The Pervasive Role of Composition in Machine Learning: From Neural Architectures to Learning Dynamics

Anders Karlsson

July 2025

Notes from a talk delivered at the University of Oxford on July 22, 2025 as part of the conference *Geometric and Combinatorial Methods in the Foundations of Computer Science and Artificial Intelligence*. Organisers: Michael Bronstein, Cornelia Drutu, Dawid Kielak, Alexander Lubotzky, Assaf Naor, Standa Zivny.

Thanks to Gargi Biswas and Paula Heim for putting together a first draft.

Title: The Pervasive Role of Composition in Machine Learning

Abstract: From the layer maps of neural networks to training procedures and reinforcement learning, compositions of transformations permeate modern AI. These compositions often involve randomly selected maps, as in weight initialization, stochastic gradient descent (SGD), and dropout. In reinforcement learning, Bellman-type operators are iterated, often with randomness, to update reward structures and strategies. I will discuss the mathematics and geometry underlying the composition of random transformations. In particular, I will explain a general limit law established in joint work with Gouëzel. Moreover, I will discuss a possible cut-off phenomenon related to the depth of neural networks and the influence of iteration order. Motivated by these observations, and in collaboration with Avelin, Dherin, Gonzalvo, Mazzawi, and Munn, we propose backward variants of SGD that improve stability and convergence while maintaining generalization.

Introduction: Compositional Structures

When trying to identify structural mathematical features of modern AI and machine learning (ML), one finds that long chains of compositions of transformations are a recurrent ingredient. For example, they are a defining feature of:

- artificial neural networks
- the training dynamics of AI models, and
- reinforcement learning algorithms.

These three classes of examples will follow us throughout the discussion.

What is meant by long chains of compositions? Let X and Y be sets, and consider a transformation, or map, $T : X \to Y$, with $T(x)$ denoting the image of $x \in X$. For simplicity, consider the case $X = Y$ and given two maps T and S one can form their composition

$$T \circ S$$

by first applying S and then T. Given a sequence of such maps $T_1, ..., T_n$ one can consider their composition product

$$Z_n := T_1 \circ T_2 \circ \cdots \circ T_n,$$

For example, say the maps have the form $T_i(x) = \sigma(W_i x + b_i)$, where σ is an activation function, like ReLU which is $\sigma(t) = \max\{0, t\}$ applied coordinate-wise, A a matrix, and b a vector. Then Z_n is called a *(feed-forward) neural network* or *multi-layer perceptron (MLP)*. More complicated neural network architectures, can often be written in a similar form, or is a juxtaposition of such. *Long* is essentially a different word

for the more commonly used *deep*. Indeed, this usage of the word deep is what has given the name *deep learning*.

The second type of example is the training of the networks. Here, most classically, $T(x) = x - \eta \nabla L(x)$, which is the gradient descent update map, when trying to minimize a given loss function $L(x)$. In the case of stochastic gradient (or other variants like Adam), one would have a random sequence of maps of this form but only using parts of the loss function $L = \sum L_i$. Training the model amounts to applying the composition $Z_n(x_0)$ to some initial point x_0 in the parameter space chosen in a good way. (Note here a point that we will come back to: the maps are composed in the opposite order to what is indicated in the above displayed formula for Z_n.) The number n of iterates is easily in the thousands.

Third, in reinforcement learning, an agent learns a value function $Q(s, a)$ that estimates the expected total reward for taking action a in state s. The update rule is given by

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\big(r(s,a,s') + \gamma \max_{a'} Q(s',a')\big) = TQ(s,a),$$

which defines the *Q-function* update map T.

Nonexpansive Maps

In the ML-context, and in mathematics more generally [Kar21], it turns out that there is often a metric (distance function) d on the set X, such that the transformations T are *nonexpansive*, meaning that

$$d(Tx, Ty) \leq d(x, y) \quad \forall x, y.$$

The ubiquity of this property has perhaps not been sufficiently emphasized. Sometimes one benefits from relaxing the metric space axioms, and other times one instead looks at the induced action of T on a naturally associated auxiliary space, see [Kar21, AK22, Kar23] for examples and references.

A first example arises from neural network layer maps:

$$T_i(x) = \sigma(W_i x + b_i),$$

where σ is an activation function. These maps are nonexpansive in the supremum-norm if W_i and b_i are nonnegative. Neural networks with such layers are sometimes called nonnegative neural networks. See [DGT25] and [AK22] for more details.

The nonexpansivity can be explained by an old observation of Blackwell: Let $T : B(S) \to B(S)$ be a map on the space of bounded functions on a set S. The space has a partial order $f \leq g$ which means that $f(s) \leq g(s)$ for all s. If T satisfies the two properties

1. $T(f + C) \leq Tf + \gamma C$, and
2. $f \leq g \implies Tf \leq Tg$,

then T is a contraction if $\gamma < 1$ and nonexpansive if $\gamma = 1$. This observation is also fundamental for reinforcement learning.

As pointed out in an appendix of [DAK+25] under the assumptions of a Polyak-Łojasiewicz (PL) condition and a Lipschitz condition on the gradients, one can define a loss pseudo-metric in the interpolation regime, where the loss $L \geq 0$ attains the minimum 0, and the gradient update map is nonexpansive. In the case of batches, as in stochastic gradient descent, the maps are nonexpansive only on average. The *loss pseudo-metric* is

$$d(x,y) = \begin{cases} L(x) + L(y) & \text{if } x \neq y, \\ 0 & \text{if } x = y. \end{cases}$$

The *Polyak-Łojasiewicz (PL)* condition asks that there exists a constant $\mu > 0$ such that

$$\frac{1}{2} \|\nabla L(\theta)\|^2 \geq \mu L(\theta)$$

for all $\theta \in S$. This condition is satisfied by strongly convex functions, but it also tends to hold for over-parametrized neural networks (which are typically not convex), as argued in [LZB22].

Reinforcement Learning

As noted earlier, in reinforcement learning (RL) the Q-update map defines a random transformation. Thanks to the above observation of Blackwell, the Bellman operators, which are fundamental for RL, are nonexpansive or even contractive. Therefore, a significant part of reinforcement learning, such the Q-update iteration, can be viewed as a composition of random nonexpansive transformations.

Random Deep Compositions: A Noncommutative Law of Large Numbers

In several ML-situations the composition products are moreover random. This means that the maps T_i as above are chosen independently at random with a common probability distribution supported on the relevant family of transformations. In our fundamental examples,

- artificial neural networks are initialized before training by selecting random layer maps
- the training uses randomly chosen batches giving rise to randomly selected gradient update maps T_i
- in RL the various update maps are random, because rewards and transitions are random in general.

Is there anything one can say about such random composition products? Consider first the following fundamental theorem in probability theory. Its first version dates back to the early 17th century in the work of J. Bernouilli and is called the Law of Large Numbers:

For i.i.d. real numbers $T_1, T_2, \dots, T_n$, we have

$$\frac{1}{n}(T_1 + T_2 + \cdots + T_n) \to \mathbb{E}[T] \quad \text{as } n \to \infty.$$

Why the notation T_i here? This is because we can view these numbers instead as translations of the real axis $X = \mathbb{R}$, and they commute since the composition is just addition. The process is a random walk on the real line. This is a key viewpoint for its generalization.

If we replace sums with compositions, we have

$$Z_n := T_1 \circ T_2 \circ \cdots \circ T_n.$$

In mathematics and physics, they have been considered at least since the 1950s (Bellman, Dyson, Furstenberg, Kesten, and others). Or one could perhaps say even earlier in finite Markov chain theory (Markov, Poincaré and others). Note that in contrast to the classical law of large numbers the maps T_i do not commute, i.e. $T_i \circ T_j$ may not be equal to $T_j \circ T_i$. This is one of the reasons that it is not so clear how the law of large numbers should be generalized.

Nevertheless, there is a good and useful such generalization:

Theorem (Gouëzel–Karlsson, [GK20, Kar23]). *Let X be a (weak) metric space and $T_i : X \to X$ be random (i.i.d) non-expansive maps under a finite first moment condition. Then there exists a metric functional h_ω such that*

$$\frac{1}{n} h_\omega(T_1 \circ T_2 \circ \cdots \circ T_n(x)) \to \mathbb{E}[T] \quad \text{as } n \to \infty.$$

Here, $\mathbb{E}[T]$ represents the average growth rate of the product, while h_ω provides the direction of iteration. One can view the composition $Z_n x$ as a random walk on X. In Banach spaces, one could instead use h_ω a linear functional of norm 1. Special cases include the Birkhoff ergodic theorem, the von Neumann mean ergodic theorem, the Oseledets multiplicative ergodic theorem (random matrix products), random walks on groups and an extension of Thurston's spectral theorem for homeomorphisms of surfaces to random products. See [Kar23] for a more complete discussion with relevant references. As remarked above, many transformations in AI are nonexpansive and chosen at random, see [AK22, Kar23].

A Cutoff Phenomenon in Deep Networks?

In [AK22], some small experiments point towards the existence of a depth threshold, when selecting random layer maps (for example, at initialization). Below this depth, the composition is out of equilibrium; beyond it, a stationary measure emerges. This phenomenon seems first to have been observed in card-shuffling (seven shuffles are enough as Aldous and Diaconis famously showed in the 1980s). For the current development on this topic more generally we refer to the lecture notes of Salez [Sal25].

Forward vs Backward Composition

When iterating a single map T, there is no order to the composition product T^n. This changes if we instead have a sequence of noncommuting maps to compose. The order of composition becomes significant, there are two basic choices:

$$\text{Forward: } T_n \circ \cdots \circ T_1(x) \quad \text{vs} \quad \text{Backward: } T_1 \circ \cdots \circ T_n(x).$$

Examples of forward iteration include the Oseledets' theorem, the backward Euler method, and stochastic gradient descent (SGD), whereas backward iteration arises in random walks (as in the theorem stated above) and continued fraction expansions. The order may get reversed when considering the induced action on an associated space. This is discussed in [Kar23]. A well-known example is the adjoint, or the transpose of matrices.

An Illustrative Example

Let $X = \{\text{dry}, \text{wet}\}$ with metric $d(\text{dry}, \text{wet}) = 1$. Define two operations of this set T_1, to clean-up, and T_2, to spill. Even a young child would recognize that these two operations do not commute. In addition they are as contractive as it gets since both map the space to a single point. Notice now that forward iterations oscillate between the states, while backward iteration converges to one them (randomly; after one step). This shows that in the presence of noncommutativity, backward composition may lead to convergence while forward composition does not.

Training via Gradient Descent

In supervised learning, we are given input-output pairs (x, y) and the aim is to learn a function $f(x)$ such that $y \approx f(x)$, where f is in some family of function with parameters w. If the function takes only a discrete set of values, the problem is called a *classification problem* and otherwise a *regression problem*.

We define a loss function $L(w)$, such as mean squared error, and update parameters using gradient descent:

$$x_n = T^n(x_0),$$

where the T has the form displayed above.

Stochastic Gradient Descent (SGD)

Instead of computing the gradient over the full training dataset, random mini-batches are most commonly used, for cost efficiency and for better generalization. This gives

$$x_n = T_n \circ T_{n-1} \circ \cdots \circ T_1(x_0).$$

This is a forward iteration.

Backward SGD

Inspired by the noncommutative and metric composition framework, *Backward SGD* would instead be

$$x_n = T_1 \circ T_2 \circ \cdots \circ T_n(x_0),$$

Based on the nonexpansive properties discussed above, backward SGD may provide better convergence. The paper by Dherin, Avelin, Karlsson et al. [DAK$^+$25] indeed shows in large experiments that backward SGD is much more stable and somewhat more convergent than standard SGD. However, it is computationally much more expensive. An open question is how to exploit the advantages of backward thinking for the training of networks in a cost-effective manner.

Conclusion

Long, or "deep", compositions of random maps appear frequently in machine learning, as they do throughout mathematics and science more generally. They are often nonexpansive in some naturally associated metric space. In this case, the non-commutative law of large numbers stated above applies to give convergence in direction. Networks may exhibit a cut-off depth, but this has not been much studied. When thinking of iterative schemes in terms of transformations, there is an a priori theoretical choice of forward versus backward composition. They behave very differently in the noncommutative setting, yet both are natural in different contexts. Implementing intermittent or approximative backward steps may have advantages in some situations, although how to do this efficiently in practice remains to be explored.

References

[AK22] Benny Avelin and Anders Karlsson. Deep limits and a cut-off phenomenon for neural networks. *J. Mach. Learn. Res.*, 23:Paper No. 191, 29, 2022.

[DAK+25] B. Dherin, B. Avelin, A. Karlsson, J. Gonzalvo, H. Mazzawi, and M. Munn. How iteration composition influences convergence and stability in deep learning. *to appear*, 2025.

[DGT25] Piero Deidda, Nicola Guglielmi, and Francesco Tudisco. Nonlinear joint spectral radius. *https://arxiv.org/pdf/2507.11314*, pages 1–56, 2025.

[GK20] Sébastien Gouëzel and Anders Karlsson. Subadditive and multiplicative ergodic theorems. *J. Eur. Math. Soc. (JEMS)*, 22(6):1893–1915, 2020.

[Kar21] Anders Karlsson. From linear to metric functional analysis. *Proc. Natl. Acad. Sci. USA*, 118(28):Paper No. e2107069118, 5, 2021.

[Kar23] Anders Karlsson. Generalized Lyapunov exponents and aspects of the theory of deep learning. In *New trends in Lyapunov exponents*, CIM Ser. Math. Sci., pages 99–118. Springer, Cham, 2023.

[LZB22] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Appl. Comput. Harmon. Anal.*, 59:85–116, 2022.

[Sal25] Justin Salez. Modern aspects of markov chains: entropy, curvature and the cutoff phenomenon. *https://arxiv.org/abs/2508.21055*, pages 1–92, 2025.