TP n°1 :

Minimisation d'une fonction de 2 variables : méthode du gradient à pas fixe

Objectifs du TP :

- Mettre en œuvre la recherche de minimum par la méthode du gradient à « pas fixe »
- Analyser l'influence du pas de la méthode et du point de départ

Ce TP illustre les slides 41-44 du cours n°1. Voir aussi la section « Minimisation sans contrainte » du poly de 2005, pages 10 et 11.

Principe de la méthode du gradient

Le problème posé est de minimiser une certaine fonction $J(X)$ définie de $\mathbb{R}^n$ dans $\mathbb{R}$. La variable X est une vecteur : $X = (x_1, \dots, x_n) \in \mathbb{R}^n$. La fonction J est supposée différentiable à l'ordre 1[1], et on suppose que l'on connait l'expression analytique du gradient.

De manière générale, les méthodes de descente sont des méthodes itératives qui génèrent une suite de points X_n tels que : $\forall n \geq 0 : J(X_{n+1}) < J(X_n)$. Au point X_n, la direction dans laquelle la fonction décroît le plus vite est la direction opposée au gradient de J. Une technique simple pour réduire J consiste donc à rechercher un nouveau point en se déplaçant d'une certaine quantité dans cette direction.

Le schéma général de la méthode du gradient est le suivant :

$$X_{n+1} = X_n - \alpha . \nabla J(X_n) \qquad \text{avec} \qquad \alpha > 0 \text{ choisi pour avoir : } J(X_{n+1}) < J(X_n)$$

Gradient à pas fixe :

La méthode dite « à pas fixe » consiste à prendre une valeur de α constante au cours des itérations. Le déplacement étant proportionnel au module du gradient, il diminue automatiquement au voisinage du minimum et on peut espérer converger vers la solution, à condition de choisir une valeur de α correcte. En effet, si α est « trop grand », on peut osciller autour du minimum sans jamais l'atteindre. A l'inverse, si α est « trop petit », on est sûr de converger vers le minimum, mais avec un nombre d'itérations inutilement grand. En pratique, à moins que l'on dispose d'informations sur la dérivée seconde de la fonction, la détermination de α est empirique (essais-erreurs). Pour un problème donné, on teste différentes valeurs de α jusqu'à trouver quelque chose de satisfaisant.

Dans le contexte de l'apprentissage automatique (machine learning), le paramètre α est appelé facteur d'apprentissage (learning rate) car il pondère l'importance de l'information apprise (le gradient de la fonction) à l'itération courante.

[1] C'est-à-dire que le vecteur gradient est défini en tous points

Travail à réaliser :

A. Implanter la méthode du gradient à pas fixe

Schéma général de l'algorithme :

- Notations :
  - X_0 : point de départ de l'algorithme
  - X_n et X_{n+1} : point initial et point final de l'itération n.
  - α : pas de recherche (ou facteur d'apprentissage)
  - dX : distance entre X_n et X_{n+1}
  - n et n_{max} : compteur d'itérations et nombre maximal d'itérations autorisé
  - ε : critère de précision
  - $converge$: indicateur booléen de convergence
- Algorithme :
  - Choix des paramètres de l'algorithme : X_0, α, ε, n_{max}
  - Initialisation : $X_n \leftarrow X_0$, $n \leftarrow 0$, $converge \leftarrow faux$
  - Tant que $\overline{converge}$ et $n < n_{max}$:
    - $X_{n+1} \leftarrow X_n - \alpha.\nabla J(X_n)$
    - $dX = \|X_{n+1} - X_n\|$
    - $X_n \leftarrow X_{n+1}$
    - $n \leftarrow n + 1$
    - $converge \leftarrow (dX \leq \varepsilon)$

Cahier des charges :

- Les expressions analytiques de J et son gradient ∇J sont connues (calculées au préalable « à la main ») et programmées dans deux fonctions distinctes.
- L'algorithme de recherche de minimum est implémenté dans une fonction nommée `descente_gradient_fixe` dont les paramètres d'entrée sont : le nom de la fonction à minimiser, le nom de la fonction gradient, le point de départ de l'algorithme, la valeur de α, la précision souhaitée ε et le nombre maximal d'itérations autorisées n_{max}. La fonction `descente_gradient_fixe` renvoie la liste des points testés $[X_0, X_1, \dots]$ et un indicateur booléen qui indique si l'algorithme à convergé.
- La fonction `descente_gradient_fixe` est appelée depuis un programme principal dans lequel les paramètres d'appel sont définis, puis les résultats sont affichés.
- Les résultats sont affichés sous deux formes : i/ message textuel indiquant le point de départ, le point d'arrivé, le nombre d'itérations et l'indicateur de convergence, ii/ graphe sur lequel la trajectoire de recherche $[X_0, X_1, \dots]$ est superposée aux isovaleurs de la fonction minimisée (voir slide 44 du cours 1).

Fonction à étudier :

- La fonction à minimiser a pour expression : $J(X) = (x_1 - 1)^2 + 2(x_1^2 - x_2)^2$.
- Dans un premier temps, vérifier que le mécanisme itératif est correctement implanté. Pour cela, faire 2 itérations sans tester le critère d'arrêt et vérifier que les résultats sont logiques : le vecteur $\overrightarrow{X_n X_{n+1}}$ doit être normal à l'isovaleur de J passant par X_n et parallèle à $-\vec{\nabla} J(X_n)$. Faire ces premiers tests en commençant avec une « petite » valeur de α et pour différents points de départ.
- Une fois le mécanisme itératif validé, implémenter le critère d'arrêt. Tester l'algorithme pour différente valeurs de α et de ε. Tests proposés : point de départ $X_0 = (-1; 2)$, valeurs de α : 0.1 / 0.05 / 0.2 , valeurs de ε : 10^{-3}, 10^{-6}, 10^{-10}.
- Mettre en évidence le fait que l'algorithme diverge si α dépasse une certaine valeur, et ne converge pas si ε est « trop petit ». Essayer de comprendre pourquoi.
- Proposer et programmer un mécanisme qui ajuste si besoin la valeur de α donnée par l'utilisateur afin d'assurer qu'on a toujours $J(X_{n+1}) < J(X_n)$. On est alors sûr d'assurer la convergence de l'algorithme.