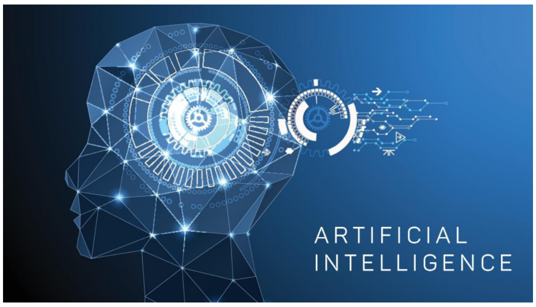
Chapitre 6 : Algorithme d'apprentissage



Pr. Mustapha BOURAHLA, Département d'Informatique, Université de M'Sila, Contact : mustapha.bourahla@univ-msila.dz

Mars 2021
v 1.0

Table des matières

Introduction

Ce sixième chapitre vous introduit la notion de l'apprentissage automatique par les réseaux de neurones

Introduction

I

On supposera à partir d'ici que vous avez déjà quelques connaissances fondamentales sur le modèle du neurone artificiel et les réseaux de neurones. Dans ce chapitre nous présentons l'algorithme des gradients descendants. Encore une fois, le but ici n'est pas d'expliquer les différentes applications possibles du machine learning, mais plutôt comment implémenter ces algorithmes.

Algorithme du gradient

II

Algorithme du gradient

- L'algorithme du gradient désigne un algorithme d'optimisation différentiable.
- Il est par conséquent destiné à minimiser une fonction réelle différentiable définie sur un espace euclidien (par exemple, $\mathbb{R}^n$, l'espace des n-uplets de nombres réels, muni d'un produit scalaire) ou, plus généralement, sur un espace hilbertien.
- L'algorithme est itératif et procède donc par améliorations successives.
- Au point courant, un déplacement est effectué dans la direction opposée au gradient, de manière à faire décroître la fonction.
- Le déplacement le long de cette direction est déterminé par la technique numérique connue sous le nom de recherche linéaire.
- Cette description montre que l'algorithme fait partie de la famille des algorithmes à directions de descente
- Les algorithmes d'optimisation sont généralement écrits pour minimiser une fonction.
- Si l'on désire maximiser une fonction, il suffira de minimiser son opposée.
- Il est important de garder à l'esprit le fait que le gradient, et donc la direction de déplacement, dépend du produit scalaire qui équipe l'espace hilbertien ; l'efficacité de l'algorithme dépend donc de ce produit scalaire.

- L'algorithme du gradient est également connu sous le nom d'algorithme de la plus forte pente ou de la plus profonde descente (steepest descent, en anglais) parce que le gradient est la pente de la fonction linéarisée au point courant et est donc, localement, sa plus forte pente (notion qui dépend du produit scalaire).
- Dans sa version la plus simple, l'algorithme ne permet de trouver ou d'approcher qu'un point stationnaire (i.e., un point en lequel le gradient de la fonction à minimiser est nul) d'un problème d'optimisation sans contrainte.
- De tels points sont des minima globaux, si la fonction est convexe.
- Des extensions sont connues pour les problèmes avec contraintes simples, par exemple des contraintes de borne.
- Malgré des résultats de convergence théoriques satisfaisants, cet algorithme est généralement lent si le produit scalaire définissant le gradient ne varie pas avec le point courant de manière convenable, c'est-à-dire si l'espace vectoriel n'est pas muni d'une structure riemannienne appropriée, d'ailleurs difficilement spécifiable a priori.
- Il est donc franchement à déconseiller, même pour minimiser une fonction quadratique strictement convexe de deux variables.
- Toutefois, ses qualités théoriques font que l'algorithme sert de modèle à la famille des algorithmes à directions de descente ou de sauvegarde dans les algorithmes à régions de confiance.
- Le principe de cet algorithme remonte au moins à Cauchy (1847).

L'algorithme

III

Notations

- $\mathbb{E}$ un espace hilbertien (produit scalaire noté $\langle\cdot,\cdot\rangle$ et norme associée notée $\|\cdot\|$) et $x \in \mathbb{E} \mapsto f(x) \in$
- $x \in \mathbb{E} \mapsto f(x) \in \mathbb{R}$ une fonction différentiable.
- On note $f'(x)$ et $\nabla f(x)$ la dérivée et le gradient de f en x, si bien que pour tout $d \in \mathbb{E}$, $f'(x) \cdot d = \langle \nabla f(x), d \rangle$.

Algorithme du gradient —

On se donne un point/itéré initial $x_0 \in \mathbb{E}$ et un seuil de tolérance $\varepsilon \geqslant 0$. L'algorithme du gradient définit une suite d'itérés $x_1, x_2, \ldots \in \mathbb{E}$, jusqu'à ce qu'un test d'arrêt soit satisfait. Il passe de x_k à x_{k+1} par les étapes suivantes.

1. Simulation : calcul de $\nabla f(x_k)$.
2. Test d'arrêt : si $\|\nabla f(x_k)\| \leqslant \varepsilon$, arrêt.
3. Calcul du pas $\alpha_k > 0$ par une règle de recherche linéaire sur f en x_k le long de la direction $-\nabla f(x_k)$.
4. Nouvel itéré : $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$.

Explication

- En pratique, il faudra prendre $\varepsilon > 0$; la valeur nulle de cette tolérance a été admise uniquement pour simplifier l'expression des résultats de convergence ci-dessous.
- Cet algorithme structurellement très simple repose sur le fait que, dans le voisinage d'un point x, la fonction f décroît le plus fortement dans la direction opposée à celle du gradient de f en x, à savoir dans la direction $-\nabla f(x)$.
- De manière plus précise, cette affirmation exprime en termes suggestifs le fait que, si $f'(x) \neq 0$, la solution du problème d'optimisation $\min_{\|d\|=1} \langle \nabla f(x), d \rangle$ est la direction $d = -\nabla f(x)/\|\nabla f(x)\|$, orientée donc vers l'opposé du gradient.
- La notion de direction de plus forte pente est en fait mal définie car elle dépend fortement du produit scalaire que l'on se donne sur l'espace hilbertien $\mathbb{E}$.
- En effet, si $\sigma : (u,v) \mapsto \sigma(u,v)$ est un autre produit scalaire sur $\mathbb{E}$, il existe un opérateur linéaire continu $S : \mathbb{E} \to \mathbb{E}$ auto-adjoint et défini positif tel que $\sigma(u,v) = \langle Su, v \rangle$, si bien que le gradient de f en x pour ce dernier produit scalaire s'écrit $S^{-1}\nabla f(x)$, ce qui montre explicitement la dépendance du gradient par rapport au produit scalaire. Il n'y a donc pas une unique direction de plus forte pente, ce qui n'apparaît pas clairement dans l'affirmation faite au début du paragraphe précédent.
- On peut même montrer que toute direction de descente de f en x, c'est-à-dire toute direction d telle $f'(x) \cdot d < 0$, est l'opposé du gradient de f en x pour un certain produit scalaire.

- L'efficacité de l'algorithme du gradient dépendra donc du choix de ce produit scalaire.
- Si $f'(x) \neq 0$, la direction $d = -\nabla f(x)$ est une direction de descente de f en x, puisque $f'(x) \cdot d = -\|\nabla f(x)\|^2 < 0$, si bien que pour tout $\alpha > 0$ assez petit, on a $f(x - \alpha \nabla f(x)) < f(x)$.
- Grâce à la recherche linéaire, tant que $f'(x_k) \neq 0$, l'algorithme fait décroître la fonction strictement à chaque itération : $f(x_{k+1}) < f(x_k)$.
- L'algorithme du gradient peut s'utiliser lorsque l'espace vectoriel sur lequel est définie la fonction à minimiser est de dimension infinie.
- Dans ce cas, l'algorithme n'est pas implémentable, mais son étude peut avoir un intérêt pour connaître son comportement en grande dimension ou pour en utiliser les propriétés de convergence à des fins théoriques.
- L'algorithme du gradient peut s'interpréter comme la méthode d'Euler explicite de résolution de l'équation différentielle ordinaire $x'(\alpha) = -\nabla f(x(\alpha))$ (flot du gradient), avec un pas α_k de discrétisation adapté à l'itération k courante par la recherche linéaire.

Conclusion

Ce chapitre vous a présenté l'algorithme d'apprentissage basé sur la technique de la descente du gradient.