

Liste de Projets du Module Algorithmes d'optimisation

Responsables : Pr. F. Benabbou, Dr. Z. Ellaky

Filière : Master DSBD M1.

Date de remise 25 juin.

Livrables :

1. Rapport Word

Un document en anglais structuré (~10 à 15 pages) contenant selon l'exemple fournit:

1. **Introduction**
 - Objectif du projet
 - Brève présentation des optimiseurs et des architectures
2. **Revue de Littérature**
3. **Méthodologie**
 - Description des modèles, des jeux de données, des outils utilisés
 - Protocole expérimental : nombre de runs, réglages des hyperparamètres, etc.
4. **Résultats**
 - Tableaux comparatifs
 - Graphiques (courbes de précision, perte, temps de convergence, mémoire, etc.)
5. **Discussions**
 - Interprétation des résultats
 - Discussion sur les forces/faiblesses de chaque optimiseur
6. **Conclusion**
 - Synthèse des observations
 - Recommandations
7. **Bibliographie**
 - Utiliser Zotero pour une meilleur gestion des références.
8. **Annexes**
 - Lien vers le dépôt de code dans github, gitlab,...
 - Un notebook (Jupyter ou Google Colab) montrant un exemple d'entraînement et de visualisation des résultats et les différentes courbes.

2. Présentation orale de 10 min(slides en PPTX)

Listes de Projets

NB : un projet peut être choisi au plus par 2 groupes

1. Comparaison des optimiseurs pour l'entraînement des réseaux de neurones profonds

Objectif : Comparer les performances de SGD, Adam, RMSprop, Adagrad et Nadam sur un réseau de neurones profond CNN (ex. ResNet-18/34, VGG-16, DenseNet) ou Transformer pour vision (ex. ViT, DeiT)) appliqué à la classification d'images.

Données : CIFAR-10, CIFAR-100 ou ImageNet

Métriques : Temps de convergence, précision, stabilité, consommation mémoire.

2. Impact des algorithmes d'optimisation sur les modèles de régression (ML vs DL)

Objectif : Evaluer L-BFGS, SGD avec momentum, Adam et Newton-CG sur :

- Un modèle de régression linéaire (ML)
- Un réseau de neurones à une couche cachée (DL)

Données : Boston Housing, California Housing.

Critères : Vitesse de convergence, généralisation, sensibilité aux hyperparamètres.

3. Optimisation pour l'apprentissage fédéré (Federated Learning)

Objectif : Comparer FedAvg, FedAdam, FedProx et SGD local dans un cadre distribué (simulé avec PyTorch ou TensorFlow Federated) pour une tâche de classification avec clients hétérogènes.

Données : Classification sur MNIST ou CIFAR-10.

Métriques : Robustesse, communication overhead, précision globale, scalabilité.

4. Optimisation stochastique pour les modèles de recommandation

Objectif : Évaluer Adagrad, Adam, FTRL (Follow-the-Regularized-Leader) et SGD sur une tâche de recommandation (matrice de factorisation ou modèle Deep Learning comme Neural Collaborative Filtering).

Données : MovieLens ou Netflix Prize.

Critères : Précision (RMSE, MAE), temps d'entraînement, scalabilité.

5. Algorithmes d'optimisation pour les GANs (Generative Adversarial Networks)

Objectif : Analyser comment Adam, RMSprop, SGD avec momentum et Lookahead impactent la stabilité et la qualité des GANs (DCGAN ou WGAN).

Critère : Fréquence des modes collapse, score FID (Frechet Inception Distance), diversité des échantillons générés.

6. Optimisation convexe vs non convexe en SVM et réseaux de neurones

Objectif : Comparer SMO (Sequential Minimal Optimization) pour les SVM et Adam/SGD pour un perceptron multicouche (MLP). La comparaison se fait sur un problème de classification binaire à partir de données non linéairement séparables.

Données : MNIST (classification binaire : chiffres 3 vs 8).

Critères : Temps d'entraînement, précision, robustesse, complexité algorithmique.

7. Comparaison des méthodes d'optimisation pour le fine-tuning de modèles pré-entraînés (BERT, GPT)

Objectif : évaluer AdamW, LAMB, SGD avec warmup et AdaFactor sur le fine-tuning de BERT pour une tâche de NLP (classification de texte ou Q&A).

Données : GLUE (MRPC, SST-2) ou SQuAD.

Critères : Précision, stabilité, consommation de ressources.

8. Optimisation dans les architectures à mémoire limitée (méthodes 1st-order vs 2nd-order)

Objectif : algorithmes d'optimisation de descente de gradient, de type 1er ordre (SGD, Adam) et 2e ordre (Shampoo, K-FAC), dans un contexte contraint en (simulé avec gradient checkpointing) pour l'entraînement d'un ResNet-18 avec mémoire limitée.

Données : CIFAR-10

Critères : Temps d'entraînement, Précision, stabilité, consommation mémoire, consommation CPU/GPU.

9. Méta-optimisation : Hyperparamètres des optimiseurs par Bayesian Optimization

Objectif : Utiliser Bayesian Optimization ou Hyperband pour trouver les meilleurs hyperparamètres (learning rate, momentum) pour Adam, SGD, et RMSprop sur un modèle de prédiction de séries temporelles (LSTM ou Time Series Transformer) en utilisant des méthodes de recherche d'hyperparamètres intelligentes comme : Bayesian Optimization et Hyperband

Données : financières (Cryptomonnaies, yFinance, etc..) ou météo (Weather dataset de Kaggle).

Critères : Temps d'entraînement, Précision, stabilité, consommation de ressources.

10. Optimisation distribuée : Comparaison entre AllReduce, Horovod et Parameter Server.

Objectif : Évaluer l'efficacité de différentes stratégies distribuées couplées à SGD, Adam dans l'entraînement d'un grand modèle (Vision Transformer (ViT) ou LLM) comme GPT,

BERT sur plusieurs GPU.

Données : Tiny ImageNet ou WikiText-2, WikiText-103 .

Critère : Scalabilité % nombre GPU, overhead de communication, accélération (temps, throughput).

11. Optimisation des tournées de collecte des déchets à Casablanca via des algorithmes combinatoires avancés.

Objectif : Casablanca doit organiser efficacement les tournées de ses camions de collecte de déchets. L'objectif est de déterminer les itinéraires optimaux des camions afin de :

- ✓ visiter chaque quartier exactement une fois,
- ✓ ne pas dépasser la capacité de chaque camion,
- ✓ minimiser la distance totale parcourue.

Extraire les données topologiques via l'API OpenStreetMapOn et modéliser le réseau de la ville comme un graphe, où :

- chaque quartier est représenté par un nœud,
- les distances entre quartiers sont les arêtes pondérées du graphe,
- chaque quartier génère une quantité fixe de déchets (connue à l'avance),
- chaque camion part d'un dépôt central et a une capacité limitée.

Hypothèses

- ✓ Un seul dépôt (point de départ/arrivée commun),
- ✓ Quantité de déchets fixe et connue pour chaque quartier,
- ✓ Tous les quartiers sont accessibles à tout moment (pas de fenêtres de temps),
- ✓ Tous les camions sont identiques et disponibles dès le départ, et ont une capacité connue.

Critères de performance :

- ✓ Nombre de camions utilisés
- ✓ Distance totale parcourue
- ✓ Temps de convergence
- ✓ Temps de calcul pour trouver une solution (complexité).
- ✓ Scalabilité