**From https://machinelearningmastery.com/difference-between-backpropagation-and-stochastic-gradient-descent/**

**Introduction**

Il existe beaucoup de confusion chez les débutants concernant l’algorithme utilisé pour entraîner les modèles de réseaux de neurones profonds.

Il est courant d’entendre que les réseaux de neurones apprennent à l’aide de l’algorithme de « rétropropagation de l’erreur » ou de la « descente de gradient stochastique ». Parfois, l’un ou l’autre de ces algorithmes est utilisé comme raccourci pour décrire la manière dont un réseau de neurones est ajusté sur un jeu de données d’entraînement, bien que, dans de nombreux cas, il existe une réelle confusion sur ce que sont ces algorithmes, comment ils sont liés et comment ils fonctionnent ensemble.

Ce tutoriel est conçu pour clarifier le rôle des algorithmes de descente de gradient stochastique et de rétropropagation dans l’entraînement des réseaux de neurones.

Dans ce tutoriel, vous découvrirez la différence entre la descente de gradient stochastique et l’algorithme de rétropropagation.

Après avoir terminé ce tutoriel, vous saurez :

* La descente de gradient stochastique est un algorithme d’optimisation pour minimiser la perte d’un modèle prédictif par rapport à un jeu de données d’entraînement.
* La rétropropagation est un algorithme de différentiation automatique pour calculer les gradients des poids dans la structure d’un réseau de neurones.
* La descente de gradient stochastique et la rétropropagation de l’erreur sont utilisées ensemble pour entraîner les modèles de réseaux de neurones.

Commençons.

**Aperçu du tutoriel**

Ce tutoriel est divisé en trois parties :

* Descente de gradient stochastique
* Algorithme de rétropropagation
* Descente de gradient stochastique avec rétropropagation

**Descente de gradient stochastique**

La **descente de gradient** est un algorithme d’optimisation qui trouve l’ensemble des variables d’entrée d’une fonction cible qui aboutit à une valeur minimale de cette fonction, appelée le minimum de la fonction.

Comme son nom l’indique, la descente de gradient implique le calcul du gradient de la fonction cible.

Vous vous souvenez peut-être du calcul différentiel que la dérivée première d’une fonction calcule la pente ou la courbure d’une fonction en un point donné. De gauche à droite, une dérivée positive suggère que la fonction cible monte et une dérivée négative suggère qu’elle descend.

**Dérivée** : pente ou courbure d’une fonction cible par rapport à des valeurs d’entrée spécifiques de la fonction.

Si la fonction cible prend plusieurs variables d’entrée, elles peuvent être considérées ensemble comme un vecteur de variables. Travailler avec des vecteurs et des matrices relève de l’algèbre linéaire, et faire du calcul différentiel avec ces structures s’appelle le calcul matriciel ou vectoriel. En calcul vectoriel, le vecteur des dérivées premières (dérivées partielles) est généralement appelé le gradient de la fonction cible.

**Gradient** : vecteur des dérivées partielles d’une fonction cible par rapport aux variables d’entrée.

L’algorithme de descente de gradient nécessite le calcul du gradient de la fonction cible par rapport aux valeurs spécifiques des variables d’entrée. Le gradient pointe vers le haut, donc on suit le négatif du gradient de chaque variable d’entrée pour obtenir de nouvelles valeurs qui aboutissent à une évaluation plus basse de la fonction cible.

Une taille de pas est utilisée pour ajuster le gradient et contrôler de combien chaque variable d’entrée doit être modifiée par rapport au gradient.

**Taille de pas** : taux d’apprentissage ou alpha, un hyperparamètre utilisé pour contrôler de combien chaque variable d’entrée est modifiée par rapport au gradient.

Ce processus est répété jusqu’à ce que le minimum de la fonction cible soit atteint, qu’un nombre maximal de solutions candidates soit évalué, ou qu’une autre condition d’arrêt soit remplie.

La descente de gradient peut être adaptée pour minimiser la fonction de perte d’un modèle prédictif sur un jeu de données d’entraînement, comme un modèle de classification ou de régression. Cette adaptation s’appelle la descente de gradient stochastique.

**Descente de gradient stochastique** : extension de l’algorithme de descente de gradient pour minimiser une fonction de perte d’un modèle prédictif sur un jeu de données d’entraînement.

La fonction cible est prise comme la fonction de perte ou d’erreur sur le jeu de données, comme l’erreur quadratique moyenne pour la régression ou l’entropie croisée pour la classification. Les paramètres du modèle sont considérés comme les variables d’entrée de la fonction cible.

**Fonction de perte** : fonction cible à minimiser. **Paramètres du modèle** : paramètres d’entrée de la fonction de perte à optimiser.

L’algorithme est dit « stochastique » car les gradients de la fonction cible par rapport aux variables d’entrée sont bruités (ex. : une approximation probabiliste). Cela signifie que l’évaluation du gradient peut comporter du bruit statistique qui peut masquer le véritable signal du gradient, à cause de la rareté et du bruit dans le jeu de données d’entraînement.

L’idée clé de la descente de gradient stochastique est que le gradient est une espérance. L’espérance peut être approximée à l’aide d’un petit ensemble d’échantillons.

— Page 151, Deep Learning, 2016.

La descente de gradient stochastique peut être utilisée pour entraîner (optimiser) de nombreux types de modèles, comme la régression linéaire et la régression logistique, même si souvent des algorithmes d’optimisation plus efficaces ont été découverts et devraient probablement être utilisés à la place.

La descente de gradient stochastique (SGD) et ses variantes sont probablement les algorithmes d’optimisation les plus utilisés en apprentissage automatique en général et en apprentissage profond en particulier.

— Page 294, Deep Learning, 2016.

La descente de gradient stochastique est l’algorithme le plus efficace découvert pour entraîner les réseaux de neurones artificiels, où les poids sont les paramètres du modèle et la fonction de perte cible est l’erreur de prédiction moyennée sur un, un sous-ensemble (batch) ou l’ensemble du jeu de données d’entraînement.

Presque tout l’apprentissage profond repose sur un algorithme très important : la descente de gradient stochastique ou SGD.

— Page 151, Deep Learning, 2016.

Il existe de nombreuses extensions populaires à la descente de gradient stochastique, conçues pour améliorer le processus d’optimisation (même perte ou meilleure en moins d’itérations), telles que Momentum, RMSProp (Root Mean Squared Propagation) et Adam (Adaptive Movement Estimation).

Un défi lors de l’utilisation de la descente de gradient stochastique pour entraîner un réseau de neurones est de savoir comment calculer le gradient pour les nœuds des couches cachées du réseau, c’est-à-dire les nœuds situés à une ou plusieurs étapes de la couche de sortie du modèle.

Cela nécessite une technique spécifique du calcul différentiel appelée la règle de la chaîne et un algorithme efficace qui implémente cette règle pour calculer les gradients de tout paramètre du réseau. Cet algorithme s’appelle la rétropropagation.

**Algorithme de rétropropagation**

La rétropropagation, aussi appelée « backpropagation » ou simplement « backprop », est un algorithme pour calculer le gradient d’une fonction de perte par rapport aux variables d’un modèle.

**Rétropropagation** : algorithme pour calculer le gradient d’une fonction de perte par rapport aux variables d’un modèle.

Vous vous souvenez peut-être du calcul différentiel que la dérivée première d’une fonction pour une valeur spécifique d’une variable d’entrée est le taux de variation ou la courbure de la fonction pour cette variable. Lorsque l’on a plusieurs variables d’entrée pour une fonction, elles forment un vecteur, et le vecteur des dérivées premières (dérivées partielles) est appelé le gradient (c’est-à-dire le calcul vectoriel).

**Gradient** : vecteur des dérivées partielles de valeurs d’entrée spécifiques par rapport à une fonction cible.

La rétropropagation est utilisée lors de l’entraînement de modèles de réseaux de neurones pour calculer le gradient de chaque poids du modèle. Le gradient est ensuite utilisé par un algorithme d’optimisation pour mettre à jour les poids du modèle.

L’algorithme a été développé explicitement pour calculer les gradients des variables dans des structures de graphe en travaillant à rebours depuis la sortie du graphe vers l’entrée, en propageant l’erreur de la sortie prédite qui est utilisée pour calculer le gradient de chaque variable.

L’algorithme de rétropropagation, souvent simplement appelé backprop, permet à l’information issue du coût de circuler ensuite en arrière dans le réseau, afin de calculer le gradient.

— Page 204, Deep Learning, 2016.

La fonction de perte représente l’erreur du modèle ou la fonction d’erreur, les poids sont les variables de la fonction, et les gradients de la fonction d’erreur par rapport aux poids sont donc appelés gradients d’erreur.

**Fonction d’erreur** : fonction de perte à minimiser lors de l’entraînement d’un réseau de neurones. **Poids** : paramètres du réseau considérés comme valeurs d’entrée de la fonction de perte. **Gradients d’erreur** : dérivées premières de la fonction de perte par rapport aux paramètres.

C’est ce qui donne à l’algorithme son nom « rétropropagation », ou parfois « rétropropagation de l’erreur ». **Rétropropagation de l’erreur** : commentaire sur la façon dont les gradients sont calculés récursivement en arrière à travers le graphe du réseau à partir de la couche de sortie.

L’algorithme implique l’application récursive de la règle de la chaîne du calcul différentiel (différente de la règle de la chaîne en probabilité) utilisée pour calculer la dérivée d’une sous-fonction étant donné la dérivée de la fonction parente dont la dérivée est connue.

La règle de la chaîne du calcul différentiel […] est utilisée pour calculer les dérivées de fonctions composées d’autres fonctions dont les dérivées sont connues. La rétropropagation est un algorithme qui calcule la règle de la chaîne, avec un ordre d’opérations spécifique qui est très efficace.

**Descente de gradient stochastique avec rétropropagation**

La descente de gradient stochastique est un algorithme d’optimisation qui peut être utilisé pour entraîner les modèles de réseaux de neurones.

L’algorithme de descente de gradient stochastique nécessite que les gradients soient calculés pour chaque variable du modèle afin que de nouvelles valeurs pour ces variables puissent être calculées.

La rétropropagation est un algorithme de différentiation automatique qui peut être utilisé pour calculer les gradients des paramètres dans les réseaux de neurones.

Ensemble, l’algorithme de rétropropagation et l’algorithme de descente de gradient stochastique peuvent être utilisés pour entraîner un réseau de neurones. On peut appeler cela « descente de gradient stochastique avec rétropropagation ».

**Descente de gradient stochastique avec rétropropagation** : une description plus complète de l’algorithme général utilisé pour entraîner un réseau de neurones, faisant référence à l’algorithme d’optimisation et à l’algorithme de calcul du gradient.

Il est courant que les praticiens disent qu’ils entraînent leur modèle avec la rétropropagation. Techniquement, c’est incorrect. Même comme raccourci, ce serait incorrect. La rétropropagation n’est pas un algorithme d’optimisation et ne peut pas être utilisée pour entraîner un modèle.

Le terme rétropropagation est souvent mal compris comme désignant l’ensemble de l’algorithme d’apprentissage pour les réseaux de neurones multicouches. En réalité, la rétropropagation ne fait que calculer le gradient, tandis qu’un autre algorithme, comme la descente de gradient stochastique, est utilisé pour réaliser l’apprentissage à l’aide de ce gradient.

— Page 204, Deep Learning, 2016.

Il serait juste de dire qu’un réseau de neurones est entraîné ou apprend avec la descente de gradient stochastique comme raccourci, car il est supposé que l’algorithme de rétropropagation est utilisé pour calculer les gradients dans le cadre de la procédure d’optimisation.

Cela dit, un autre algorithme peut être utilisé pour optimiser les paramètres d’un réseau de neurones, comme un algorithme génétique qui ne nécessite pas de gradients. Si l’algorithme d’optimisation utilisé est la descente de gradient stochastique, un algorithme différent peut être utilisé pour calculer les gradients de la fonction de perte par rapport aux paramètres du modèle, comme d’autres algorithmes qui implémentent la règle de la chaîne.

Néanmoins, la combinaison « descente de gradient stochastique avec rétropropagation » est largement utilisée car c’est l’approche générale la plus efficace et efficiente développée à ce jour pour ajuster les modèles de réseaux de neurones.

**Résumé**

Dans ce tutoriel, vous avez découvert la différence entre la descente de gradient stochastique et l’algorithme de rétropropagation.

Plus précisément, vous avez appris :

* La descente de gradient stochastique est un algorithme d’optimisation pour minimiser la perte d’un modèle prédictif par rapport à un jeu de données d’entraînement.
* La rétropropagation est un algorithme de différentiation automatique pour calculer les gradients des poids dans la structure d’un réseau de neurones.
* La descente de gradient stochastique et la rétropropagation de l’erreur sont utilisées ensemble pour entraîner les modèles de réseaux de neurones.