lambda \in \mathbb{R}^+$ est un hyper-paramètre contrôlant l'amplitude de la régularisation et L est une fonction positive qui ne dépend que des paramètres θ du modèle. Ainsi, les deux fonctions les plus fréquemment utilisées sont les normes L_1 et L_2 . Intuitivement, pour différents réseaux avec la même performance de classification, l'optimiseur privilégiera les modèles avec une norme plus petite et donc, de plus faible complexité.

La dernière technique que nous allons voir s'appelle «dropout». Alors que les régularisations précédentes s'appliquent lors de la mise à jour des paramètres,

dropout est une méthode qui modifie la procédure forward et la procédure de prédiction lors du test. Lors de la procédure forward, pour chaque couche, nous utilisons un masque binaire dont chaque composante est tirée selon une loi de Bernoulli de paramètre p . Ce paramètre est généralement fixé à 0.8 pour la couche d'entrée et 0.5 pour les couches cachées. Ensuite, ce masque est multiplié composante par composante à la matrice des activations et donc, le masque est différent pour chaque exemple. Finalement, lors de la prédiction, nous utilisons le réseau moyen, c'est-à-dire dont les poids sortants d'une unité d'activation sont pondérés par le paramètre p utilisé pour cette unité lors de l'entraînement. Pour plus de détails sur cette méthode, vous pouvez consulter SRIVASTAVA et al., 2014.

2 Description

Le but du TP4 est de tester différentes méthodes de régularisation avec les réseaux de neurones, à l'aide du code développé lors du TP2 et TP3, afin de minimiser l'erreur de validation sur le base d'apprentissage MNIST.

En résumé, voici les étapes du TP4 :

1. Implémenter les méthodes de régularisation L_1 et L_2
2. Tester différentes valeurs de λ et commenter son influence sur les courbes d'apprentissage
3. Implémenter la méthode de régularisation dropout
4. Tester un réseau 800-800 avec ReLU et dropout

Le rapport du TP4 doit contenir les figures ainsi qu'une brève analyse des phénomènes observés. Vous pouvez aussi tenter de répondre aux questions ouvertes que vous avez éventuellement notées dans le rapport du TP3. Dans tous les cas, le rapport du TP4 doit inclure les rapports du TP2 et TP3.

3 Livrable

Date du livrable : avant le 15 janvier 2016

Format du livrable : un fichier compressé nommé *DL_tp4_prénom_nom.zip* contenant le code et le résumé

Dépôt : à l'adresse `gaetan.marceau-caron@inria.fr` avec comme objet du message *DL_tp4_prénom_nom*.

Description :

Le livrable associé au TP4 doit contenir le code de MiniNN complété et accompagné d'un résumé de quatre pages à cinq pages incluant les résumés du TP2 et TP3. Le code doit s'exécuter avec la commande `python miniNN.py` et afficher l'évolution de l'apprentissage (sortie par défaut du programme). Le résumé doit être succinct et se focaliser uniquement sur les points essentiels reliés à l'entraînement des réseaux de neurones. Ce document doit décrire les difficultés que vous avez rencontrées et, dans le cas échéant, les solutions utilisées pour les

résoudre. Vous pouvez aussi y décrire vos questions ouvertes et proposer une expérience sur MNIST afin d’y répondre.

Références

SRIVASTAVA, Nitish et al. (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In : *J. Mach. Learn. Res.* 15.1, p. 1929–1958. ISSN : 1532-4435.