C. R. Acad. Sci. Paris, Ser. I 356 (2018) 395–405

Contents lists available at ScienceDirect

C. R. Acad. Sci. Paris, Ser. I

www.sciencedirect.com

Partial differential equations/Dynamical systems

On the convergence of formally diverging neural net-based classifiers ☆

Convergence de classifieurs par réseaux de neurones formellement divergents

Leonid Berlyand [a], Pierre-Emmanuel Jabin [b]

[a] *Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA*
[b] *CSCAMM and Department of Mathematics, University of Maryland, College Park, MD 20742, USA*

ARTICLE INFO

Article history:
Received 2 March 2018
Accepted 8 March 2018
Available online 16 March 2018

Presented by Haïm Brezis

ABSTRACT

We present an analytical study of gradient descent algorithms applied to a classification problem in machine learning based on artificial neural networks. Our approach is based on entropy–entropy dissipation estimates that yield explicit rates. Specifically, as long as the neural nets remain within a set of “good classifiers”, we establish a striking feature of the algorithm: it mathematically diverges as the number of gradient descent iterations (“time”) goes to infinity but this divergence is only logarithmic, while the loss function vanishes polynomially. As a consequence, this algorithm still yields a classifier that exhibits good numerical performance and may even appear to converge.

RÉSUMÉ

Nous étudions dans cette note le comportement asymptotique d’algorithmes du gradient appliqués à des problèmes de classification basés sur des modèles élémentaires de réseaux neuronaux à apprentissage supervisé. Nous prouvons que ces algorithmes divergent au sens mathématique strict, puisque la suite de paramètres définissant le classifieur est non bornée. Toutefois, en développant des méthodes d’entropie–production d’entropie, notre approche conduit à des taux explicites qui montrent, au moins lorsque les classes peuvent être bien séparées, que les paramètres divergent seulement logarithmiquement alors que la fonction coût converge vers 0 polynomialement. En conséquence, d’un point de vue pratique, l’algorithme permet effectivement d’obtenir un classifieur avec de bonnes performances, et peut même sembler converger.

☆ LB was partially supported by NSF DMREF grant DMS-1628411; PEJ was partially supported by NSF Grant DMS-161453, NSF Grant RNMS (Ki-Net) DMS-1107444 and by LTS grants DO 0048-0049-0050 and 0052.

E-mail addresses: lvb2@psu.edu (L. Berlyand), pjabin@umd.edu (P.-E. Jabin).

https://doi.org/10.1016/j.crma.2018.03.003
1631-073X/© 2018 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

Version française abrégée

Depuis leur introduction (voir notamment déjà [14]), les réseaux de neurones artificiels ont montré des performances remarquables sur les problèmes de classification. Cette note a pour but d'apporter des premiers éléments de réponse rigoureuse sur le comportement de certains de ces réseaux, dans un cadre nécessairement très simplifié.

Plus précisément, nous considérons une famille de classifieurs $\phi(\alpha, s)$, paramétrée par α et associant à chaque objet $s \in S \subset \mathbb{R}^N$, un vecteur $\phi(\alpha, s) = (p_1, \ldots, p_K)$, K étant le nombre de classes, où intuitivement chaque coefficient $p_i(\alpha, s)$ représente la probabilité prédite par le classifieur que s appartienne à la classe i. Cette famille est construite à l'aide d'un réseau neuronal élémentaire où chacune des L couches est, soit entièrement connectée, soit de type convolutif – voir la description complète plus bas avec (1), (2), (3), (4), (5).

L'apprentissage supervisé du réseau consiste, dans le cas le plus simple, à définir une suite de paramètres α obtenus par un algorithme de gradient stochastique qui est basé sur une fonction coût construite à l'aide d'un sous-ensemble d'entraînement T où la classe $i(s)$ de chaque objet $s \in T$ est connue. Un exemple commun consiste à prendre

$$\bar{L}(\alpha) = -\sum_{s\in T} \nu(s)\, \log p_{i(s)}(\alpha, s),$$

pour une certaine loi de probabilité sur T. Toujours par souci de simplicité dans cette note, nous étudions le flot gradient issu de $\bar{L}$, c'est-à-dire le comportement quand $t \to \infty$ du flot

$$\frac{\mathrm{d}}{\mathrm{d}t}\alpha(t) = -\nabla_\alpha \bar{L}(\alpha(t)).$$

Dans le cas où les classes peuvent être parfaitement séparées, on peut espérer que $\bar{L}(\alpha(t)) \to 0$. Toutefois, du fait de la construction des classifieurs $\phi(\alpha, s)$, ceci empêcherait la convergence de $\alpha(t)$ et, plus précisémment, on aurait $|\alpha(t)| \to \infty$.

Ceci amène à la définition de l'ensemble des paramètres α pour lesquels on a séparation des classes

$$\alpha \in \mathcal{A} \quad \text{iff} \quad \forall s \in T, \quad p_{i(s)}(\alpha, s) = \max_k p_k(\alpha, s).$$

Notre analyse montre que $\nabla_\alpha \bar{L}$ a une structure non dégénérée sur $\mathcal{A}$ permettant de le relier à $\bar{L}$. Ceci conduit à des taux explicites sur $\mathcal{A}$.

Théorème 0.1. *Supposons que pour tout $t \in [t_0, \infty)$, $\alpha(t) \in \mathcal{A}$. Il existe alors une constante C dependant seulement de K, ν, $\bar{L}(t_0)$ et $|\alpha(t_0)|$ telle que*

$$\frac{C}{t\,|\log t|^{2M-4}} \le \bar{L}(\alpha(t)) \le \frac{C}{t}, \quad |\alpha(t)| \le C \,\log t.$$

Ce résultat montre que, bien que formellement divergent, l'algorithme fonctionne en pratique, puisqu'il nous permet de sélectionner des paramètres α qui définissent des classifieurs efficaces. Une telle divergence logarithmique est d'ailleurs probablement difficilement observable d'un point de vue numérique. Il faut également souligner que $|\alpha|$ borne directement la norme Lipschitz de $\phi(\alpha, s)$, ce qui est une importante mesure de la robustesse du classifieur.

Notre méthode repose toutefois sur des estimations très précises, de façon à obtenir cette échelle logarithmique. Une inégalité clé est notamment le fait que, pour une certaine constante C,

$$\frac{1}{L}\frac{\mathrm{d}}{\mathrm{d}t}\frac{|\alpha|^2}{2} \ge (\bar{L} - C)\,(|\log \bar{L}| - C).$$

Ces premiers résultats laissent encore un grand nombre de questions non résolues, que nous espérons aborder dans un travail ultérieur. Nous mentionnons seulement brièvement ici la stabilité de l'ensemble $\mathcal{A}$, la dépendance du taux vis-à-vis du choix de la mesure ν, le comportement de l'algorithme lorsque les classes ne sont pas complètement séparées, ou la performance du classifieur obtenu sur tout l'ensemble des objets dans S.

1. Introduction

Neural networks have demonstrated their effectiveness on classifying problems, as exhibited as early as in [14] for handwriting recognition. They are now widely used in many contexts for which we briefly refer to the general presentation for deep neural networks in [13]. We also mention more specifically their performance on image classification, see for example [12], on speech recognition, see [11], on applications to the BioSciences as [16], or natural language understanding, see [22] among many other contributions.

Neural networks have already started to be studied from a more mathematical point of view, providing a mathematical framework to formulate their properties in particular in terms of multiscale contractions involving wavelets and scattering

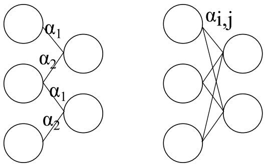

Fig. 1. Left: a convolutional layer. Right: a fully connected layer.

transforms. Those studies have, for example, led to a better understanding of how to embed desired natural invariants in them (for instance translation and rotation invariants in image recognition should not impact the classifiers); we refer the reader in particular to [5,18,17] and to [6] for the use of scattering transforms.

The focus of this work is on the dynamics of how the network learns and our goal is to provide rigorous mathematical understanding of the performance of canonical gradient descent algorithm.

1.1. The neural network and the cost function

It would of course not be possible to give here even a rough overview of the various constructions and approaches used in deep learning networks (see [9] for such an introduction). Instead, we quickly describe the supervised learning neural networks that we consider on a well-posed classification problem, that a set of K classes $S_j \subset \mathbb{R}^N$ with $S_i \cap S_j = \emptyset$ if $i \neq j$. We denote the set of all possible objects by $S = \bigcup_{i=1}^K S_i$ and for all $s \in S$, we denote by $i(s)$ the unique i s.t. $s \in S_i$. We denote by T our given training set, which is a finite subset of S.

Construct a classifier $\phi(\alpha, s)$ in the usual manner by defining a sequence $X^l \in \mathbb{R}^{N_l}$ for $l = 0 \dots M$, M being the number of layers and N_l is the dimension of the layer, where $\alpha \in \mathbb{R}^\mu$ in which μ is large. Put $X^0 = s$ (an object from S or T), $N_0 = N$ and

$$X_i^{l+1} = \lambda\left(Y_i^{l+1}\right), \quad l = 0 \dots M-2, \quad i = 1, 2, \cdots N_{l+1} \tag{1}$$

where Y_i^{l+1} is the input of the activation function λ and $\lambda(\xi) = |\xi|$ or $\lambda(\xi) = \xi_+$ so that we have the following key identity (crucial in the proof)

$$\lambda'(\xi)\, \xi = \lambda(\xi). \tag{2}$$

In our simplified setting, we only consider for the first $M-1$ layers, either a convolutional layer for which

$$Y_i^{l+1} = \sum_{j=0}^{N_l - N_{l+1}} \alpha_j^{l+1} X_{i+j}^l, \tag{3}$$

or a fully connected layer (see Fig. 1), where one has

$$Y_i^{l+1} = \sum_{j=1}^{N_l} \alpha_{i,j}^{l+1} X_j^l. \tag{4}$$

The last layer, $l = M$ uses the softmax function

$$p_i(\alpha, s) = X_i^M = \frac{e^{X_i^{M-1}}}{\sum_{j=1}^K e^{X_j^{M-1}}}, \quad i = 1 \dots K. \tag{5}$$

The performance of the network on a given object s is evaluated through the cross-entropy, leading to the definition of the following error function L, which defines the classification error

$$L(\alpha, s) = -\log p_{i(s)}(\alpha, s). \tag{6}$$

The total classification error is obtained by averaging over the training set, yielding

$$\bar{L}(\alpha) = \sum_{s \in T} \nu(s)\, L(\alpha, s), \tag{7}$$

i.e. the expectation of the error with respect to a probability measure $\nu(s)$ on T such as $\inf_{s \in T} \nu(s) > 0$. This has now become an optimization problem in very large dimension with a non-smooth, non-convex cost function $\bar{L}$, since λ and hence $\bar{L}$ are Lipschitz, but not C^1.

Observe that a perfect classification on the training set T entails that for any $s \in T$, $p_{i(s)}(\alpha, s) = 1$, while $p_k(\alpha, s) = 0$ for all $k \neq i(s)$. This is obviously equivalent to having $\bar{L}(\alpha) = 0$, which justifies the definition of $\bar{L}$. However, by (5), such a perfect classification would require $X_{i(s)}^{M-1} = +\infty$, which in turn implies that $|\alpha| = \infty$.

Still by the form (5), one could conjecture that, provided that there exist good classifiers, then there should exist acceptable parameters α s.t. $\bar{L} \leq C\, \mathrm{e}^{-|\alpha|}$. One would hope that any reasonable method that one chooses to try to solve this optimization problem will lead to such an acceptable α, which is precisely what we prove for a simple gradient algorithm.

1.2. *The gradient algorithms*

Nowadays, training the network is often performed through a stochastic gradient algorithm. Assuming for simplicity that the so-called batch is 1, this yields a Markov chain α^n defined by

$$\alpha^{n+1} = \alpha^n - \tau\, \nabla_\alpha L(\alpha^n, s^n), \tag{8}$$

where the s^n is a sequence of independent random objects chosen according to ν, a discrete probability measure such that $\nu(s) > 0$ for $s \in T$, defined on T.

Equivalently, the law $f_n(\alpha)$ of α^n solves

$$f_{n+1}(\alpha) = \Lambda_\tau\, f_n(\alpha), \quad \Lambda_\tau^*\, \phi(\alpha) = \sum_{s \in T} \nu(s)\, \phi(\alpha - \tau\, \nabla_\alpha L(\alpha, s)). \tag{9}$$

There are several general approaches to obtain rates of convergence for Markov processes. One such approach is to prove convergence in total variation for so-called Harris recurrent chains, see for example [10] or the general presentation in [19]. This does not apply in this case, not only since the sequence α^n is *a priori* unbounded, but more importantly since, even after re-scaling, the limiting measure should be singular.

The possibility of singular limiting measure is what led to the study of the convergence in Wasserstein distances; we refer in particular the reader to the recent [2,3], to [7], or [8]. Interestingly, in such settings, the convergence is often subgeometric (polynomial as is our case here).

Since there is in general no contraction with respect to the Wasserstein distance, applying those methods is challenging. Instead, we have to rely on adhoc estimates derived from the specific structure of the neural network and $\bar{L}$. This is not so surprising, as the behavior of neural net classifier as $n \to \infty$ has long been seen as potentially complex, see in particular [15], while the convergence of stochastic gradient descent has only been investigated in limited, specific framework such as in [21].

Given the limited scope of this note though, we present our methods in the even more simplified setting of the deterministic limit $\tau \to 0$ and $n\,\tau \to t$. The law f_n then converges to a continuous time-dependent law $f(t, \alpha)$ solving the deterministic equation

$$\partial_t f(t, \alpha) = \Lambda\, f(t, \alpha) = \mathrm{div}_\alpha \left(f \sum_{s \in T} \nu(s)\, \nabla_\alpha L(\alpha, s) \right), \tag{10}$$

which, as a first order advection equation, can also be represented by the deterministic characteristics system or flow,

$$\partial_t \alpha(t, \alpha^0) = -\sum_{s \in T} \nu(s) \nabla_\alpha L(\alpha(t, \alpha^0), s) = -\nabla_\alpha \bar{L}(\alpha(t, \alpha^0)), \quad \alpha(t = 0) = \alpha^0. \tag{11}$$

Just as in the random case, the general study of convergence of gradient flows like (10) or (11) often requires convexity as in [4]. Estimates in total variation also exist in deterministic settings, see [20], but for reason similar to the stochastic case, do not seem to be applicable here.

1.3. *Statement of the result*

Our analysis relies on the "non-degeneracy" property of $\nabla_\alpha \bar{L}$ on the set of parameters that defines good classifiers in the following precise sense.

Definition 1. Define the set of parameters $\mathcal{A}$ as follows, $\alpha \in \mathcal{A}$ iff for every $s \in T$

$$X_{i(s)}^{M-1}(\alpha, s) = \max_k X_k^{M-1}(\alpha, s), \quad \text{or equivalently } p_{i(s)}(\alpha, s) = \max_k p_k(\alpha, s).$$

If $\alpha \in \mathcal{A}$, the neural network classifier $\phi(\alpha, s)$ yields the highest probability of belonging to a class k of every object $s \in T$ to the class $i(s)$.

The set $\mathcal{A}$ has some obviously useful properties. For example note that $X_i^{M-1}(\mu\, \alpha, s) = \mu^{M-1}\, X_i^{M-1}(\alpha, s)$ if $\mu \geq 0$, so that $\mathcal{A}$ is a cone: $\alpha \in \mathcal{A} \implies \mu\, \alpha \in \mathcal{A}$ (this remains true if one only scales the parameters on a given layer α^m). But the key property of $\mathcal{A}$ called "non-degeneracy" is that on the interior $\mathring{\mathcal{A}}$, $\nabla_\alpha \bar{L} = 0$ iff $\bar{L} = 0$. For simplicity, we hence introduce a quantified interior set.

Definition 2. The parameters $\alpha \in \mathcal{A}_\eta$ for some $\eta > 0$ iff for every $s \in T$

$$X^{M-1}_{i(s)}(\alpha, s) \geq \max_{k \neq i(s)} X^{M-1}_k(\alpha, s) + \eta. \tag{12}$$

This non-degeneracy is of course critical for the convergence of a gradient flow: the main contribution of this note is entropy–entropy production estimates, which quantify the non-degeneracy property by bounding $|\nabla_\alpha \bar{L}|$ via $\bar{L}$ on the set $\mathcal{A}_\eta$, leading to the rate of decay of $\bar{L}$.

Theorem 3. *Assume that for any $t \in [t_0, \infty)$, one has $\alpha(t) \in \mathcal{A}_\eta$ for some $\eta > 0$. Then there exists a constant C depending on K, ν, $\bar{L}(t_0)$, $|\alpha(t_0)|$ and the diameter of T s.t.*

$$\frac{1}{C\,t\,|\log t|^{2M-4}} \leq \bar{L}(\alpha(t)) \leq \frac{C}{t}, \quad |\alpha(t)| \leq C \log t. \tag{13}$$

Remark 1. Up to a log correction, which for practical purposes is essentially of order 1, we obtain the same asymptotic behavior for the lower bound as for the upper bound.

Remark 2. The rates in Theorem 13 are under implicit conditions on T, $\bar{L} < \inf_{s\in T} \nu(s)$, but we can give an upper bound for the time to reach the point when this inequality holds because we have polynomial rate (32) independent on T. Therefore the constant C in Theorem 13 has a complicated dependence on K, ν, $\bar{L}(t_0)$, and $|\alpha(t_0)|$.

The exponential scaling $\bar{L} \leq C\,\mathrm{e}^{-|\alpha|}$ in (5) (end of Section 1.1) between $|\alpha|$ and $\bar{L}$ fits with the decay rates in (13), which is what we had hoped. In addition of proving that the algorithm performs well on the set $\mathcal{A}_\eta$, we also want to emphasize that the size of α directly bounds the Lipschitz norm of the classifier as $|\nabla_s \phi(\alpha, s)| \lesssim |\alpha|^{M-2}$. Theorem 3 hence guarantees some minimal robustness in the network, in the sense that the classifier will be stable under small changes in an object.

Such robustness is another key feature of acceptable classifier, but we should note that measuring accurately $|\nabla_\alpha \phi(\alpha, s)|$ (instead of the very rough estimate above) is in fact quite challenging, as exemplified in [1].

Of course, Theorem 3 leaves many important open questions that we hope to tackle in a further publication, such as whether the set $\mathcal{A}$ is an attractor or what happens if one starts outside of $\mathcal{A}$. But we also point out that the dependence on ν in the constant C in (13) grows at least linearly with respect to the size of the training set T.

Finally, it may not be possible to have any good classifier (if $\mathcal{A}_\eta = \emptyset$), but we should still expect similar results if condition (12) fails for only few objects in the training set. Then it is not clear how the classifier would perform on the whole set S (instead of the training set T).

2. The proof of Theorem 3

2.1. A uniform relation for the rate of change in ℓ^2 norm of the parameters

The structure of the neural networks allows for some algebraic relations (e.g., (14)) which are key to our proof, and are central to the so-called backpropagation that allows for simplified calculations of the algorithm.

Note that convolution networks are a special case of fully connected ones for the following calculations. Indeed, in the case of a convolution network denote $\alpha^{n+1}_{i,j} = \alpha^{n+1}_{j-i}$ with the convention $\alpha^{n+1}_{i,j} = 0$ if $j - i < 0$. Then $\alpha^{n+1}_{i,j}$ would then form an upper triangular Toeplitz matrix with bandwidth $N_l - N_{l+1} + 1$. Hence, in the following it is sufficient to assume that the layers are fully connected and solve (4).

We then have the following relations for ℓ^2 norm of α

Lemma 2.1. *Assume that the vector α solves (11), then for any m, one has that*

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i,j} |\alpha^m_{i,j}|^2/2 = \sum_{s\in T} \nu(s) \sum_{i \neq i(s)} p_i\,(X^{M-1}_{i(s)} - X^{M-1}_i), \quad m = 1, 2, \cdots M-1 \tag{14}$$

so that in particular $\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i,j} |\alpha^m_{i,j}|^2/2$ is independent of m.

Remark 3. The network has obvious scaling properties since $\phi(\alpha^1, \ldots, \alpha^{m-1}, \mu\,\alpha^m, \alpha^{m+1}, \ldots, \alpha^{M-1}, s)$ is equal to $\phi(\alpha^1, \ldots, \alpha^{l-1}, \mu\,\alpha^l, \alpha^{l+1}, \ldots, \alpha^{M-1}, s)$ for any l, m. This necessarily means that $\sum_{i,j} |\alpha^m_{i,j}|^2 - \sum_{i,j} |\alpha^l_{i,j}|^2$ is invariant under the dynamics. Lemma 2.1 is a more precise and explicit version of this property.

Remark 4. From Lemma 2.1, we immediately find that $\nabla_\alpha \bar{L}$ is non-degenerate on $\mathring{\mathcal{A}}$ (recall that $\mathring{\mathcal{A}}$ is the interior of $\mathcal{A}$). Indeed, if $\alpha \in \mathcal{A}_\eta$ and $\nabla_\alpha \bar{L} = 0$, then since $X_{i(s)}^{M-1} > X_i^{M-1}$ relation (14) implies that $p_i = 0$ for $i \neq i(s)$ and thus $p_{i(s)} = 1$ and $\bar{L} = 0$.

Proof. The equation (11) implies that

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i,j} |\alpha_{i,j}^m|^2/2 = -\sum_{i,j} \alpha_{i,j}^m \partial_{\alpha_{i,j}^m} \bar{L}.$$

In order to prove this lemma, we need to calculate $\partial_{\alpha_{i,j}^m} \bar{L}$.

We follow the basic ideas in backpropagation. Denote by $L_m(\alpha, X^m, k)$ the function obtained by solving (1) with (4) starting from $l = m$ to $l = M-2$ and finishing with (5) and (6) for $i(s) = k$.

With this notation $L(\alpha, s) = L_0(\alpha, X^0 = s, i(s))$. Moreover, L_m does not depend on the $\alpha_{i,j}^l$ for any $l \leq m$, i.e. the coefficients in the previous layer. Also note that $L_m(\alpha, X^m, k) = L_{m+1}(\alpha, X^{m+1}, k)$ for the given input s. Differentiating this equality in X_j^m and in $\alpha_{i,j}^m$ as well as using (1) and (4) for $l = m-1$ and $l = m$, we obtain the two following induction relations which are at the heart of backpropagation

$$\partial_{X_j^m} L_m(\alpha, X^m, k) = \sum_i \partial_{X_i^{m+1}} L_{m+1}(\alpha, X^{m+1}, k)\, \lambda'(Y_i^{m+1}) \alpha_{i,j}^{m+1}, \quad m \leq M-2, \tag{15}$$

$$\partial_{\alpha_{i,j}^m} L(\alpha, s) = \partial_{X_i^m} L_m(\alpha, X^m, i(s))\, \lambda'(Y_i^m)\, X_j^{m-1}, \quad m \leq M-1. \tag{16}$$

We use (2), (11) and (16) to calculate

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \sum_j |\alpha_{i,j}^m|^2/2 &= -\sum_s \nu(s) \sum_{i,j} \partial_{\alpha_{i,j}^m} L(\alpha, s)\, \alpha_{i,j}^m \\
&= -\sum_s \nu(s) \sum_i \partial_{X_i^m} L_m(\alpha, X^m, i(s))\, \lambda'(Y_i^m) \sum_j \alpha_{i,j}^m X_j^{m-1} \\
&= -\sum_s \nu(s) \sum_i \partial_{X_i^m} L_m(\alpha, X^m, i(s))\, \lambda'(Y_i^m)\, Y_i^m \\
&= -\sum_s \nu(s) \sum_i \partial_{X_i^m} L_m(\alpha, X^m, i(s))\, X_i^m.
\end{aligned}$$

On the other hand, by (15) for any k

$$\begin{aligned}
\sum_j \partial_{X_j^m} L_m(\alpha, X^m, k)\, X_j^m &= \sum_{i,j} \partial_{X_i^{m+1}} L_{m+1}(\alpha, X^{m+1}, k)\, \lambda'(Y_i^{m+1}) \alpha_{i,j}^{m+1}\, X_j^m \\
&= \sum_i \partial_{X_i^{m+1}} L_{m+1}(\alpha, X^{m+1}, k)\, \lambda'(Y_i^{m+1})\, Y_i^{m+1} \\
&= \sum_i \partial_{X_i^{m+1}} L_{m+1}(\alpha, X^{m+1}, k)\, X_i^{m+1}.
\end{aligned}$$

Therefore, if one denotes $S_k^m = \sum_i \partial_{X_i^m} L_m(\alpha, X^m, k)\, X_i^m$, then one has that $S_k^m = S_k^{m+1}$ for $m \leq M-2$, proving that

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i,j} |\alpha_{i,j}^m|^2/2 = -\sum_s \nu(s)\, S_{i(s)}^{M-1} = -\sum_s \nu(s) \sum_i \partial_{X_i^{M-1}} L_{M-1}(\alpha, X^{M-1}, i(s))\, X_i^{M-1}. \tag{17}$$

By definition, L_{M-1} is given by (5) and (6). Thus L_{M-1} is independent of α and

$$L_{L-1}(\alpha, X^{M-1}, k) = -\log p_k = -\log \frac{\mathrm{e}^{X_k^{M-1}}}{\sum_j \mathrm{e}^{X_j^{M-1}}} = -X_k^{M-1} + \log \sum_j \mathrm{e}^{X_j^{M-1}}.$$

Therefore,

$$\partial_{X_i^{M-1}} L_{M-1}(\alpha, X^{M-1}, k) = -\delta_{k,i} + p_i. \tag{18}$$

From (17) and (18), we deduce

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i,j}|\alpha^m_{i,j}|^2/2=\sum_s \nu(s)\,\sum_i(\delta_{i,i(s)}-p_i)\,X_i^{M-1}=\sum_s \nu(s)\,\sum_{i\neq i(s)} p_i\,(X_{i(s)}^{M-1}-X_i^{M-1}),$$

since $1-p_{i(s)}=\sum_{i\neq i(s)} p_i$, which concludes the proof of Lemma 2.1. □

2.2. The key bounds

In this subsection we derive bounds on the right hand side of (14) in terms of $\bar L$. First, we prove the following auxiliary lemma.

Lemma 2.2. *Assume that $\alpha\in\mathcal{A}$ and $K\geq 2$. Then*

$$\frac{\bar L}{2\,\log K}\leq\sum_s \nu(s)\,(1-p_{i(s)})\leq \bar L, \tag{19}$$

and

$$\sum_s \nu(s)\,(1-p_{i(s)})\geq \bar L-\frac{K^2}{2\,\inf_s \nu(s)}\,\bar L^2. \tag{20}$$

Proof. If $\alpha\in\mathcal{A}$ then $p_{i(s)}\geq \max_{i=1\dots K} p_i$ and since $\sum_i p_i=1$, one has that $p_{i(s)}\geq 1/K$. Next, consider the following auxiliary inequalities

$$1-z\leq|\log z|\leq 2(\log K)(1-z),\ \text{for all } 1\geq z\geq 1/K \text{ and } K\geq 2, \tag{21}$$

$$|\log z|\leq 1-z+\frac{K^2}{2}|\log z|^2. \tag{22}$$

Inequality (19) follows from (21). To prove (20), we use the definition of $\bar L$ and (22) to obtain that

$$\bar L=-\sum_s \nu(s)\,\log p_{i(s)}=\sum_s \nu(s)\,|\log p_{i(s)}|\leq\sum_s \nu(s)\,(1-p_{i(s)})+\frac{K^2}{2}\,\sum_s \nu(s)\,|\log p_{i(s)}|^2,$$

and combine this estimate with the following inequality which follows directly from the definition of $\bar L$:

$$\bar L^2\geq\sum_s \nu^2(s)\,|\log p_{i(s)}|^2\geq \inf_s \nu(s)\,\sum_s \nu(s)\,|\log p_{i(s)}|^2. \quad □$$

The following estimate is critical to derive explicit rates in Theorem 3.

Lemma 2.3. *Assume that $\alpha\in\mathcal{A}_\eta$. Then*

$$\sum_s \nu(s)\,\sum_{i\neq i(s)} p_i\,(X_{i(s)}^{M-1}-X_i^{M-1})\geq\eta\,\sum_s \nu(s)\,(1-p_{i(s)}). \tag{23}$$

Moreover, if $\bar L$ is small enough, then

$$\sum_s \nu(s)\,\sum_{i\neq i(s)} p_i\,(X_{i(s)}^{M-1}-X_i^{M-1})\geq\sum_s \nu(s)\,(1-p_{i(s)})\,(|\log\bar L|-|\log\inf_s \nu(s)|-\log K). \tag{24}$$

Proof. The bound (23) follows from $X_{i(s)}^{M-1}-X_i^{M-1}\geq\eta$ for $i\neq i(s)$ and $1-p_{i(s)}=\sum_{i\neq i(s)} p_i$.

Since $X_{i(s)}^{M-1}\geq X_i^{M-1}$, for any $i\neq i(s)$ we have that

$$p_i=\frac{1}{\sum_{j=1}^K \mathrm{e}^{X_j^{M-1}-X_i^{M-1}}}\geq\frac{1}{K\,\mathrm{e}^{X_{i(s)}^{M-1}-X_i^{M-1}}}.$$

Hence

$$|\log p_i|=-\log p_i\leq\log\left(K\,\mathrm{e}^{X_{i(s)}^{M-1}-X_i^{M-1}}\right)\leq\log K+X_{i(s)}^{M-1}-X_i^{M-1}.$$

Consequently,

$$\sum_s \nu(s) \sum_{i\neq i(s)} p_i\,(X^{M-1}_{i(s)} - X^{M-1}_i) \geq \sum_s \nu(s) \sum_{i\neq i(s)} p_i\,(|\log p_i| - \log K). \tag{25}$$

Since $1-p_{i(s)} = \sum_{i\neq i(s)} p_i$, then

$$\sum_s \nu(s) \sum_{i\neq i(s)} p_i \log K = \log K \sum_s \nu(s)(1-p_{i(s)}). \tag{26}$$

Similarly since $p_i \leq 1-p_{i(s)}$, one has that

$$\sum_{i\neq i(s)} p_i\,|\log p_i| \geq \sum_{i\neq i(s)} p_i\,|\log(1-p_{i(s)})| = (1-p_{i(s)})\,|\log(1-p_{i(s)})|. \tag{27}$$

If f is a convex increasing function then one can simply bound by Jensen inequality

$$\sum_s \nu(s)\, f(1-p_{i(s)}) \geq f\left(\sum_s \nu(s)\,(1-p_{i(s)})\right) \geq f\left(\frac{\bar L}{2\,\log K}\right),$$

by Lemma 2.2. The corresponding estimate would be much better than (24). But unfortunately for $x\leq 1$, $x\,|\log x| = -x\,\log x$ is concave...

Still observe that, by using the right inequality from (19), we obtain that

$$-\log(1-p_{i(s)}) = -\log(\nu(s)\,(1-p_{i(s)})) + \log\nu(s) \geq -\log\left(\sum_{s'} \nu(s')\,(1-p_{i(s')})\right) + \log\nu(s)$$
$$\geq -\log\bar L + \log\nu(s).$$

By using this inequality to bound from below the right-hand side of (27) and combining the resulting bound with (25) and (26), we derive (24) provided that $\bar L$ is small enough, specifically $\bar L < 1$. □

2.3. Proof of the upper bounds on $\bar L$ and $|\alpha|$

We are now in position to prove the first half of Theorem 3, that is, upper bounds for $\bar L$ and $|\alpha|$ in (13). Using the chain rule and the equation for α (11), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) = -|\nabla_\alpha \bar L(\alpha(t))|^2. \tag{28}$$

Denote $R^2(t) = \sum_{n=1}^{M-1}\sum_{i,j}|\alpha^n_{i,j}|^2$ and observe that

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) \leq -\frac{\alpha\cdot\dot\alpha}{R(t)}\,|\dot\alpha| \leq -\alpha\cdot\dot\alpha\,\frac{\dot R(t)}{R(t)}. \tag{29}$$

Using (23) and then (19) one can obtain that

$$-\alpha\cdot\dot\alpha \leq -\eta M \sum_s \nu(s)(1-p_{i(s)}) \leq -\frac{C_0\eta}{2\log K}\bar L.$$

For simplicity of notation, denote $q := \frac{M\eta}{2\log K}$ so $-\alpha\cdot\dot\alpha \leq -q\bar L$. Since $\dot R \geq 0$, we can now bound the right-hand side of (29):

$$\dot{\bar L} \leq -q\bar L\,\frac{\dot R}{R}.$$

One can divide both sides of this differential inequality by $\bar L$ and integrate to obtain

$$\bar L \leq C_1 R^{-q}. \tag{30}$$

Next, we re-visit inequality (29) and rewrite its right-hand side using $\dot R = R^{-1}(\alpha\cdot\dot\alpha)$ and bound using (19) as well as (30):

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) \leq -\frac{(\alpha\cdot\dot\alpha)^2}{R^2} \leq -C_2\bar L^{2+\frac{2}{q}}. \tag{31}$$

Integrating this inequality, we obtain an upper bound for $\bar{L}$ vanishing as $t \to \infty$:

$$\bar{L}(\alpha(t)) \le C_3 t^{-\frac{1}{1+\frac{2}{q}}}. \tag{32}$$

One can notice that this upper bound is weaker than the desired one in (13). In order to proceed towards the desired upper bound choose time t_1 such that $\bar{L}(\alpha(t_1)) \le \inf_{s\in T} \nu(s)$, which then holds for all $t \ge t_1$ since $\bar{L}(\alpha(t))$ is decreasing in time. Note that (32) shows that $t_1 - t_0$ can be bounded from above in terms of K, ν, $\bar{L}(t_0)$ and $|\alpha(t_0)|$.

Apply estimates (20) and (24) from Lemma 2.2 and 2.3 respectively to obtain that for $C_4 = |\log \inf \nu(s)| + \log K$, we have

$$\begin{aligned} \alpha \cdot \dot{\alpha} &\ge \sum_{s\in T} \nu(s)\,(1 - p_{i(s)})\,(|\log \bar{L}(\alpha(t))| - C_4) \\ &\ge (\bar{L} - C_5\,\bar{L}^2)\,(|\log \bar{L}| - C_5), \end{aligned} \tag{33}$$

where $C_5 = \max(C_4,\ K^2/2 \inf_s \nu(s))$. Define $U(t) = \frac{1}{C_6\,\bar{L}(t)} - 1$ with $C_6 = 2\,C_5$ to find consequently that

$$-\frac{\dot{U}(t)}{U(t)\,\log U(t)} \le -\frac{\dot{R}(t)}{R(t)},$$

so that

$$-\log\log U(t) + \log\log U(t_0) \le -\log R(t) + \log R(t_0).$$

This implies that for some $C_7 > 0$ depending on $\bar{L}(\alpha(t_1))$, $R(t_1)$ and C_5

$$R(t) \le C_7\,|\log \bar{L}(\alpha(t))|.$$

To derive a bound for $\bar{L}$, differentiate $\bar{L}$ in t and use (33) as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{L}(\alpha(t)) \le -\frac{|\alpha\cdot\dot{\alpha}|^2}{R^2(t)} \le -\frac{1}{C_8}\,|\bar{L}|^2,$$

for some C_8 with the same dependence as C_7. Rearrange the terms and integrate to obtain

$$\frac{\mathrm{d}\bar{L}(\alpha(t))}{|\bar{L}(\alpha(t)|^2} \le \mathrm{d}t \implies \frac{-1}{\bar{L}} \le \frac{-t}{C_8} \implies \bar{L} \le C_8/t.$$

This lets us deduce that for some C_8 depending on K and $\bar{L}(\alpha(t_1))$, $R(t_1)$, one has that $\bar{L} \le -C_8/t$ while $R(t) \le C_8 \log t$ for $t \ge t_1$. Thus, upper bounds in (13) are proved.

2.4. *The end of the proof of Theorem 3: the lower bound on $\bar{L}(\alpha(t))$*

The upper bound on $|\alpha|$ derived in the previous section will be the key estimate for obtaining a lower bound on $\bar{L}$. The following analogue of Lemma 2.1 holds.

Lemma 2.4. *Assume that all conditions of Theorem 3 are satisfied. Then there exists a constant C depending on K, ν, $\bar{L}(t_0)$, $\alpha(t_0)$ and the diameter of T s.t. for $t \ge t_0$*

$$|\nabla_\alpha \bar{L}(\alpha(t))| = |\dot{\alpha}(t)| \le C\,(\log t)^{M-2} \sum_{s\in T} \nu(s)\,(1 - p_{i(s)}). \tag{34}$$

Proof. Using the key estimate $|\alpha| \le C \log t$ to deduce from (15) that for $0 \le m \le M-2$, we have

$$\sup_j |\partial_{X_j^m} L_m(\alpha, X^m, k)| \le C \sup_j |\partial_{X_j^{m+1}} L_m(\alpha, X^m, k)| \log t.$$

From (18) for $m = M-1$, we obtain by induction that

$$\sup_j |\partial_{X_j^m} L_m(\alpha, X^m, k)| \le C \sum_i |\delta_{k,i} - p_i|\,(\log t)^{M-1-m}. \tag{35}$$

At the same time, $|X^{m+1}| \le |\alpha^{m+1}|\,|X^m|$. Since $X^0 = s \in T$, by induction and the finite diameter of T, we also have that

$$|X^m| \le C\,(\log t)^m.$$

Use (16) and combine it with (35) and the previous inequality to find

$$|\nabla_{\alpha^m} L(\alpha, s)| \leq C \sum_i |\delta_{k,i} - p_i| \, (\log t)^{M-2}. \tag{36}$$

Observe that, as in Lemma 2.1, the estimate (36) is of the same order for each m. After appropriately summing over s, (36) leads to

$$|\dot\alpha(t)| \leq C \sum_{s\in T} \nu(s) \sum_i |\delta_{i(s),i} - p_i| \, (\log t)^{M-2}.$$

Now one can derive (34) by using that $\sum_i p_i = 1$. □

We can then proceed and conclude the proof of Theorem 3. Use (28) and (34)

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) = -|\nabla_\alpha \bar L(\alpha(t))|^2 \geq -C\,(\log t)^{2M-4} \left|\sum_{s\in T} \nu(s)\,(1-p_{i(s)})\right|^2 .$$

Apply estimate (19) from Lemma 2.2 to find

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) \geq -C\,(\log t)^{2M-4}\,\bar L^2.$$

Rewrite this differential inequality as follows:

$$\frac{1}{\bar L^2}\frac{\mathrm{d}}{\mathrm{d}t}\bar L(\alpha(t)) \geq -C\,(\log t)^{2M-4} > -C\,\left((\log t)^{2M-4} + (2M-4)(\log t)^{2M-5}\right) = -C\,\frac{\mathrm{d}}{\mathrm{d}t}((\log t)^{2M-4} t).$$

Integration in t yields

$$\frac{1}{\bar L(\alpha(t_0))} - \frac{1}{\bar L(\alpha(t))} \geq -C\,(\log t)^{2M-4}\,t + C\,(\log t_0)^{2M-4}\,t_0.$$

For $C' = \frac{1}{\bar L(\alpha(t_0))} - C\,(\log t_0)^{2M-4}\,t_0$, we get

$$\bar L(\alpha(t)) \geq \frac{1}{C' + C\,(\log t)^{2M-4}\,t},$$

which finishes the proof of Theorem 3.

Acknowledgements

We are grateful to H. Chi, R. Creese, and M. Potomkin for useful comments.

References

[1] R. Balan, M. Singh, D. Zou, Lipschitz properties for deep convolutional networks, Contemp. Math. (2018), in press.

[2] O. Butkovsky, Subgeometric rates of convergence of Markov processes in the Wasserstein metric, Ann. Appl. Probab. 24 (2) (2014) 526–552.

[3] O.A. Butkovsky, A.Yu. Veretennikov, On asymptotics for Vaserstein coupling of Markov chains, Stoch. Process. Appl. 123 (9) (2013) 3518–3541.

[4] J.A. Carrillo, M. Di Francesco, A. Figalli, T. Laurent, D. Slepcev, Global-in-time weak measure solutions and finite-time aggregation for nonlocal interaction equations, Duke Math. J. 156 (2011) 229–271.

[5] X. Cheng, X. Chen, S. Mallat, Deep Haar scattering networks, Inf. Inference 5 (2) (2016) 105–133.

[6] W. Czaja, W. Li, Analysis of time-frequency scattering transforms, Appl. Comput. Harmon. Anal. (2018), https://doi.org/10.1016/j.acha.2017.08.005, in press.

[7] R. Douc, G. Fort, E. Moulines, P. Soulier, Practical drift conditions for subgeometric rates of convergence, Ann. Appl. Probab. 14 (2004) 1353–1377.

[8] A. Durmus, G. Fort, E. Moulines, Subgeometric rates of convergence in Wasserstein distance for Markov chains, Ann. Inst. Henri Poincaré Probab. Stat. 52 (4) (2016) 1799–1822.

[9] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[10] M. Hairer, J.C. Mattingly, M. Scheutzow, Asymptotic coupling and a general form of Harris' theorem with applications to stochastic delay equations, Probab. Theory Relat. Fields 149 (1–2) (2011) 223–259.

[11] G. Hinton, et al., Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Process. Mag. 29 (2012) 82–97.

[12] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 2012, 2014, pp. 109–1098.

[13] Y. Le Cun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[14] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackelt, Handwritten digit recognition with a back-propagation network, in: D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems 2, Morgan Kaufmann, San Francisco, CA, 1990, pp. 396–404.

[15] Y. Le Cun, L. Bottou, G. Orr, K. Muller, Efficient BackProp, in: G. Orr, K. Muller (Eds.), Neural Networks: Tricks of the Trade, Springer, 1998.

[16] M.K. Leung, H.Y. Xiong, L.J. Lee, B.J. Frey, Deep learning of the tissue regulated splicing code, Bioinformatics 30 (2014), i121–i129.

[17] S. Mallat, Group invariant scattering, Commun. Pure Appl. Math. 65 (10) (2012) 1331–1398.

[18] S. Mallat, Understanding deep convolutional networks, Phil. Trans. R. Soc. A 374 (2065) (2016) 20150203.

[19] S. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, 2nd ed., Cambridge University Press, Cambridge, UK, 2009.

[20] P. Michel, S. Mischler, B. Perthame, General relative entropy inequality: an illustration on growth models, J. Math. Pures Appl. 84 (9) (2005) 1235–1260.

[21] O. Shamir, Convergence of stochastic gradient descent for PCA, preprint, arXiv:1509.09002v2.

[22] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proc. 28th Annual Conf. on Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014.