Rapport TDL : Étude et reproduction expérimentale de *Catapults in SGD*

Valentin Dugay, Jibril El Hassani

Février 2026

1 Introduction

Ce rapport s'intéresse au phénomène de *catapultes* observé lors de l'optimisation de réseaux de neurones, en particulier avec la descente de gradient stochastique (SGD), tel qu'analysé dans le papier de référence. L'idée centrale est que certains pics transitoires de loss proviennent d'un learning rate trop élevé sur un petit nombre de directions associées à de grandes valeurs propres, et qu'ils peuvent néanmoins améliorer la généralisation.

Dans le notebook associé, nous reproduisons ces effets sur MNIST et évaluons leur impact empirique sur la performance (train/test) en comparant plusieurs optimiseurs : SGD, SGD + Catapults, AdaGrad, AdaGrad + Catapults, et Adam. Nous complétons ces expériences par une analyse dans le régime NTK via le suivi de métriques spectrales (conditionnement et norme spectrale) afin de relier les phases de catapultes à des changements de dynamique. La question principale que nous explorons est : *comment obtenir des catapultes avec AdaGrad, et sont-elles aussi bénéfiques que celles obtenues avec SGD ?*

2 Origine du phénomène

D'après le descent lemma, on a :

$$\mathcal{L}(\mathbf{w}^{t+1}) \leq \mathcal{L}(\mathbf{w}^t) - \eta \left(1 - \frac{\eta\beta}{2}\right) \|\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}^t)\|^2.$$

Avec beta la plus grand valeur propre de la hessienne. On voit que si le learning rate dépasse $\frac{2}{\beta}$, alors la loss va augmenter. Il est important de noter que la loss augmente uniquement dans la direction des valeurs propres pour lesquelles le learning rate est plus grand que le learning critique de 2 divisé par la valeur propre. Ainsi on peut avoir des cas où la loss augmente pour les directions vers lesquelles les valeurs propre de la hessienne sont maximale tout en ayant une diminution de la loss dans les autres directions. Lorsqu'un tel phénomène se produit sans que la loss diverge, on parle de catapulte.

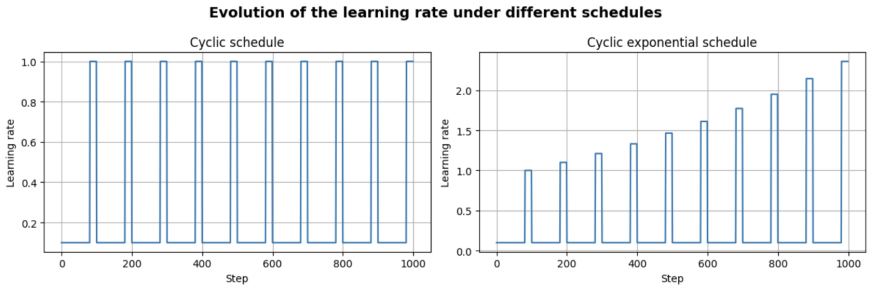

Figure 1: Évolution du learning rate en fonction du step avec un schedule "Cyclique" et un schedule "exponentiel".

3 Méthode de reproduction

Nous détaillons ici notre protocole pour reproduire les catapultes et conduire nos expérimentations. On entraîne un réseau de neurones fully-connected à 2 couches (FC2N) sur MNIST avec un batch size de 1024. Nous utilisons deux schedules du learning rate, présentés dans la Figure 1 : un schedule "Cyclique" et un schedule "Cyclique exponentiel". Le cyclique exponentiel permet d'obtenir un plus grand nombre de catapultes tandis que le cyclique est plus stable.

Dans le notebook, l'objectif principal est d'étudier l'effet des catapultes sur la performance (notamment la généralisation). On met en place ces schedules et on compare, à conditions d'entraînement identiques (mêmes epochs, batch size et protocole), les scénarios suivants : SGD, SGD + Catapults, AdaGrad, AdaGrad + Catapults, Adam. On constate que les catapultes apportent un gain net pour SGD (amélioration de la test accuracy), tandis qu'elles sont beaucoup plus difficiles à exploiter avec AdaGrad et peuvent même dégrader les performances.

Méthode d'optimisation	Test accuracy (%)
SGD	95.38
SGD + Catapult	97.65
AdaGrad	97.09
AdaGrad + Catapult	95.91
Adam	98.12

Table 1: Comparaison de différentes méthodes d'optimisations sur un réseaux de neurone à 2 couches entrainé sur MNIST. Les catapultes ne permettent pas de battre Adam mais améliorent grandement la généralisation de la descente de gradient stochastique.

Enfin, en complément des résultats de performance, le protocole expérimental du notebook inclut une analyse dans le régime NTK. Pour rendre le calcul praticable, nous utilisons un modèle plus petit et nous suivons l'évolution de

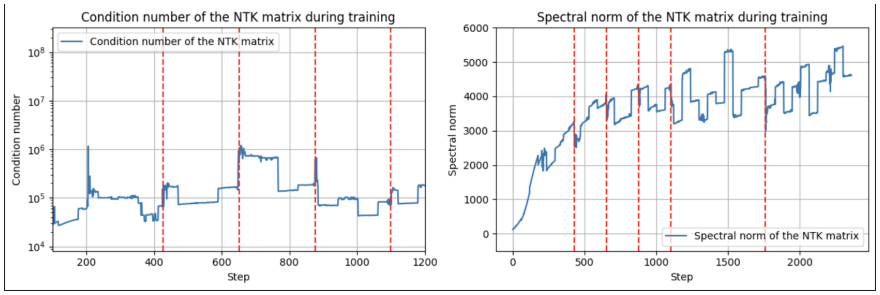

Figure 2: Évolution du condition number de la matrice NTK (gauche) et de la norme spectrale de la NTK (droite). Les lignes rouges indiquent les phases de catapultes.

métriques spectrales de la matrice NTK afin d'observer l'effet des phases de catapultes sur la dynamique d'apprentissage.

Sur la Figure 2, le *condition number* est très bruité (échelle log), ce qui rend son interprétation difficile dans notre setting. En revanche, la norme spectrale $\|K\|_2 = \lambda_{\max}(K)$ montre une corrélation nette avec les catapultes : chaque pic de learning rate coïncide avec une chute marquée de la norme spectrale, suivie d'une remontée rapide dans les itérations suivantes. Les détails expérimentaux et les courbes associées sont présentés plus complètement dans le notebook.

4 Études des catapultes sur Adagrad

Étant donné que l'article n'explore pas ce phénomène de catapultes sur Adagrad, nous voulions explorer dans quelle mesure nous pourrions étendre ces résultats sur des optimizers plus proches de l'état de l'art. Lors de nos premiers essais, nous nous rendîmes vites compte de la difficulté d'obtenir beaucoup de catapultes sans créer de grosses instabilités dans l'entrainement avec Adagrad. Nous avons donc exploré ce phénomène d'un point théorique et empirique.

4.1 Étude théorique

On se place dans le régime NTK, la descente de gradient stochastique a pour update :

$$f^{t+1} - y = (I - 2\eta \frac{K}{b})(f^t - y)$$

Utiliser Adagrad aulieu de SGD revient à changer le learning rate pour chaque paramètre du modèle, sans Adagrad, le learning rate est le même pour tous les paramètres :

$$(\eta)_{i \leq n}$$

Avec Adagrad, on a pour learning rate :

$$(\frac{\eta}{[\sqrt{g \circ g}]_i})_{i \leq n}$$

Ou le vecteur $\sqrt{\Sigma_p g_p \circ g_p}$ est l'historique des carré des gradients pour chaque poid du modèle. Ainsi, toute la difficulté est résumée ici : avec Adagrad on peut maintenant avoir des catapultes pour différents poids du modèle. Ceci complique donc notre analyse et explique les difficultés vues lors de l'entrainement. De plus, nous n'avons aucune hypothèse ou borne sur la norme des gradients ce qui demande à ce que le schedule du learning rate soit adapté. On voit d'ailleurs dans la Figure 4 que l'historique des gradients est d'abord très petit puis augmente de manière linéaire à partir du 500 ème step. Afin de permettre une analyse théorique, nous nous restreignons à une analyse simple, dans laquelle le learning rate ne dépend pas du paramètre :

Comme on sait qu'il y a une catapule si $\eta > \eta_{crit}$, une manière naïve d'analyser les catapultes sur Adagrad est de regarder :

$$(\frac{\eta}{s_t})_{i \leq n}$$

avec

$$s_t = min(\Sigma_p \sqrt{g_p \circ g_p})$$

. On sait que si ce learning rate reste en dessous du learning rate critique, il n'y aura pas de catapultes.

Étant donné que $\eta_{crit,SGD} = \frac{2}{\lambda_{max}}$, on a :

$$\eta^t_{crit,ADAGRAD} \geq \frac{2s_t}{\lambda^t_{max}}$$

Ainsi, en supposant la suite des $(\lambda^t_{max})_{t \in \mathbf{N}^*}$ bornée et en supposant $s_t \to +\infty quand t \to +\infty$, on peut démontrer plusieurs théorèmes :

1. Si le learning rate est borné alors il y aura un nombre fini de catapultes
2. Le learning rate critique d'Adagrad tend vers l'infini lorsque t tend vers l'infini

Ces théorèmes permettent surtout d'avoir une intuition de 'pourquoi' les catapultes sont plus difficiles à obtenir avec Adagrad.

4.2 En pratique

En pratique, on remarque dans la Figure 4 que nos hypothèses sur l'historique des gradient semble vérifiée, on comprend ainsi mieux la Figure 3. Pour aller plus loin, on peut utiliser l'information de l'historique des gradients pour obtenir un schedule adapté à Adagrad.

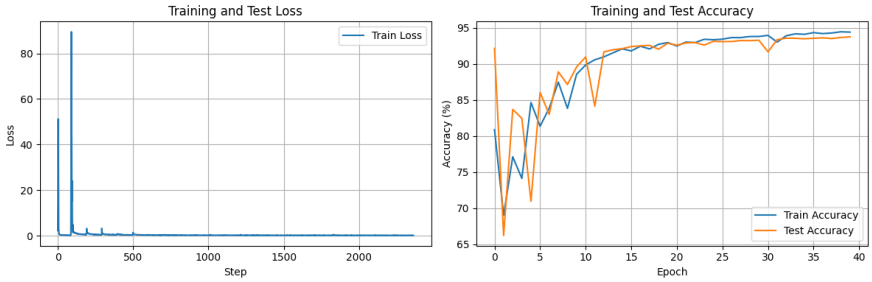

Figure 3: Évolution de la train loss et de l'accuracy pendant un entrainement utilisant Adagrad avec un scheduler cyclique : on remarque qu'au début, les catapultes sont très fortes et créent donc de l'instabilité (ce qui correspond au début de l'entrainement ou le minimum de l'historique des gradients a une faible valeur), puis que les catapultes disparaissent complètement, car l'historique des gradients tend vers l'infini.

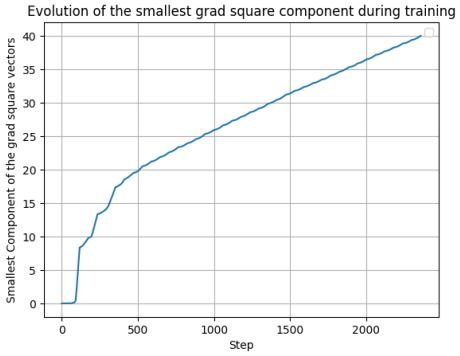

Figure 4: Évolution de la plus petite composante de l'historique des gradients pendant l'entrainement.

4.3 Conclusion

Notre étude caractérise la difficulté d'obtenir des catapultes avec d'autres optimizer, Adagrad étant de loin le plus facile à étudier d'un point de vue théorique. La Table 3 suggère que les catapultes ne sont peut-être pas souhaitable sur des optimizers qui performent déjà correctement, car les catapultes sur Adagrad offrent très peu de bénéfices en pratique. Une piste que nous n'avons pas pu explorer serait de regarde l'alignement AGOP des différentes méthodes pour avoir une meilleure compréhension de ces résultats.

5 Références

References

[1] L. Zhu, C. Liu, A. Radhakrishnan, and M. Belkin, "Catapults in SGD: spikes in the training loss and their impact on generalization through feature learning," *arXiv preprint* arXiv:2306.04815, 2023. `https://arxiv.org/abs/2306.04815`