French-German-Spanish Conference on Optimization
Gijón, June 18-21, 2024
(pp. 11-16)

Controllability of neural ODEs for data classification

Antonio Álvarez-López

Departamento de Matemáticas, Universidad Autónoma de Madrid, 28049 Madrid, Spain.

Abstract

In this work, we explore the capacity of neural ordinary differential equations (ODEs) for supervised learning from a control perspective. Specifically, we rely on the property of simultaneous controllability and explicitly construct the controls that achieve this as piecewise constant functions in time.

First, we analyze the expressivity of the model for cluster-based classification by estimating the number of neurons required for the classification of a set constituted by N points. We consider a worst-case scenario where these points are independently sampled from $U([0,1]^d)$. Assuming only that the initial points are in general position, we propose an algorithm that classifies clusters of d points simultaneously, employing $O(N/d)$ neurons.

Secondly, we examine the impact of the architecture, determined by the depth p and width L, for interpolating a set of N pairs of points. Our findings reveal a balance where L scales as $O(1+N/p)$. For the autonomous model, with constant controls ($L=0$), we relax the problem to approximate controllability of N pairs of points, establishing an explicit error decay with respect to p. Finally, we extend the problem to the approximate control of measures in the Wasserstein space, finding another balance between p and L.

1. Introduction

Supervised learning is one of the main paradigms in machine learning. Given some spaces $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ with $d, m \geq 1$, the problem can be formulated as the approximation of an unknown function $f : \mathcal{X} \to \mathcal{Y}$ using a parametric model built from the information contained in a training dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathcal{X} \times \mathcal{Y}$, where $\mathbf{y}_n = f(\mathbf{x}_n)$ for all n.

Neural networks constitute a widely used class of models, and among them, residual networks have been shown to be particularly effective. A residual neural network, defined for a fixed depth $L \in \mathbb{N}$, operates as a discrete system given by:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \sum_{i=1}^{p} \mathbf{w}_{l,i} \sigma(\mathbf{a}_{l,i} \cdot \mathbf{x}_l + b_{l,i}), \qquad l = 0, \dots, L, \tag{1.1}$$

where $\mathbf{x}_l \in \mathbb{R}^d$ is the sequence of states, $\cdot$ denotes the scalar product, and:

- $\mathbf{w}_{l,i}, \mathbf{a}_{l,i} \in \mathbb{R}^d$ and $b_{l,i} \in \mathbb{R}$ are the parameters;
- p is the width of the model;
- σ is a predefined nonlinearity, frequently the Rectified Linear Unit (ReLU) function, defined by:

$$\sigma(z) = \max\{z, 0\}, \qquad \text{for } z \in \mathbb{R}. \tag{1.2}$$

Neural ODEs are essentially the continuous-time limit of residual networks [5]. They are obtained by multiplying the nonlinear term in (1.1) by a constant $h > 0$ and taking the limit when $h \to 0$, resulting in:

$$\dot{\mathbf{x}} = \sum_{i=1}^{p} \mathbf{w}_i(t)\, \sigma(\mathbf{a}_i(t) \cdot \mathbf{x} + b_i(t)), \qquad t \in (0, T), \tag{1.3}$$

where the parameters can now be seen as p control functions $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p \subset L^\infty\left((0,T), \mathbb{R}^{2d+1}\right)$, for some $T > 0$. Note that the time horizon T does not play a major role, since equation (1.3) admits a time-rescaling property: one can equivalently fix $T = 1$ and absorb a factor T into $\mathbf{w}_i$.

One of the main advantages of neural ODEs is that they enable the reinterpretation and study of various machine learning paradigms using the tools from differential equations and dynamical systems [10]. For instance, data classification can be formulated as a problem of simultaneous control of the system (1.3). The objective is to design p controls that drive every initial data point $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^d$ to its corresponding target point via the flow map at time T of the system (1.3).

To facilitate the geometric interpretation of the dynamics, achieve a layered structure similar to (1.1), and reduce the problem to finite dimensions, it is often assumed that the controls are piecewise constant in time [7, 9]. The discrete network's depth can then be interpreted as the number of distinct values that these controls take, and each of the finite-jump discontinuities, whose total number we denote by L, corresponds to a layer transition.

Within each layer $t \in (t_{k-1}, t_k) \subset (0, T)$, the controls $\mathbf{a}_i(t) \equiv \mathbf{a}_i \in \mathbb{R}^d$ and $b_i(t) \equiv b_i \in \mathbb{R}$ define p hyperplanes $H_1, \dots, H_p$. The ReLU function in (1.2) then activates or deactivates the corresponding half-spaces:

$$H_i^+ := \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a}_i \cdot \mathbf{x} + b_i > 0\} \quad \text{and} \quad H_i^- := \mathbb{R}^d \setminus H_i^+, \qquad \text{for all } i = 1, \dots, p, \tag{1.4}$$

Meanwhile, each control $\mathbf{w}_i(t) \equiv \mathbf{w}_i \in \mathbb{R}^d$ determines a vector field acting solely on the points inside the half-space H_i^+. The total field in (1.3) acts on each point $\mathbf{x} \in \mathbb{R}^d$ as a weighted superposition of the form $\sum_{i=1}^p \operatorname{dist}(\mathbf{x}, H_i^-)\mathbf{w}_i$, where the i-th term is null when $\mathbf{x} \in H_i^-$. By appropriately defining the controls, we can thus fix any hyperplane H_i in $\mathbb{R}^d$ and generate three basic dynamics, as represented in Figure 1.

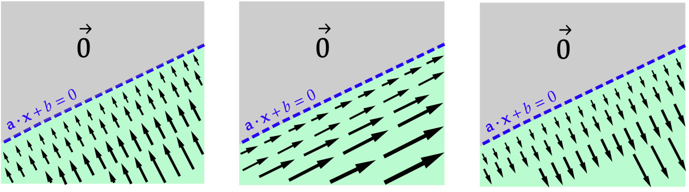


Fig. 1 Basic movements that we can generate: Compression, laminar motion, expansion (from left to right).

2. Controlled cluster-based classification

First, we address binary classification, where $\mathcal{Y} = \{1, 0\}$. In this context, the values y_n are commonly referred to as labels. We associate the two labels with a pair of target regions that are linearly separable and form a partition of $\mathbb{R}^d$. For example, the two half-spaces defined by $x^{(k)} \neq 1$. Our goal is to design controls for the neural ODE that generate a flow mapping each initial point $\mathbf{x}_n$ to the corresponding target region $x^{(k)} > 1$ or $x^{(k)} < 1$.

Furthermore, for optimal classification, the complexity of the model, represented by the number of neurons defining the network, should not grow excessively large. By fixing $p = 1$ in (1.3), the complexity is thus determined solely by the number of discontinuities in the controls over time:

$$\dot{\mathbf{x}} = \mathbf{w}(t)\, \sigma(\mathbf{a}(t) \cdot \mathbf{x} + b(t)), \qquad t \in (0, T). \tag{2.1}$$

In [7], classification of any finite dataset was achieved through a constructive algorithm that leverages the nonlinear dynamics of (2.1) to simultaneously control the N points inductively. The main result in this work is the following:

Theorem 2.1 *Let $N \geq 1$, $d \geq 2$, and $T > 0$. Consider any dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^N \subset \mathbb{R}^d \times \{1, 0\}$ with $\mathbf{x}_n \neq \mathbf{x}_m$ if $n \neq m$. Then, there exists a piecewise constant control $(\mathbf{w}, \mathbf{a}, b) \in L^\infty\left((0, T), \mathbb{R}^{2d+1}\right)$ such that the flow map Φ_T generated by (2.1) satisfies, for all $n = 1, \dots, N$:*

$$\Phi_T(\mathbf{x}_n)^{(1)} > 1 \quad \text{if } y_n = 1, \qquad \text{and} \qquad \Phi_T(\mathbf{x}_n)^{(1)} < 1 \quad \text{if } y_n = 0,$$

Furthermore, the number of discontinuities in the controls is $L = 3N$.

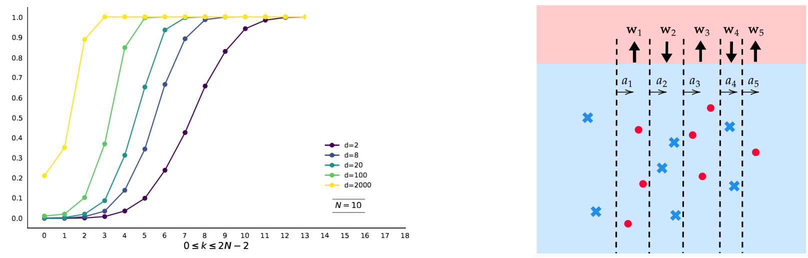
Theorem 2.1 opens new pathways for methodologies in data classification. However, it requires high complexity since the number of neurons scales with N due to the inductive nature of the algorithm. In [1], we propose new algorithms that consider the spatial structure of the data distribution to reduce the number of parameters needed. Specifically, by assuming that the points are randomly sampled from $U([0, 1]^d)$—a worst-case scenario of pure noise—we construct controls that provide the following probabilistic bound on the model's depth:

Theorem 2.2 *Let $N \geq 1$, $d \geq 2$, and $T > 0$. Consider any dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^{2N}$ with $\mathbf{x}_n \sim U([0, 1]^d)$ and $y_n \in \{1, 0\}$ for all n, satisfying $\#\{n : y_n = 1\} = \#\{n : y_n = 0\} = N$. Then, there exist a direction $j \in \{1, \dots, d\}$, a* *piecewise constant control* $(\mathbf{w}, b) \in L^\infty\left((0,T), \mathbb{R}^{d+1}\right)$ *and* $\mathbf{a} \in \{\mathbf{e}_1, \dots, \mathbf{e}_d\}$*, such that the flow map* Φ_T *generated by* (2.1) *satisfies, for all* $n = 1, \dots, 2N$*:*

$$\Phi_T(\mathbf{x}_n)^{(j)} < 1 \quad \textit{if } y_n = 1, \quad \textit{and} \quad \Phi_T(\mathbf{x}_n)^{(j)} > 1 \quad \textit{if } y_n = 0.$$

Furthermore, the number of discontinuities L follows the probability distribution, for $0 \leq k \leq 2N-2$*,*

$$\mathbb{P}(L \geq k) = \left(\sum_{p=\lceil \frac{k+1}{2} \rceil}^{N} \binom{N-1}{p-1}^2 + \sum_{p=\lceil \frac{k}{2} \rceil}^{N-1} \binom{N-1}{p}\binom{N-1}{p-1} \right)^d 2^d \binom{2N}{N}^{-d}. \tag{2.2}$$



(a) Visualization of (2.2) for $N = 10$ and different values of d.

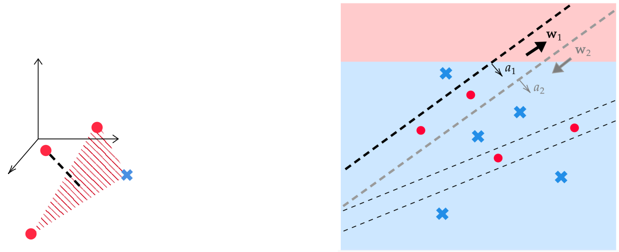
(b) Representation of the algorithm for classification from Theorem 2.2.

Fig. 2

The maximum number of $L = 2N-2$ discontinuities corresponds to the configuration where the $2N-1$ points lie on a single line and are interspersed according to their labels. Although these scenarios are typically unrealistic, they hold a positive probability in Theorem 2.2 due to the strong constraint on $\mathbf{a}$. However, if we assume that the points are in general position, meaning no $d+1$ points lie on the same hyperplane (see figure 3a), we can build new controls that refine the maximum value of L:

Theorem 2.3 *Let* $d \geq 2$*,* $N \geq 1$*, and* $T > 0$*. Consider any dataset* $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d \times \{1, 0\}$ *in general position and any direction* $j \in \{1, \dots, d\}$*. Then, there exists a piecewise constant control* $(\mathbf{w}, \mathbf{a}, b) \in L^\infty\left((0,T), \mathbb{R}^{2d+1}\right)$ *with* $L = 4\lceil m/d \rceil - 1$ *discontinuities, where* $m = \min\left(\#\{n : y_n = 1\}, \#\{i : y_n = 0\}\right)$*, such that the flow map generated by* (2.1) *satisfies, for all* $n = 1, \dots, N$*:*

$$\Phi_T(\mathbf{x}_n)^{(j)} < 1 \quad \textit{if } y_n = 1 \quad \textit{and} \quad \Phi_T(\mathbf{x}_n)^{(j)} > 1 \quad \textit{if } y_n = 0.$$



(a) General position setting.

(b) Representation of the algorithm for classification from Theorem 2.3.

Fig. 3

3. Interplay between depth and width

3.1. In simultaneous control

As an extension of Theorem 2.1, the property of simultaneous control was also proven in [7] by constructing the necessary controls in (2.1):

Theorem 3.1 *Let $N \geq 1$, $d \geq 2$, and $T > 0$. Consider any dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d$ with $\mathbf{x}_n \neq \mathbf{x}_m$ and $\mathbf{y}_n \neq \mathbf{y}_m$ for $n \neq m$. Then, there exists a piecewise constant control $(\mathbf{w}, \mathbf{a}, b) \in L^\infty\left((0,T), \mathbb{R}^{2d-1}\right)$ such that the flow map Φ_T generated by (2.1) satisfies:*

$$\Phi_T(\mathbf{x}_n) = \mathbf{y}_n, \qquad \text{for all } n = 1, \dots, N.$$

Furthermore, the number of discontinuities in the controls is $L = 4N$.

In our second work [2], we focus on the role that the architecture can play in this task by allowing the width to be $p \geq 1$ and studying its interplay with the depth L. Our findings reveal a balancing trade-off between these two parameters, as shown in the following result:

Proposition 3.2 *Let $N \geq 1$, $d \geq 2$, and $T > 0$. Consider any dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d$ with $\mathbf{x}_n \neq \mathbf{x}_m$ and $\mathbf{y}_n \neq \mathbf{y}_m$ for $n \neq m$. Then, for any $p \geq 1$, there exist piecewise constant controls $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p \subset L^\infty\left((0,T), \mathbb{R}^{2d+1}\right)$ such that the flow map Φ_T generated by (1.3) satisfies:*

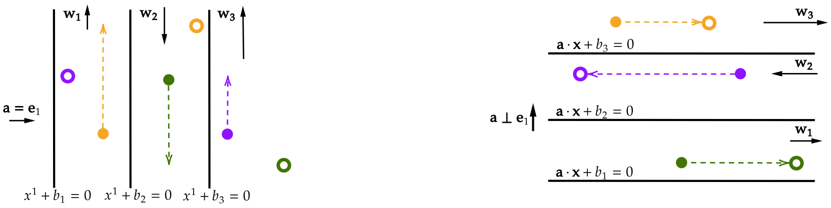
$$\Phi_T(\mathbf{x}_n) = \mathbf{y}_n, \qquad \text{for all } n = 1, \dots, N.$$

Furthermore, the number of discontinuities in the controls is $L = 2\left\lceil \frac{N}{p} \right\rceil - 1$.

We can see that as the width p increases, the parameter L decreases at the same rate, indicating that both play a similar role in the steering process. However, whenever $p \geq N$, the constructed control will exhibit only one switch ($L = 1$), which precludes a complete transition to the autonomous model

$$\dot{\mathbf{x}} = \sum_{i=1}^{p} \mathbf{w}_i \sigma(\mathbf{a}_i \cdot \mathbf{x} + b_i) \tag{3.1}$$

This is because the proof is algorithmically divided into two phases, represented in Figure 4. First, we control $d-1$ coordinates of each batch of p points, and then we control the remaining coordinate. Therefore, at least one discontinuity is inevitable to transition between these two phases.



$\mathbf{w}_1$
$\mathbf{w}_2$
$\mathbf{w}_3$
$\mathbf{a} = \mathbf{e}_1$
$x^1 + b_1 = 0$
$x^1 + b_2 = 0$
$x^1 + b_3 = 0$
$\mathbf{w}_3$
$\mathbf{a} \cdot \mathbf{x} + b_3 = 0$
$\mathbf{w}_2$
$\mathbf{a} \perp \mathbf{e}_1$
$\mathbf{a} \cdot \mathbf{x} + b_2 = 0$
$\mathbf{w}_1$
$\mathbf{a} \cdot \mathbf{x} + b_1 = 0$

Fig. 4 Left: Step 1. Control of $d-1$ coordinates. Right: Step 2. Control of the remaining coordinate.

Motivated by this observation, we now pose the following question:

Is it possible to achieve exact control using $L = 0$ discontinuities?

There are some remarks that can be made as a first approach:

1. **Semi-autonomous neural ODE**: If we consider the semi-autonomous neural ODE where only the controls b_i depend on time,

$$\dot{\mathbf{x}} = \sum_{i=1}^{p} \mathbf{w}_i \sigma(\mathbf{a}_i \cdot \mathbf{x} + b_i(t)), \tag{3.2}$$

we can adapt the proof of Theorem 3.2, obtaining the same result and the same number of discontinuities for some controls $(b_i)_{i=1}^p \subset L^\infty\left((0,T), \mathbb{R}\right)$, but with constant $(\mathbf{w}_i, \mathbf{a}_i)_{i=1}^p \subset \mathbb{R}^{2d}$.

2. **High dimensions**: When $d > N$, the second step in the proof of Theorem 3.2 can be omitted because we can find a new basis of $\mathbb{R}^d$ in which each point $\mathbf{x}_n$ shares the first coordinate with its target $\mathbf{y}_n$. Thus, we reduce L to $2\left\lceil \frac{N}{p} \right\rceil - 2$.

3. **Probabilistic**: Additionally, we can estimate the probability that the points will appear in certain spatial configurations that facilitate their autonomous control. For instance, if $\mathbf{x}_n$ and $\mathbf{y}_n$ are randomly sampled from $U([0,1]^d)$ for all $n = 1, \dots, N$, then with probability P bounded as

$$1 \geq P \geq 1 - \left[1 - \frac{1}{\sqrt{2}}\left(\frac{e}{2N}\right)^N\right]^d \to 1,$$

there exist p controls $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p \subset \mathbb{R}^{2d+1}$ such that $\Phi_T(\mathbf{x}_n) = \mathbf{y}_n$.

In general, another option is to relax the problem statement to approximate controllability, which means allowing a uniform error $\varepsilon > 0$ that can be made arbitrarily small. Thus, we can obtain the following result:

Theorem 3.3 *Let $N \geq 1$, $d \geq 2$, and $T > 0$. Consider any dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d$ with $\mathbf{x}_n \neq \mathbf{x}_m$ for $n \neq m$. For each $p \geq 1$, there exist controls $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p \subset \mathbb{R}^{2d+1}$ such that the flow map Φ_T generated by* (1.3) *satisfies*

$$\sup_{n=1,\dots,N} \left|\mathbf{y}_n - \Phi_T(\mathbf{x}_n)\right| \leq C\,\frac{\log_2(\kappa)}{\kappa^{1/d}},$$

where $\kappa = (d+2)dp$ is the number of parameters in the model, and $C > 0$ is a constant independent of p.

The strategy consists of applying an approximation theorem for shallow neural networks in the space of Lipschitz functions with respect to the uniform norm, providing explicit convergence rates, as derived from [3]. The vector field to be approximated will be a time-independent Lipschitz field whose integral curves guide each input point $\mathbf{x}_n$ in $\mathcal{D}$ to its corresponding target $\mathbf{y}_n$ within a fixed time T. The construction of this field is described in Figure 5.

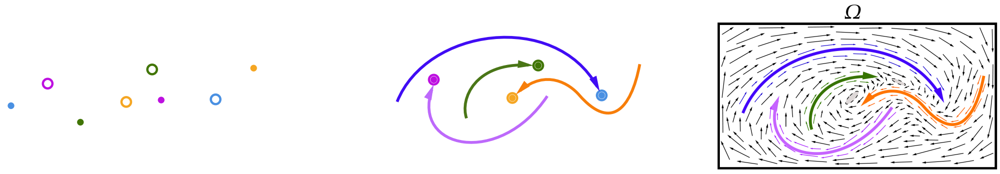


Fig. 5 Construction of a Lipschitz field which interpolates $\mathcal{D}$ in a compact domain Ω that contains all the points and curves.

3.2. In neural transport

As an extension of the results we present in this section, we also consider the reformulation of the model (1.3) as a semilinear hyperbolic equation, known as the neural transport equation:

$$\partial_t \mu + \mathrm{div}_x(\mathbf{V}(\mathbf{x})\mu) = 0, \qquad \text{with} \quad \mathbf{V}(\mathbf{x}) = \sum_{i=1}^{p} \mathbf{w}_i \sigma(\mathbf{a}_i \cdot \mathbf{x} + b_i). \tag{3.3}$$

This equation defines the evolution of a measure μ in $\mathbb{R}^d$ following an advection vector field $\mathbf{V}$ given by the neural ODE. The case of N initial data points is recovered by taking N Dirac deltas as the base measure, which evolve according to the characteristic equation given by (1.3).

We will work in the space $\mathcal{P}_{ac}^c(\mathbb{R}^d)$ of compactly supported, absolutely continuous probability measures in $\mathbb{R}^d$, with the metric given by the Wasserstein distance, which is rooted in the theory of optimal transport. For any pair of measures $\mu, \nu \in \mathcal{P}_{ac}^c(\mathbb{R}^d)$ and $q \geq 1$, the Wasserstein-q distance between μ and ν is defined by

$$\mathcal{W}_q(\mu,\nu) := \left(\min_{\gamma \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} |\mathbf{x} - \mathbf{y}|^q \, d\gamma(\mathbf{x},\mathbf{y})\right)^{1/q}, \tag{3.4}$$

where $\Pi(\mu,\nu)$ is the space of all couplings of μ and ν:

$$\Pi(\mu,\nu) := \left\{\gamma \in \mathcal{P}_{ac}^c(\mathbb{R}^d \times \mathbb{R}^d) \mid \gamma(\cdot \times \mathbb{R}^d) = \mu(\cdot) \quad \text{and} \quad \gamma(\mathbb{R}^d \times \cdot) = \nu(\cdot)\right\}.$$

Since the vector field $\mathbf{V}(\mathbf{x})$ in (3.3) is Lipschitz in $\mathbf{x}$, the classic Cauchy-Lipschitz theorem guarantees that the curve $\mu(t)(\cdot) := \Phi_t(\cdot;\theta)\#\mu_0$ in $\mathcal{P}_{ac}^c(\mathbb{R}^d)$ is well-defined, where $\Phi_T\#\mu_0$ denotes the pushforward measure under Φ_T.

The objective now is to study the controllability problem of the equation (3.3), aimed at transforming one given probability measure into another, up to an arbitrarily small error ε. As in simultaneous control, the case with $p = 1$ was resolved for both the total variation metric in [8] and the Wasserstein-1 space in [7]. In the latter work, the following result was obtained:

Theorem 3.4 *Let $d \geq 2$ and $T > 0$. For any $\mu_0, \mu_* \in \mathcal{P}_{ac}^c(\mathbb{R}^d)$ and $\varepsilon > 0$, there exists a piecewise constant control $(\mathbf{w}, \mathbf{a}, b) \in L^\infty\left((0,T), \mathbb{R}^{2d+1}\right)$ such that the solution $\mu(t)$ of* (3.3)*, taking μ_0 as initial condition, satisfies*

$$\mathcal{W}_1(\mu(T), \mu_*) < \varepsilon.$$

In our work [2], we study the case with $p \geq 1$ for the uniform measure in the hypercube $[0,1]^d$ as the target. The control algorithm we develop is explicit and allows us to obtain an explicit expression for the number of discontinuities L in terms of p, d, and the order of Wasserstein q:

Theorem 3.5 *Let $d \geq 2$ and $T > 0$. For any $\mu_0 \in \mathcal{P}_{ac}^c(\mathbb{R}^d)$, $\varepsilon > 0$, $q \in \left[1, \frac{d}{d-1}\right)$, and $p \geq 1$, there exist piecewise constant controls $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p \subset L^\infty\left((0,T), \mathbb{R}^{2d+1}\right)$ such that the solution $\mu(t)$ of* (3.3)*, taking μ_0 as the initial condition, satisfies*

$$\mathcal{W}_q(\mu(T), \mu_*) < \varepsilon,$$

and the number of discontinuities in the controls is

$$L = \left\lceil \frac{2d}{p} \right\rceil + \left\lceil \frac{1}{p-d+1} \left(\frac{3^{1+d/q}\sqrt{d}}{\varepsilon} \right)^{\frac{d}{1+d/q-d}} \right\rceil - 1.$$

As a final remark, when $q = 1$ then the number of discontinuities simplifies to:

$$L = \left\lceil \frac{2d}{p} \right\rceil + \left\lceil \frac{1}{p-d+1} \left(\frac{3^{1+d}\sqrt{d}}{\varepsilon} \right)^{d} \right\rceil - 1.$$

Acknowledgements

The author A. Álvarez-López has been funded by contract FPU21/05673 from the Spanish Ministry of Universities and supported by the Government of Madrid under the multiannual agreement with UAM for the Excellence of University Research Staff within the context of the V PRICIT Programme.

References

[1] Antonio Álvarez-López, Rafael Orive-Illera, and Enrique Zuazua. Optimized classification with neural ODEs via separability. *Preprint arXiv:2312.13807*, 2023.

[2] Antonio Álvarez-López, Arselane Hadj Slimane, and Enrique Zuazua. Interplay between depth and width for interpolation in neural ODEs. *Preprint arXiv:2401.09902*, 2024.

[3] Francis Bach. Breaking the Curse of Dimensionality with Convex Neural Networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[4] Borjan Geshkovski and Enrique Zuazua. Turnpike in optimal control of PDEs, ResNets, and beyond. *Acta Numerica*, 31:135–263, 2022. Cambridge University Press.

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583, Curran Associates Inc., 2018.

[6] Domènec Ruiz-Balet, Elisa Affili, and Enrique Zuazua. Interpolation and approximation via Momentum ResNets and Neural ODEs. *Systems & Control Letters*, 162:105182, 2022.

[7] Domènec Ruiz-Balet and Enrique Zuazua. Neural ODE Control for Classification, Approximation, and Transport. *SIAM Review*, 65(3):735–773, 2023.

[8] Domènec Ruiz-Balet and Enrique Zuazua. Control of neural transport for normalising flows. *J. Math. Pures Appl. (9)*, 181:58-90, 2024.

[9] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural ODEs. *Advances in Neural Information Processing Systems*, 33:3952–3963, 2020.

[10] E Weinan. A Proposal on Machine Learning via Dynamical Systems. *Communications in Mathematics and Statistics*, 5:1-11, 2017.