

STAGES M2- INFORMATIQUE

Description Scientifique

ACRONYME et titre du projet : Amélioration des performances de la rétro propagation dans les réseaux de neurones

Noms et coordonnées des porteurs : Marot M. Afifi H¹, T. Bonald, J Badosa, H MOUNGLA¹
contact : hossam.afifi@telecom-sudparis.eu,

Laboratoires ou équipes :

(Coordonnées des laboratoires ou équipes de l'institut de convergence qui bénéficieront des financements)

**Laboratoire Samovar, département RST , Télécom
SudParis, IPP;
Laboratoire LTCI, TP
Laboratoire LMD , X**

Description :

Introduction

Le but du stage est d'étudier les solutions pour accélérer l'apprentissage d'un réseau de neurones en général. Nous nous focaliserons sur les RNNs en premier lieu.

Un réseau de neurones fonctionne en deux phases. D'abord dans le sens 'avant', il exécute des calculs simples mais massifs pour sortir un résultat. Ce résultat est évalué et comparé par une fonction de coût à un résultat attendu. La différence entre le résultat de ces calculs simples et du résultat attendu est utilisée dans la seconde phase.

La seconde phase est la plus complexe en temps et en calcul car il est nécessaire d'ajuster tous les poids de chaque neurone par rapport à l'erreur. Il s'agit de la rétro propagation.

Etat de l'art

Nous utilisons dans la plupart du temps un algorithme Canadien pour la rétro propagation. ADAM optimizer est le nom de ce système. Il est implémenté dans Tensorflow, la librairie la plus courante dans l'apprentissage. ADAM est codé en Python pour différentes familles de réseaux de neurones : les réseaux convolutionnels et les réseaux récurrents. Il est évident que l'approche est totalement différente pour chacune de ces deux familles vu que leurs rétro propagation est différente.

Rétro propagation dans les RNNs (Gradient descent)

Les RNNs servent pour les séries temporelles.

Ils permettent à Google de prédire les mots d'une phrase, à traduire, à corriger. Nous les utilisons dans le cadre de prédiction diverses et variées (météo, consommation, trafic routier, télécom...).

Un RNN de taille moyenne comporte un millier de neurones (gates). Chaque rétro propagation doit ajuster l'erreur pour chaque neurones d'une certaine manière (à reculons)

Cela prend beaucoup de temps. Les équations

1 à 6 correspondent aux paramètres du réseau de la figure ci-contre. Il est nécessaire de dériver ces paramètres par rapport à l'erreur et ajuster les poids pour que le résultat égale à zéro. Il s'agit de la descente du gradient.

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \quad (1)$$

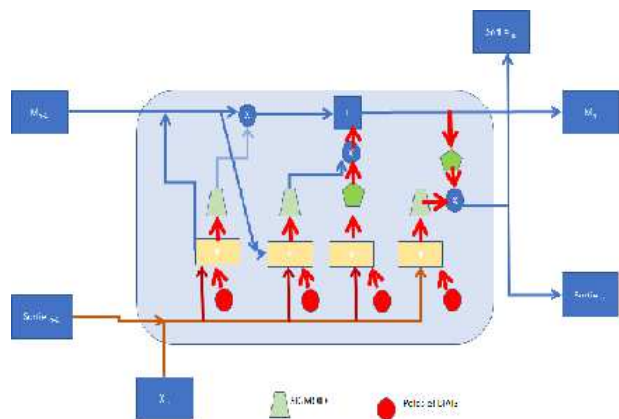
$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i) \quad (2)$$

$$o_t = \tanh(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t \quad (4)$$

$$g_t = \sigma(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (5)$$

$$h_t = o_t \circ \tanh(c_t) \text{ and } z_t = \text{softmax}(W_{hz} h_t + b_z) \quad (6)$$



Rétro propagation dans les CNNs

Les CNNs sont plus courants encore et servent dans le traitement d'image et dans le renforcement. Ils sont plus simples mais largement plus nombreux que les RNNs. Par

exemple, le réseau utilisé dans YOLO comporte 53 couches de neurones. Donc un rétro apprentissage prend énormément de temps

Optimisation des RNNs

Les temps d'apprentissage sont, tout le monde le sait, énormes. On sait que les réseaux sont totalement maillés et donc le nombre d'arcs à optimiser est très grand. On peut faire des calculs pendant une semaine pour certains cas.

On sait aussi que tous les poids ne sont pas nécessaires dans l'amélioration de l'apprentissage. On souhaite donc étudier des méthodes statistiques qui pourraient 'isoler' certains arcs et ne pas les mettre à jour pour améliorer le temps d'apprentissage.

Comment choisir les zones à éliminer est complexe et n'est pas approprié pour un stage court ; cela doit se faire par analyse des zones du réseau de neurones et des résultats.

Nous nous contenterons d'un petit démonstrateur de choix de zones à isoler et des résultats qui en découlent par rapport au temps et au résultat.

Résultats attendus pour le stage

D'abord, le stagiaire doit comprendre le système de gradient des RNNs et plus particulièrement du réseau LSTM (chaque réseau a ses propres formules).

Ensuite il va étudier ADAM dans le cadre des RNNs et dans TensorFlow 1.5.

Enfin, il utilisera les séries temporelles d'énergie du laboratoire LMD (DrahiX) pour voir l'effet d'isolation sur les résultats en temps de calcul et en précision de prédiction.

Références

1 Diederik P. Kingma*University of Amsterdam, OpenAIdpkingma@openai.com Jimmy Lei Ba ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION *University of Toronto

2 Deep Learning Semantic Compression: IoT Support over LORA Use Case
A Dridi, A Debar, V Gauthier, HI Khedher, H Afifi
2019 2nd IEEE Middle East and North Africa COMMunications Conference