Méthodes d’optimisation et Apprentissage Automatique

Table des matières

[1. Types de problèmes d’optimisation 1](#_Toc190796431)

[2. Méthodes de résolution d’optimisation 1](#_Toc190796432)

[3. Descente de Gradient 2](#_Toc190796433)

[4. Apprentissage Automatique (Machine Learning) 2](#_Toc190796434)

[5. Deep Learning et Réseaux de Neurones 3](#_Toc190796435)

[6. Algorithmes d’optimisation avancés 4](#_Toc190796436)

[7. Applications des méthodes d’optimisation et de l’apprentissage automatique 4](#_Toc190796437)

[Conclusion 4](#_Toc190796438)

# 1. Types de problèmes d’optimisation

L’optimisation vise à déterminer la meilleure solution possible à un problème sous des contraintes spécifiques. On distingue plusieurs types d’optimisation :

**1.1 Optimisation linéaire**

* La fonction objective et les contraintes sont **linéaires**.
* Exemple : Maximiser un bénéfice ou minimiser un coût sous des ressources limitées.
* Méthodes : Simplex, programmation linéaire en nombres entiers.

**1.2 Optimisation non linéaire**

* La fonction objective ou les contraintes sont **non linéaires**.
* Exemple : Réduction de consommation énergétique dans un réseau.
* Méthodes : Descente de gradient, Newton, Quasi-Newton.

**1.3 Optimisation combinatoire**

* Nécessite de choisir **la meilleure combinaison** d’éléments parmi un **ensemble fini de solutions**.
* Exemple : Problème du voyageur de commerce (TSP).
* Méthodes : Branch and Bound, A\*, Recuit simulé.

**1.4 Optimisation stochastique**

* Certaines variables sont incertaines ou aléatoires.
* Exemple : Gestion de portefeuille boursier.
* Méthodes : Algorithmes génétiques, optimisation par colonies de fourmis.

# 2. Méthodes de résolution d’optimisation

On distingue deux grandes familles de méthodes :

**2.1 Méthodes exactes**

* **Programmation linéaire (Simplex)** : utilisée pour l’optimisation linéaire.
* **Branch and Bound** : utilisée en optimisation combinatoire.
* **Programmation dynamique** : divise un problème en sous-problèmes résolus récursivement.

**2.2 Méthodes approchées**

* **Algorithmes heuristiques** : produisent des solutions rapides mais non garanties optimales (exemple : Hill Climbing).
* **Méta-heuristiques** :
  + Recuit simulé (inspiré du refroidissement des métaux).
  + Algorithmes génétiques (inspirés de la sélection naturelle).
  + Optimisation par essaims de particules (simule un groupe d’individus cherchant collectivement une solution).

# 3. Descente de Gradient

**3.1 Objectif**

La descente de gradient est une méthode utilisée pour trouver le **minimum d’une fonction**.

**3.2 Étapes de l’algorithme**

1. Choisir un point de départ.
2. Calculer le **gradient** (pente) de la fonction.
3. Effectuer un pas dans la direction de la plus forte descente.
4. Répéter l’opération jusqu’à convergence.

**3.3 Variantes de la descente de gradient**

* **Descente de gradient classique** : pas fixe.
* **Gradient stochastique (SGD)** : mise à jour après chaque point de données.
* **Gradient par mini-batch** : mise à jour après un petit groupe de points.

# 4. Apprentissage Automatique (Machine Learning)

L’apprentissage automatique permet à un ordinateur d’apprendre **à partir de données** sans être explicitement programmé.

**4.1 Types d’apprentissage**

1. **Supervisé**
   * Les données d’entraînement incluent des **entrées et des sorties**.
   * Objectif : apprendre une relation entre X et Y.
   * Exemples :
     + Classification (prédire une catégorie, ex: détection de spams).
     + Régression (prédire une valeur, ex: estimation du prix d’une maison).
   * Algorithmes : Arbres de décision, régression linéaire, SVM, réseaux de neurones.
2. **Non supervisé**
   * Les données ne contiennent que des **entrées**, sans sortie associée.
   * Objectif : **découvrir des structures cachées** dans les données.
   * Exemples :
     + Clustering (regroupement de données similaires).
     + Réduction de dimension (ex: PCA).
   * Algorithmes : K-Means, ACP.
3. **Apprentissage par renforcement**
   * Un agent interagit avec son environnement et apprend par **essais-erreurs**.
   * Utilisé dans des applications comme AlphaGo ou les voitures autonomes.
   * Algorithmes : Q-Learning, Deep Q-Networks (DQN).

# 5. Deep Learning et Réseaux de Neurones

Le deep learning est une sous-catégorie du machine learning utilisant des **réseaux de neurones artificiels**.

**5.1 Fonctionnement d’un neurone artificiel**

1. Il reçoit des **entrées** (ex: pixels d’une image).
2. Il effectue une **somme pondérée** des entrées.
3. Il applique une **fonction d’activation** (ex: sigmoïde, ReLU).
4. Il transmet **une sortie**.

**5.2 Types de réseaux de neurones**

* **Perceptron multi-couches (MLP)** : utilisé pour la classification et la régression.
* **Réseaux convolutionnels (CNN)** : utilisés en vision par ordinateur.
* **Réseaux récurrents (RNN, LSTM)** : utilisés en traitement du langage naturel.

# 6. Algorithmes d’optimisation avancés

**6.1 Branch and Bound (B&B)**

* Technique utilisée pour **diviser un problème en sous-problèmes** et éliminer les solutions non prometteuses.
* Appliquée aux problèmes combinatoires (ex: voyageur de commerce).

**6.2 Algorithme A\***

* Recherche du **chemin optimal** dans un graphe en combinant **coût réel et heuristique**.
* Appliqué dans les **GPS, jeux vidéo et robotique**.

# 7. Applications des méthodes d’optimisation et de l’apprentissage automatique

* **Transport et logistique** : optimisation des itinéraires (ex: Uber, Amazon).
* **Finance** : optimisation de portefeuille.
* **IA et robotique** : planification et apprentissage automatique.
* **Industrie** : maximisation de la production sous contraintes.

# Conclusion

Les méthodes d’optimisation et l’apprentissage automatique sont essentielles dans divers domaines, allant de la logistique à la finance en passant par l’intelligence artificielle. Il est important de maîtriser les concepts clés tels que la programmation linéaire, les algorithmes heuristiques, le deep learning et les méthodes de clustering pour résoudre efficacement des problèmes complexes.