# Matrix Optimization and Muon: A Natural Perspective on Neural Network Training, part 2

Fall into ML

Aleksandr Beznosikov

MIPT

24 October 2025

# Optimization problem

- Consider optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x).$$

# Optimization problem

- Consider optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x).$$

- If we speak about ML, then the good example is ERM:

$$\min_{x \in \mathbb{R}^d} \left[ f(x) := \frac{1}{n} \sum_{i=1}^{n} [\ell(g(x, \xi_{x,i}), \xi_{y,i})] \right],$$

where $\{\xi_i\}_{i=1}^n$ – train set from nature $\mathcal{D}$, $g$ – model, $\ell$ – loss.

# Iterative procedures

- Consider optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x).$$

- For real-world tasks, we cannot solve this problem directly.
- I want to solve it iteratively, using some current properties.

# Iterative procedures

- Consider optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x).$$

- For real-world tasks, we cannot solve this problem directly.
- I want to solve it iteratively, using some current properties.
- We are at some point $x^k$. Look at the quadratic approximation of the function at this point:

$$f(x) \approx f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle x - x^k, \nabla^2 f(x^k)(x - x^k) \rangle.$$

# Iterative procedures

- Consider optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x).$$

- For real-world tasks, we cannot solve this problem directly.
- I want to solve it iteratively, using some current properties.
- We are at some point $x^k$. Look at the quadratic approximation of the function at this point:

$$f(x) \approx f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2} \langle x - x^k, \nabla^2 f(x^k)(x - x^k) \rangle.$$

- Instead of

$$x^* = \arg\min_{x \in \mathbb{R}^d} f(x)$$

we use

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^d} f_k(x)$$

# Method

- **Q:** if we do this way

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^d} f_k(x)$$

with

$$f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2}\langle x - x^k, \nabla^2 f(x^k)(x - x^k)\rangle.$$

which method we have?

# Method

- **Q:** if we do this way

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^d} f_k(x)$$

with

$$f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2}\langle x - x^k, \nabla^2 f(x^k)(x - x^k) \rangle.$$

which method we have?

- Newton's method:

$$\boxed{x^{k+1} = x^k - \left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k)}$$

# Pros and cons

# Pros and cons

## Pros

- Very fast quadratic convergence

### Convergence of Newton's method

$$\|x^{k+1} - x^*\|_2 \leq \|x^k - x^*\|_2^2.$$

$$\|x^0 - x^*\|_2 = \frac{1}{2} \to \left(\frac{1}{2}\right)^2 \to \left(\left(\frac{1}{2}\right)^2\right)^2 \to \ldots$$

# Pros and cons

Pros

- Very fast quadratic convergence

**Convergence of Newton's method**

$$\|x^{k+1} - x^*\|_2 \leq \|x^k - x^*\|_2^2.$$

$$\|x^0 - x^*\|_2 = \frac{1}{2} \to \left(\frac{1}{2}\right)^2 \to \left(\left(\frac{1}{2}\right)^2\right)^2 \to \dots$$

Cons

- Only local convergence

# Pros and cons

## Pros

- Very fast quadratic convergence

### Convergence of Newton's method

$$\|x^{k+1} - x^*\|_2 \leq \|x^k - x^*\|_2^2.$$

$$\|x^0 - x^*\|_2 = \frac{1}{2} \to \left(\frac{1}{2}\right)^2 \to \left(\left(\frac{1}{2}\right)^2\right)^2 \to \ldots$$

## Cons

- Only local convergence
- Expensive iteration: we don't want to compute Hessian and inverse it

# Idea

To use the cheap matrix instead of the real Hessian in

$$f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2}\langle x - x^k, \nabla^2 f(x^k)(x - x^k) \rangle.$$

# Idea 1

- Let us use the identity matrix:

$$\nabla^2 f(x^k) \to I.$$

# Idea 1

- Let us use the identity matrix:

$$\nabla^2 f(x^k) \to I.$$

- But how to make it more correctly?

# Idea 1

- Let us use the identity matrix:

$$\nabla^2 f(x^k) \to I.$$

- But how to make it more correctly? Let for all $x, y \in \mathbb{R}^d$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2,$$

then

$$LI \succeq \nabla^2 f(x).$$

# Idea 1

- Let us use the identity matrix:

$$\nabla^2 f(x^k) \to I.$$

- But how to make it more correctly? Let for all $x, y \in \mathbb{R}^d$

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2,$$

then

$$LI \succeq \nabla^2 f(x).$$

- So the good way

$$\nabla^2 f(x^k) \to LI.$$

# Idea 1

- We have not only approximation but upper bound of $f$:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2}\|x - x^k\|_2^2.$$

# Idea 1

- We have not only approximation but upper bound of $f$:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2}\|x - x^k\|_2^2.$$

- **Q:** which method we have?

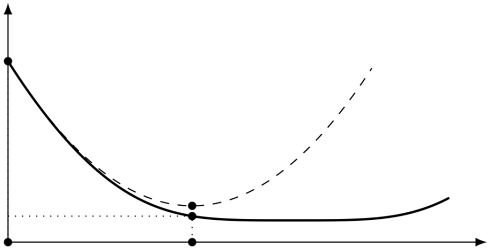
# Idea 1

- We have not only approximation but upper bound of $f$:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2}\|x - x^k\|_2^2.$$

- **Q:** which method we have? Gradient descent with the step $\gamma = \frac{1}{L}$:

$$x^{k+1} = x^k - \gamma \nabla f(x^k) = x^k - \frac{1}{L}\nabla f(x^k)$$



# Idea 2

- Let us play and use other upper bound of $f$:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L_1}{2}\|x - x^k\|_\infty^2.$$

# Idea 2

- Let us play and use other upper bound of $f$:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L_1}{2}\|x - x^k\|_\infty^2.$$

- We have the sign method $\gamma = \frac{1}{L_1}$:

$$x^{k+1} = x^k - \frac{1}{L_1} \cdot \|\nabla f(x^k)\|_1 \cdot \text{sign}(\nabla f(x^k))$$

or

$$x^{k+1} = x^k - \gamma \cdot \text{sign}(\nabla f(x^k))$$

# Idea 2

$$x^{k+1} = x^k - \gamma \cdot \text{sign}(\nabla f(x^k))$$

- **Q:** Why the sign method can be nice?

# Idea 2

$$x^{k+1} = x^k - \gamma \cdot \text{sign}(\nabla f(x^k))$$

- **Q:** Why the sign method can be nice?
- We usually use stochastic methods instead of an full gradient. Clipping techniques are standard way to control noise:

$$\text{clip}(g, \lambda) = \min\left\{ \frac{\|g\|_2}{\lambda}; 1 \right\} g$$

  In fact we project the stochastic gradient $g$ to the ball. But we need to tune $\lambda$ and schedule for it.
- Sign it as an alternative to this approach!

# Idea 2

Llama experiment:

| Method | Perplexity ↓ | | |
|---|---|---|---|
| Model size | 130M | 350M | 1.3B |
| M-SignSGD | **18.37**$_{\pm.01}$ | **13.73** | **11.56** |
| M-NSGD | 19.28$_{\pm.03}$ | 14.60 | 12.62 |
| M-ClippedSGD | 18.95$_{\pm.03}$ | 14.30 | 12.30 |
| AdamW | 18.67$_{\pm.00}$ | 13.78 | 11.57 |
| Training tokens | 10B | 30B | 30B |
| Number of iterations | 100k | 300k | 300k |

# Idea 3

- Let us do something more complicated than the identity matrix!

# Idea 3

- Let us do something more complicated than the identity matrix!
- Let us ask to contain some properties of the Hessian:

$$\nabla f(x^k) = \nabla f(x^{k+1}) + \nabla^2 f(x^{k+1})(x^k - x^{k+1}) + o(\|x^{k+1} - x^k\|_2)$$

or

$$\nabla f(x^k) - \nabla f(x^{k+1}) \approx \nabla^2 f(x^{k+1})(x^k - x^{k+1})$$

# Idea 3

- Let us do something more complicated than the identity matrix!
- Let us ask to contain some properties of the Hessian:

$$\nabla f(x^k) = \nabla f(x^{k+1}) + \nabla^2 f(x^{k+1})(x^k - x^{k+1}) + o(\|x^{k+1} - x^k\|_2)$$

or

$$\nabla f(x^k) - \nabla f(x^{k+1}) \approx \nabla^2 f(x^{k+1})(x^k - x^{k+1})$$

- Use some $B_{k+1}$:

$$\nabla^2 f(x^{k+1}) \to B_{k+1}$$

such that

$$\nabla f(x^k) - \nabla f(x^{k+1}) = B_{k+1}(x^k - x^{k+1}) \quad \text{and} \quad B_{k+1} = B_{k+1}^T$$

# Idea 3

- These conditions are called quasi-Newton conditions:

$$\nabla f(x^k) - \nabla f(x^{k+1}) = B_{k+1}(x^k - x^{k+1}) \quad \text{and} \quad B_{k+1} = B_{k+1}^T$$

- Due to the fact that a whole family of methods satisfies these conditions, there is a huge potential for creativity in how to select $B_{k+1}$.

# Idea 3

- These conditions are called quasi-Newton conditions:

$$\nabla f(x^k) - \nabla f(x^{k+1}) = B_{k+1}(x^k - x^{k+1}) \quad \text{and} \quad B_{k+1} = B_{k+1}^T$$

- Due to the fact that a whole family of methods satisfies these conditions, there is a huge potential for creativity in how to select $B_{k+1}$.
- The most popular – BFGS:

$$B_{k+1} = B_k + \frac{y^k (y^k)^T}{(y^k)^T s^k} + \frac{B_k s^k (B_k s^k)^T}{(s^k)^T B_k s^k}$$

# Idea 3

Pros

- Cheap iteration
- Global superlinear convergence

# Idea 3

Pros

- Cheap iteration
- Global superlinear convergence

Cons

- Too good for big neural networks!

# Idea 4

- Need more simple approximation of the Hessian.
- Quadratic example:

$$\min_{x_1 \in \mathbb{R}, x_2 \in \mathbb{R}} f(x_1, x_2) = x_1^2 + 1000 x_2^2$$

# Idea 4

- Need more simple approximation of the Hessian.
- Quadratic example:

$$\min_{x_1 \in \mathbb{R}, x_2 \in \mathbb{R}} f(x_1, x_2) = x_1^2 + 1000 x_2^2$$

$$\nabla f(1,1) = \begin{pmatrix} 2 \\ 2000 \end{pmatrix}, \quad \nabla^2 f(1,1) = \begin{pmatrix} 2 & 0 \\ 0 & 2000 \end{pmatrix}$$

# Idea 4

- Need more simple approximation of the Hessian.
- Quadratic example:

$$\min_{x_1 \in \mathbb{R}, x_2 \in \mathbb{R}} f(x_1, x_2) = x_1^2 + 1000 x_2^2$$

$$\nabla f(1,1) = \begin{pmatrix} 2 \\ 2000 \end{pmatrix}, \quad \nabla^2 f(1,1) = \begin{pmatrix} 2 & 0 \\ 0 & 2000 \end{pmatrix}$$

- Simple idea:

$$\nabla^2 f(x) \to \text{diag}(|\nabla f(x)|) \text{ or } \text{diag}\left(\sqrt{(\nabla f(x))^2}\right)$$

# Idea 4

- Let us make it a little smoother and more robust.:

$$G_i^{k+1} = \beta_2 G_i^k + (1 - \beta_2)([\nabla f(x^k)]_i)^2 \quad \text{with} \quad \beta_2 \approx 0.999$$

$$\nabla^2 f(x^k) \to \text{diag}(\sqrt{G^k})$$

# Idea 4

- Let us make it a little smoother and more robust.:

$$G_i^{k+1} = \beta_2 G_i^k + (1 - \beta_2)([\nabla f(x^k)]_i)^2 \quad \text{with} \quad \beta_2 \approx 0.999$$

$$\nabla^2 f(x^k) \to \text{diag}(\sqrt{G^k})$$

**Q:** why we need this update?

# Idea 4

- Let us make it a little smoother and more robust.:

$$G_i^{k+1} = \beta_2 G_i^k + (1-\beta_2)([\nabla f(x^k)]_i)^2 \quad \text{with} \quad \beta_2 \approx 0.999$$

$$\nabla^2 f(x^k) \to \text{diag}(\sqrt{G^k})$$

**Q:** why we need this update? Recall that in practice, the gradient is stochastic - it is worth summing several stochastic gradients at once.

# Idea 4

- RMSProp:

$$G_i^{k+1} = \beta_2 G_i^k + (1 - \beta_2)([\nabla f(x^k)]_i)^2$$
$$x^{k+1} = x^k - \gamma(\text{diag}(G^{k+1}))^{-1}\nabla f(x^k)$$

# Idea 4

- RMSProp:

$$G_i^{k+1} = \beta_2 G_i^k + (1 - \beta_2)([\nabla f(x^k)]_i)^2$$
$$x^{k+1} = x^k - \gamma(\text{diag}(G^{k+1}))^{-1} \nabla f(x^k)$$

- Adam:

$$v^{k+1} = \beta_1 v^k + (1 - \beta_1) \nabla f(x^k)$$
$$G_i^{k+1} = \beta_2 G_i^k + (1 - \beta_2)([\nabla f(x^k)]_i)^2$$
$$x^{k+1} = x^k - \gamma(\text{diag}(G^{k+1}))^{-1} v^{k+1}$$

# Idea 5

- Before that, we used the inner product of gradients, why not take the outer one:

$$\nabla^2 f(x) \to \sqrt{\nabla f(x^k)(\nabla f(x^k))^T}$$

# Idea 5

- Before that, we used the inner product of gradients, why not take the outer one:
$$\nabla^2 f(x) \to \sqrt{\nabla f(x^k)(\nabla f(x^k))^T}$$
or to make more robust consider
$$\nabla^2 f(x) \to \sqrt{\sum_{t=1}^{k} \nabla f(x^t)(\nabla f(x^t))^T}$$

# Idea 5

- Before that, we used the inner product of gradients, why not take the outer one:

$$\nabla^2 f(x) \to \sqrt{\nabla f(x^k)(\nabla f(x^k))^T}$$

or to make more robust consider

$$\nabla^2 f(x) \to \sqrt{\sum_{t=1}^{k} \nabla f(x^t)(\nabla f(x^t))^T}$$

- Unlike the inner product, this product is very expensive to store.

# Idea 5

- Here it is time to memorize that typically in ML:

$$X, \nabla f(X) \in \mathbb{R}^{d_1 \times d_2}$$

and we work with matrix optimization.

# Idea 5

- Here it is time to memorize that typically in ML:

$$X, \nabla f(X) \in \mathbb{R}^{d_1 \times d_2}$$

  and we work with matrix optimization.
- And in fact we deal with

$$\sum_{t=1}^{k} \text{vec}[\nabla f(x^t)](\text{vec}[\nabla f(x^t)])^T$$

# Idea 5

- It can be proved that

$$\sum_{t=1}^{k} \text{vec}[G^t](\text{vec}[G^t])^T \preceq \left(\sum_{t=1}^{k} G^t (G^t)^T\right)^{\frac{1}{2}} \otimes \left(\sum_{t=1}^{k} (G^t)^T G^t\right)^{\frac{1}{2}}$$

Here $G^t = \nabla f(X^t)$.

# Idea 5

- It can be proved that

$$\sum_{t=1}^{k} \text{vec}[G^t](\text{vec}[G^t])^T \preceq \left( \sum_{t=1}^{k} G^t (G^t)^T \right)^{\frac{1}{2}} \otimes \left( \sum_{t=1}^{k} (G^t)^T G^t \right)^{\frac{1}{2}}$$

Here $G^t = \nabla f(X^t)$.

- It means that we can use two matrices: $d_1^2$ and $d_2^2$ instead of $d_1^2 \cdot d_2^2$.

# Idea 5

- Let us introduce:

$$L^{k+1} = L^k + G^k(G^k)^T, \quad R^{k+1} = R^k + (G^k)^T G^k$$

to collect this two matrices.

# Idea 5

- Let us introduce:

$$L^{k+1} = L^k + G^k (G^k)^T, \quad R^{k+1} = R^k + (G^k)^T G^k$$

to collect this two matrices. And use

$$\nabla^2 f(X^k) \to (L^{k+1})^{1/4} \otimes (R^{k+1})^{1/4}.$$

# Idea 5

- Like in Newton's method:

$$\text{vec}[X^{k+1}] = \text{vec}[X^k] - \gamma(\text{"Hessian"})^{-1} \cdot \text{vec}[\nabla f(X^k)]$$

# Idea 5

- Like in Newton's method:

$$\mathrm{vec}[X^{k+1}] = \mathrm{vec}[X^k] - \gamma(\text{"Hessian"})^{-1} \cdot \mathrm{vec}[\nabla f(X^k)]$$

- For our new approximation

$$\mathrm{vec}[X^{k+1}] = \mathrm{vec}[X^k] - \gamma \cdot ((L^k)^{-\frac{1}{4}} \otimes (R^k)^{-\frac{1}{4}})\mathrm{vec}(G^k)$$

# Idea 5

- Like in Newton's method:

$$\text{vec}[X^{k+1}] = \text{vec}[X^k] - \gamma(\text{"Hessian"})^{-1} \cdot \text{vec}[\nabla f(X^k)]$$

- For our new approximation

$$\text{vec}[X^{k+1}] = \text{vec}[X^k] - \gamma \cdot ((L^k)^{-\frac{1}{4}} \otimes (R^k)^{-\frac{1}{4}})\text{vec}(G^k)$$

$$\text{vec}[X^{k+1}] = \text{vec}[X^k] - \gamma \cdot \text{vec}((L^k)^{-\frac{1}{4}} G^k (R^k)^{-\frac{1}{4}})$$

This is Shampoo method!

# Idea 6

- We have already come across several times that simplifying is not always a bad thing!

# Idea 6

- We have already come across several times that simplifying is not always a bad thing!
- Let us consider the simplified version of the Shampoo – without collecting $L$ and $R$:

$$X^{k+1} = X^k - \gamma(G^k[G^k]^T)^{-1/4} \cdot G^k \cdot ([G^k]^T G^k)^{-1/4}$$

Here $G^k = \nabla f(X^k)$.

# Idea 6

- We have already come across several times that simplifying is not always a bad thing!
- Let us consider the simplified version of the Shampoo – without collecting $L$ and $R$:

$$X^{k+1} = X^k - \gamma(G^k[G^k]^T)^{-1/4} \cdot G^k \cdot ([G^k]^T G^k)^{-1/4}$$

Here $G^k = \nabla f(X^k)$.

- It can be proved that

$$X^{k+1} = X^k - \gamma U^k V^k,$$

where $U^k V^k$ from SVD for $\nabla f(X^k) = U^k \Sigma^k V^k$.

# Idea 6

- And we come to the Muon method:

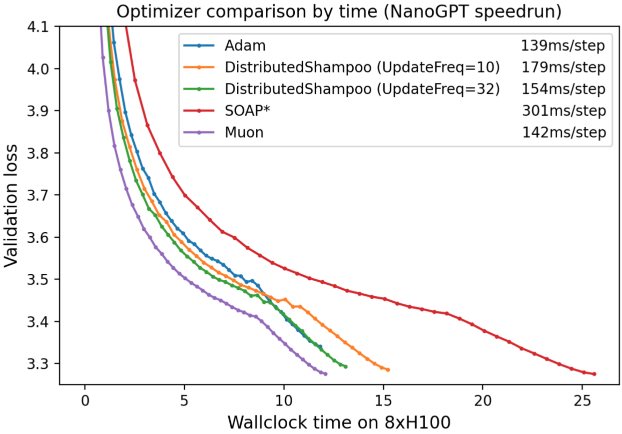
$$V^{k+1} = \beta_1 V^k + (1 - \beta_1)\nabla f(X^k)$$

$$H^{k+1} = Newton - Schulz(V^{k+1}) \quad \text{to get} \quad U^k V^k$$

$$X^{k+1} = X^k - \gamma H^{k+1}$$

# Idea 6

Original experiment on nanoGPT:



# Final idea

- We start from

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2}\|x - x^k\|_2^2.$$

# Final idea

- We start from

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2}\|x - x^k\|_2^2.$$

- That we can very the upped bound:

$$f(x) \leq f_k(x) = f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L_{\text{norm}}}{2}\|x - x^k\|_{\text{norm}}^2.$$

And find new methods with this trick.

# Final idea

- For matrix we have the same approximation:

$$f(X) \leq f_k(X) = f(X^k) + \langle \nabla f(X^k), X - X^k \rangle + \frac{L_{\text{norm}}}{2} \|X - X^k\|_{\text{norm}}^2.$$

# Final idea

- For matrix we have the same approximation:

$$f(X) \leq f_k(X) = f(X^k) + \langle \nabla f(X^k), X - X^k \rangle + \frac{L_{\text{norm}}}{2} \|X - X^k\|_{\text{norm}}^2.$$

But for the matrices we have more opportunities in terms of norms:

**Matrix induced norm**

$$\|M\|_{\alpha,\beta} = \max_{x \in \mathbb{R}^d} \frac{\|Mx\|_\beta}{\|x\|_\alpha}$$

# Final idea

- For matrix we have the same approximation:

$$f(X) \leq f_k(X) = f(X^k) + \langle \nabla f(X^k), X - X^k \rangle + \frac{L_{\text{norm}}}{2} \|X - X^k\|_{\text{norm}}^2.$$

But for the matrices we have more opportunities in terms of norms:

**Matrix induced norm**

$$\|M\|_{\alpha,\beta} = \max_{x \in \mathbb{R}^d} \frac{\|Mx\|_\beta}{\|x\|_\alpha}$$

$\alpha = 2$, $\beta = 2$ gives Muon, Shampoo, SOAP.

# Final idea

How to compute different LMO:

$$\arg\min_{X \in \mathbb{R}^{d\times d}} f_k(X)$$

| | $1 \to$ RMS (ColNorm) | $1 \to \infty$ (Sign) | RMS $\to$ RMS (Spectral) | RMS $\to \infty$ (RowNorm) |
|---|---|---|---|---|
| **Norm** | $\max_j \frac{1}{\sqrt{d_{\text{out}}}} \lVert \operatorname{col}_j(A) \rVert_2$ | $\max_{i,j} \lvert A_{i,j} \rvert$ | $\sqrt{d_{\text{in}}/d_{\text{out}}} \lVert A \rVert_{\mathcal{S}_\infty}$ | $\max_i \sqrt{d_{\text{in}}} \lVert \operatorname{row}_i(A) \rVert_2$ |
| **LMO** | $\operatorname{col}_j(A) \mapsto -\sqrt{d_{\text{out}}} \frac{\operatorname{col}_j(A)}{\lVert \operatorname{col}_j(A) \rVert_2}$ | $A \mapsto -\operatorname{sign}(A)$ | $A \mapsto -\sqrt{d_{\text{out}}/d_{\text{in}}}\, UV^\top$ | $\operatorname{row}_i(A) \mapsto -\frac{1}{\sqrt{d_{\text{in}}}} \frac{\operatorname{row}_i(A)}{\lVert \operatorname{row}_i(A) \rVert_2}$ |

RMS = 2

# Final idea

Recommendations for choosing optimizers for each layer:

**Recommendation 3.1.** We refer to the instantiation of uSCG and SCG using operator norms as UNCONSTRAINED SCION and SCION respectively (*cf.* Algorithm 3), which stands for **S**tochastic **C**onditional Grad**i**ent with **O**perator **N**orms. We recommend the following configurations of the layer norms (First layer → Intermediary layers → Last layer):

(i) image domains: Spectral → Spectral → Sign

(ii) 1-hot input: ColNorm → Spectral → Sign

# Final idea

Recommendations for choosing optimizers for each layer:

**Recommendation 3.1.** We refer to the instantiation of uSCG and SCG using operator norms as UNCONSTRAINED SCION and SCION respectively (*cf.* Algorithm 3), which stands for **S**tochastic **C**onditional Grad**i**ent with **O**perator **N**orms. We recommend the following configurations of the layer norms (First layer $\to$ Intermediary layers $\to$ Last layer):

(i) image domains: Spectral $\to$ Spectral $\to$ Sign

(ii) 1-hot input: ColNorm $\to$ Spectral $\to$ Sign

It looks like the beginning of research!