Learning Associative Memories with Gradient Descent

Vivien Cabannes [1] **Berfin Şimşek** [2] **Alberto Bietti** [2]

Abstract

This work focuses on the training dynamics of one associative memory module storing outer products of token embeddings. We reduce this problem to the study of a system of particles, which interact according to properties of the data distribution and correlations between embeddings. Through theory and experiments, we provide several insights. In overparameterized regimes, we obtain logarithmic growth of the "classification margins." Yet, we show that imbalance in token frequencies and memory interferences due to correlated embeddings lead to oscillatory transitory regimes. The oscillations are more pronounced with large step sizes, which can create benign loss spikes, although these learning rates speed up the dynamics and accelerate the asymptotic convergence. In underparameterized regimes, we illustrate how the cross-entropy loss can lead to suboptimal memorization schemes. Finally, we assess the validity of our findings on small Transformer models.

1. Introduction

Modern machine learning often involves discrete data, whether it is labels in a classification problem, sequences of text tokens in language modeling, or sequences of discrete codes when dealing with other modalities. In such settings it is common to consider cross-entropy objectives, and to embed each input and output token into high-dimensional embedding vectors. Deep learning architectures consist in transforming the embedding vectors by a cascade of linear matrix multiplications together with non-linear operations. This work aims at obtaining a fine-grained understanding of training a single such linear layer with the cross-entropy loss and fixed embeddings. Indeed, one could then see the training of deep models as the joint training of multiple such associative memory models. Although our setup admits a standard convex analysis treatment, we felt the need to provide a finer picture, more inline with behaviors observed when training large neural networks.

We consider N input tokens $x \in [N]$, each associated with some output $y = f^*(x) \in [M]$ for a deterministic function $f^* : [N] \to [M]$ and some number M of classes.[1] The input variable x is assumed to be drawn from a data distribution $p(x)$. The goal is to learn the input-output relationship f^* with a model of the form

$$f_W(x) = \operatorname*{arg\,max}_{y \in [M]} \langle u_y, W e_x \rangle, \quad \text{with} \quad W \in \mathbb{R}^{d \times d} \quad (1)$$

where $e_x, u_y \in \mathbb{R}^d$ are fixed input/ouput token embeddings with $d \geq 2$, and W is a parameter to be learned. The quality of a model f_W is typically measured through the 0-1 loss

$$\mathcal{L}_{01}(f_W) = \mathbb{E}_{X,Y}[\mathbf{1}_{f_W(X) \neq Y}] = \sum p(x) \mathbf{1}_{f_W(x) \neq f^*(x)},$$

while W is learned by optimizing a surrogate loss with gradient methods. We will focus on the cross-entropy loss

$$\mathcal{L}(W) = \mathbb{E}_{X,Y}\left[\ell(W; x, y)\right] \quad (2)$$

$$\ell(W; x, y) = \log\Big(\sum_{z \in [M]} e^{\langle u_z, W e_x \rangle}\Big) - \langle u_y, W e_x \rangle. \quad (3)$$

This loss is also known as softmax, multinomial logistic or negative log-likelihood loss.

The model (1) can be seen as an associative memory, which stores or "memorizes" pairwise associations (x, y). Associative memories originate in the neural computation literature, where they were used to model how the brain stores information (see, e.g., Willshaw et al., 1969; Longuet-Higgins et al., 1970), and solve algorithmic tasks with a machine learning perspective (see, e.g., Hopfield, 1982; Hopfield & Tank, 1985). These models have gained in popularity recently, notably as candidates to explain the inner workings of some deep neural networks (Geva et al., 2021; Schlag et al., 2021; Ramsauer et al., 2021; Bietti et al., 2023; Cabannes et al., 2024). This line of work motivates the thorough study of the training dynamics behind associative memory models. This is the goal of our paper, formalized as Problem 1.

Problem 1. *Understand the training dynamics of the associative memory model with the cross-entropy loss.*

[1] Meta AI [2] Flatiron. Correspondence to: <vivc@meta.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] We use the notation $[p] = \{1, 2, \ldots, p\}$.

Training dynamics on the logistic loss have been the subject of a vast line of work (see, e.g., Soudry et al., 2018; Ji & Telgarsky, 2019; 2021; Lyu & Li, 2019; Wu et al., 2023). In contrast to these works, we focus on the specific structure arising from our associative memory setup. In particular, we characterize phenomena observed empirically in deep learning, such as loss spikes and oscillations, which are often necessary for faster optimization. While oscillatory behaviors have been studied in the literature on large learning rates and *edge-of-stability* (Cohen et al., 2020; Nakkiran, 2020; Beugnot et al., 2022; Agarwala et al., 2023; Bartlett et al., 2023; Chen & Bruna, 2023; Rosenfeld & Risteski, 2023; Wu et al., 2023), our setup provides a new perspective on the matter, involving correlated embeddings and imbalanced token frequencies.

Contributions. With the goal of resolving Problem 1, we make the following contributions:

- We show that all gradient dynamics (i.e. stochastic or deterministic, continuous-time or discrete-time) reduce to a non-linear system of interacting particles.
- We solve the deterministic dynamics for orthogonal embeddings, where the memories do not interfere much.
- We illustrate typical training behaviors in the overparameterized case $d \geq N$ by solving the system with two-particles. In particular, we show that competition between memories can lead to benign oscillations and loss spikes, especially when considering large step-size, although large learning rates accelerate the asymptotic convergence toward robust solutions of the underlying classification problem.
- In limited capacity regimes ($d < N$), we illustrate precisely the deterministic dynamics for $d = M = 2$, when $N \geq 2$ particles interact. We showcase how the competition between memories can ultimately erase most of them.

We complement our analysis with experiments, investigating small multi-layer Transformer models with our associative memory viewpoint and identifying similar behaviors to those pinpointed in the simpler models.

2. The Many Faces of Problem 1

This section focuses on the disambiguation of Problem 1.

What type of understanding. Although the training dynamics are the result of deterministic computations on a computer, which could be described exhaustively, the causal factors behind their behavior are often too numerous for us to fully comprehend, even for the simple model (1). To overcome this issue, one can abstract coarse quantities that play important roles in many training scenarios, such as (i) the dimension and geometry of embeddings; (ii) data distributional properties, such as imbalanced token frequencies and heavy tails; (iii) the optimization algorithms and their hyperparameters, particularly the learning rate. We aim to highlight the effect of these factors, whose understanding would help predict the outcome of alternative training choices such as bigger learning rates, or different data curricula. Levels of understanding vary from rigorous math on small models, to controlled experiments on more complex models, or insights from the training of large-scale models without many ablation studies. This paper aims for a theoretical study in between these two extremes.

Which dynamics. We classify dynamics into five types, providing coarse and fine approximations of the dynamics used in practice to train neural networks.

Gradient flow. The gradient flow dynamics consists in letting the weight matrix W evolves according to the equation

$$\mathrm{d}W = -\nabla\mathcal{L}(W_t)\,\mathrm{d}t. \tag{4}$$

From initialization $W = W_0$, this deterministic evolution pushes W_t towards the lower value of the "potential" $\mathcal{L}$. In our case, $\mathcal{L}$ is convex, but does not always have a minimum as it might be minimized by a sW_* for s going to infinity.

Gradient descent. Gradient descent is the discrete-time approximation of the gradient flow dynamics, namely

$$\Delta_t W = -\eta_t \nabla\mathcal{L}(W_t), \tag{5}$$

where $\Delta_t W = W_{t+1} - W_t$ and η_t is a learning rate. In the continuous case, the learning rate corresponds to a reparameterization of the time with $ds = \eta_t\,\mathrm{d}t$. However, in the discrete-time regime, the learning rate is an important parameter that does influence the dynamics.

Stochastic gradient flow. In practice, following full batch dynamics, i.e., dynamics that involve processing all the data at all time to compute the gradient of $\mathcal{L}(W)$, is quite costly and inconvenient. In those cases, one can process a random subset of data instead to get a good estimate of $\nabla\mathcal{L}(W)$. This randomness can be modeled as a perturbation of the dynamics due to some random noise with zero mean. For gradient flow, the stochasticity is naturally modeled with a Brownian motion $\mathcal{E}_t$,

$$\mathrm{d}W = -\nabla\mathcal{L}(W_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}\mathcal{E}_t \tag{6}$$

where σ_t is the variance of the updates, i.e. of the gradient $\nabla\mathcal{L}(W_t; B) := \sum_{(x,y)\in B} \nabla\ell(W_t; x, y)$, when considering random batches B of data.

Stochastic gradient descent. For discrete-time dynamics, stochastic gradient descent can be written as

$$\Delta_t W = -\eta_t(\nabla\mathcal{L}(W_t) + \varepsilon_t), \tag{7}$$

where ε_t is some random variable. When the descent is "unbiased", this variable has zero mean. Typically,

$$\nabla \mathcal{L}(W_t) + \varepsilon_t = \sum_{(x,y)\in B} \nabla \ell(W_t; x, y),$$

for some random mini-batch B of data with $y = f^*(x)$.

Practical descent. Practitioners often use variants of stochastic gradient descent that are known to perform well empirically. These typically involve momentum in the descent, re-conditioning the gradient (Kingma & Ba, 2015), and the addition of normalization layers in the architecture (see, e.g., Ioffe & Szegedy, 2015; Ba et al., 2016).

This paper focuses on gradient flow and gradient descent, postponing the study of other dynamics for future work. We discuss the consistency of these methods, i.e., if they are reaching the best possible performance, their asymptotic convergence behaviors, as well as finite-time behaviors.

3. Memories as Interacting Particles

The section reduces the training dynamics to a system of interacting particles, where the particles correspond to input-output associations.

To simplify the analysis of our model, a few simple observations can be made. First, one can identify matrices with two-dimensional tensors, highlighting the linearity of our model $\langle u_j, We_i\rangle = \langle W, u_j \otimes e_i\rangle$ in the tensor space $\mathbb{R}^d \otimes \mathbb{R}^d \cong \mathbb{R}^{d\times d}$. Secondly, recall that the loss (3) corresponds to the negative log-likelihood

$$\ell(W; x, y) = -\log p_W(y|x),$$

of the probability p_W whose conditionals are parameterized as a soft-max over the scores $\langle e_x, W u_y\rangle$,

$$p_W(y|x) \propto \exp(\langle W, e_x \otimes u_y\rangle). \tag{8}$$

The chain rule leads to the following formula,

$$\nabla \ell(W; x, y) = \sum_{z\in[M]} p_W(z\,|\,x)(u_z - u_y) \otimes e_x. \tag{9}$$

The gradient formula shows that the dynamics take place on the span of the $(u_j - u_k) \otimes e_i$ with $i \in [N]$ and $j, k \in [M]$ up to an affine shift due to initialization.

The resulting training dynamics can be studied by tracking projections onto the input and output embeddings, or onto another family generating the tensor space $\mathbb{R}^d \otimes \mathbb{R}^d$, which leads to a system of particles with non-linear interactions.

Theorem 1 (Particle system)**.** *Define the particle w_{ij},*

$$w_{ij} = \langle W, u_j \otimes e_i\rangle = u_j^\top W e_i, \tag{10}$$

as well as the constant correlation parameters

$$\alpha_{ij} = \langle e_i, e_j\rangle\,, \quad \beta_{ijk} = \langle u_i, u_j - u_k\rangle\,. \tag{11}$$

The projected gradient can be rewritten as

$$\langle \nabla \ell(W; x, y), u_j \otimes e_i\rangle = \alpha_{ix} \sum_z \frac{\beta_{jzy} \exp(w_{xz})}{\sum_{k\in[M]} \exp(w_{xk})}.$$

Hence, all variations of gradient dynamics, (4), (5), (6) *and* (7)*, can be expressed as a (stochastic) system of interacting particles. For example, the gradient descent dynamics* (5) *is*

$$\Delta_t w_{ij} = \eta_t \sum_x p(x)\alpha_{ix} \sum_z \frac{\beta_{jf^*(x)z} \exp(w_{xz})}{\sum_{k\in[M]} \exp(w_{xk})}. \tag{12}$$

Similarly, the dynamics for the stochastic gradient descent consists in replacing $\sum_x p(x)$ by the summation over x in a random mini-batch in (12)*.*

Proof. The proof follows directly from (8) and (9). □

There are two reasons for interactions in this particle system; either the input embeddings are not orthogonal and the α's mix the particles, or there are more than two classes and β's mix the particles. Moreover, when the embeddings are not orthogonal, particles are not independent, since an increase of w_{ij} changes w_{ik}, as soon as $u_i^\top u_k \neq 0$. Note that multiple factors could lead to correlated embeddings, such as under-parameterization (viz., embeddings are necessarily correlated in low dimension), or semantic similarity in the case of trained embeddings (e.g., Mikolov et al., 2013).

Interacting particle systems commonly arise in other machine learning settings, e.g., to describe parameters in the mean-field regime of two-layer networks (Chizat & Bach, 2018; Mei et al., 2018; Rotskoff & Vanden-Eijnden, 2018), samples in certain approaches to generative modeling (Liu & Wang, 2016; Arbel et al., 2019), or both (Domingo-Enrich et al., 2021). However, these systems typically involve particles as discretizations of an underlying measure evolution, while we make no such connection here. Our dynamics may also be seen as training the middle layer of a three-layer linear network, and infinite-width dynamics for related models have been studied in (Jacot et al., 2021; Chizat et al., 2022). Yet, our focus is on the finite width ($d < \infty$) case, and we note that this suffices for optimal storage when d is sufficiently large compared to N and M.

The particle w_{xi} corresponds to the score assigned by W to the class i for the token x. Another set of sufficient statistics for the problem are the margins, which are defined by

$$m_i(x) = w_{xf^*(x)} - w_{xi} = (u_{f^*(x)} - u_i)^\top W e_x. \tag{13}$$

It corresponds to the difference between the scores assigned by the model (1) to the classes $f^*(x)$ and i, for the input x. When all the margins $(m_i(x))_i$ are positive, the token x is classified correctly.

4. Overparameterized Regimes

This section focuses on the case where $N \leq d$ and the (e_x) form a linearly independent family. In this setting, the optimization of the convex loss (2) will ensure perfect accuracy for our model, i.e. $f_W = f^*$.

4.1. Orthogonal Embeddings

We first solve the case where the embedding families (e_x) and (u_y) are both orthogonal. The orthogonality of the inputs implies $\alpha_{ix} = \mathbf{1}_{i=x} \left\| e_i \right\|^2$, in which case (12) shows that the gradient dynamics for W decouple on the $\mathbb{R}^d \otimes e_i$. In other terms, our model is implicitly fitting in parallel N parameters, the $(We_i)_i$, of N independent exponential families, the $p_W(y|x)$. As a consequence, we can forget the context variable and fix a $x \in [N]$ for the remainder of the section. For simplicity, we assume $f^*(x) = 1$.

Binary classification. Let us consider the binary case first. When $M = 2$, the dynamics on We_x evolves on the line $\mathbb{R} \cdot (u_1 - u_2)$, and is fully characterized by the margin

$$m_t = (u_1 - u_2)^\top W_t e_x.$$

An algebraic manipulation of (12) shows that this scalar quantity evolves according to the dynamics

$$(1 + \exp(m_t))\Delta_t m = c_x \eta_t,$$

$c_x = p(x) \left\| e_x \right\|^2 \left\| u_1 - u_2 \right\|^2$, and $\Delta_t m = m_{t+1} - m_t$. This discrete-time evolution can be solved recursively. A nice formula can be derived for the continuous-time version, i.e. for the flow (4) instead of the descent (5), where

$$(1 + \exp(m)) \,\mathrm{d}m = c_x \,\mathrm{d}t.$$

In particular when w is initialized at zero,

$$m_t + \exp(m_t) = c_x t.$$

This equation is inverted with the product logarithm, giving an exact expression for the margin evolution, proven in Appendix B.1.

Theorem 2 (Binary orthogonal)**.** *Let $M = 2$, and the input embeddings be orthogonal. The dynamics* (4), (5), (6) *and* (7) *lead to*

$$W_t = \sum_x m_t(x) \frac{u_1 - u_2}{\left\| u_1 - u_2 \right\|^2} \otimes \frac{e_x}{\left\| e_x \right\|^2} + \Pi_\perp W_0, \quad (14)$$

where $\Pi_\perp$ is the projection on the orthogonal of the span of the gradient updates. For gradient flow (4)*, there exists a $t_0 \in \mathbb{R}$ that depends on initial condition, e.g. $t_0 = -1/c_x$ when $W_0 = 0$, such that when $t \geq t_0$, the exact evolution is given by,*

$$m_t(x) = \log(c_x(t - t_0)) - h(c_x(t - t_0)), \quad (15)$$

where $c_x = p(x) \left\| e_x \right\|^2 \left\| u_1 - u_2 \right\|^2$ and h is a function such that $0 \leq h(x) \leq 2\log(x)/x$ for all $x \geq 1$. Similarly, for gradient descent (5) *with a learning rate η,*

$$m_t(x) \geq \log(\eta c_x(t - t_0)).$$

In particular, when $W_0 = 0$, it leads to the following bound on the loss,

$$\mathcal{L}(W_t) \leq \frac{1}{t\eta} \sum_x \frac{p(x)}{c_x}.$$

The setting of Theorem 2 is a special case of logistic regression with linearly separable data where all margins grow logarithmically without inhibiting each other. The loss monotonically decays in $O(1/\eta t)$ which corresponds to the rate of Wu et al. (2023), with the addition of the explicit dependence on the learning rate.

Multi-class. We now attack the multi-class case. Let

$$w_i = w_{xi} = u_i^\top W e_x,$$

and denote the partition function $A(w) = \sum \exp(w_i)$ with $w = (w_i)$. Let us focus for now on gradient flow for simplicity. When the (u_i) are orthogonal, the dynamics (12) becomes

$$\frac{A(w)}{A(w) - \exp(w_1)} \,\mathrm{d}w_1 = p(x) \left\| e_x \right\|^2 \left\| u_1 \right\|^2 \mathrm{d}t,$$
$$A(w) \exp(-w_i) \,\mathrm{d}w_i = -p(x) \left\| e_x \right\|^2 \left\| u_i \right\|^2 \mathrm{d}t,$$

for $i \neq 1$. In the multi-class case, the evolution of the margins $m_i = w_1 - w_i$ do not directly decouple from each other as in the binary case.

One can combine these differential equations to find many invariants of this system of interacting particles. In particular, $\exp(-w_i) - \exp(-w_j)$ stays constant over time.[2] Some algebraic manipulations implies the following evolution of the tightest margin (corresponding to $\arg\max_{i \neq 1} p_W(i|x)$)

$$(c_i + \exp(m_i)) \,\mathrm{d}m_i = p(x) \left\| e_x \right\|^2 (c_i + 1) \,\mathrm{d}t.$$

for some bounded function c_i that depends on initial conditions. This leads to the same logarithm convergence as in the binary case. For simplicity, we only report the asymptotic behavior in Theorem 3, which we prove in Appendix B.2.

Theorem 3 (Multi-class orthogonal)**.** *Assume that the input and output embeddings are orthonormal. For any initialization, the gradient flow dynamics* (4) *converges as*

$$\lim_{t \to \infty} \frac{W_t}{\log(t)} \propto \sum_x \Pi(u_{f^*(x)}) \otimes e_x,$$

where Π is the projection on the span of the $(u_i - u_j)$.

[2] This generates all the invariants of the dynamics but one, the remaining one is a consequence of $\mathrm{d}W \in \mathrm{Span}\left\{u_i - u_j\right\}_{ij} \otimes R^d$.

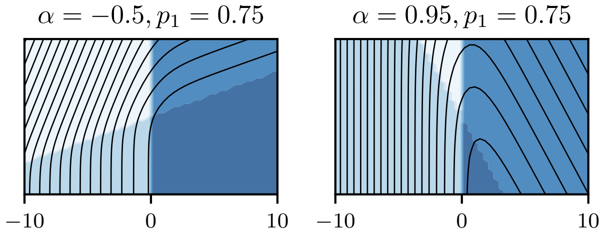

Figure 1: Level lines of $\mathcal{L}(W)$ for $N = d = 2$ as a function of $\gamma_i(W) := (u_2 - u_1)^\top W f_i$ where (f_i) is a basis of $\mathbb{R}^2$. Token embeddings have correlation α (16). We equally plot the value of $\mathcal{L}_{01}(W)$, dark blue meaning perfect accuracy, and white meaning null accuracy.

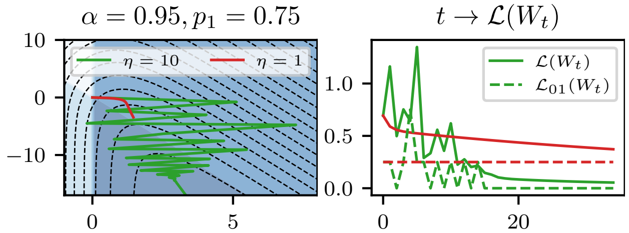

Figure 2: *Loss spikes.* Trajectories of W_t in the setting of Figure 1 for two learning rates η, $\eta = 10$ in green, $\eta = 1$ in red, and their traces in term of losses as a function of the number of epochs, here $t \in [35]$.

For gradient descent, one can express the updates for the margins similarly

$$A(w)\Delta_t m_i = \eta_t (A(w) - e^{w_1} + e^{w_i}) p(x) \|e_x\|^2 > 0.$$

Hence, the margins only increase during training, and the larger the learning rate, the faster the evolution. Indeed, one gradient step is enough to learn all the associations $x \to f^*(x)$, i.e. $f_{W_1}(x) = f^*(x)$ for any initialization. Continuing the training will continue to increase the margins, ultimately ensuring the convergence of W_t to the max-margin solution of the classification problem, as characterized by Theorem 3, making the final classifier robust to embedding displacements (Cortes & Vapnik, 1995).

To conclude, when the embeddings are orthogonal, the memories do not interfere much, and one can learn all the associations with one giant gradient step. This case still presents several behaviors of interest. First of all, Equation (15) shows that the association $x \to y$ is learned faster when x is frequent, i.e. $p(x)$ is large. Indeed, early in training, one can envision $W \simeq \sum_x p(x) u_y \otimes e_x$. However, later in training, the training dynamics will start saturating in the direction $u_y \otimes e_x$ for the frequent tokens, allowing the less frequent ones to catch up. The catch-up is facilitated by large learning rates. Ultimately, as shown by Equation (3), the final W does not depend on the token frequencies (see Byrd & Lipton, 2019 for related observations). In other terms, if the model has enough capacity to learn all the data (in our case, orthogonality implies $d \geq N$), then at the end of the training, it allocates equal capacity to every token even though some tokens are much rarer. Nonetheless, curating data to make them less redundant can make learning more efficient.

4.2. Particles Interfering

Let us now consider the case where $N \leq d$, but where memories interfere between them. We first notice that in the case when the input embeddings are orthogonal, correlated output embeddings introduce limited competition, and this case can largely be understood as a simplified version of the interaction between input embeddings.

Let us analyze the simple but instructive case $N = 2$ of two input tokens with $f^*(x) = x$, and

$$\alpha_{ij} = \langle e_i, e_j \rangle = \mathbf{1}_{i=j} + \alpha \mathbf{1}_{i \neq j}, \qquad \alpha \in [-1, 1]. \quad (16)$$

In other terms, the input embeddings are normalized and are α-correlated. Two margins are at play:

$$m_i = w_{ii} - w_{ij} = (u_i - u_j)^\top W e_i, \quad \{i, j\} = \{1, 2\}.$$

The interacting system (12) becomes,

$$\Delta m_i = c\eta_t \left(\frac{p_i}{1 + \exp(m_i)} - \frac{\alpha p_j}{1 + \exp(m_j)} \right), \quad (17)$$

where $c = \|u_1 - u_2\|^2$ and we denote $p_i = p(i)$ for readability. In the gradient dynamics, $x = 1$ pushes W in the direction $(u_1 - u_2) \otimes e_1$, which, when $\alpha \leq 0$, is positively correlated with the direction $(u_2 - u_1) \otimes e_2$ promoted by $x = 2$. As can be seen in Equation (17), when $\alpha \leq 0$, both margins increase during training, there is no competition between the memories, and a single gradient step is enough to reach perfect accuracy. To solve Equation (17), let us introduce the orthogonal family

$$f_1 = e_1 + e_2, \qquad f_2 = e_1 - e_2,$$

and project the dynamics on those directions with

$$\gamma_i = \frac{1}{2}(u_1 - u_2)^\top W f_i. \quad (18)$$

The evolution of the γ_i is governed by

$$\begin{aligned} \frac{\mathrm{d}\gamma_1}{\mathrm{d}ct} &= \frac{(1+\alpha)p_1}{1 + \exp(\gamma_2 + \gamma_1)} - \frac{(1+\alpha)p_2}{1 + \exp(\gamma_2 - \gamma_1)} \\ \frac{\mathrm{d}\gamma_2}{\mathrm{d}ct} &= \frac{(1-\alpha)p_1}{1 + \exp(\gamma_2 + \gamma_1)} + \frac{(1-\alpha)p_2}{1 + \exp(\gamma_2 - \gamma_1)}, \end{aligned} \quad (19)$$

From the second differential equation, we see that γ_2 always increases during the dynamics. The growth of γ_2 will slow down the growth of γ_1. These together imply that W grows logarithmically in one direction (f_2, which turns out to be the max-margin direction) and stays bounded in the orthogonal direction, which we prove in Appendix C.1 and is the object of the following theorem.

Theorem 4 (Two particles interacting)**.** *Let $N = 2$ with $f^*(x) = x$. Assume without restriction that $p_1 \geq p_2$. When Equation* (16) *holds, if W is initialized at zero, i.e. $W_0 = 0$, for gradient flow,*

$$\gamma_2(t) = \log(c_t t + 1) + O\left(\frac{\log(c_2 t + 1)}{c_2 t + 1}\right),$$
$$\gamma_1(t) = \frac{1}{2}\log\big(p_1/p_2\big) + O(1/t),$$

where $2p_2 \leq c_t/c_0 \leq 8p_1^3/p_2^2$ with $c_0 = (1-\alpha)c$. Similarly for gradient descent, with any step-size $\eta \geq 0$,

$$\gamma_2(t) \geq \log(\eta p_2 c_0 t + 1) + O\left(\frac{\log(t)}{t}\right),$$
$$\left|\gamma_1(t) - \frac{1}{2}\log(p_1/p_2)\right| \leq \eta(1-\alpha)p_1 + \frac{p_1}{2p_2} + O(1/t).$$

These results are consistent with Wu et al. (2023; 2024), although our focus is to obtain a fine-grained dependence on the quantities relevant to our setting (α, p_1, p_2, η). For any learning rate η, when t grows large, both margins eventually become positive (since they are proportional to $\gamma_2 \pm \gamma_1$ with γ_1 bounded), leading to perfect accuracy of our model.

In the dynamics analyzed so far, we observe a stationary regime where $W_t \simeq \log(t)W_\infty$. However, transitory regimes can hide under the big-O in Theorem 4 –we characterize the big-O precisely in the appendix. When considering discrete-time dynamics such as gradient descent (5), or stochastic dynamics, i.e., (6) or (7), those transitory regimes can showcase weight oscillations and loss spikes. For example, when $N = 2$ and there is strong association imbalanced and correlation, viz. $\alpha p(1) \gg p(2)$, the dynamics at the beginning of training can be approximated by

$$\Delta m_1 = \frac{\eta_t p(1)}{1 + \exp(m_1)}, \quad \Delta m_2 = \frac{-\eta_t \alpha p(1)}{1 + \exp(m_1)}.$$

Hence, in terms of the association stored in W, when the learning rate is large, the token $x = 1$ will "erase" the token $x = 2$. Since $p_W(f^*(x)|x = 2)$ approaching to zero implies that $\ell(W; x, f^*(x))$ goes to infinity, this can lead to arbitrarily big loss spikes, as captured by Proposition 5, proved in Appendix C.2. However, later in training, $p_W(2)$ catches up and W ultimately aligns in the max-margin direction, while $m_1 - m_2$ remains bounded.

Proposition 5 (Loss spikes)**.** *Let $N = 2$ with $f^*(x) = x$. Assume that Equation* (16) *holds, and $\alpha p_1 - p_2 > 0$. From a null initialization $W_0 = 0$, one gradient update* (5) *with learning rate η leads to*

$$\mathcal{L}(W_1) \geq \eta(\alpha p_1 - p_2)p_2, \tag{20}$$

which can be arbitrarily large.

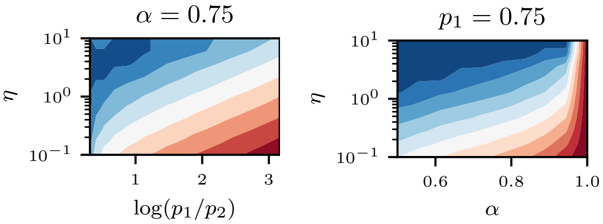

Figure 3: Level lines of the (logarithm of the) number of steps needed to reach perfect accuracy in the setting of Theorem 4, as a function of the learning rates η, the interaction parameter α, and the class imbalance $\log(p_1/p_2)$. Red means more steps to reach perfect accuracy.

To conclude, for overparameterized models, the dynamics is initially governed by memory interactions, before settling in a stationary regime similar to the orthogonal case described in Theorem 3. The oscillatory regime is due to the competition between two groups of tokens where increasing the margins of the high-frequency tokens causes a decrease in the margins of the others, similar to the opposing signals in Rosenfeld & Risteski (2023). The settling down of the dynamics can be understood intuitively. Since the max-margin will grow, all the partition function $A(We_x)$ of the $p_W(y|x)$ will grow, which will slow down the dynamics. Hence the oscillation will fade, and dynamics will enter the stationary logarithmic regime. In the stationary regime, bigger learning rates act as a speed-up of time, ensuring faster convergence. From a learning efficiency point of view, there is a trade-off between large learning rates implying longer oscillatory transitory regimes, and small learning rates implying slow speed of the dynamics. We illustrate this trade-off in Figure 3. We observe that class imbalance and interference make the problem harder, and that large learning rates are beneficial, although very large learning rates can be detrimental (top left of the left plot).

4.3. Graphical Understanding

Now that we have a good understanding of the mechanisms at play, we can verify these phenomena more generally through simulations. Let us first leverage the previous derivations to explain how to read measures of performance that can be obtained from experiments. When $d = M = 2$ and any value of $N \geq 2$, the problem reduces to a two-dimensional ones with

$$\gamma_i = (u_2 - u_1)^\top W f_i, \quad (f_i)_k = \mathbf{1}_{i=k}. \tag{21}$$

In the resulting two-dimensional space, we can plot the level lines of the loss function, the level lines of its Hessian eigenvalues, as well as the trajectories followed by W_t for different optimizers.

Figure 1 shows the level lines of the loss for $p(1) = 3/4$ and $\alpha \in \{-1/2, 0.95\}$. The deterministic gradient trajectories can be deduced from this picture: they are always orthogonal to level lines, and their speed is proportional

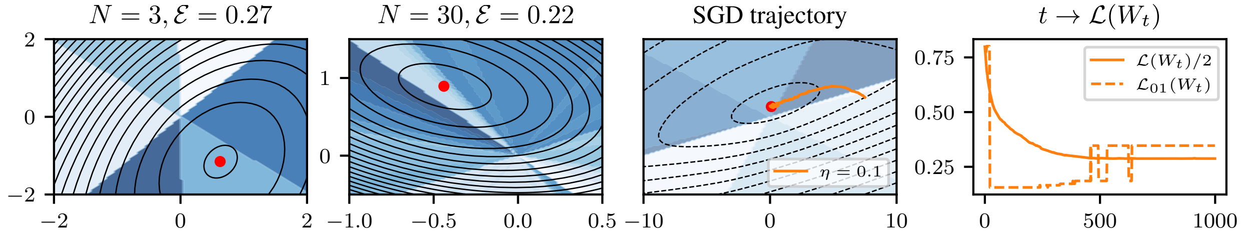

Figure 4: *Forgetting.* Similar plots as in Figures 1 and 2, yet in the limited capacity case $d < N$. In those situations, competition between the memories can lead to sub-optimal minimizer of $\mathcal{L}$, which we illustrate with SGD on the bottom plots. The sub-optimality is reflected in the excess of risk $\mathcal{E} = \mathcal{L}_{01}(\arg\min_W \mathcal{L}(W)) - \min_W \mathcal{L}_{01}(W)$.

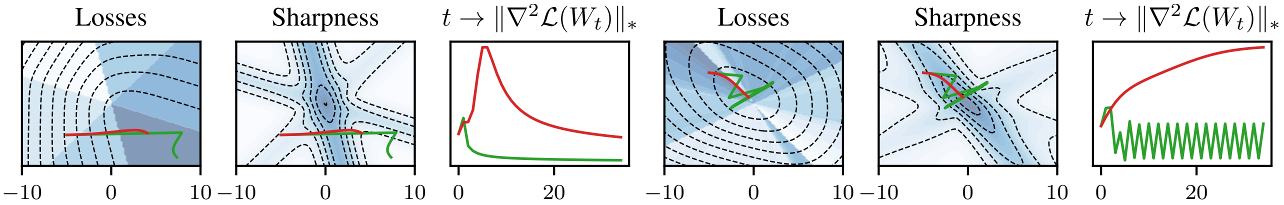

Figure 5: *Sharpness profile.* Gradient descent trajectories in the setting of Figures 2 and 4 with learning rates $\eta = 10$ (green) and $\eta = 1$ (red). We plot the level lines of the sharpness, i.e. the operator norm of $\nabla^2 \mathcal{L}(W)$, as well as the trace of the trajectories in terms of sharpness. The left plots are in the overparameterized regime, the right ones in the underparameterized one.

to the number of lines crossed locally. The fact that there is not much level line in the region $\{W | \mathcal{L}_{01}(W) = 0\}$ is due to the logarithmic convergence illustrated by Theorem 3. The right of Figure 1 shows that, although gradients are always positively correlated with the max-margin direction (formally $\mathrm{d}\gamma_2 \geq 0$ in (19)), they can point in directions that do not lead to perfect accuracy. Indeed, Figure 2 illustrates how large learning rates are likely to result in spikes of both the loss and the accuracy. This latter figure shows the trajectories of W_t for two different learning rates, and the trace of these trajectories in the training loss and accuracy plots which are usually monitored by practitioners training neural networks.

5. Numerical Analysis

This section complements previous derivations with numerical analysis. It discusses underparameterized regimes, large versions of model (1), as well as more complicated ones.

5.1. Limited capacity

Let us start the numerical analysis with the case where $N > d$. In those cases, one can not necessarily store all associations in memory, and the model has to favor some of them. It was shown in Cabannes et al. (2024) that the ideal W can usually store about d memories similarly to Hopfield network scalings. However, this ideal W is not always the one minimizing the cross-entropy loss.

We plot our problem in the case $M = d = 2$ thanks to the statistics γ_i of (21). Figure 4 reveals a striking fact: the cross-entropy loss is not calibrated for our model, i.e., minimizing $\mathcal{L}(W)$ does not always minimize $\mathcal{L}_{01}(W)$. Indeed, even in the case $N = 3$, one can find examples where competition between the memories leads the minimizer of $\mathcal{L}$ to “forget” the most frequent association. When N becomes large in front of d, these cases become the norm. On these landscapes, one can come up with examples of catastrophic forgetting, where the dynamics is first dominated by frequent tokens that are well memorized until rare classes come into play, perturbing the minimizer of $\mathcal{L}$, ultimately leading to convergence to a sub-optimal place. We illustrate it on the right of Figure 4.

To further illustrate the differences between the dynamics in over- and under-parameterized regimes, Figure 5 illustrates the sharpness, as defined by the operator norm of the Hessian of $\mathcal{L}(W)$ along two descent trajectories. We compute the Hessian in closed-form to show its level lines, illustrating that the sharpness of the logistic loss is mainly high for small values of the norm of W. We observe three types of behaviors. In the separable case, e.g. when $d \geq N$, the transitory regime goes through relatively sharp regions, before the stationary regime where the sharpness decreases until reaching zero at infinity. In the non-separable case, with is typical when $d < N$, either the learning rate is small enough and we converge to the minimum of $\mathcal{L}$ presenting a sharpness H_* greater than $2/\eta$, or the learning rate is greater than $2/H_*$ and we oscillate around the minimizer of $\mathcal{L}$.

5.2. Larger dimension

When the dimension d is larger, although we can not plot the weight-space, we can plot the evolution of certain statistics, such as the margin, along descent trajectories. In Fig-

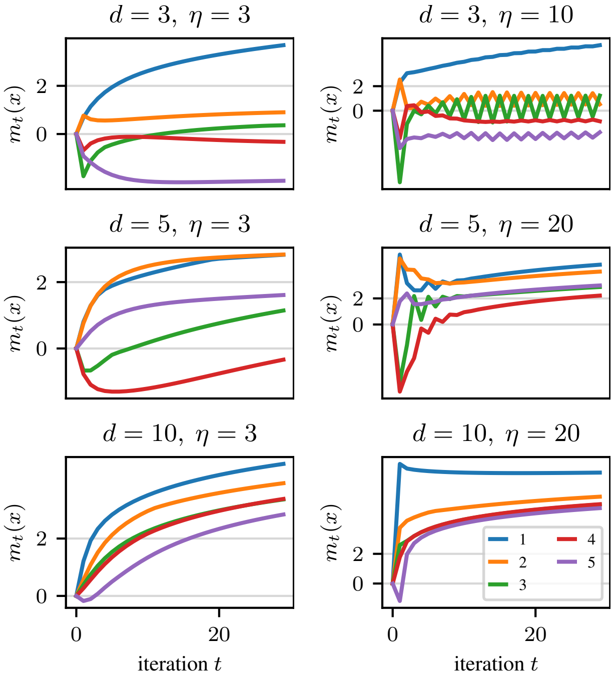

Figure 6: Margins $m_t(x)$ for $N = 5$ tokens, with varying dimensions d and learning rates η. The embeddings were sampled uniformly at random on the sphere. Large learning rates learn faster, although they lead to more oscillation, especially in low dimension. When $d < N$, the model does not have enough capacity to learn all the associations, and it favors the most frequent ones.

ure 6, we consider a setup with $N = M = 5$, $f^*(x) = x$, and $p(x) \propto 1/x$, in different dimensions (with random embeddings). We show the evolution of the margins

$$m_t(x) = \langle u_{f^*(x)}, We_x \rangle - \max_{j \neq x} \langle u_j, We_x \rangle. \qquad (22)$$

Perfect accuracy is achieved when $m_t(x) > 0$ for all x. We see the faster increase of margins for more frequent tokens, faster convergence with large step-size η, at the cost of oscillations, and benefits of larger d. The latter are likely due to less interference thanks to more orthogonality between random embeddings in higher dimension.

5.3. Simplified Transformer model

Finally, we empirically study training dynamics on a more complex model involving multiple associative memory mechanisms like the ones above. In particular, we consider a simplified two-layer Transformer architecture trained on an in-context learning task (described in Appendix D) that requires copying a bigram from the context depending on the current token. A two-layer attention-only transformer can solve this by implementing an "induction head" mechanism (Elhage et al., 2021; Olsson et al., 2022), and Bietti et al. (2023) show that this can be achieved by training only three matrices W_O^2, W_K^2, and W_K^1. These were found to behave as associative memories with appropriate embeddings,

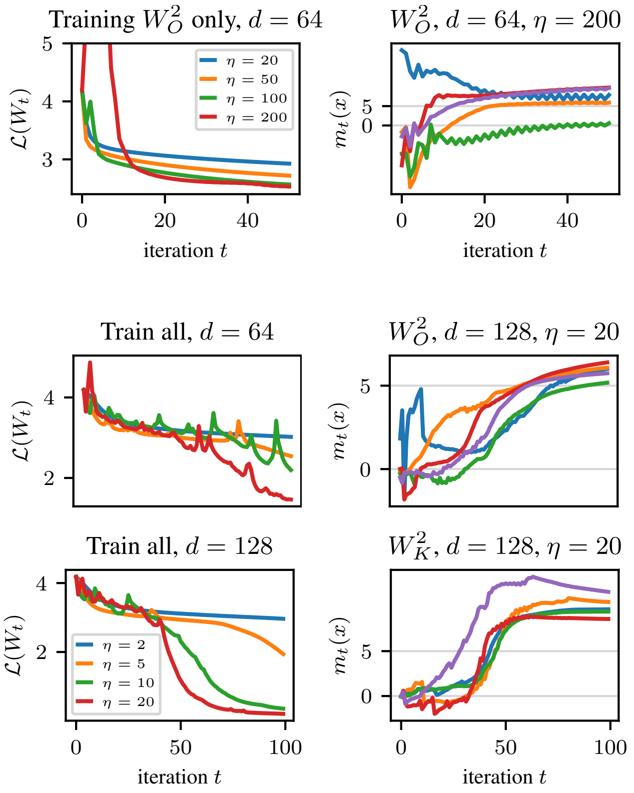

Figure 7: Full-batch training of selected transformer layers on the bigram task. (top) Loss and margins when training W_O^2 alone. (bottom) Loss and margins when training the three layers (W_O^2, W_K^2, and W_K^1) sufficient for the task, for two widths d. Training losses are shown for different step-sizes η, and margins are shown for 5 different tokens.

and we may thus empirically assess their margins.

We consider full-batch gradient descent on a dataset of 16 384 sequences of length 256 generated from the model described above with $N = 64$ tokens. The top of Figure 7 shows the training loss and margins when only training W_O^2, which can learn an appropriate associative memory by itself. The objective for this problem is convex and similar to the ones considered in this paper, up to some noise in the input embeddings due to attention. We see that margins tend to increase during training, and that large learning rates lead to faster optimization of the loss, at the cost of some spikes in the loss, and oscillations in the margins, which are due to correlations between embeddings.

The bottom of Figure 7 shows loss curves and margin evolution when training all three matrices. Here we see more frequent spikes in the loss for large learning rates, yet their gains are much more significant later in training, with small final losses that suggest the induction head mechanism is learned. The increasing margins confirm that the desired associative behaviors have indeed been recovered. Compared to the top of Figure 7, the margins for W_O^2 display more significant oscillations initially, likely due to additional interactions across different parameter matrices. In later iterations, when the attention heads are in place and inputs to W_O^2 are less noisy, the margins increase together to large values, leading to a similar learning speed on all memories. This uniform convergence behavior was facilitated by the relatively low output tokens imbalance considered in our tasks where the "copied" tokens were sampled uniformly. Finally, we see the effect of larger embedding dimensions d, accelerating the convergence thanks to more orthogonality.

6. Discussion

In this paper, we studied the gradient dynamics of associative memory models trained with cross-entropy loss, by viewing memory associations as interacting particles. This leads to new insights on the role of the data distribution and correlated embeddings on convergence speed as well as training "instabilities" in large learning rate regimes, such as oscillations and loss spikes. We also showed that some of these insights may transfer to some more realistic scenarios such as training small Transformers. Nonetheless, our simple model is only a first step, and there are many additional factors at play in larger models, which may lead to different behaviors and instabilities. This includes factorized parameterizations, normalization layers, adaptive optimizers, noisy data, and interactions between different layers which may change at different timescales. Studying the impact of these on training dynamics could unlock new improvements to the practice and reliability of training large models.

Impact Statement

This theoretical work aims to advance our understanding of training dynamics. Its short-time impact is limited. In the long run, our stream of research could help improve the training of large language models, from an energy or alignment standpoint. It indirectly relates to the quest for "general artificial intelligence," which is not without consequences, although discussing them is beyond the scope of this paragraph.

Acknowledgements

The authors would like to thank Léon Bottou and Hervé Jegou for fruitful discussions that led to this line of work.

References

Agarwala, A., Pedregosa, F., and Pennington, J. Second-order regression models exhibit progressive sharpening to the edge of stability. In *International Conference on Machine Learning (ICML)*, 2023.

Arbel, M., Korba, A., Salim, A., and Gretton, A. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bartlett, P. L., Long, P. M., and Bousquet, O. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research (JMLR)*, 24(316):1–36, 2023.

Beugnot, G., Rudi, A., and Mairal, J. On the benefits of large learning rates for kernel methods. In *Conference on Learning Theory (COLT)*, 2022.

Bietti, A., Cabannes, V., Bouchacourt, D., Jegou, H., and Bottou, L. Birth of a transformer: A memory viewpoint. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Byrd, J. and Lipton, Z. What is the effect of importance weighting in deep learning? In *International conference on machine learning (ICML)*, 2019.

Cabannes, V., Dohmatob, E., and Bietti, A. Scaling laws for associative memories. In *International Conference on Learning Representations*, 2024.

Chen, L. and Bruna, J. Beyond the edge of stability via two-step gradient updates. In *International Conference on Machine Learning (ICML)*, 2023.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Chizat, L., Colombo, M., Fernández-Real, X., and Figalli, A. Infinite-width limit of deep linear neural networks. *arXiv preprint arXiv:2211.16980*, 2022.

Cohen, J., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2020.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 1995.

Domingo-Enrich, C., Bietti, A., Gabrié, M., Bruna, J., and Vanden-Eijnden, E. Dual training of energy-based models with overparametrized shallow neural networks. *arXiv preprint arXiv:2107.05134*, 2021.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.

Hoorfar, A. and Hassani, M. Inequalities on the lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008.

Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 1982.

Hopfield, J. and Tank, D. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 1985.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Ji, Z. and Telgarsky, M. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory (COLT)*, 2019.

Ji, Z. and Telgarsky, M. Characterizing the implicit bias via a primal-dual analysis. In *Algorithmic Learning Theory*, 2021.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization, 2015.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems (NIPS)*, 2016.

Longuet-Higgins, C., Willshaw, D., and Buneman, P. Theories of associative recall. *Quarterly Reviews of Biophysics*, 1970.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems (NIPS)*, 2013.

Nakkiran, P. Learning rate annealing can provably help generalization, even for convex problems. *arXiv preprint arXiv:2005.07360*, 2020.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021.

Rosenfeld, E. and Risteski, A. Outliers with opposing signals have an outsized effect on neural network optimization. *arXiv preprint arXiv:2311.04163*, 2023.

Rotskoff, G. M. and Vanden-Eijnden, E. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*, 2018.

Schlag, I., Irie, K., and Schmidhuber, J. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research (JMLR)*, 19(1):2822–2878, 2018.

Willshaw, D., Buneman, P., and Longuet-Higgins, C. Non-holographic associative memory. *Nature*, 1969.

Wu, J., Braverman, V., and Lee, J. D. Implicit bias of gradient descent for logistic regression at the edge of stability. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Wu, J., Bartlett, P., Telgarsky, M., and Yu, B. Large stepsize gradient descent for logistic loss: Non-monotonicity of the loss improves optimization efficiency, 2024.

A. Gradient derivations

In the following, to be consistent with pytorch convention, we redefine the model as

$$f_W(x, y) = \langle e_x, W u_y \rangle$$

Recall that the loss can be understood intuitively as the negative log-likelihood

$$\mathcal{L}(W) = -\mathbb{E}_{X,Y}[\log p_W(Y \,|\, X)], \qquad \text{where} \qquad p_W(y \,|\, x) = \frac{\exp(\langle e_x, W u_y \rangle)}{\sum_z \exp(\langle e_x, W u_z \rangle)},$$

of the probability p_W whose conditional distributions are parameterized as a soft-max over the score $\langle e_x, W u_y \rangle$. In information theory, this negative log-likelihood is known as the cross-entropy of the model probability p_W relative to the real data distribution p.

To analyze the training dynamics, we will monitor quantities related to the gradient and the Hessian of the loss function. The gradient of the loss is easy to compute with simple derivation rules.

$$\begin{aligned} \nabla \ell(W; x, y) &= \nabla \log \left(\sum_z \exp\langle e_x, W u_z \rangle \right) - \nabla \langle e_x, W u_y \rangle \\ &= \sum_z \frac{\exp\langle e_x, W u_z \rangle}{\sum_{z'} \exp\langle e_x, W u_{z'} \rangle} e_x u_z^\top - e_x u_y^\top. \end{aligned}$$

This gradient can be understood with the probabilistic perspective on the loss as

$$\nabla \ell(W; x, y) = \sum_z p_W(z \,|\, x) e_x u_z^\top - e_x u_y^\top. \tag{23}$$

HESSIAN COMPUTATION

Notice that when the gradient can be written as

$$\nabla \ell(\theta) = g(\theta) \cdot a = (\partial_i \ell(\theta))_i, \qquad \text{with} \qquad g(\theta) \in \mathbb{R}, \quad a \in \mathbb{R}^d,$$

the Hessian follows as

$$\nabla^2 \ell(\theta) = (\partial_{ij} \ell(\theta)) = (\partial_i \partial_j \ell(\theta)) = (\partial_i g(\theta)\, a_j) = (\nabla g(\theta))\, a^\top.$$

In our case, we want to use the Euclidean structure on the matrix space, which leads to

$$\nabla^2 \ell(W; x, y) = \sum_z \nabla p_W(z \,|\, x)(e_x \otimes u_z)^\top. \tag{24}$$

To compute $\nabla p_W(z \,|\, x)$, notice that we could equally have expressed the loss gradient as

$$\nabla \ell(W; x, y) = -\nabla \log p_W(y \,|\, x) = -\frac{\nabla p_W(y \,|\, x)}{p_W(y \,|\, x)},$$

from which we deduce that

$$\begin{aligned} \nabla p_W(z \,|\, x) &= -p_W(z \,|\, x) \nabla \ell(W; x, z) \\ &= p_W(z \,|\, x) \left(e_x \otimes u_z - \sum_{z'} p_W(z' \,|\, x) e_x \otimes u_{z'} \right). \end{aligned} \tag{25}$$

Plugging this into Equation (24), we will have to deal with quantities such as[3]

$$(e_x \otimes u_y)(e_x \otimes u_z)^\top = e_x \otimes u_y \otimes e_x \otimes u_z.$$

[3] Recall that a tensor $M = a_1 \otimes a_2 \ldots \otimes a_p \in (\mathbb{R}^d)^{\otimes p}$ can be understood as a p-dimensional matrix $\mathtt{M}$. If (e_i) denotes the basis of $\mathbb{R}^d$, then $e_{j_1} \otimes e_{j_2} \ldots \otimes e_{j_p}$ is a basis of the tensor space, and M assimilates to $\mathtt{M}$ such that $\mathtt{M}[\mathtt{j}_1, \mathtt{j}_2, \ldots, \mathtt{j}_\mathtt{p}] = \prod_{i \in [p]} \langle a_i, e_{j_i} \rangle$.

The last operation can be understood from the fact that, when f_i is the canonical basis of $\mathbb{R}^d$, and a and b are in $\mathbb{R}^d$, we have the matrix identification

$$(ab^\top)_{i,j} = \langle a, f_i \rangle \langle b, f_j \rangle$$

In our case, using ij as the matrix indexation for $\mathbb{R}^{d\times d}$ and $f_i f_j^\top$ as the canonical basis of $\mathbb{R}^{d\times d}$,

$$\begin{aligned} \big((e_x \otimes u_y)(e_x \otimes u_z)^\top\big)_{ij,kl} &= \langle e_x \otimes u_y, f_i \otimes f_j \rangle \langle e_x \otimes u_z, f_k \otimes f_l \rangle \\ &= (f_i^\top e_x)(f_j^\top u_y)(f_k^\top e_x)(f_i^\top u_z) \\ &= \langle f_i \otimes f_j \otimes f_k \otimes f_l \otimes, e_x \otimes u_y \otimes e_x \otimes u_z \rangle. \end{aligned}$$

Using Equations (24) and (25), we deduce

$$\nabla^2 \ell(W; x, y) = \sum_{z,z'} p_W(z \,|\, x)(\delta_{z,z'} - p_W(z' \,|\, x)) e_x \otimes u_z \otimes e_x \otimes u_{z'}. \tag{26}$$

We implemented this formula vectorially in our code, and checked its correctness based on automatic differentiation libraries.

B. Dynamics without interference

To study the dynamic, using Equation (9), with the notation of the main text, i.e. $f_W(x, y) = \langle u_y, We_x \rangle$, we have

$$\nabla \ell(W; x, y) e_i = \sum_z p_W(z \,|\, x)(u_z - u_y) e_x^\top e_i.$$

In particular, when the (e_i) are orthogonal, summing over x leads to

$$\nabla \mathcal{L}(W) e_x = p(x) \left\| e_x \right\|^2 \sum_z p_W(z \,|\, x)(u_z - u_{f^*(x)}). \tag{27}$$

B.1. Binary classification - Proof of Theorem 2

Let us consider the binary case where $y \in \{1, 2\}$. Assume that $f^*(x) = 1$, Equation (27) simplifies as

$$\nabla \mathcal{L}(W) e_x = p(x) \left\| e_x \right\|^2 p_W(2 \,|\, x)(u_2 - u_1).$$

We can project it on the line where it evolves, reducing this equation to a scalar evolution

$$\begin{aligned} (u_1 - u_2)^\top \nabla \mathcal{L}(W) e_x &= -p(x) \left\| e_x \right\|^2 p_W(2 \,|\, x) \left\| u_2 - u_1 \right\|^2 \\ &= -p(x) \left\| e_x \right\|^2 \left\| u_2 - u_1 \right\|^2 \frac{\exp(u_2^\top W e_x)}{\exp(u_1^\top W e_x) + \exp(u_2^\top W e_x)} \\ &= -p(x) \left\| e_x \right\|^2 \left\| u_2 - u_1 \right\|^2 \frac{1}{\exp((u_1 - u_2)^\top W e_x) + 1}. \end{aligned}$$

Let us consider the evolution equation, for some learning rate scheduling $(\eta_t)_{t\geq 1}$

$$W_{t+1} = W_t - \eta_t \nabla \mathcal{L}(W_t).$$

This leads to

$$(u_1 - u_2)^\top (W_{t+1} - W_t) e_x = \eta_t p(x) \left\| e_x \right\|^2 \left\| u_2 - u_1 \right\|^2 \frac{1}{\exp((u_1 - u_2)^\top W_t e_x) + 1}.$$

Let us set

$$m_t = (u_1 - u_2)^\top W_t e_x, \qquad c = p(x) \left\| e_x \right\|^2 \left\| u_2 - u_1 \right\|^2.$$

The evolution equation becomes

$$(\exp(m_t) + 1)(m_{t+1} - m_t) = \eta_t c. \tag{28}$$

B.1.1. Gradient flow.

Since the updates are in the span of $(u_1 - u_2) \otimes e_x$, we have

$$W_t = \sum_x \alpha_t(x)(u_1 - u_2) \otimes e_x + \Pi_\perp W_0,$$

where $\Pi_\perp$ is the projection on the orthogonal of the gradient updates, and $\alpha_t(x)$ can be inferred from the margin

$$m_t(x) = (u_1 - u_2)^T W_t e_x = \alpha_t(x) \|u_1 - u_2\|^2 \|e_x\|^2,$$

In the following, we will solve the ODE for each margin using the product logarithm. The gradient flow limit of the previous derivation in Eq. 28 leads to

$$(\exp(m) + 1)\,\mathrm{d}m = c\,\mathrm{d}t.$$

Integrating this differential equation gives

$$\exp(m_t) + m_t = ct + \exp(m_0) + m_0.$$

For $x, y \in \mathbb{R}$, we can solve

$$\exp(x) + x = y \quad \Leftrightarrow \quad (y - x)\exp(y - x) = \exp(y) \quad \Leftrightarrow \quad x = y - \mathcal{W}_0(e^y),$$

where $\mathcal{W}_0$ is the product logarithm. This allows us to solve the previous equation in closed-form

$$m_t = ct + e^{m_0} + m_0 - \mathcal{W}_0(e^{ct} e^{\exp(m_0) + m_0}). \tag{29}$$

We can simplify this profile using the following asymptotic development of the product logarithm (Hoorfar & Hassani, 2008) when $x \geq e$,

$$\mathcal{W}_0(x) = \log(x) - \log\log(x) + \frac{\tilde{h}(\log(x)) \log\log(x)}{\log(x)}, \qquad \text{with} \qquad \tilde{h}(x) \in [1/2, 2].$$

We deduce that as soon as $ct + \exp(m_0) + m_0 \geq 0$,

$$m_t = \log(ct + \exp(m_0) + m_0) - \frac{\tilde{h}(ct + \exp(m_0) + m_0) \log(ct + \exp(m_0) + m_0)}{ct + \exp(m_0) + m_0},$$

since

$$\mathcal{W}_0(e^{ct} e^{\exp(m_0) + m_0}) = ct + \exp(m_0) + m_0 - \log(ct + \exp(m_0) + m_0) + \frac{\tilde{h}(\cdots) \log(ct + \exp(m_0) + m_0)}{ct + \exp(m_0) + m_0}.$$

The first part of theorem is found with

$$t_0 = -\frac{\exp(m_0) + m_0}{c},$$

and with the substitution of $\tilde{h}$ by $h(x) = \tilde{h}(x) \log(x)/x$.

B.1.2. Gradient descent

In the case of gradient descent, we will work with the discrete update equation

$$m_{t+1} - m_t = \frac{c\eta_t}{\exp(m_t) + 1}.$$

Since we expect a logarithmic growth of the margins, we exponentiate this equation and rearrange terms

$$\exp(m_{t+1}) - \exp(m_t) = \exp(m_t)\left(\exp\left(\frac{c\eta_t}{\exp(m_t) + 1}\right) - 1\right).$$

Notably, when we initialize the weights at zero, all margin updates are positive in this case with no interferences. This implies that $m_t \geq 0$ at all times.

$$\exp(m_{t+1}) - \exp(m_t) \geq c\eta_t \frac{\exp(m_t)}{\exp(m_t)+1} \geq c\eta_t \frac{1}{2}$$

where we used $e^x \geq x+1$ and $e^x/(e^x+1) \geq 1/2$ for $x \geq 0$. Telescoping this summation, we get

$$\exp(m_t) \geq c \sum_{t=0}^{t-1} \eta_t + 1$$

which yields the following logarithmic growth for the fixed learning rate schedule, i.e. $\eta_t = \eta$

$$m_t \geq \log(\eta c t + 1).$$

When te weights are not initialized at zero, there will be a moment t_0 where the margin will become positive and the same picture will hold.

We can then control the growth of the loss given by

$$\mathcal{L}(W_t) = \sum_x \log(1 + \exp(-(u_i - u_j)^T W_t e_x)) p(x)$$

which can be expressed in terms of the margins

$$\mathcal{L}(W_t) = \sum_x \log(1 + \exp(-m_t(x))) p(x).$$

Since $\log(1+e^{-x})$ is decreasing, we can directly install lower bound on $m_t(x)$ and get an upper bound on the loss

$$\mathcal{L}(W_t) \leq \sum_x \log(1 + \exp(-\log(\eta c_x(t - t_0(x))))) p(x) = \sum_x \log(1 + \frac{1}{\eta c_x(t-t_0(x))}) p(x) \leq \sum_x \frac{p(x)}{\eta c_x(t - t_0(x))}.$$

When $W_0 = 0$, $t_0(x) \leq 0$, which implies the second part of the theorem.

B.2. Multi-class - Proof of Theorem 3

For the multi-class, consider $x \in [N]$ and assume that $f^*(x) = 1$. Because the dynamics decouple, we can simplify notation with $w = We_x \in \mathbb{R}^d$, and forget about the context variable x. Let us denote

$$p_w(j) \propto \exp(w^\top u_j), \qquad \ell(w) = -\log(p_w(1)).$$

Consider the gradient flow dynamics

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -\nabla \ell(w) = -\sum_{j \in [M]} p_w(j)(u_j - u_1).$$

Developing the probabilities, we get

$$\sum_{j\in[M]} \exp(w^\top u_j) \frac{\mathrm{d}w}{\mathrm{d}t} = \sum_{j \in [M]} \exp(w^\top u_j)(u_1 - u_j).$$

Let us denote $w_j = \langle w, u_j \rangle$.

$$\sum_{j\in[M]} \exp(w_j) \frac{\mathrm{d}w}{\mathrm{d}t} = \sum_{j \in [M]} \exp(w_j)(u_1 - u_j).$$

When the (u_j) are orthonormal, we can project the last equation on the (u_i), which leads to the following coupled differential equations

$$\sum_{j=1}^{M} \exp(w_j)\,\mathrm{d}w_1 = \sum_{j=2}^{M} \exp(w_j)\,\mathrm{d}t, \qquad \text{and} \qquad \sum_{j=1}^{M} \exp(w_j)\,\mathrm{d}w_i = -\exp(w_i)\,\mathrm{d}t \qquad \forall\, i \neq 1.$$

We can rewrite it with the partition function $A(w) = \sum \exp(w_j)$,

$$\frac{A(w)}{A(w) - \exp(w_1)} \,\mathrm{d}w_1 = \mathrm{d}t, \qquad \text{and} \qquad A(w)\exp(-w_i)\,\mathrm{d}w_i = -\,\mathrm{d}t \qquad \forall\, i \neq 1.$$

Subtracting any two instances of the coupling equations for $i, j \neq 1$, we get the following invariant

$$\begin{aligned} A(w)\left(\exp(-w_i)\,\mathrm{d}w_i - \exp(-w_j)\,\mathrm{d}w_j\right) = 0 \qquad &\Leftrightarrow \qquad \exp(-w_i)\,\mathrm{d}w_i - \exp(-w_j)\,\mathrm{d}w_j = 0 \\ \Leftrightarrow \qquad \exp(-w_i) - \exp(-w_j) &= \exp(-u_i^\top w_0) - \exp(-u_j^\top w_0) =: c_{ij} \end{aligned}$$

The last invariant is found with

$$A(w)\,\mathrm{d}w_1 = (A(w) - \exp(w_1))\,\mathrm{d}t = \sum_{i>1} \exp(w_i)\,\mathrm{d}t = -\sum_{i>1} A(w)\,\mathrm{d}w_i \qquad \Leftrightarrow \qquad \sum_{i\in[M]} \mathrm{d}w_i = 0.$$

This is the transcription of the fact that the update of w are in the span of the $(u_i - u_j)$ which is the orthogonal $(\sum u_i)^\perp$ when the (u_i) are orthonormal.

The first invariant allows us to characterize the partition function using only w_1 and the logit of the most probable incorrect class. Let $k = \arg\min_{j\in\{2,..,M\}} \exp(-w_j)$, hence $c_{jk} \geq 0$ for $j \neq 1$, and

$$\begin{aligned} A(w) &= \exp(w_1) + \sum_{j\neq 1} \exp(w_j) = \exp(w_1) + \sum_{j\neq 1} \frac{1}{c_{jk} + \exp(-w_k)} \\ &= \exp(w_1) + \sum_{j\neq 1} \frac{\exp(w_k)}{c_{jk}\exp(w_k) + 1} \\ &= \exp(w_1) + (M_k + \theta_k)\exp(w_k) \end{aligned}$$

where

$$M_k = |\{k \neq 1 | c_{jk} = 0\}| \geq 1, \qquad \theta_k \in [0, |\{k \neq 1 | c_{jk} > 0\}|] = [0, M - M_k],$$

where we have used that when $c_{jk} > 0$, $\exp(w_k)/(c_k \exp(w_k) + 1) \leq \exp(w_k)$.

Note that for any $j \neq 1$, we can write the differential equations for the margin as

$$A(w)\,\mathrm{d}(w_1 - w_j) = (A(w) - \exp(w_1) + \exp(w_j))\,\mathrm{d}t.$$

In particular, for the tightest margin, we get

$$\frac{A(w)}{A(w) - \exp(w_1) + \exp(w_k)} = \frac{A(w)\exp(-w_k)}{A(w)\exp(-w_k) - \exp(w_1 - w_k) + 1} = \frac{\exp(w_1 - w_k) + M_k + \theta_k}{M_k + \theta_k + 1}.$$

Using the bounds on θ_k, we get

$$\left(\frac{\exp(w_1 - w_k)}{M+1} + \frac{M_k}{M_k+1}\right)\mathrm{d}(w_1 - w_k) \leq \mathrm{d}t \leq \left(\frac{\exp(w_1 - w_k)}{M_k+1} + \frac{M}{M+1}\right)\mathrm{d}(w_1 - w_k)$$

Let us introduce constants to ease notations,

$$(c_1 \exp(w_1 - w_k) + c_2)\,\mathrm{d}(w_1 - w_k) \leq \mathrm{d}t \leq (c_3 \exp(w_1 - w_k) + c_4)\,\mathrm{d}(w_1 - w_k).$$

We can integrate these inequalities

$$c_1 \exp(w_1 - w_k) + c_2(w_1 - w_k) + b_1 \leq t \leq c_3 \exp(w_1 - w_k) + c_4(w_1 - w_k) + b_2.$$

This implies

$$w_1 - w_k = \log(t)(1 + o(1)).$$

Let us denote $h(t) = \exp(w_1)$, using the first invariant, we get

$$\exp(w_1 - w_k) = \exp(w_1)(\exp(-w_j) + c_{kj}) = \exp(w_1 - w_j) + c_{kj}h(t) = c_t t(1 + o(t)) + c_{kj}h(t),$$

for $c_t \in [1/c_3, 1/c_1]$ a bounded function. We can characterize $h(t)$ with the last invariant

$$\sum_{i\in[M]} w_i =: C = o(1).$$

The previous equations are solved with

$$w_i = \frac{-1}{M}\log(t)(1 + o(1)), \qquad w_1 = \frac{M-1}{M}\log(t)(1 + o(1)),$$

which leads to $c_{kj}h(t) = c_{kj}t^{1-1/M} = o(t)$, and, since the (u_i) are orthonormal,

$$w = \sum_{i\in[M]} \langle w, u_i\rangle\, u_i = \sum_{i\in[M]} w_i u_i = \sum_{i\in[M]} \frac{\log(t)}{M}(u_1 - u_i).$$

We can simplify the last equation by realizing that it is proportional to the projection of u_1 on the span of the $u_i - u_j$, which is also the span of the $u_1 - u_i$. If we denote Π the projection on this span, we have the existence of (b_i) such that

$$\Pi(u_1) = \sum_{i>2} b_i(u_1 - u_i),$$

and since $(\Pi(u_1) - u_1)^\top(u_i - u_j) = 0$ for all $i, j > 1$, we deduce $b_i = b_j = b$.

The value of b can be computed explicitly, the triangle formed by 0, u_1 and $\Pi(u_1)$ is both isosceles and rectangular, which leads to $\|\Pi(u_1) - u_1)\| = \|\Pi(u_1)\|$, $1 = \|u_i\|^2 = \|\Pi(u_1) - u_1\|^2 + \|\Pi(u_1)\|^2$, hence $\|\Pi(u_1)\| = 1/\sqrt{2}$. We also have $\|\Pi(u_1)\| = \Pi(u_1)^\top u_1 = (M-1)b$, from which we deduce that the proportionality constant in Theorem 3 is $(M-1)/\sqrt{2}M$.

Indications for a proof in the case of gradient descent

For gradient descent, we expect the same theorem to hold for two simple reasons, which, for simplicity, we do not formalize.

- By convexity, there could only be one directional convergence for gradient descent (regardless of the initialization), and it has to be the same as the one for gradient flow.
- Because the level lines of loss are exponentially spaced, for any fixed learning rates, gradient descent will become a finer and finer approximation of gradient flow as $\|W\|$ grows large.

Another way to proceed is to retake the previous arguments in the discrete setting. For example, when initializing gradient descent with $W = 0$, one can check by recurrence that $\exp(w_i) = \exp(w_j)$ for all $i, j \neq 1$, this allows reducing the dynamics to a scalar evolution, which can be treated as in Theorem 2.

C. Dynamics with two particles interfering

In the setting of Theorem 4, Theorem 1 plus a few lines of omitted derivations lead to the couplings

$$\langle \nabla\mathcal{L}(W), (u_j - u_i)\otimes e_j\rangle = \|u_2 - u_1\|^2\left(\frac{p(i)\alpha}{1 + \exp((u_i - u_j)^\top We_i)} - \frac{p(j)}{1 + \exp((u_j - u_i)^\top We_j)}\right). \tag{30}$$

We remark that if $\alpha \leq 0$, there is no competition between the memory associations, the dynamics always advances in the cone $\mathbb{R}_+ \cdot e_1 + \mathbb{R}_+ \cdot e_2$, reinforcing both associations simultaneously.

Let us introduce the margin

$$m_j = (u_j - u_i)^\top We_j = \langle W, (u_j - u_i)\otimes e_j\rangle, \qquad \text{for} \qquad \{i, j\} = \{1, 2\}.$$

The previous equation can be rewritten as

$$\langle \nabla \mathcal{L}(W), (u_j - u_i) \otimes e_j \rangle = \|u_2 - u_1\|^2 \left(\frac{p(i)\alpha}{1 + \exp(m_i)} - \frac{p(j)}{1 + \exp(m_j)} \right),$$

For the gradient flow, it leads to the evolution

$$\mathrm{d}m_j = -\langle \nabla \mathcal{L}(W), (u_j - u_i) \otimes e_j \rangle = \|u_2 - u_1\|^2 \left(\frac{p(j)}{1 + \exp(m_j)} - \frac{p(i)\alpha}{1 + \exp(m_i)} \right) \mathrm{d}t. \tag{31}$$

Similarly, if we define the orthogonal vectors

$$f_1 = e_1 + e_2, \qquad f_2 = e_1 - e_2,$$

as well as the statistics,

$$\gamma_i = \frac{1}{2}(u_1 - u_2)^\top W f_i,$$

we get $m_1 = \gamma_1 + \gamma_2$ and $m_2 = \gamma_2 - \gamma_1$, and $\gamma_1 = (m_1 - m_2)/2$, $\gamma_2 = (m_1 + m_2)/2$. Hence,

$$\begin{aligned}
\frac{2}{\|u_2 - u_1\|^2} \frac{\mathrm{d}\gamma_2}{\mathrm{d}t} &= \frac{1}{\|u_2 - u_1\|^2} \frac{\mathrm{d}m_1 + \mathrm{d}m_2}{\mathrm{d}t} \\
&= \left(\frac{p(1)}{1 + \exp(m_1)} - \frac{p(2)\alpha}{1 + \exp(m_2)} \right) + \left(\frac{p(2)}{1 + \exp(m_2)} - \frac{p(1)\alpha}{1 + \exp(m_1)} \right) \\
&= \frac{p(1)(1 - \alpha)}{1 + \exp(m_1)} + \frac{p(2)(1 - \alpha)}{1 + \exp(m_2)} \\
&= (1 - \alpha) \left(\frac{p(1)}{1 + \exp(\gamma_1 + \gamma_2)} + \frac{p(2)}{1 + \exp(\gamma_2 - \gamma_1)} \right)
\end{aligned}$$

Similarly

$$\begin{aligned}
\frac{2}{\|u_2 - u_1\|^2} \frac{\mathrm{d}\gamma_1}{\mathrm{d}t} &= \frac{1}{\|u_2 - u_1\|^2} \frac{\mathrm{d}m_1 - \mathrm{d}m_2}{\mathrm{d}t} \\
&= \left(\frac{p(1)}{1 + \exp(m_1)} - \frac{p(2)\alpha}{1 + \exp(m_2)} \right) - \left(\frac{p(2)}{1 + \exp(m_2)} - \frac{p(1)\alpha}{1 + \exp(m_1)} \right) \\
&= \frac{p(1)(1 + \alpha)}{1 + \exp(m_1)} - \frac{p(2)(1 + \alpha)}{1 + \exp(m_2)} \\
&= (1 + \alpha) \left(\frac{p(1)}{1 + \exp(\gamma_1 + \gamma_2)} - \frac{p(2)}{1 + \exp(\gamma_2 - \gamma_1)} \right)
\end{aligned}$$

This explains the evolution in the main text. We see that γ_2 will grow at least logarithmically, while γ_1 will be contained eventually because of the growth of γ_2.

C.1. Proof of Theorem 4

We start by focusing on the gradient flow dynamics. Recall that the evolution of the max-margin and orthogonal directions γ_2 and γ_1 is given by the following ODEs:

$$\frac{\mathrm{d}\gamma_1}{\mathrm{d}ct} = \frac{(1 + \alpha)p_1}{1 + \exp(\gamma_2 + \gamma_1)} - \frac{(1 + \alpha)p_2}{1 + \exp(\gamma_2 - \gamma_1)} \tag{32}$$

$$\frac{\mathrm{d}\gamma_2}{\mathrm{d}ct} = \frac{(1 - \alpha)p_1}{1 + \exp(\gamma_2 + \gamma_1)} + \frac{(1 - \alpha)p_2}{1 + \exp(\gamma_2 - \gamma_1)}. \tag{33}$$

LOWER BOUND IN THE MAX-MARGIN DIRECTION γ_2

From the evolution equation (33) of the margin direction γ_2 we deduce, using the fact that either $e^{\gamma_1} \leq 1$ or $e^{-\gamma_1} \leq 1$ for all $\gamma_1 \in \mathbb{R}$,

$$\frac{\mathrm{d}\gamma_2}{\mathrm{d}ct} \geq \frac{(1 - \alpha)\min(p_1, p_2)}{1 + \exp(\gamma_2)} = \frac{(1 - \alpha)p_2}{1 + \exp(\gamma_2)},$$

since we have assumed without restriction that $p_1 \geq p_2$. We have solved the differential equation in Appendix B.1 (with a different constant). Using Grönwall's inequality, integrating this out leads to, when initialized at $W_0 = 0$,

$$\gamma_2 \geq \log(c_1 t + 1) - h(c_1 t + 1).$$

where $c_1 = (1-\alpha)cp_2$ and h as defined in Appendix B.1, i.e. $h(x) = \tilde{h}(x)\log(x)/x$ with $\tilde{h} \in [1/2, 2]$.

Upper bound in the orthogonal direction γ_1

Let us now consider γ_1. First, note that whenever $\gamma_1 \leq 0$, then we have $\mathrm{d}\gamma_1 \geq 0$, thanks to our assumption $p_1 \geq p_2$. In particular, with zero initialization $W(0) = 0$, we then have $\gamma_1(t) \geq \gamma_1(0) = 0$ throughout.

Let us know look for an upper bound. For γ_1 to grow, we need $\mathrm{d}\gamma_1 \geq 0$. Denoting $\bar{\gamma} := \log(\sqrt{p_1/p_2})$, this only possible when

$$\begin{aligned}\frac{p_1}{1+\exp(\gamma_2+\gamma_1)} - \frac{p_2}{1+\exp(\gamma_2-\gamma_1)} \geq 0 \quad &\Leftrightarrow \quad p_1 - p_2 + p_1\exp(\gamma_2-\gamma_1) \geq p_2\exp(\gamma_2+\gamma_1) \\ &\Leftrightarrow \quad (p_1-p_2)\exp(-\gamma_2) \geq p_2\exp(\gamma_1) - p_1\exp(-\gamma_1) \\ &\Leftrightarrow \quad (p_1-p_2)\exp(-\gamma_2) \geq \sqrt{p_1p_2}(\exp(\gamma_1-\bar{\gamma}) - \exp(-\gamma_1+\bar{\gamma})) \\ &\Leftrightarrow \quad \sinh(\gamma_1-\bar{\gamma}) \leq \frac{(p_1-p_2)\exp(-\gamma_2)}{2\sqrt{p_1p_2}}.\end{aligned}$$

We define

$$C(\gamma_2) := \frac{(p_1-p_2)\exp(-\gamma_2)}{2\sqrt{p_1p_2}}. \tag{34}$$

We thus have

$$\sinh(\gamma_1-\bar{\gamma}) \geq C(\gamma_2) \qquad \Leftrightarrow \qquad \mathrm{d}\gamma_1 \leq 0.$$

This implies that gradient flow will be bounded. In particular, the lower bound on γ_2 gives us

$$\exp(-\gamma_2) \leq \frac{\exp(h(c_1t+1))}{c_1t+1} \leq \frac{\exp(2/e)}{c_1t+1}$$

We conclude that, when γ_1 is initialized at zero, we have that $\mathrm{d}\gamma_1(0) \geq 0$, and γ_1 will grow until reaching the point where $\mathrm{d}\gamma_1 \leq 0$, which leads to a bound on γ_1 characterized by

$$\sinh(\gamma_1-\bar{\gamma}) \leq C(\gamma_2) \leq \frac{p_1-p_2}{2\sqrt{p_1p_2}}\frac{\exp(2/e)}{c_1t+1}.$$

This yields

$$\gamma_1(t) \leq \bar{\gamma} + \sinh^{-1}\left(\frac{p_1-p_2}{\sqrt{p_1p_2}}\frac{1.05}{c_1t+1}\right) \leq \frac{1}{2}\log\left(\frac{p_1}{p_2}\right) + \frac{p_1-p_2}{\sqrt{p_1p_2}}\frac{1.05}{c_1t+1},$$

using that $\sinh^{-1}(x) \leq x$ for $x \geq 0$.

Upper bound in the max-margin direction γ_2

We can upper bound γ_2 based on Equation (33),

$$\frac{\mathrm{d}\gamma_2}{\mathrm{d}ct} \leq \frac{2(1-\alpha)\max(p_1,p_2)}{1+\min(\exp(\gamma_1),\exp(-\gamma_1))\exp(\gamma_2)} = \frac{2(1-\alpha)p_1}{1+\exp(-\gamma_1)\exp(\gamma_2)}.$$

We have seen that

$$\begin{aligned}\exp(\gamma_1) &\leq \sqrt{\frac{p_1}{p_2}}\exp(\sinh^{-1}(\frac{p_1-p_2}{p_2}\frac{1.05}{c_1t+1})) = \sqrt{\frac{p_1}{p_2}}\left(\frac{p_1-p_2}{p_2}\frac{1.05}{c_1t+1} + \sqrt{\frac{(p_1-p_2)^2}{p_2^2}\frac{1.05}{(c_1t+1)^2}+1}\right) \\ &\leq \sqrt{\frac{p_1}{p_2}}\left(1.05\frac{p_1-p_2}{p_2} + \sqrt{1.05\frac{(p_1-p_2)^2}{p_2^2}+1}\right) \leq \frac{p_1}{p_2}\left((1.05+\sqrt{2.05})\frac{p_1-p_2}{p_2} + \sqrt{2.05}\right) \leq 4\left(\frac{p_1}{p_2}\right)^2.\end{aligned}$$

We deduce that

$$\frac{\mathrm{d}\gamma_2}{\mathrm{d}ct} \leq \frac{2(1-\alpha)p_1}{1+\frac{1}{4}\left(\frac{p_2}{p_1}\right)^2\exp(\gamma_2)} \leq \frac{8(1-\alpha)p_1^3/p_2^2}{1+\exp(\gamma_2)}.$$

This allows us to conclude that γ_2 does not grow faster than logarithmically.

$$\gamma_2 \leq \log(c_2 t+1) - h(c_2 t+1).$$

with $c_2 = 8c(1-\alpha)p_1^3/p_2^2$. Using the intermediate value theorem, we deduce the form of γ_2 given in the theorem.

Lower bound in the orthogonal direction γ_1

If γ_1 was initialized such that $\mathrm{d}\gamma_1 \leq 0$, we would have that γ_1 would decrease until reaching the point found in the upper bound for γ_1. The difficulty consists in showing that γ_1 increases fast enough toward $\bar{\gamma}$. Retaking the derivations made to characterize the sign of $\mathrm{d}\gamma_1$, we can rewrite the evolution equation as

$$\begin{aligned}\frac{\mathrm{d}\gamma_1}{\mathrm{d}ct} &= (1+\alpha)\frac{p_1-p_2+2\sqrt{p_1p_2}\exp(\gamma_2)\sinh(\bar{\gamma}-\gamma_1)}{(1+\exp(\gamma_2+\gamma_1))(1+\exp(\gamma_2-\gamma_1))} \\ &= (1+\alpha)\frac{p_1-p_2+2\sqrt{p_1p_2}\tilde{c}_t t\sinh(\bar{\gamma}-\gamma_1)}{(1+\tilde{c}_t t\exp(\gamma_1))(1+\tilde{c}_t t\exp(-\gamma_1))},\end{aligned}$$

where $\tilde{c}_t/c_t \in [\exp(e/2), 1]$ is found with the intermediate value theorem, and we have used that

$$\exp(\gamma_2) = c_t t\exp(-h(c_t t+1)) \in c_t t\cdot[\exp(e/2), 1].$$

We can lower bound the growth of γ_1 when $\gamma_1 \leq \bar{\gamma}$, which implies $\mathrm{d}\gamma_1 \geq 0$. Using that γ_1 is bounded, we get the existence of a constant c_3 such that

$$\frac{\mathrm{d}\gamma_1}{\mathrm{d}ct} \geq (1+\alpha)\frac{p_1-p_2+2\sqrt{p_1p_2}\tilde{c}_t t\sinh(\bar{\gamma}-\gamma_1)}{(1+c_3 t)^2} \geq (1+\alpha)\frac{p_1-p_2}{(1+c_3 t)^2}.$$

This leads to a growth of γ_1 in $O(1/t)$, from which we deduce that

$$\gamma_1 \geq \bar{\gamma} + O(1/t),$$

which ends the characterization of the dynamics for gradient flow.

Gradient Descent

The dynamics of γ_1 and γ_2 for gradient descent with a step-size η are given by

$$\gamma_1(t+1) = \gamma_1(t) + \eta c\Delta\gamma_1, \qquad \gamma_2(t+1) = \gamma_2(t) + \eta c\Delta\gamma_2,$$

with

$$\Delta\gamma_1 = \frac{(1+\alpha)p_1}{1+\exp(\gamma_2+\gamma_1)} - \frac{(1+\alpha)p_2}{1+\exp(\gamma_2-\gamma_1)} \tag{35}$$

$$\Delta\gamma_2 = \frac{(1-\alpha)p_1}{1+\exp(\gamma_2+\gamma_1)} + \frac{(1-\alpha)p_2}{1+\exp(\gamma_2-\gamma_1)}. \tag{36}$$

Similar to gradient flow, we can lower bound the update equation of γ_2 for descent, with $c_1 = (1-\alpha)cp_2$

$$\gamma_2(t+1) - \gamma_2(t) \geq \frac{\eta c_1}{1+\exp(\gamma_2)}.$$

Since we expect logarithmic growth from the study of flow, we want to study $\exp(\gamma_2(t))$. In particular

$$\exp(\gamma_2(t+1)) \geq \exp\left(\frac{\eta c_1}{1+\exp(\gamma_2)}\right)\exp(\gamma_2(t))$$

Using $e^x \geq 1 + x$, we furthermore get

$$\exp(\gamma_2(t+1)) - \exp(\gamma_2(t)) \geq \eta c_1 \cdot \frac{\exp(\gamma_2(t))}{1 + \exp\left(\gamma_2(t)\right)}.$$

Since γ_2 is always non-negative we have that $\exp(\gamma_2)/(1 + \exp(\gamma_2)) \geq 1/2$ hence we get a recursion

$$\exp(\gamma_2(t+1)) - \exp(\gamma_2(t)) \geq \eta c_1/2, \tag{37}$$

Using a telescopic sum, we get the desired lower bound $\gamma_2(t) \geq \log(\eta c_1 t/2 + 1)$.

Let us know focus on γ_1. In comparison to gradient flow, γ_1 can grow large because of potentially large steps taken from values where $\Delta\gamma_1$ is positive. Similar to the gradient flow case, we have that $\Delta\gamma_1 \leq 0$ if and only if

$$\sinh(\gamma_1 - \bar{\gamma}) \geq C(\gamma_2),$$

where $C(\gamma_2)$ is given in (34). Now consider a time t. If $\sinh(\gamma_1(t) - \bar{\gamma}) \leq C(\gamma_2(t))$, we have

$$\begin{aligned}\gamma_1(t) \leq \gamma_1(t+1) &\leq \gamma_1(t) + \eta c \Delta\gamma_1 \leq \bar{\gamma} + \sinh^{-1}(C(\gamma_2(t))) + \eta c(1-\alpha)p_1 \\ &\leq \bar{\gamma} + C(\gamma_2(0)) + \eta c(1-\alpha)p_1 \leq \bar{\gamma} + \frac{p_1}{2p_2} + \eta c(1-\alpha)p_1 =: \gamma_{\max}.\end{aligned}$$

If $\sinh(\gamma_1(t) - \bar{\gamma}) \geq C(\gamma_2(t))$, then $\gamma_1(t+1) \leq \gamma_1(t)$, and

$$\gamma_1(t+1) \geq \gamma_1(t) + \eta c \Delta\gamma_1 \geq \bar{\gamma} - \eta c(1-\alpha)p_2 =: \gamma_{\min},$$

where $\gamma_1(t) \geq \bar{\gamma}$ follows from $\sinh(\gamma_1(t) - \bar{\gamma}) \geq 0$.

By induction, we then have that $\gamma_1(t) \in [\min(0, \gamma_{\min}), \gamma_{\max}]$ for all t (assuming $\gamma_1(0) = 0$).

For simplicity, we skip the upper bound on γ_2, as well as the convergence of γ_1 towards $\bar{\gamma}$.

INDICATIONS TO PROVE THAT γ_1 CONVERGES TO $\bar{\gamma}$

Note that in the case of gradient descent with large learning rates, γ_1 might be oscillating around $\bar{\gamma}$. This case does not happen in gradient flow and requires extra derivations to handle it. Using the fact that $\gamma_2(t) \geq \log(\eta c t + 1)$, we get

$$C(\gamma_2(t)) \leq \frac{p_1 - p_2}{2\sqrt{p_1 p_2}} \frac{1}{\eta c_3 t + 1}.$$

If $\sinh(\gamma_1(t) - \bar{\gamma}) \leq C(\gamma_2(t))$, we then have

$$\begin{aligned}\gamma_1(t) \leq \gamma_1(t+1) &= \gamma_1(t) + \eta c \Delta\gamma_1 \\ &\leq \bar{\gamma} + \frac{p_1 - p_2}{2\sqrt{p_1 p_2}} \frac{1}{\eta c_3 t + 1} + \frac{\eta c(1-\alpha)p_1}{1 + \exp(\gamma_{\min})\eta c_3 t} = \bar{\gamma} + O(1/t)\end{aligned}$$

If $\sinh(\gamma_1(t) - \bar{\gamma}) \geq C(\gamma_2(t))$, then

$$\begin{aligned}\gamma_1(t) \geq \gamma_1(t+1) &= \gamma_1(t) + \eta c \Delta\gamma_1 \\ &\geq \bar{\gamma} - \frac{\eta c(1-\alpha)p_2}{1 + \exp(-\gamma_{\max})\eta c_3 t} = \bar{\gamma} + O(1/t).\end{aligned}$$

This ensures that when the dynamics are in an oscillating regime, the bound on $|\gamma_1(t) - \bar{\gamma}|$ will decrease as $O(1/t)$, thus inducing faster progress towards perfect accuracy than as guaranteed by the looser bound $[\gamma_{\min}, \gamma_{\max}]$.

C.2. Loss Spike

Proposition 5 follows from

$$\mathcal{L}(W_1) \geq p_2 \ell(W_1; 2, 2) = p_2 \log(1 + \exp(-m_2)) \geq -p_2 m_2.$$

In particular, when initialized at zero, after one gradient update

$$m_2 = \eta(p_2 - \alpha p_1).$$

D. Transformer experiments

In this section, we provide more details on the setup for the transformer experiments in Section 5.3.

We follow Bietti et al. (2023) and consider a simplified two-layer Transformer architecture trained on a simple in-context learning task. The task consists of sequences of tokens $z_{1:T} \in [N]^T$, where any occurrence of a so-called *trigger* token $q \in Q$ is followed by the same *output* token o_q, but where o_q is resampled uniformly across different sequences. The tokens following all non-trigger tokens are randomly sampled from a sequence-independent Markov model (namely, a character-level bigram model estimated from Shakespeare text data).

We focus on the prediction of the output tokens o_q given a sequence $[z_1, \dots, q, o_q, \dots, q]$, where we assume q has appeared at least once before the last token. Correctly predicting the token o_q then requires finding previous occurrences of q in the input sequence and copying the token just after it. Bietti et al. (2023) show that this task can be solved with a two-layer transformer with no feed-forward blocks, and all layers fixed at random except three trained matrices, by implementing an "induction head" mechanism (Elhage et al., 2021; Olsson et al., 2022). The three trained matrices were found to behave as associative memories, each with different sets of embeddings, as we now detail:

- W_K^1 (first layer key-query matrix), which implements a previous token lookup, satisfying
$$\arg\max_j \langle p_j, W_K^1 p_t \rangle = t - 1,$$
where p_t are positional embeddings;
- W_K^2 (second layer key-query matrix), which implements lookup of the previous trigger that matches the current token, with
$$\arg\max_j \langle e_j, W_K^2 \varphi_k(e_i) \rangle = i,$$
where e_i are input token embeddings, and $\varphi_k(e_i) = W_O^1 W_V^1 e_i$ is a remapping of the input embeddings by the first attention head;
- W_O^2 (second layer output matrix), which implements a copy of the output token into the unembedding space, with
$$\arg\max_j \langle u_j, W_O^2 \varphi_o(e_i) \rangle = i,$$
where u_j are output embeddings, and $\varphi_o(e_i) = W_V^2 e_i$ is a remapping of input embeddings by the (random) second value matrix.

We may then define the W_O^2 margins (the ones for $W_K^{1/2}$ are defined analogously):

$$m_i = \langle u_i, W_O^2 \varphi(e_i) \rangle - \max_{j \neq i} \langle u_j, W_O^2 \varphi(e_i) \rangle.$$

As explained in (Bietti et al., 2023), we note that input embeddings to each matrix are often sums/superpositions of embeddings, some of which are typically noise that gets filtered out during training. For instance, training W_O^2 alone may recover the desired associations in high-dimension, even though its input at initialization is an average over all tokens in the sequence, due to the initially flat attention pattern. Our training setup is the following: we consider full-batch gradient descent on a dataset of 16 384 sequences of length 256 generated from the model described above with $N = 64$ tokens. The loss considers only predictions on tokens o_q, ignoring the very first occurrence since it is not predictable from context.