Miguel Alejandro Martinez Herrera
Advanced Machine Learning and Deep Learning
3872000

Algorithmes d'optimisation de descente de gradient

Il existe trois variantes de l'algorithme de Descente de Gradient: Batch, Mini-Batch et Stochastique. Pour la première, le calcul du gradient est réalisé à partir de toutes les données; la méthode Mini-Batch consiste à séparer l'ensemble d'apprentissage en sous-ensembles qui seront ensuite utilisés individuellement pour le calcul du gradient. Finalement, la méthode stochastique calcule le gradient pour chaque donnée disponible et met à jour les paramètres en permanence.
La méthode Batch assure une convergence plus fine, bien qu'elle mette plus de temps vu la petite fréquence des mises à jours. La méthode stochastique permet de mieux approcher les minima et surtout à moins tendance dans des minima locaux sous-optimaux, mais tout au long de l'entraînement il y a plus de fluctuations. Le Mini-Batch est un compromis entre les deux méthodes.

Ces variantes se focalisent sur la manière de traiter les données lors de l'entraînement, il existe aussi des méthodes d'optimisation qui se focalisent sur l'amélioration de l'efficacité.

- La méthode des "Momentum" correspond à privilégier les directions des plus forts gradients, ce qui permet d'accélérer l'apprentissage et diminuer les fluctuation. Intuitivement, cette méthode correspondrait à une pierre dont le moment augmente le long d'une pente
- La méthode NAG peut-être vue comme une autre version des moments, où on cherche à éviter l'accumulation de "vitesse" qui pourrait avoir un impact négatif lorsque la pente diminue. Globalement, ceci consiste à ajuster la direction du gradient en tenant en compte la dernière direction calculée.
- Adagrad se focalise sur le "learning rate" de l'algorithme, qui pour l'algorithme de base est normalement fixe. Adagrad essaye de tenir en compte les situations où pas tous les paramètres sont identiquement fréquents. Le "learning rate" est mis à jour à chaque calcul de gradient, décroissant le long de l'entraînement, néanmoins ceci donne la possibilité d'avoir un taux d'apprentissage approchant zéro, ce qui empêcherait l'algorithme d'apprendre.
- Adadelta est une adaptation d'Adagrad, qui essaye de contrôler la décroissance du taux d'apprentissage afin d'éviter la situation où l'algorithme n'évolue plus. RMSprop est une autre méthode similaire à Adadelta en objectif.
- La méthode Adam reste une méthode adaptatif du taux d'apprentissage qui utilise des calculs similaires aux moments ainsi qu'Adadelta. La méthode Adamax est une généralisation de la méthode Adam, qui utilise la norme infinie au lieu de la norme 2 qui peut-être vu dans Adam.
- Nadam est une variante qui consiste à combiner deux méthodes, de manière similaire à Adam, mais au lieu d'utiliser les moments, ceci utilise la version améliorée NAG
- Finalement, l'algorithme AMSGrad utilise la même logique qu'Adam, mais cette fois utilisant le maximum parmi le carré des gradients passés. Ceci permet de corriger certains des problèmes d'Adam, surtout lorsqu'on dispose pas de beaucoup de données, cependant, cet algorithme est moins efficace qu'Adam.