Reaching for Resilience: Understanding How Optimizers Affect the Stability Gap in Continual Learning

Author: Chris Obis
Responsible Professor: Dr. Tom J. Viering
Supervisor: Dr. Gido M. van de Ven

Introduction

- **Continual learning** defines the accumulation of knowledge performed by an agent, be it natural or artificial, exposed to data-generating distriburtions, with the goal of solving future pattern identification problems.
- **Artificial (deep neural network - DNN) models** have been shown to struggle to integrate incoming information without disrupting existing memory (**"stability-plasticity trade-off"**).
- Recent analyses of model performance during transitions between different tasks have revealed a recurring sharp performance drop, followed by a gradual recovery; This has been termed the **Stability Gap** [1].
- Understanding the shape of the forgetting phenomenon can potentially contribute to reducing resource consumption and computational time needed to train models exposed to complex non-stationary environments.

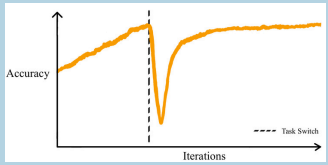

Figure 1: The ***Stability Gap*** *phenomenon exhibited by models at task transitions, as theorized by [1].*

Optimizers benchmarked

Stochastic Gradient Descent - **SGD** with **momentum**

- pioneered a **velocity** term that guides parameter updates, much like a ball rolling downhill, accumulating inertia to **overcome** small bumps (**local minima**) and **smooth transitions**

Figure 2: SGD

Nesterov Accelerated Gradient - **NAG**

- close variant of **SGD** with **momentum,** calculating the next gradient trajectory at a **look-ahead** position, after applying velocity

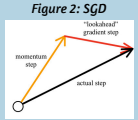

Figure 3: NAG

Adaptive Gradient Algorithm - **AdaGrad**

- reduces per-parameter **learning rate** in a directly proportional way to its summed **squared gradient history**

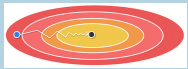

Figure 4: AdaGrad [2]

Root Mean Square Propagation - **RMSprop**

- adjusts per-parameter **learning rate** using a **decay factor** that makes older gradients less influential on current updates

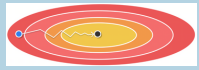

Figure 5: RMSprop [2]

Adaptive Moment Estimation - **Adam**

- has a **hybrid** momentum-based and adaptive design, including **bias correction**, and is often the preferred option in deep learning

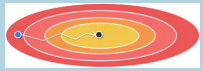

Figure 6: Adam [2]

Research Question

- **RQ**: "What is the impact of **momentum** and different **optimizer** choices on the **stability gap** of deep neural networks in continual learning problems?
- **SQ1**: How does increasing momentum in SGD and NAG impact the shape of the gap?
- **SQ2**: How do adaptive optimizers (AdaGrad, RMSprop, Adam) compare?

Results and discussion

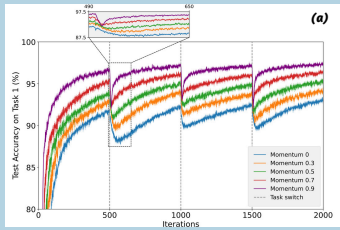

Figure 9: Performances of (a) SGD and (b) NAG on Task 1 under different momentum values.

Metric	Trend $\mu \uparrow$	*SGD* ($0.3\mu \rightarrow 0.9\mu$)	*NAG* ($0.3\mu \rightarrow 0.9\mu$)
TBP (it.)	↓	208.8 → 166.2	206.5 → 180.3
GD (%)	Peaks at 0.9μ	2.33 → 5.02	2.49 → 3.56
DS	↓	−0.092 → −0.384	−0.068 → −0.309
RS	↑	0.029 → 0.231	0.044 → 0.168

Figure 10: Trends of the stability gap metrics for SGD and NAG, based on momentum.

Metric	Trend $\mu \uparrow$	*SGD* ($0.3\mu \rightarrow 0.9\mu$)	*NAG* ($0.3\mu \rightarrow 0.9\mu$)
ACC (%)	↑	93.49 → 96.58	93.44 → 96.82
FORG (%)	↑	−1.47 → −0.62	−1.4 → −0.67
min-ACC (%)	↑	89.25 → 90.12*	89.1 → 91.92

Figure 11: Trends of the stability-plasticity metrics for SGD and NAG, based on momentum.

SQ1: Momentum

Insights:

- **increasing** momentum leads to a **steeper** performance drop (**DS**), a **sharper** and **earlier** recovery (**RS**), a **deeper** stability gap (**GD**), which together ultimately **shorten** its duration (**TBP**)
- these volatile updates of parameters are a result of a **stronger inertia force** that pulls the configuration **away** from the previous **optimum** point, but also allows a **faster** finding of the **joint trajectory**
- there exists a **trade-off** between an accelerated **convergence** (at a higher overall performance) and a **stable** performance evolution after a task transition point
- **NAG's** more **preemptive** updates help it **prevent overshooting** and correct its course earlier, scoring a lower **GD**

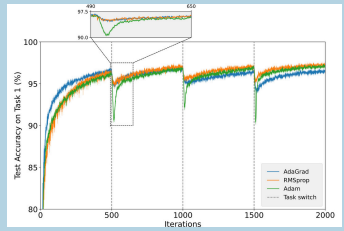

Figure 12: Performance of the adaptive optimizers on Task 1.

Metric	*AdaGrad*	*RMSprop*	*Adam*
TBP (it.) ↓	208.4 ± 7.9	**134.1** ± 8.0	150.1 ± 5.4
GD (%) ↓	1.68 ± 0.06	**1.4** ± 0.08	4.31 ± 0.13
DS	-0.084 ± 0.005	-0.141 ± 0.035	-0.296 ± 0.010
RS	0.031 ± 0.002	0.070 ± 0.017	0.191 ± 0.011

Figure 13: Stability gap metrics for AdaGrad, RMSprop and Adam.

Metric	*AdaGrad*	*RMSprop*	*Adam*
ACC (%) ↑	95.49 ± 0.03	96.69 ± 0.10	**96.82** ± 0.07
FORG (%) ↑	-0.36 ± 0.10	-0.74 ± 0.13	**-0.83** ± 0.14
min-ACC (%) ↑	92.61 ± 0.06	**93.61** ± 0.05	90.76 ± 0.17

Figure 14: Stability-plasticity metrics for AdaGrad, RMSprop and Adam.

SQ2: Adaptive methods

Insights:

- **AdaGrad** and **RMSprop** exhibit the most **controlled** stability gap **slopes** and **depths**, although **struggling** more than Adam with **joint accuracy** performance
- **RMSprop** proves most capable to **minimize** the **depth** and **recovery time** of the stability gap
- **AdaGrad's prolonged recovery** can be explained by its monotonically decreasing learning rates
- **Adam** performs similarly to the previous momentum-based optimizers, **increasing** the **amplitude** of the gap and overall **volatility**

We acknowledge the experimental **time** and **space complexity** (model size, number of tasks) **limitations**, as well as the simplified rotated-digit identification tasks. We hypothesize that, while **RMSprop** (adaptiuity) maintains a balance of plasticity and stability in this case, **SGD and NAG** (high-momentum values) generalize more effectively. This can result in an overall more stable learning trajectory, as the number and complexity of the tasks, as well as the noise presence, increase.

Setup and Metrics

Evaluating optimizer-induced stability gap shapes in a **domain-incremental learning** [3] process of a DNN with

- **Dataset**: Rotated-MNIST
- **Training regime**: Incremental-Joint training
- **Mini-batch size**: grows incrementally 128-256-384-512
- **Model architecture**: 3 fully-connected hidden layers with 400 ReLU neurons each
- **Evaluation periodicity**: 1
- **Test size**: 2000
- **Optimizer yperparameters**: tuned in a **gridsearch** way
- **Iterations per task**: 500

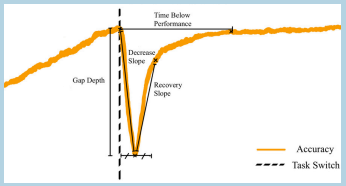

Figure 7: Stability gap metrics. Alongside these, we compute general indicators of average accuracy ***ACC****, average forgetting* ***FORG****, and average minimum accuracy* ***min-ACC****.*

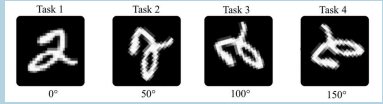

Figure 8: Task-specific training samples, with rotational degrees applied.

Conclusions

- Higher **momentum** increases the **steepness** and **magnitude** of the gap, while **narrowing** it.
- **AdaGrad** expectedly experiences a **mild** drop, but **limited plasticity**.
- **Adam** mirrors the volatility of **SGD** and **NAG**.
- **RMSprop** strikes the best **balance** between **controlling** the drop and scoring considerably high in **joint accuracy**.
- **Safety-critical systems** such as autonomous vehicles or adaptive healthcare tools, where even **minimal** performance **drops** can have **major consequences**, would highly benefit from a deeper understanding of the stability gap.
- **Future research** avenues include evaluating the **generalizability** of the observed trends across larger-scale learning scenarios, broader datasets and more complex DNN architectures.
- Ultimately, deeper insights into the **stability gap** phenomenon can enable robust and resource-efficient continual learning, by effectively preserving acquired knowledge.

References

[1] Lange, M. D., van de Ven, G. M., and Tuytelaars, T. (2023). Continual evaluation for lifelong learning: Identifying the stability gap. In The *Eleventh International Conference on Learning Representations*, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.
[2] Efimov, V. (2023). Understanding Deep Learning Optimizers: Momentum, AdaGrad, RMSprop and Adam. *Towards Data Science*.
[3] van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022). Three types of incremental learning. Nature Machine Intelligence, 4(12):1185–1197.
Poster template was provided by **PosterNerd**.