

# REBUBLIQUE DU SENEGAL

\*\*\*\*\*

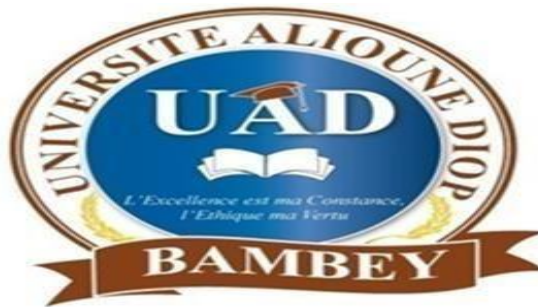
**Un Peuple – Un But –Une Foi**



**Ministère de l'Enseignement supérieur et de la Recherche**

\*\*\*\*\*

**Université Alioune Diop de Bambey**



**UFR : Sciences Appliquées & Technologie de l'Information et de la Communication**

**Département : Technologie de l'Information et de la Communication**

**Filière : SR et SI**

**OPTIMMISEUR ADADELTA**

**Présenté par :**

Groupe4

**Sous la direction de**

Dr Nouru Sylla

## **Approche Thématique:**

- I. Historique de l'algorithme
- II. Objectif principal
- III. Fonctionnement
- IV. Cas d'application
- V. Interprétation
- VI. Conclusion

### **I. Historique de l'algorithme d'optimisation Ada delta**

L'algorithme d'optimisation Ada delta est une méthode populaire utilisée dans le domaine de l'apprentissage automatique et du deep Learning. Il a été proposé en 2012 par Matthew Zeiler, chercheur en intelligence artificielle et fondateur de la société Clarifai.

Ada delta est une extension de l'algorithme d'optimisation du gradient stochastique (SGD) qui vise à atténuer certains de ses inconvénients, tels que la nécessité de choisir manuellement un taux d'apprentissage adapté. L'idée principale derrière Ada delta est d'ajuster automatiquement le taux d'apprentissage en fonction des caractéristiques de l'ensemble de données et de la progression de l'entraînement.

L'algorithme Ada delta utilise un mécanisme de mise à jour adaptative pour ajuster les taux d'apprentissage pour chaque paramètre du modèle de manière individuelle. Contrairement à d'autres méthodes d'optimisation, Ada delta ne nécessite pas de réglage manuel d'un taux d'apprentissage global. Au lieu de cela, il utilise une moyenne mobile exponentielle des carrés des gradients précédents pour normaliser la mise à jour du paramètre.

L'algorithme tire son nom du terme grec "ada delta", qui signifie "delta adaptatif". Le "delta" dans le nom fait référence à la façon dont les mises à jour des paramètres sont calculées en utilisant une estimation locale de l'échelle des gradients.

L'introduction de l'algorithme Ada delta a contribué à améliorer les performances des modèles d'apprentissage automatique en permettant une convergence plus rapide et plus stable lors de l'entraînement. Depuis sa proposition en 2012, Ada delta est devenu l'une des méthodes d'optimisation les plus utilisées dans les réseaux de neurones profonds et a été adopté par de nombreux Framework d'apprentissage automatique populaires.

Il convient de noter que l'algorithme Ada delta fait partie d'une famille plus large d'algorithmes d'optimisation adaptative, tels que Adam (Adaptive Moment Estimation) et RMSprop (Root Mean Square Propagation), qui utilisent des techniques similaires pour ajuster les taux d'apprentissage de manière adaptative. Ces méthodes ont joué un rôle essentiel dans l'avancement des techniques d'apprentissage profond et ont permis de réaliser des progrès significatifs dans de nombreux domaines de l'intelligence artificielle.

## II. Les objectifs

L'algorithme d'optimisation Ada delta a plusieurs objectifs :

1. Éliminer le besoin de régler manuellement le taux d'apprentissage : L'un des principaux problèmes avec l'algorithme de gradient stochastique standard (SGD) est la nécessité de choisir un taux d'apprentissage approprié. Un taux d'apprentissage trop élevé peut conduire à des oscillations ou à une divergence de l'algorithme, tandis qu'un taux d'apprentissage trop faible peut entraîner une convergence lente. Ada delta vise à résoudre ce problème en ajustant automatiquement le taux d'apprentissage en fonction des caractéristiques de l'ensemble de données et de la progression de l'entraînement.
2. Convergence plus rapide : Ada delta utilise une technique appelée moyenne mobile exponentielle pour normaliser les mises à jour des paramètres. Cette normalisation permet une convergence plus rapide en évitant les grands écarts de mise à jour des paramètres qui peuvent ralentir le processus d'optimisation.
3. Stabilité de l'optimisation : L'algorithme Ada delta est conçu pour être plus stable que le SGD standard. Il utilise une estimation locale de l'échelle des gradients pour ajuster les taux d'apprentissage de manière adaptative. Cette approche aide à maintenir une progression régulière de l'optimisation, évitant ainsi les sauts brusques et les oscillations indésirables.
4. Utilisation efficace de la mémoire : Ada delta maintient une fenêtre glissante de la moyenne mobile exponentielle des carrés des gradients précédents. Cette approche permet de réduire la quantité de mémoire nécessaire pour stocker les gradients précédents, ce qui est particulièrement important lorsque l'on travaille avec de grands ensembles de données ou des modèles complexes.

En résumé, les objectifs de l'algorithme Ada delta sont d'éliminer le réglage manuel du taux d'apprentissage, d'accélérer la convergence, de maintenir la stabilité de l'optimisation et d'utiliser efficacement la mémoire. Ces caractéristiques en font une méthode d'optimisation populaire dans le domaine de l'apprentissage automatique et du deep Learning.

Cependant, l'algorithme bien qu'il soit efficace, a des limites. Ainsi on peut citer quelques inconvénients de celui-ci :

1. Sensibilité aux hyper paramètres : Ada delta comporte certains hyper paramètres, tels que le paramètre de décroissance et l'ajustement initial du taux d'apprentissage, qui doivent être réglés de manière appropriée pour obtenir de bonnes performances. Trouver les bonnes valeurs d'hyper paramètres peut nécessiter des expérimentations et une recherche approfondie.
2. Performance sensible aux jeux de données : Les performances d'Ada delta peuvent varier selon les caractéristiques du jeu de données. Il peut ne pas être aussi efficace que d'autres optimiseurs dans certains contextes spécifiques.

## III. Fonctionnement

L'algorithme Ada Delta est une méthode d'optimisation adaptative utilisée pour entraîner des réseaux de neurones profonds dans le cadre de l'apprentissage profond (deep Learning). Il a été proposé par Matthew D. Zeiler en 2012 et améliore certains inconvénients de l'algorithme AdaGrad.

L'idée principale derrière l'algorithme Ada Delta est d'ajuster dynamiquement le taux d'apprentissage (learning rate) en fonction des mises à jour des poids du réseau, ce qui permet

d'atténuer les problèmes de convergence lente ou d'oscillations lors de l'entraînement. En effet, dans de nombreux cas, un taux d'apprentissage fixe peut être problématique, car il peut entraîner des mises à jour de poids trop grandes ou trop petites, ce qui peut ralentir la convergence ou même empêcher la convergence du modèle.

Le fonctionnement de l'algorithme Ada Delta est le suivant :

1. Calcul des moyennes mobiles exponentielles des gradients précédents :

\_\_- À chaque itération de l'entraînement, l'algorithme calcule une moyenne mobile exponentielle des carrés des gradients précédents. Cette moyenne est similaire à celle utilisée dans l'algorithme RMSProp.

2. Calcul des mises à jour des poids :

\_\_- Ada Delta utilise cette moyenne mobile des carrés des gradients pour calculer la racine carrée de la moyenne mobile des mises à jour précédentes des poids. Ce calcul permet d'obtenir une estimation du "taux d'apprentissage effectif" pour chaque poids du réseau.

3. Calcul des mises à jour des gradients :

\_\_- Pour chaque poids du réseau, AdaDelta calcule une nouvelle estimation du gradient en prenant en compte le taux d'apprentissage effectif calculé précédemment. Cela permet de réguler la taille des mises à jour des poids en fonction de l'historique des mises à jour précédentes.

4. Mise à jour des poids du réseau :

\_\_- Enfin, les poids du réseau sont mis à jour en utilisant les nouveaux gradients calculés.

L'un des avantages d'Ada Delta est qu'il n'a pas besoin de spécifier un taux d'apprentissage global, ce qui rend l'algorithme plus robuste et moins sensible aux choix de paramètres. De plus, Ada Delta atténue le problème de la diminution du taux d'apprentissage au fil du temps, ce qui peut être un problème avec certaines méthodes d'optimisation.

En résumé, l'algorithme Ada Delta est une méthode d'optimisation adaptative qui ajuste dynamiquement le taux d'apprentissage en fonction de l'historique des mises à jour des poids, ce qui permet de mieux réguler la taille des mises à jour et d'améliorer la convergence lors de l'entraînement de réseaux de neurones profonds.

L'un des avantages d'Ada Delta est qu'il n'a pas besoin de spécifier un taux d'apprentissage global, ce qui rend l'algorithme plus robuste et moins sensible aux choix de paramètres. De plus, Ada Delta atténue le problème de la diminution du taux d'apprentissage au fil du temps, ce qui peut être un problème avec certaines méthodes d'optimisation.

En résumé, l'algorithme Ada Delta est une méthode d'optimisation adaptative qui ajuste dynamiquement le taux d'apprentissage en fonction de l'historique des mises à jour des poids, ce qui permet de mieux réguler la taille des mises à jour et d'améliorer la convergence lors de l'entraînement de réseaux de neurones profonds.

- Il est important de noter que les résultats obtenus peuvent varier d'une exécution à l'autre en raison de l'aléa initial de l'entraînement et des variations dans

la répartition des données lors de la division en ensembles d'entraînement et de test.

- 
- En conclusion, les résultats actuels du modèle utilisant l'optimiseur ADADELTA indiquent des performances insatisfaisantes en termes d'accuracy et de perte. Des améliorations supplémentaires sont nécessaires pour obtenir un modèle plus précis et fiable dans la tâche de classification du diabète.

## V) Cas D'application (voir Code)

## VI. Conclusion

Ada Delta est une amélioration de l'algorithme AdaGrad qui adresse le problème de décroissance du taux d'apprentissage. Il est efficace pour optimiser les réseaux neuronaux dans des tâches d'apprentissage profond. Cependant, il existe d'autres algorithmes d'optimisation adaptatifs tels que Adam qui peuvent également être utilisés en fonction des caractéristiques spécifiques du problème d'apprentissage