

# Optimisation des Hyperparamètres : Analyse et Recommandations

## 1. Analyse de l'Impact des Hyperparamètres dans Différentes Applications

L'optimisation des hyperparamètres joue un rôle clé dans l'amélioration des modèles d'apprentissage automatique, mais son importance varie selon l'application et la nature des données. Nous explorons ici plusieurs cas d'étude :

### 1.1 Classification et Régression

- **Hyperparamètres clés** : *Learning rate, régularisation (L1/L2), profondeur du modèle, nombre d'arbres (Random Forest, Gradient Boosting)*
- **Impact** :
  - Un **learning rate trop élevé** peut empêcher la convergence, tandis qu'un taux trop faible ralentit l'entraînement.
  - Une **régularisation excessive** peut sous-ajuster le modèle en limitant sa capacité d'apprentissage.
- **Exemple** : Comparaison d'un modèle de classification **Logistic Regression** avec et sans régularisation pour observer l'effet du paramètre  $c$  sur la précision.

### 1.2 Vision par Ordinateur (Computer Vision)

- **Hyperparamètres clés** : *Taille des batches, nombre de filtres, fonction d'activation, taux d'apprentissage*
- **Impact** :
  - Une **taille de batch élevée** accélère l'entraînement mais peut réduire la généralisation.
  - Le **nombre de filtres** dans un réseau convolutionnel influe sur la capacité du modèle à capturer des motifs complexes.
- **Exemple** : Étudier l'évolution de la perte (  $loss$  ) et de la précision (  $accuracy$  ) en fonction de la taille des batches sur **CIFAR-10**.

### 1.3 Traitement du Langage Naturel (NLP)

- **Hyperparamètres clés** : *Longueur des séquences, taille des embeddings, taux de dropout, nombre de couches LSTM/Transformer*
- **Impact** :
  - Une **taille de séquence trop courte** peut tronquer des informations importantes.
  - Un **dropout trop élevé** empêche le modèle d'apprendre efficacement.
- **Exemple** : Observer l'effet de la taille des embeddings sur la précision d'un modèle BERT pour la classification de texte.

## 1.4 Détection d'Anomalies

- Hyperparamètres clés : *Nombre de clusters (k-means), profondeur des arbres (Isolation Forest), taux de contamination*
- Impact :
  - Un **nombre de clusters mal choisi** peut affecter la séparation des anomalies et des données normales.
  - Une **profondeur excessive des arbres** peut entraîner un surajustement dans des méthodes comme Isolation Forest.
- Exemple : Expérimenter sur un dataset de **fraude bancaire** et comparer les performances des méthodes supervisées et non supervisées.

## 2. Contexte de Pertinence des Hyperparamètres vs Paramètres Ajustables

Les hyperparamètres sont définis avant l'entraînement et contrôlent l'architecture et l'apprentissage, tandis que les paramètres ajustables sont **appris à partir des données**. Voici quelques distinctions clés :

Critère	Hyperparamètres	Paramètres Ajustables
Définition	Fixés avant l'entraînement	Appris à partir des données
Exemples	Learning rate, nombre de couches	Poids du réseau, coefficients d'une régression
Impact	Affecte la convergence et la généralisation	Influence directement la performance
Optimisation	Grid Search, Random Search, Bayesian Optimization	Backpropagation, descente de gradient

### Cas où les Hyperparamètres sont plus pertinents :

- Petites bases de données** → La régularisation et le dropout sont plus critiques pour éviter le surajustement.
- Modèles complexes** → Réseaux neuronaux profonds nécessitent une optimisation fine des hyperparamètres.
- Systèmes en ligne (Streaming)** → Une mise à jour dynamique des hyperparamètres peut être plus efficace qu'une réestimation périodique des paramètres internes.

## 3. Comparaison des Techniques d'Optimisation

L'optimisation des hyperparamètres peut être réalisée par plusieurs méthodes, chacune avec ses avantages et ses limites.

Méthode	Avantages	Inconvénients
Grid Search	Exploration exhaustive, fiable	Coût computationnel élevé
Random Search	Moins coûteux, efficace sur grands espaces	Ne garantit pas la meilleure solution
Bayesian Optimization	Recherche intelligente, efficace	Implémentation plus complexe
Descente de Gradient	Adaptatif, rapide	Peut converger vers un minimum local

Comparaison expérimentale :

- **Problème** : Classification MNIST avec un MLP
  - **Objectif** : Trouver la meilleure combinaison (learning rate, batch size)
  - **Résultats attendus** :
    - Grid Search : Résultats précis mais coûteux.
    - Random Search : Plus rapide, efficace pour des espaces larges.
    - Bayesian Optimization : Meilleur équilibre entre performance et coût.
- 

4. Visualisation de l’Impact des Optimisations

Les outils de visualisation sont essentiels pour comprendre et justifier les choix d’optimisation.

Techniques recommandées :

1. **Cartes de chaleur (Heatmaps)** → Visualiser l’impact combiné de plusieurs hyperparamètres.
2. **Courbes d’apprentissage** → Suivre la perte et l’accuracy au fil des itérations.
3. **Diagrammes de distribution** → Étudier la répartition des poids avec différentes valeurs de régularisation.
4. **Graphiques comparatifs** → Comparer les performances entre différentes techniques d’optimisation.