Report : Training Deep Learning Models with Norm-Constrained LMOs

Tristan Beruard
CentraleSupélec
code : https://github.com/TristanoB/SCION-method-review.git
tristanberuard@gmail.com

February 24, 2026

1 Understanding of the paper

lmo-optimization The goal is to find a method to solve a very general optimization problem $\min_{x \in X} f(x)$ avec $X = \mathbb{R}^d$ or in the case of a constrained probleme $X = \mathcal{D}$ with $\mathcal{D}$ is a ball. Classical gradient descents as Adam and SGD are often the most used to solve this problem. Another whole field of method is based on linear minimization oracle (lmo) principle, that takes not the simple opposite direction of the gradient, but the direction of the steepest descent according to any chosen norm. More precisely the lmo is defined as $\min\limits_{s \in \mathcal{D}} \langle g, s \rangle$, and one common method to use it is the Conditional Gradient (CG) Method as described in the paper :

$$x_{k+1} = x_k + \eta(\text{lmo}(\nabla f(x_k)) - x_k)$$

lmo-based optimization method will therefore optimize according to a chosen geometry, whereas SGD correspond to the l_2 norm, which is not always adapted, specially in the application case where we will want to force the feature learning regime (see below). It is worth noting that the chosen norm appears in the definition of the unit ball.

One natural question arises : what are the differences between this method and more classical one as L_2 regularization, as this also constraints the choice of the parameters. In fact classical steepest descent with for any norm can be seen as $x_{k+1} = x_k + \gamma_k \|g_k\|_* \text{lmo}(g_k)$, so with the update being scaled with the dual norm of the gradient, contrary to CG, where the gradient only control the direction, achieving better control without weight explosion for feature learning condition.

Conditional Gradient CG works well for l_1 ball or other atomic set constrained problem, because the linear minimization oracle promotes sparse or structured solutions. But in the case of unconstrained problem, or for l_∞ balls, the method may suffer from zig-zagging behavior, slow convergence, and poor conditioning.

SGC introduced stochasticity with random batch selection. But the update term when using stochastic lmo term is biased, as $lmo(\cdot)$ is not linear. In classical SGD, the inversion between the expectation and the sum of over the mini-batch can be made, making the stochastic estimor unbiased. With the lmo it is not possible.

Then to solve this, the authors propose to use an accumulation of the gradients, using for the gradients $d_k = \alpha_k \nabla f(x^k, \xi_k) + (1 - \alpha_k) x_k$ which reduces the variance of the gradient estimator, and so the effect of the bias induced by the lmo.

The authors gives two different general optimization algorithm, depending on the problem type :

- We mentionned the limitations of the CG on unconstrained problems. To solve that, the authors "simply" changes the convex combination of the above equation with the accumulation of the gradients : $x^{k+1} = x^k + \gamma_k lmo(d_k)$, γ_k being the learning rate. As $\text{lmo}(\cdot)$ operator give a vector over the unit ball, the magnitude of the update does not depend on the magnitude of d_k. For computing the gradient, batch of input are sampled at every iteration k They name the algorithm uSGC

- For the constrained case, they reuse the update of SGC using the mentionned convex combination, but also replace the gradient by it's accumulated sum : $x^{k+1} = (1 - \gamma_k) x^k + \gamma_k \text{lmo}(d_k)$

In the constrained case the iterate must remain inside the feasible set, so the update is written as a convex combination, whereas in the unconstrained case no such restriction exists and the method can directly take a step in the LMO direction without mixing with the current iterate.

These optimizers permit to control the norm of the parameters explicitely :

- for SGC, as we take a convex combination of x_k, d_k both in $\mathcal{D}$ (reasonning via recusrivity over x_k and by the choice of unit ball for d_k), then $x_{k+1} \in \mathcal{D}$, so we get that the parameters are bounded by the radius ρ of the unit ball
- for uSGC, as $x_k = x_0 + \sum_{i=1}^{k} \gamma_k lmo(d_i)$, we get $\|x_k\| \leq \rho \sum_{i=1}^{k} \gamma_k$

In pratice this will control the norm of the weight matrices of a NN during training, with the utility of controlling their spectral norm, forcing the feature learning as we will see below.

Feature Learning and norm choice The advantages of uSGC and SGC have been introduced for any norm choice. The authors then focus on a specific norm. By applying to the operator norm for Deep Neural Networks, they name the application case "Stochastic Conditional gradIent with Operator Norms" (SCION). The objective is to achieve feature learning, that be defined as when activation $h^{(l)}$ and their update $\nabla h^{(l)}$ for a particular layer are in $O(1)$ regarding the parameter layer l. More particulary we want one of the three different bound condition :

$$\frac{1}{d_{out}} \|h_l(z)\|_1, \|h_l(z)\|_{RMS}, \|h_L(z)\|_\infty \leq 1$$

Depending on the layer, the norm selected for the condition will vary. Typically for intermediate layer the RMS norm will be used, for output layer it will be more often the ∞ norm. These norm can be controlled in the case of linear layers, as $h_{l+1} = \sigma(W_l h_l)$. For that using the operator norm $\|A\|_{\alpha \to \beta} = \sup_{\|z\|_\alpha = 1} \|Ah\|_\beta$, and using $\|Ah\|_\beta \leq \|A\|_{\alpha \to \beta} \|h\|_\alpha$, we get :

- input layer : $\|h^{(1)}\|_{RMS_m} \leq \|W_1\|_{RMS_p \to RMS_m} \|h^0\|_{RMS_p}$, denoting with p, m the dim for layer $0, 1$
- intermediate layer : $\|h_l\|_{RMS_m} \leq \|W_l\|_{RMS_m \to RMS_m} \|h_{l-1}\|_{RMS_p}$
- for the final layer : $\|h_L\|_{RMS_1} \leq \|W_L\|_{RMS_m \to RMS_1} \|h_{L-1}\|_{RMS_m}$

Therefore by reasonning recursively, if $\|h_{l-1}\|$ is bounded, and that $\|W_l\|$ also is, then $\|h_l\|$ will be, and we will have feature learning. We can apply the same reasonning for Δh. For the RMS norm, this links to the spectral norm of the weight matrices by writting $\|W_l\|_{RMS_{d_{in}} \to RMS_{d_{out}}} = \frac{\sqrt{d_{in}}}{\sqrt{d_{out}}} \|W\|_{2 \to 2} = \frac{\sqrt{d_{in}}}{\sqrt{d_{out}}} \|W\|_{S_\infty}$, and so we get the general condition for feature learning :

$$\|W_l\|_{2 \to 2} = O\left(\sqrt{\frac{d_{out}}{d_{in}}}\right)$$

We now have a condition on the layer weights to get into feature learning. If we want to apply it, and use the constraint over the NN geometry we just found, we need to use it for defining the norm of the unit ball for the lmo step. So from what we said and taking account the biases, the weight norm $\sqrt{\frac{d_{in}}{d_{out}}} max(\|W\|_{2 \to 2}, \|b\|)$ should be at maximum 1, and so we can define the norm as :

$$\|x\| = max_l(\sqrt{\frac{d_{in}}{d_{out}}} max(\|W\|_{2 \to 2}, \|b\|)$$

with this norm being forced to be ≤ 1 Therfore, for each the layer the norm operator is used, and for taking into account all the layers, we apply a ∞-norm, by taking the max. The unit ball for the lmo update is therefore an intersection of different unit ball for each layer $\mathcal{D} = \{x : \forall l, \|W_l\|_l \leq \rho_l\}$, the ρ_l radius parameter corresponding in our case to typically $\sqrt{\frac{d_{out}}{d_{in}}}$. The global lmo problem separate within each layer : $lmo(g) = (lmo_1(g_1), \ldots, lmo_L(g_L))$ So each layer will receive a different lmo update, depending on the norm used for bounding.

As the condition for feature learning was computed recursively starting from the input, we need to pay attention to the norm used for the input h_0, to be sure that it will be less than 1. For example for images, as the input is often rescaled and between $[-1, 1]$, than we can use the classical $\|\cdot\|_{RMS}$, ensuring to be below 1. Other choices will apply for text or other applications

SCION lmo update term For the image domain, the lmo term is :

- for the Spectral term (input layer and all intermediate layers) : $lmo(A) = -\sqrt{\frac{d_{out}}{d_{in}}}UV^T$
- For the sign term (output layer) : $lmo(A) = -sign(A)$

We can compute the lmo spectral term by using the exact SVD, as the lmo for spectral term requires U, V or using a newton-schulz approximation, by $\text{sign}(G) \approx G(G^\top G)^{-1/2}$.

Hyperparameter Transfer For SGD and Adam, when increasing the width of a NN, the norm of the weights and gradients changes. So also the magnitude of the updates, so the learning rate need to adapt, and therefore change with the width.

Hyperparameter transfer is defined when the same hyperparameter (for example the learning rate) can be used after having changed another hyperparameter (here the width). For SCION method, the update does not depend on the magnitude of the gradient. For the spectral term (the most used update term), it is linked to UV^T, which has always the same scale indepentently of the layer, as it is scaled by the ratio $\sqrt{\frac{d_{out}}{d_{in}}}$. In fact they show in appendix that the magnitude of the preactivation changes is proportionnal to $\gamma^2 d_l$, and as a result the optimal learning rate becomes independant of the witdh d_l. They confirm this intuition experimentally.

Convergence analysis The paper studies the stochastic optimization problem $\min_{x \in \mathcal{X}} f(x) = \mathbb{E}_\xi[f(x, \xi)]$ where either $\mathcal{X} = \mathbb{R}^d$ (unconstrained case, uSCG) or $\mathcal{X} = \mathcal{D} := \{x : \|x\| \leq \rho\}$. Assuming smoothness with respect to a chosen norm and an unbiased stochastic gradient with bounded variance, the authors derive two convergence regimes. With constant averaging $\alpha \in (0, 1)$ and step size $\gamma = \frac{1}{\sqrt{n}}$, both uSCG and SCG achieve a stochastic $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ rate, but only up to a noise-dominated region, corresponding to a residual term proportional to the gradient noise level σ.

2 Context and novelty

Classical NN Optimization Neural network have historically often been framed as stochastic optimization, as $min_x \mathbb{E}_\omega[f(x, \omega)]$. The standard pratice is to use SDG, which implies by the update term an implicit euclidean geometry, so optimization was historically geometric agnostic. Adam did introduced an adaptative scaling, equivalent to a Mahlanobis norm. But even if it does take account of the past gradient, it is still geometric agnostic and fails to preserve feature learning regime. (Chizat et al., 2019 (lazy regime / NTK regime) νP (Maximal Update Parametrization) was introduced by Yang et al., enabling hyperparameter transfer across widths, with an architectural scaling rule. But the optimization geometry itself was not redisigned.

spectral feature learning theory Yang et al., 2023 developped spectral conditions for feature learning, with a condition (cite equation) on the weight matrices to guarantee feature learning. But until SCION no optimizer was explictey designed to enforce them.

Conditional Gradient Litterature To force the condition, SCION is based partly on the conditional gradient literature, as Frank-Wolf CG, LMO and recent stochastic version of this optimaztion methods such as uSCG and SCG from (Pethick et al., 2025). The key idea is that the update direction depend on a fix norm fixed a priori.

Unification SCION uses these optimization method; and combines Spectral feature learning theory, operator-norm geometry, stochastic Conditional Gradient with momentum with an Architecture-aware norm choice layer-by-layer Instead of adapting step sizes (Adam) or scaling init (uP), SCION uses the optimization geometry itself and align it with spectral feature learning conditions. the authors link infinite-width theory and practical optimization, and provided convergence guarntees.

Other Optimizers Other optimizers are SGD that does a steepest descent but on the l_2 unit ball, signSGD does it on l_1 unit ball. Shampoo uses matrix precondionneurs, and without the preconditionnteur accumulation is equivaent to a steep descent in the operator norm. Muon does also correspond to an operator norm based steepest decent, but is not applied to initial and output layers. SCION show that these optimizers correspond to differnt norm of the lmo update, but link them to feature learning and consequently choose specifically an RMS scaling norm. More recent work as MARS-Shampoo combine spectral geometry and STORM gradient (which a upgraded lower variance gradient estimator), and can be seen as SCG $x_{k+1} = x_k + \eta lmo(d_k)$ but with d_k built with STORM and not via simple momemtum, making it a specific case of SCION. Moonlight goes from muon, but adds weight decay, to control the norm, but without being linked to feature learning.

3 Optimization and Vision Connection

Works for Deep Neural Newtorks used in vision. The optimization choices that make SCION forces the feature learning regime of the NN. For some well known architectures as large CNN as ResNet on datasets large datasets like ImageNet, these choices permit to get more stable representations and non-lazy feature learning. This makes the optimizer architecture-aware and particularly suited to vision models where representation learning relies on stable signal propagation across deep layers.

- For large vision dataset, feature learning permits to scale the performance (they show some results on ImageNet and CIFAR10)
- For selecting the input and output layers norms, the authors propose specific choices for images, making SCION ready-to-use for Computer Vision.
- The choice of a RMS norm for the hidden layers of a NN in vision is pertinent, as the activations are distributed across all the dimensions, and RMS corresponds to a mean energy. Controlling it empêche the domination of single neurons in the activations, lowering overfitting.

4 Critical Evaluation

Strenghts They show that lot of optimizers can be seen as instances of a same lmo schema. They propose convergence results of uSGC and SCG with momemtum, with a differentation for constant momemtum and decreasing. The geomerty aware recipe that they propose is linked to working experimental results. The SCION method is memory efficient, as it requires only 1 set of weight and 1 set of gradients.

Limitations One key point of the method is the computation of the U, V matrices, via Netwon-Schulz or SVD, that are some approximations ad could lead to some results reproducibility issues. There are some key hypothesis in the paper, for example for the convergence analysis (Lipshichtz for exmaple), that could be questionned in some extreme cases; Some factors as architerture tricks and learning rate scheduling are not modelized, which is a limitation in pratice, and could hide the gains of SCION. There might be some regime for DL where the tasks requires a increasing norm, that could be refrained via SCION. The paper gives a method for classical architetures in Vision for example, but do not extend the results to more complex one, as ConvNext, diffusion U-Nets and exotic losses. The global generalization of outperforming Adam is yet to be studied.

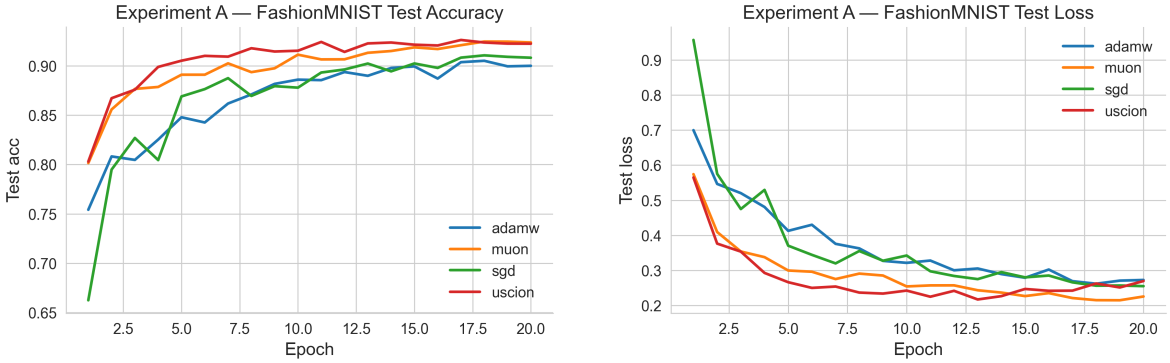
5 Implementation

For the image domain, we implemented the Spectral version of SCION, with a focus for computer vision, where the Linear Minimization Oracle (LMO) is defined with respect to the operator (spectral) norm. Convolutional kernels were reshaped into 2D matrices of size $(d_{out}, d_{in} \cdot k_H \cdot k_W)$, allowing us to apply the polar LMO consistently across both convolutional and fully connected layers. The LMO direction is computed as the negative polar factor of the momentum-averaged gradient, approximated via Newton–Schulz iterations. To preserve the layer-wise scaling properties described in the paper, we introduced a scaling coefficient $\rho_\ell = \sqrt{\frac{d_{out}}{d_{in}^{\text{eff}}}}$ with $d_{in}^{\text{eff}} = d_{in} \cdot k_H \cdot k_W$ for convolutional layers. Bias parameters and other vector-valued tensors were handled using an RMS-normalized LMO. For this work, we propose to reimplement the SCION optimizer and to compare it with 3 other optimizers : AdamW, SGD and Muon. We propose 3 experiments :

- Experiment A is about comparing performance when tuning a small CNN over the fashionMNIST dataset
- Experiment B is hyperparameter transfer as it was done in the paper, but this time on a smallCNN on fashionMNIST
- Experiment C is about following the norm matrices for SCION vs uSCION on every layer's weight, using the smallCNN and fashionMNIST

The results are presented in Appendix.

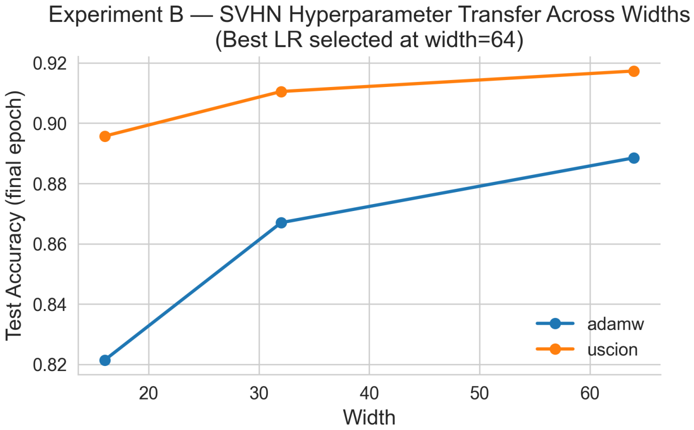
A Appendix



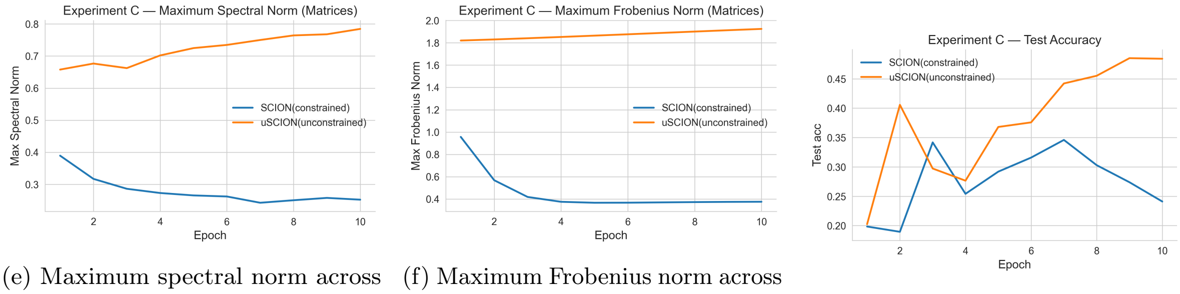
(a) Training and test loss on Fashion-MNIST.

(b) Test accuracy comparison across optimizers.

(c) Experiment A: Performance comparison of uSCION against SGD, Adam and Muon on Fashion-MNIST. uSCION outperforms the other methods. SCION is included in the next page, as I did not manage to have good performance, due to an issue of learning rate tuning (I did not find a working learning rate)



(d) Experiment B: Hyperparameter transfer across widths on fashionMNIST. We observe that uSCION maintains stable performance for one given learning rate as width increases, supporting the architecture-aware design principle. In contrast, AdamW require retuning, indicating weaker hyperparameter transfer. Tested for 10 epochs.



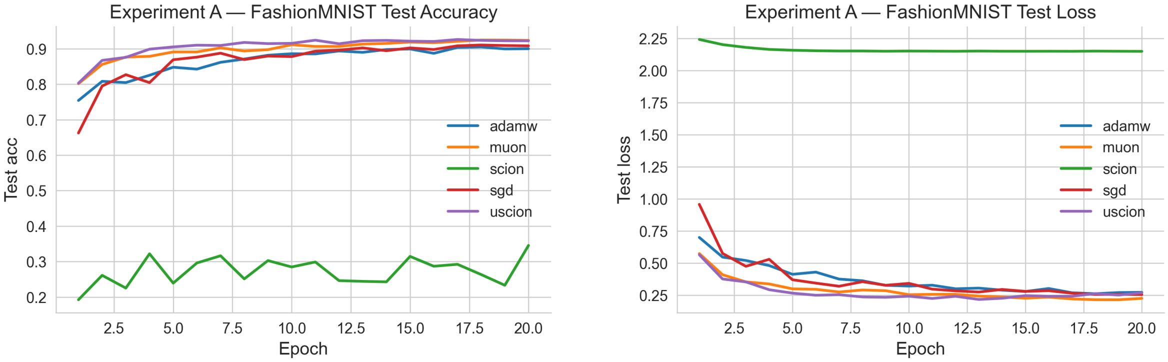
(e) Maximum spectral norm across all weight matrices during training.

(f) Maximum Frobenius norm across all weight matrices during training.

(g) Test accuracy evolution.

(h) Experiment C — Norm control under SCION vs uSCION. We compare the constrained variant (SCION) and the unconstrained variant (uSCION) on Fashion-MNIST. SCION explicitly enforces a norm-ball constraint through the conditional gradient update, resulting in stable operator norms throughout training. In contrast, uSCION allows freer norm growth. The spectral norm plot highlights the geometry induced by the operator-norm LMO, while the Frobenius norm provides a complementary scale-sensitive view. These results empirically illustrate the theoretical norm-control guarantees required for spectral feature learning.

Figure 1: Empirical evaluation of SCION and uSCION. Experiment A compares optimization performance on Fashion-MNIST. Experiment B studies hyperparameter transfer across widths on fashionMNIST. Experiment C analyzes layer-wise operator norm dynamics, highlighting the geometry-enforcing effect of the constrained formulation.



(a) Training and test loss on Fashion-MNIST.

(b) Test accuracy comparison across optimizers.

Figure 2: Experiment A - bis : Performance comparison of uSCION and SCION against SGD, Adam and Muon on Fashion-MNIST. SCION does not converge, after having test a lot of learning rate parameters. This might be due to an issue in my implementation