IFAC Journal of Systems and Control 30 (2024) 100290

Contents lists available at ScienceDirect

IFAC Journal of Systems and Control

journal homepage: www.elsevier.com/locate/ifacsc

Full length article

On the turnpike to design of deep neural networks: Explicit depth bounds☆

Timm Faulwasser [a,*], Arne-Jens Hempel [b], Stefan Streif [c]

[a] Institute of Control Systems, Hamburg University of Technology, 20179 Hamburg, Germany
[b] Chair for Automation, Berufsakademie Sachsen Staatliche Studienakademie Glauchau, 08371 Glauchau, Germany
[c] Automatic Control and System Dynamics Laboratory, Technische Universität Chemnitz, 09126 Chemnitz, Germany

ARTICLE INFO

Article history:
Received 21 May 2024
Received in revised form 30 September 2024
Accepted 23 October 2024
Available online 2 November 2024

Keywords:
Dissipativity
Turnpike properties
Deep learning
Artificial neural networks
Machine learning

ABSTRACT

It is well-known that the training of Deep Neural Networks (DNN) can be formalized in the language of optimal control. In this context, this paper leverages classical turnpike properties of optimal control problems to attempt a quantifiable answer to the question of how many layers should be considered in a DNN. The underlying assumption is that the number of neurons per layer—i.e., the width of the DNN—is kept constant. Pursuing a different route than the classical analysis of approximation properties of sigmoidal functions, we prove explicit bounds on the required depths of DNNs based on asymptotic reachability assumptions and a dissipativity-inducing choice of the regularization terms in the training problem. Numerical results obtained for the two spiral task data set for classification indicate that the proposed constructive estimates can provide non-conservative depth bounds.

1. Introduction

Machine Learning (ML) and Optimal Control (OC)—despite being seemingly distant topics—share fruitful and established interconnections. For example, the well-known concept of back propagation, popularized by Rumelhart et al. (1986),[1] enables efficient gradient computation in the training of various ML algorithms. This concept arises naturally in discrete-time OC disguised as adjoint/co-state dynamics, see Bryson and Denham (1962) and the overview by Baydin et al. (2017). Indeed the recently exponentially growing interest in deep learning and artificial neural networks was preceded by seminal results obtained in systems and control. Consider, e.g., the analysis of approximation properties of sigmoidal (activation) functions (Cybenko, 1989), the works on controllability and reachability properties of recurrent neural networks (Sontag & Sussmann, 1997) and the more recent works (Ruiz-Balet et al., 2022; Ruiz-Balet & Zuazua, 2023). Moreover, it has been proposed to analyze the training of Deep Neural Nets (DNN) in optimal control frameworks (Esteve et al., 2020; Li et al., 2017).

The classical works of Cybenko (1989), Hornik (1991), and a large number of follow-up papers, have shown that the universal approximation properties of multilayer feed-forward networks hold when condensed to huge single hidden layer networks. However, so-called *shallow* networks may encounter problems when it comes to learning as well as generalization. They might also have more parameters/weights then their multi-layered counterparts, cf. the empirical study (Ba & Caruana, 2014). It stands to reason that there is a substantial depth vs. width trade-off in designing neural networks.

When training DNNs network depth and width are usually chosen empirically by try-and-error. Another approach is the additive construction of DNNs which requires many learning cycles or conceiving the network depth as uncertain/random variable (Antoran et al., 2020). Alternatively, one can address the depth-design problem within the optimization/learning process, e.g., by choosing zero weights or by choosing other network architectures like deep residual networks featuring bypasses of layers, see Goodfellow et al. (2016). In principle, one could also re-phrase the depth-choice as a so-called hyper-parameter tuning problem and apply Bayesian regression (Maclaurin et al., 2015). An abstract view of network depth based on upper bounds of empirical margin error along with possible measures is given by Sun et al. (2016a).

In principle, the solution of supervised ML tasks via deep learning and neural networks entails two different aspects. The *formulation of the training* optimization problem and its *numerical*

☆ This paper was not presented at any IFAC meeting. An earlier version appeared as preprint on arxiv https://arxiv.org/abs/2101.03000. Major parts of this work have been conducted while TF was with the Institute of Energy Systems, Energy Efficiency and Energy Economics (ie3) at TU Dortmund University.

* Corresponding author.
E-mail addresses: timm.faulwasser@ieee.org (T. Faulwasser), arne-jens.hempel@ba-sachsen.de (A.-J. Hempel), stefan.streif@etit.tu-chemnitz.de (S. Streif).

[1] In 2024, John J. Hopfield and Geoffrey E. Hinton received the Nobel Prize in Physics *for foundational discoveries and inventions that enable machine learning with artificial neural networks.*

https://doi.org/10.1016/j.ifacsc.2024.100290

solution. While the former refers to the choice of loss functions, regularization penalties and network DNN architectures, the latter is concerned with computing approximations to solutions of the training problem in scalable and efficient fashion. In this paper, we investigate the formulation of DNN training from the perspective of Optimal Control Problem (OCP) similar to Esteve et al. (2020) and Li et al. (2017). That is, we focus on the first aspect mentioned above. Eventually, this allows us to propose and prove constructive depth bounds using optimal control theory.

This way, we also do not take the functional analytic and function approximation route paved by Cybenko (1989) and Hornik (1991) and others. Rather we regard a feed-forward neural network as a dynamic system on a finite horizon and invoke reachability properties to derive depth bounds, while the width—i.e., the number of neurons per layer—is kept constant. We show that reachability properties of the neural network refer the ability of the network to infer a suitable label from a given data point, i.e., to the ability to solve a particular classification or regression task. Specifically, we rely on dissipativity-based analysis techniques of OCPs which leverage turnpike properties. The term turnpike refers to a similarity property of solutions of OCPs for varying initial conditions and horizon lengths. It has been coined by Dorfman et al. (1958) and has received considerable interest in economics, see Carlson et al. (1991) and McKenzie (1976). There has been interest in turnpike properties and their relation to dissipation inequalities (Faulwasser et al., 2017; Grüne & Müller, 2016), in turnpikes of PDE-constrained OCPs (Gugat et al., 2016), mixed-integer problems (Faulwasser & Murray, 2020) and economic MPC (Faulwasser et al., 2018). We refer to Faulwasser and Grüne (2022) and Grüne (2022) for overviews. Moreover, the results of Esteve et al. (2020) suggest analyzing turnpike properties in continuous-time OCPs arising from continuous training problems of DNNs. Therein, ODE and PDE turnpike properties are established using a finite-time reachability assumption for the underlying DNNs and leveraging the structure of the optimality conditions.

The present paper also takes a turnpike approach to the deduction of depth-bounds as well as the design of DNNs. Therefore and in contrast to Esteve et al. (2020), we consider the training of discrete DNNs with a constant number of units per layer. We show, based on an established dissipativity notion for OCPs, that the training of DNNs exhibits the turnpike phenomenon. Moreover, we provide a constructive procedure to ensure that the regularization stage cost induces strict dissipativity of the OCP. Put differently, we investigate how suitable regularization terms for the DNN training simplify the analysis of the DNN training. Based on an asymptotic reachability assumption, we derive explicit and constructive bounds on the DNN depth—i.e., the number of layers—, while we limit the assumptions on the considered loss functions. To the best of our knowledge, the present paper appears to be the first, which derives explicit depth bounds for DNNs. Additionally, we comment on the application of our results to classification problems. We draw upon the example of the classic Two Spiral Task (TST) to illustrate our findings numerically. Numerical a-posteriori verification indicates that the proposed depth bounds are not overly conservative.

The remainder of the paper is structured as follows: Section 2 introduced the problem statement and present preliminary results. Section 3 presents the main results, while in Section 4 we turn towards a numerical example. The paper concludes with a discussion and conclusions in Section 5.

2. An optimal control formulation of neural network training

We focus on Residual artificial neural Networks (ResNets), not only because they can fulfill classification and regression tasks, e.g., image processing in the form of CNNs, but also because they can be understood as dynamic data filters. Accordingly, they form a bridge between artificial intelligence and control theory. A ResNet can be modeled as the system

$$x_{k+1} = x_k + \sigma(A_k x_k + b_k), \quad x_0 = x^i \in \mathbb{R}^d \tag{1}$$

where the discrete (time) index $k \in \mathbb{N}$ enumerates the layers of the network. The weights $A_k \in \mathbb{R}^{d\times d}$, $b_k \in \mathbb{R}^d$ are chosen in the training process. The initial condition x_0 of the state variable $x \in \mathbb{R}^d$ corresponds to data propagated through the net. Specifically, we consider a given set

$$\mathbb{D} = \{(x^1, y^1), \ldots, (x^D, y^D)\}$$

of D data points (x^i, y^i) where the vectors $y^i \in \mathbb{R}^m$ serve as labels of the training data. We denote the x-projection of $\mathbb{D}$ as $\mathbb{X}$ and, respectively, the y-projection as $\mathbb{Y}$. The sequence of integers $k_0, k_0+1, \ldots, k_1$ is written as $\mathbb{N}_{[k_0,k_1]}$. Notice that, with slight abuse of notation, we write $\sigma(A_k x_k + b_k)$ to denote the element-wise application of the scalar activation function $\sigma : \mathbb{R} \to \mathbb{R}$ to the vector $A_k x_k + b_k$.

2.1. Problem statement

Training a neural net can be understood as an optimization problem wherein, in principle, the depth of the network $N \in \mathbb{N}$ and the width $\dim(x) = d$, the weight sequences $A_k, b_k, k \in \mathbb{N}_{[0,N-1]}$, and the activation function σ are design parameters. Here, we suppose a fixed and constant dimension $\dim(x) = d$, which implies that the number of neurons per layer is kept constant. Primarily, we are interested in deriving a non-conservative constructive bound on the depth N.

Assumption 1 (*Activation Function* σ)**.** The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous on $\mathbb{R}$ and satisfies $\sigma(0) = 0$. □

Similar to Esteve et al. (2020) and Li et al. (2017) we later rewrite the training problem as an OCP. To this end, consider the stacked data vectors

$$\mathbf{x}_0 \doteq \left[x^{1\top}, \ldots, x^{D\top}\right]^\top, \quad \mathbf{y} \doteq \left[y^{1\top}, \ldots, y^{D\top}\right]^\top,$$

which allow writing $\mathbb{D} = \{\mathbf{x}_0, \mathbf{y}\}$. Moreover, the stacked or ensemble variant of the dynamics (1) reads

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \sigma\left((I^D \otimes A_k)\mathbf{x}_k + 1^D \otimes b_k\right), \ \mathbf{x}_0 \in \mathbb{R}^{d\cdot D} \\ &\doteq f(\mathbf{x}_k, u_k), \qquad\qquad \mathbf{x}_0 \in \mathbb{R}^{d\cdot D}\end{aligned} \tag{2}$$

with $u_k \doteq \left[\mathrm{vect}(A_k)^\top, b_k^\top\right]^\top \in \mathbb{R}^{d^2+d}$ and I^D is the identity matrix of $\mathbb{R}^D$ and 1^D is vector of all ones in $\mathbb{R}^D$. The initial condition can be understood as the vectorization of the available data, i.e., $\mathbf{x}_0 = \mathrm{vect}(\mathbb{X})$. For a given depth N the problem can be written as the following discrete-time OCP

$$V_N^\gamma(\mathbf{x}_0) \doteq \min_{\{u_k\}} \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, u_k) + \gamma \cdot \ell_\mathrm{f}(\mathbf{x}_N, \mathbf{y}) \tag{3a}$$

$$\text{subject to } \forall k \in \mathbb{N}_{[0,N-1]}$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k), \ \mathbf{x}_0 = \mathrm{vect}(\mathbb{X}) \in \mathbb{R}^{d\cdot D}. \tag{3b}$$

The Lagrange term (or stage cost) $\ell : \mathbb{R}^{d\cdot D} \times \mathbb{R}^{d^2+d} \to \mathbb{R}_0^+$ captures regularization terms (details see further below), while the Mayer term (or loss function) $\ell_\mathrm{f} : \mathbb{R}^{d\cdot D} \to \mathbb{R}_0^+$ describes the quality of the neural net in terms of the given (classification/learning) task at hand. Typically, one aims at minimizing the empirical loss

$$\ell_\mathrm{f}(\mathbf{x}_N, \mathbf{y}) \doteq \frac{1}{D}\sum_{i=1}^{D} \ell_\mathrm{f}^i(x_N(x^i), y^i), \tag{4}$$

where $\ell_{\mathrm{f}}^i(x_N(x^i), y^i)$ denotes the loss associated to the data sample (x^i, y^i). The scalar penalty parameter $\gamma \in \mathbb{R}^+$ is used to trade-off the importance of the regularization against the loss function ℓ_{f}. The horizon $N \in \mathbb{N}$ corresponds to the depth of the net. Optimal solutions to (3) are denoted as $\mathbf{x}_k^\star(\mathbf{x}_0)$ and $u_k^\star(\mathbf{x}_0)$ where, whenever necessary, the argument $\mathbf{x}_0$ highlights the dependence on the given data $\mathbb{D}$.

The key challenge in solving (3) is that the dimensionality of the data $\mathbf{x}_0$ might be large, while the weights A_k and b_k, collected in $u_k \doteq \left[\mathrm{vect}(A_k)^\top, b_k^\top\right]^\top$, have to provide suitable propagation for all data points x_0^i. Indeed the weights should provide suitable propagation of data points even beyond $\mathbb{X}$ such as to allow using the trained neural net as a predictor. This desired predictive or extrapolating capability is also known as *generalization* in machine learning. A priori it is not clear what a suitable depth N shall be. However, it is well-known—in systems and control as well as in machine learning—that large values of N might lead to over-fitting, which jeopardizes generalization.

2.2. Preliminaries

Throughout the remainder we assume that the loss function $\ell_{\mathrm{f}} : \mathbb{R}^{d\cdot D} \to \mathbb{R}_0^+$ is continuous and non-negative. We require

$$\mathbf{X}^\star(\mathbf{y}) \doteq \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{d\cdot D}} \ \ell_{\mathrm{f}}(\mathbf{x}, \mathbf{y}) \neq \emptyset, \tag{5}$$

i.e., the set of unconstrained minimizers of the loss function ℓ_{f} is non-empty. Moreover, the set $\mathbf{R}_N(\mathbf{x}_0) \subseteq \mathbb{R}^{d\cdot D}$

$$\mathbf{R}_N(\mathbf{x}_0) \doteq \left\{\mathbf{x} \mid \mathbf{x} = \mathbf{x}_N(\mathbf{x}_0),\, \|u_k\| < \infty,\, k \in \mathbb{N}_{[0,N-1]}\right\} \tag{6}$$

collects all states reachable from $\mathbf{x}_0$ within horizon N by bounded control sequences $\{u_k\}$. Similarly, the set $\mathbf{R}_N^\star(\mathbf{x}_0) \subset \mathbf{R}_N(\mathbf{x}_0)$ denotes the set of states reached by optimal control sequences. Similar to Esteve et al. (2020) we initially assume the following:

Assumption 2 (*Zero-Loss DNN*)**.** Given data $\mathbb{D} = \{\mathbf{x}_0, \mathbf{y}\}$ and $N \in \mathbb{N}$ it holds that $\mathbf{R}_N(\mathbf{x}_0) \cap \mathbf{X}^\star(\mathbf{y}) \neq \emptyset$. □

The previous assumption can be understood as a realizability assumption, i.e., for sufficiently deep networks zero loss can be attained for the given data $\mathbb{D}$.

If the reachability assumption above holds, and assuming with only minor loss of generality[2] that

$$\mathbf{x} \in \mathbf{X}^\star(\mathbf{y}) \quad \Leftrightarrow \quad \ell_{\mathrm{f}}(\mathbf{x}, \mathbf{y}) = 0, \tag{7}$$

we can rewrite the problem of attaining zero loss in (3) as follows:

$$V_N(\mathbf{x}_0) \doteq \min_{\{u_k\}} \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, u_k) \tag{8a}$$

$$\text{subject to } \forall k \in \mathbb{N}_{[0,N-1]}$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k), \quad \mathbf{x}_0 = \mathrm{vect}(\mathbb{X}) \tag{8b}$$

$$0 = \ell_{\mathrm{f}}(\mathbf{x}_N, \mathbf{y}), \tag{8c}$$

wherein the loss function is replaced by the terminal equality constraint $0 = \ell_{\mathrm{f}}(\mathbf{x}_N, \mathbf{y})$.

Proposition 1 (*Exact Penalization of Losses*)**.** *Suppose that Assumptions 1 and 2 hold, and let the optimal pairs $(\mathbf{x}_k^*, u_k^*)$ of OCP (8) and $(\mathbf{x}_k^\star, u_k^\star)$ of OCP (3) satisfy second-order sufficient conditions.*[3] *Then, for sufficiently large penalty values $\gamma \geq \gamma^\star$ and $N \in \mathbb{N}$, the optimal state trajectory $\mathbf{x}_k^\star$, $k \in \mathbb{N}_{[0,N]}$ of (3) satisfies $\mathbf{x}_N^\star \in \mathbf{X}^\star(\mathbf{y})$, respectively, $\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) = 0$.* □

Proof. Observe that $\ell_{\mathrm{f}} : \mathbb{R}^{d\cdot D} \to \mathbb{R}_0^+$ is scalar and non-negative, hence $\ell_{\mathrm{f}}(\mathbf{x}_N, \mathbf{y}) = |\ell_{\mathrm{f}}(\mathbf{x}_N, \mathbf{y})|$. Consequently, OCP (3) can be considered as an exact penalty reformulation of OCP (8). Let μ^* be the Lagrange multiplier of (8c) for the optimal solution $(\mathbf{x}_k^*,\ u_k^*)$ of OCP (8). Applying Theorem 14.3.1 given by Fletcher (2013) yields that for $\gamma \geq \gamma^\star = |\mu^\star|$ the assertion holds. ■

Though conceptually interesting, the above result has a number of pitfalls: (a) Proposition 1 relies on second-order sufficient optimality conditions which may not hold for arbitrary loss functions and which induce differentiability requirements, cf. Appendix. (b) Moreover, Proposition 1 does not provide an explicit estimate for the required depth N. (c) Finally, the exact reachability condition of Assumption 2 is quite strong and in general difficult to check. It may even be violated in some cases (Steinberger & Zinner, 2000). For first results on the controllability of neural ODEs we refer to Ruiz-Balet and Zuazua (2023).

3. A dissipativity approach to DNN design

We turn towards an alternative approach, which does not require exact reachability, while it provides an explicit depth estimate. To this end, we recall a definition of dissipativity of OCPs, which can be traced back to Angeli et al. (2012), while the notion of dissipative dynamical systems was coined in the seminal work of Willems (1972).

Recall that a scalar function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ which is strictly increasing and satisfies $\alpha(0) = 0$ is said to belong to class $\mathcal{K}$. If

$$\lim_{s\to\infty} \alpha(s) = \infty$$

the function α is said to belong to class $\mathcal{K}_\infty$. We refer to Kellett (2014) for a compendium of useful properties of these functions.

Definition 1 (*Strict Dissipativity*)**.**

(1) System (2) is said to be *dissipative with respect to a steady-state pair* $(\bar{\mathbf{x}}, \bar{u})$, if there exists a non-negative function $\lambda : \mathbb{X} \to \mathbb{R}_0^+$ such that for all $(\mathbf{x},\ u)$

$$\lambda(f(\mathbf{x}, u)) - \lambda(\mathbf{x}) \leq \ell(\mathbf{x}, u) - \ell(\bar{\mathbf{x}}, \bar{u}). \tag{9a}$$

(2) If, additionally, there exists $\alpha_\ell \in \mathcal{K}$ such that

$$\lambda(f(\mathbf{x}, u)) - \lambda(\mathbf{x}) \leq$$
$$-\alpha_\ell\left(\|(\mathbf{x}, u) - (\bar{\mathbf{x}}, \bar{u})\|\right) + \ell(\mathbf{x}, u) - \ell(\bar{\mathbf{x}}, \bar{u}). \tag{9b}$$

then (2) is said to be *strictly $x-u$ dissipative with respect to* $(\bar{\mathbf{x}},\ \bar{u})$.

(3) If, for all $N \in \mathbb{N}$ and all $\mathbf{x}_0 \in \mathbf{X}_0$, the dissipation inequalities (9) hold along any optimal pair of (3), then *OCP (3) is said to be (strictly) $x-u$ dissipative with respect to* $(\bar{\mathbf{x}},\ \bar{u})$.

(4) Moreover, if (2) or (3) hold with $\alpha_\ell\left(\|(\mathbf{x}, u) - (\bar{\mathbf{x}}, \bar{u})\|\right)$ replaced by $\alpha_\ell\left(\|\mathbf{x} - \bar{\mathbf{x}}\|\right)$, then system (2), respectively, OCP (3) are said to be strictly x dissipative with respect to $(\bar{\mathbf{x}},\ \bar{u})$. □

Observe the fact that for the dynamics (2) any state $\bar{\mathbf{x}}$ constitutes a controlled equilibrium with corresponding $\bar{u} = 0$. Moreover, note that the dissipativity of OCP (3) only depends on the regularization ℓ and not on the loss function ℓ_{f}.

Finally, it is straightforward to show that strict dissipativity with respect to $(\bar{\mathbf{x}},\ \bar{u})$ implies that $(\bar{\mathbf{x}},\ \bar{u})$ is a global steady-state minimizer of ℓ. We remark that the first part of our analysis of Section 3 holds independent of how the regularization stage cost ℓ is designed and thus is independent of how $\bar{\mathbf{x}}$ is chosen. We will detail this after Eq. (10) below.

[2] Indeed as ℓ_{f} is non-negative, we can always consider $\ell_{\mathrm{f}}(\mathbf{x}, \mathbf{y}) - \ell_{\mathrm{f}}(\bar{\mathbf{x}}, \mathbf{y})$ with $\bar{\mathbf{x}} \in \mathbf{X}^\star(\mathbf{y})$ and $\ell_{\mathrm{f}}(\bar{\mathbf{x}}, \mathbf{y}) < \infty$.

[3] The definition of these conditions is standard in nonlinear programming. It is given in Theorem 8 in Appendix.

3.1. Turnpikes in DNN training

Consider the sets

$$\mathcal{Q}_\varepsilon \doteq \left\{k \in \mathbb{N}_{[0,N-1]} \mid \left\|\mathbf{x}_k^\star - \bar{\mathbf{x}}\right\| \leq \varepsilon\right\},$$
$$\widehat{\mathcal{Q}}_\varepsilon \doteq \mathbb{N}_{[0,N-1]} \setminus \mathcal{Q}_\varepsilon,$$

which capture the time points optimal trajectories stay ε-close to $\bar{x}$ and the complement, i.e., the time points optimal trajectories stay outside of an ε-neighborhood of $\bar{\mathbf{x}}$.

Assumption 3 (*Strict Dissipativity of OCP (3)*)**.** For the given data $\mathbb{D} = \{\mathbf{x}_0, \mathbf{y}\}$, with $\mathbf{x}_0 \in \mathbf{X}_0$, OCP (3) is strictly x dissipative with respect to $(\bar{\mathbf{x}}, \bar{u})$. □

While Assumption 2 requires finite-time reachability of some unconstrained minimizer of the loss function, the next assumption defines an exponential reachability property with respect to a specific equilibrium $(\bar{\mathbf{x}}, \bar{u})$. To this end, we distinguish the x-projection of the data $\mathbb{X}$—which corresponds to the initial condition $\mathbf{x}_0$ in the OCPs (3) and (8)—from the set of possible initial conditions $\mathbf{X}_0 \subseteq \mathbb{R}^{d \cdot D}$ of the ensemble dynamics (2). Put differently, we have that $\text{vect}(\mathbb{X}) \in \mathbf{X}_0 \subseteq \mathbb{R}^{d \cdot D}$.

Assumption 4 (*Exponential Reachability*)**.** There exist constants $\rho \in [0, 1)$ and $\beta > 0$, an infinite-horizon control $\tilde{u} : \mathbb{N}_{[0,\infty)} \to \mathbb{R}^{d^2+d}$, and a class $\mathcal{K}$ function $\hat{\alpha}_\ell : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that, for all $\mathbf{x}_0 \in \mathbf{X}_0$,

$$\hat{\alpha}_\ell(\left\|(\tilde{\mathbf{x}}_k, \tilde{u}_k) - (\bar{\mathbf{x}}, \bar{u})\right\|) \leq \beta\rho^k$$

and $\ell(\mathbf{x}, u) \leq \hat{\alpha}_\ell(\|(\mathbf{x}, u) - (\bar{\mathbf{x}}, \bar{u})\|)$ on $\mathbb{R}^D \times \mathbb{R}^{d^2+d}$. □

First we analyze the structure of optimal solutions implied by strict dissipativity, i.e., the next result establishes a turnpike property in OCP (3).

Proposition 2 (*Turnpikes in DNN Training*)**.** *Suppose that Assumptions 3–4 hold. Let λ be bounded on $\mathbf{X}_0$. Then, there exist constants $\Lambda, \hat{V} > 0$ such that, for any $\gamma \in \mathbb{R}$*

- $\#\mathcal{Q}_\varepsilon \geq N - \dfrac{\Lambda + \hat{V}}{\alpha_\ell(\varepsilon)}$, *respectively,* $\#\widehat{\mathcal{Q}}_\varepsilon \leq \dfrac{\Lambda + \hat{V}}{\alpha_\ell(\varepsilon)}$,

where $\#\mathcal{Q}_\varepsilon$ is the cardinality of the set $\mathcal{Q}_\varepsilon$. □

Proof. Without loss of generality, we set $\ell(\bar{\mathbf{x}}, \bar{u}) = 0$. Moreover, from Assumption 4 it follows that there exists $\hat{V} \in \mathbb{R}$ such that $V_N^\gamma(\mathbf{x}_0) \leq \hat{V}$. The strict dissipation inequality (9b) implies that

$$\lambda(\mathbf{x}_N^\star) - \lambda(\mathbf{x}_0) \leq -\sum_{k=0}^{N-1} \alpha_\ell(\left\|\mathbf{x}_k^\star - \bar{\mathbf{x}}\right\|) + \ell(\mathbf{x}_k^\star, u_k^\star).$$

Hence

$$\lambda(\mathbf{x}_N^\star) - \lambda(\mathbf{x}_0) + \sum_{k=0}^{N-1} \alpha_\ell(\left\|\mathbf{x}_k^\star - \bar{\mathbf{x}}\right\|) \leq V_N^\gamma(\mathbf{x}_0) - \gamma\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}).$$

Observe that by assumption the storage is bounded from above on $\mathbf{X}_0$ and from below due to Definition 1. Hence there exists $\Lambda \in \mathbb{R}$, $\lambda(\mathbf{x}_N^\star) - \lambda(\mathbf{x}_0) \geq -\Lambda$. From the above, we obtain

$$(N - \#\mathcal{Q}_\varepsilon)\alpha_\ell(\varepsilon) \leq \sum_{k=0}^{N-1} \alpha_\ell(\left\|\mathbf{x}_k^\star - \bar{\mathbf{x}}\right\|) \leq \Lambda + \hat{V}.$$

Rearranging gives $\#\mathcal{Q}_\varepsilon \geq N - \frac{\Lambda+\hat{V}}{\alpha_\ell(\varepsilon)}$ and $\#\widehat{\mathcal{Q}}_\varepsilon \leq \frac{\Lambda+\hat{V}}{\alpha_\ell(\varepsilon)}$. ■

It is worth to be remarked that the result above establishes a turnpike property not only for $\mathbf{x}_0 = \text{vect}(\mathbb{X})$ but indeed for a set of initial conditions $\mathbf{X}_0$. This in turn implies that small perturbations of the data $\mathbb{D}$ should not affect the statement (provided Assumption 4 holds). So far, our analysis has not made any explicit assumption on the structure of ℓ—besides Assumption 4—which could easily be re-formulated without reference to ℓ. Moreover, the turnpike property established in the previous result does not depend crucially on the considered loss function ℓ_{f} but mostly on the regularization stage cost ℓ. Hence it makes sense to choose ℓ having the considered loss function in mind.

Subsequently, we consider

$$\ell(\mathbf{x}, u) = q\|\mathbf{x} - \bar{\mathbf{x}}\|_{p_x}^{s_x} + r\|u\|_{p_u}^{s_u}, \quad \bar{\mathbf{x}} \in \mathbf{X}^\star(\mathbf{y}_0), \tag{10}$$

with $q, r > 0$. We summarize the hyperparameters of the stage cost ℓ by writing $\pi \doteq (s_x, s_u, p_x, p_u, q, r)$. We say a choice of hyperparameters π is admissible if $q, r > 0$, $s_x, s_u > 1$ and $p_x, p_u \in \mathbb{N}_{[1,\infty]}$. Temporarily we set $s_x = s_u = 2$, for the further analysis. We will see in Proposition 6 that one may want to consider $p_x = \infty$ to reduce the dependence of the actual bounds on the cardinality of the data set $\mathbb{D}$. *That is, henceforth, we consider a strictly convex regularization with respect to some unconstrained minimizer $\bar{\mathbf{x}}$ of ℓ_{f}.*

Lemma 3 (*Strict Dissipativity of OCP (3)*)**.** *Suppose that the regularization stage cost satisfies (10) with $q, r > 1$, that Assumption 4 holds with respect to $(\bar{\mathbf{x}}, \mathbf{0})$ penalized in ℓ. Then system (2) and OCP (3) are strictly $x - u$ dissipative with respect to $(\bar{\mathbf{x}}, \mathbf{0})$. Moreover, $\lambda(\mathbf{x}) \equiv 0$ and*

$$\alpha_\ell(\|(\mathbf{x}, u) - (\bar{\mathbf{x}}, \bar{u})\|) = \nu\ell(\mathbf{x}, u), \quad \nu \in (0, 1].$$ □

The proof follows directly from the available storage characterization of dissipativity (Willems, 1972). It is thus omitted.

3.2. Depth bounds for DNNs

While Proposition 1 leverages an exact penalty function and a finite time reachability condition to ensure zero loss, it fails to provide an estimate on the required depth of the network under weaker (i.e. asymptotic) assumptions. The next result addresses this gap, i.e., it presents a quantitative bound on the network depth ensuring ε-loss. Let $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ denote an ε-neighborhood of $\bar{\mathbf{x}}$.

Theorem 4 (*Upper Bound on DNN Depth N*)**.** *Consider OCP (3) with some user-chosen penalty parameters $\gamma > 0$ and let Assumptions 1 and 4 hold. Suppose that the mixed input-state regularization (10) is used with the parameters $s_x = s_u = 2 = p_x = p_u$ and $q, r > 1$, and that the loss function ℓ_{f} is locally Lipschitz on $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ with constant $L_{\ell_{\mathrm{f}}}$.*

Then, for any desired $\varepsilon > 0$, the optimal solutions of OCP (3) with network depth satisfying

$$N \geq \hat{N}(\varepsilon) \doteq \frac{\hat{V}}{\alpha_\ell(\varepsilon)} = \frac{\beta}{(1-\rho)q\varepsilon^2}, \tag{11a}$$

whereby $\beta, \rho \geq 0$ are the controllability constants from Assumption 4, satisfy

$$\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) \leq \frac{L_{\ell_{\mathrm{f}}}}{\gamma}\varepsilon. \tag{11b}$$

Moreover, it holds that $\text{dist}(\mathbf{x}_N^\star, \mathbf{X}^\star(\mathbf{y})) \leq \varepsilon$. □

It deserves to be noticed this result entails a number of parameters which can be chosen. This includes the penalty parameter γ and the constant ε. The former is used in (3a) to trade-off between the regularization and the loss function. Note that $\gamma > 0$ is needed in the performance estimate (11b). The latter constant

$\varepsilon > 0$ does not affect the optimization. Rather it specifies the size of the neighborhood the optimal solutions shall reach at the last hidden layer $k = N$. Note that in (11a) small ε implies large N. However, (11b) shows that large γ can be used to obtain a reasonable performance even if $\varepsilon \not\approx 0$. Finally note that the controllability constants β, ρ from Assumption 4 appear in (11a). Not surprisingly a small overshoot constant β and $\rho \approx 1$ improve the bound. Likewise values of the weight $q \gg 1$, which penalize the deviation from $\bar{\mathbf{x}}$ in the regularization cost ℓ from (10), reduce the bound $\hat{N}(\varepsilon)$.

Proof. First note that due to Assumption 4 and the choice of the regularizing stage cost (10), we have that the constant $\hat{V}$ in Proposition 2 is given by $\hat{V} = \frac{\beta}{1-\rho} > 0$. Moreover, Lemma 3 gives that

$$\alpha_\ell(\|\mathbf{x} - \bar{\mathbf{x}}\|) \leq \ell(\mathbf{x}, u) = \hat{\alpha}_\ell(\|(\mathbf{x}, u) - (\bar{\mathbf{x}}, \bar{u})\|)$$

and $\Lambda = 0$. As $\bar{u} = 0$ and $s_x = 2$ we have $\alpha_\ell(\varepsilon) = q\varepsilon^2$. Hence the bound of Proposition 2 equates to

$$\#\widehat{\mathcal{Q}}_\varepsilon \leq \frac{\hat{V}}{\alpha_\ell(\varepsilon)} = \frac{\beta}{(1-\rho)q\varepsilon^2}.$$

In other words, the time the optimal solutions can spend outside of the ε-neighborhood of $\bar{\mathbf{x}}$ is bounded from above by $\frac{\beta}{(1-\rho)q\varepsilon^2}$. Hence, for sufficiently large N, there exists $l \in \mathbb{N}_{[0,N-1]}$ for which $\left\|\mathbf{x}_l^\star - \bar{\mathbf{x}}\right\| \leq \varepsilon$. Moreover, observe that if $\mathbf{x}_N^\star \in \mathcal{N}_\varepsilon(\bar{\mathbf{x}})$, then the estimate $\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) \leq \frac{L_{\ell_{\mathrm{f}}}}{\gamma}\varepsilon$ follows immediately. Therefore, we now show $\mathbf{x}_N^\star \in \mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ for sufficiently large N.

As the entire state space $\mathbb{R}^{d\cdot D}$ corresponds to equilibria of (2), for any $\varepsilon \geq 0$, the choice of $A_k = 0$, $b_k = 0$ renders $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ forward invariant under (2). The performance associated to this choice implies the following upper bound on the performance on the truncated horizon $\mathbb{N}_{[l,N]}$

$$\sum_{k=l}^{N-1} \ell(\mathbf{x}_k^\star, u_k^\star) + \gamma\,\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) \leq (N-l)q\varepsilon^2 + \gamma L_{\ell_{\mathrm{f}}}\varepsilon, \tag{12}$$

provided that ℓ_{f} is Lipschitz on $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$. By assumption, if ε is small enough this will be the case.

Now, we distinguish three cases:

(i) the optimal trajectory $\mathbf{x}_k^\star$ leaves $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$;
(ii) $\mathbf{x}_k^\star$ leaves and re-enters $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$;
(iii) $\mathbf{x}_k^\star$ remains in $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$.

Observe that cases of leave-enter-leave can be reduced to enter-leave by considering the final exit time. Moreover, Case (iii) corresponds to the assertion and does not require further analysis.

Case (i): At time $m \geq l$, the optimal trajectory $\mathbf{x}_k^\star$ leaves $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$, i.e., $\left\|\mathbf{x}_k^\star - \bar{\mathbf{x}}\right\| > \varepsilon \quad \forall k \geq m$. This implies the following performance bound $\sum_{k=m}^{N-1} \ell(\mathbf{x}_k^\star, u_k^\star) + \gamma\,\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) > (N-m)q\varepsilon^2 + \gamma L_{\ell_{\mathrm{f}}}\varepsilon$, which in turn contradicts the bound from (12) for $l = m$. Hence, solutions permanently leaving $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ are suboptimal.

Case (ii): At time $m_1 \geq l$, the optimal trajectory $\mathbf{x}_k^\star$ leaves $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$ and at $m_2 = m_1 + \Delta m > m_1 \geq l$ it re-enters. This implies the lower bound on $\mathbb{N}_{[m_1,N]}$

$$\sum_{k=m}^{N-1} \ell(\mathbf{x}_k^\star, u_k^\star) \geq \Delta m q(\varepsilon+\eta)^2 + \sum_{k=m_2}^{N-1} \ell(\mathbf{x}_k^\star, u_k^\star) \tag{13}$$

where $\eta > 0$ corresponds to a lower bounds of the distance of $\mathbf{x}_k^\star$, $k \in \mathbb{N}_{[m_1,m_2)}$ to $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$. For the left hand side of (13) to not exceed the right hand side of (12), the optimal trajectory $\mathbf{x}_k^\star$ has to enter $\mathcal{N}_\delta(\bar{\mathbf{x}})$ with $\delta \leq \varepsilon - \eta$. So either, case (ii) does not happen, or the optimal trajectory $\mathbf{x}_k^\star$ enters $\mathcal{N}_\delta(\bar{\mathbf{x}})$ on $\mathbb{N}_{[N-\tilde{m},N]}$ for some $\tilde{m} \geq m_1 + \Delta m$. Now, repeat the analysis for $\mathcal{N}_\delta(\bar{\mathbf{x}})$. Induction shows that $\mathbf{x}_N^\star(\mathbf{x}_0) \in \mathcal{N}_\varepsilon(\bar{\mathbf{x}})$.

Finally, we have to show that $\mathrm{dist}(\mathbf{x}_N^\star, \mathbf{X}^\star(\mathbf{y})) \leq \varepsilon$. Above we have derived that $\mathbf{x}_N^\star(\mathbf{x}_0) \in \mathcal{N}_\varepsilon(\bar{\mathbf{x}})$. As $\bar{\mathbf{x}} \in \mathbf{X}^\star(\mathbf{y})$, we have that $\|\mathbf{x}_N^\star(\mathbf{x}_0) - \bar{\mathbf{x}}\| \geq \mathrm{dist}(\mathbf{x}_N^\star, \mathbf{X}^\star(\mathbf{y}))$. This concludes the proof. ■

Let $N^\star$ denote the minimal horizon length N for which Assumption 2 holds and recall that $\gamma^\star$ denotes the minimal value of γ from Proposition 1, which ensures exact loss minimization. This allows to formulate the following corollary to Proposition 1 and Theorem 4.

Corollary 5 (*Zero Loss With Finite Depth*). *Let Assumptions 1 – 2 hold, suppose that the mixed input-state regularization (10) is used with $s_x = s_u = 2 = p_x = p_u$ and $q, r > 1$, that the loss function φ is locally Lipschitz on $\mathcal{N}_\varepsilon(\bar{\mathbf{x}})$, and that second-order sufficient conditions for OCP (3) hold at $(\mathbf{x}_k^\star, u_k^\star)$. If $\gamma \geq \gamma^\star$ with $\gamma^\star$ from Proposition 1 and*

$$N \geq \hat{N}(\varepsilon) \doteq \frac{\beta}{(1-\rho)q\varepsilon^2} \geq N^\star$$

holds in OCP (3), with $\beta, \rho \geq 0$ from Assumption 4, then $\mathrm{dist}(\mathbf{x}_N^\star, \mathbf{X}^\star(\mathbf{y})) = 0$. □

The proof follows directly from combining Proposition 1 and Theorem 4. The result shows that if the horizon bound from Theorem 4 is combined with a large value of γ, then one may even achieve zero loss, cf. (7). Observe that these results do not directly depend on the considered loss function ℓ_{f}. Indeed, any loss function for which one is able to solve (7) can be considered.

3.3. Bounds independent of the number of samples

At this point, it is fair to ask how to specify the remaining degrees of freedom in the regularization ℓ from (10)? It is straightforward to see that the choice $s_x = s_u = 2$ is not crucial in the proofs of the results above. On other hand, from a numerical point of view, it is promising to use $p_x = p_u = s_x = s_u = 2$ and $q \gg r = 1$, i.e., a convex quadratic regularization stage cost. Yet, employing the two-norm means that the bound $\hat{N}(\varepsilon)$ will scale with $\dim(\mathbf{x}) = d \cdot D$, i.e., it increases with the number of considered data samples $D = \#\mathbb{D}$. Observe that normalizing the stage cost ℓ with D is not a viable remedy as in this case α_ℓ has to be normalized by D as well. This suggests to consider $p_x = \infty$ in ℓ from (16) as this choice does not scale with $D = \#\mathbb{D}$. However, optimizing over ∞-norm objectives with non-linear equality constraints is not straightforward.

In this context, the next result shows how one may compute an a-posteriori estimate $\hat{N}$ using the ∞-norm, while the training considers a numerically more favorable norm. To this end, let $\ell(\mathbf{x}, u; \pi)$ denote the stage cost (10) with parameters $\pi = (s_x, s_u, p_x, p_u, q, r)$. Any choice with $q, r > 0$, $p_x, p_u > 1$ and $s_x, s_u > 1$ is said to be admissible. Moreover, let $N(\varepsilon; \pi)$ and $V_N^\gamma(\mathbf{x}_0; \pi)$ denote the respective dependence on π.

Proposition 6 (*Transferring Depth Bounds*). *Let $(\tilde{\mathbf{x}}_k, \tilde{u}_k)$ be an optimal solution of OCP (3) with a mixed input-state regularization $\ell(\mathbf{x}, u; \tilde{\pi})$ from (10). Except for the choice of $\tilde{\pi}$, let the conditions of Corollary 5 hold. Then, for any admissible parametrization $\pi = (s_x, s_u, p_x, p_u, q, r)$, the horizon*

$$N \geq \hat{N}(\varepsilon; \pi) \doteq \frac{1}{q \cdot \varepsilon^{s_x}} \sum_{k=0}^{\hat{N}(\varepsilon;\tilde{\pi})} \ell(\hat{\mathbf{x}}_k, \hat{u}_k; \pi) \geq N^\star \tag{14}$$

in OCP (3) ensures $\ell_{\mathrm{f}}(\mathbf{x}_N^\star, \mathbf{y}) = 0$. □

Proof. Observe that the depth bound can be written as $\hat{N}(\varepsilon) \doteq \frac{\hat{V}}{\alpha_\ell(\varepsilon)}$, i.e., as the quotient of an upper bound on the value function $V_N^\gamma(\mathbf{x}_0;\pi)$ with the lower bound on ℓ. The conditions of Corollary 5 imply that $\ell_\mathrm{f}(\hat{\mathbf{x}}_N, \mathbf{y}) = 0$ and hence, for any admissible stage cost parametrization π, the sum $\tilde{V} \doteq \sum_{k=0}^{\hat{N}(\varepsilon;\tilde{\pi})} \ell(\hat{\mathbf{x}}_k, \hat{u}_k;\pi)$ gives an upper bound on the value function $V_N^\gamma(\mathbf{x}_0;\pi)$. Moreover, for any admissible choice π, we have $\alpha_\ell(\varepsilon) = q\|\varepsilon\|_{p_x}^{s_x} = q\cdot\varepsilon^{s_x}$. ■

While at the first glance the above result is of merely technical value, it enables to use numerically favorable choices for the parametrization of ℓ for training purposes, while the depth bounds can rely on the ∞-norm for the state regularization, which does not depend on the cardinality of $\mathbb{D}$, i.e., it is independent on the number of labeled samples.

3.4. Empirical risk minimization and the choice of ε

In principle, the choice $\varepsilon > 0$ may lead to a slight performance degradation in terms of loss minimization. However, a value of ε too small might render the bound of (11a) conservative as $\hat{N} = \infty$ for $\varepsilon \to 0$. Hence, depending on the considered ML task at hand (e.g., classification or regression), it may be advisable to choose ε not too small. In essence, in classification problems the choice of ε is governed by the size of the neighborhood of $\bar{\mathbf{x}} \in \mathbf{X}^\star(\mathbf{y})$ which still allows to classify the data points. To elaborate this, let $g : \mathbb{R}^d \to \mathbb{R}^m$, $y = g(x)$ denote map from the Nth layer of (1) to the label prediction y. That is, $y^\star(x_0^i) = g(x_N^\star(x_0^i))$ is the multi-step propagation of the data point x_0^i through the DNN (1) using optimal weights obtained via OCP (3) concatenated with g.

Suppose, as before, that (7) holds and let

$$\bar{\mathbf{x}} = \left[\bar{x}^{1\top}, \dots, \bar{x}^{i\top}, \dots, \bar{x}^{D\top}\right]^\top,$$

i.e., $\bar{\mathbf{x}}$ achieves perfect classification (= zero loss). Moreover, g may be defined as follows

$$g(x^i) \doteq \begin{cases} \|x^i - \bar{x}^i\| < \delta y = y^i \\ \|x^i - \bar{x}^i\| \geq \delta y \neq y^i \end{cases}, \tag{15}$$

where $y^i \in \mathbb{Y}$ is a finite set of classification labels and $\delta > 0$ is the maximal radius of $\mathcal{N}_\delta(\bar{x}^i)$ which still allows to exactly distinguish the underlying classes for the entire data set $\mathbb{D}$ (provided it exists). This choice of g suggests the loss function for the ith data sample to be $\ell_\mathrm{f}^i(x_N(x^i), y^i) = \|g(x_N^\star(x_0^i)) - y^i\|^{s_x}$, which is not differentiable on $\|x^i - \bar{x}^i\| = \delta$ due to (15). In view of Theorem 4—$\mathrm{dist}(\mathbf{x}_N^\star, \mathbf{X}^\star(\mathbf{y})) \leq \varepsilon$—one may consider the following differentiable substitute

$$\ell_\mathrm{f}^i(x_N(x_0^i), y^i) = \|x_N^\star(x_0^i) - \bar{x}^i\|^2 \tag{16}$$

which reformulates the loss in terms of the squared distance to $\bar{x}^i$. Recall that in the proof of Theorem 4 we have shown that $\mathbf{x}_N^\star(\mathbf{x}_0) \in \mathcal{N}_\varepsilon(\bar{\mathbf{x}})$. Hence the choice $\varepsilon < \delta$ with δ from (15) leads to exact classification of all sample points in $\mathbb{D}$.

Consider the index set

$$\mathcal{I}_N(\mathbb{D}) \doteq \{i \in \{1,\dots,D\} \mid g(x_N^\star(x_0^i)) \neq y^i\}$$

of misclassified samples from $\mathbb{D}$. Then the empirical risk (of misclassification) is defined as

$$\mathcal{R}_N(\mathbb{D}) \doteq \frac{\#\mathcal{I}_N(\mathbb{D})}{\#\mathbb{D}},$$

see Shalev-Shwartz and Ben-David (2014). The next result translates the findings of Theorem 4 to classification with zero empirical risk.

Proposition 7 (*Zero Empirical Risk Classification*). *Consider OCP (3) with $\gamma = 1$ and let Assumptions 1 and 4 hold. Suppose that the mixed input-state regularization (10) is used in same fashion as in Theorem 4 and consider the map g from (15) as well as the quadratic loss ℓ_f^i from (16). Suppose that there exists a $\delta > 0$ which provides perfect classification on the training data $\mathbb{D}$. Then, if $N > \hat{N}(\varepsilon = \delta)$ in OCP (3), it holds that $\mathcal{R}_N(\mathbb{D}) = 0$ for trained network.* □

Proof. The considered setting is such that Theorem 4 holds. Hence if $N > \hat{N}(\varepsilon = \delta)$ is considered, we have that $\|\mathbf{x}_N^\star - \bar{\mathbf{x}}\| \leq \delta$. The definition of g in (15) implies zero empirical risk. This concludes the proof. ■

Naturally, if the depth bounds $\hat{N}$ in (11a) are too conservative, the potential price to pay for zero empirical risk classification as per Proposition 7 are insufficient generalization properties—i.e., a larger (true) risk of misclassification—due to over-fitting. Thus, in Section 4 we will turn towards a numerical example to analyze this aspect.

3.5. Comments and remarks

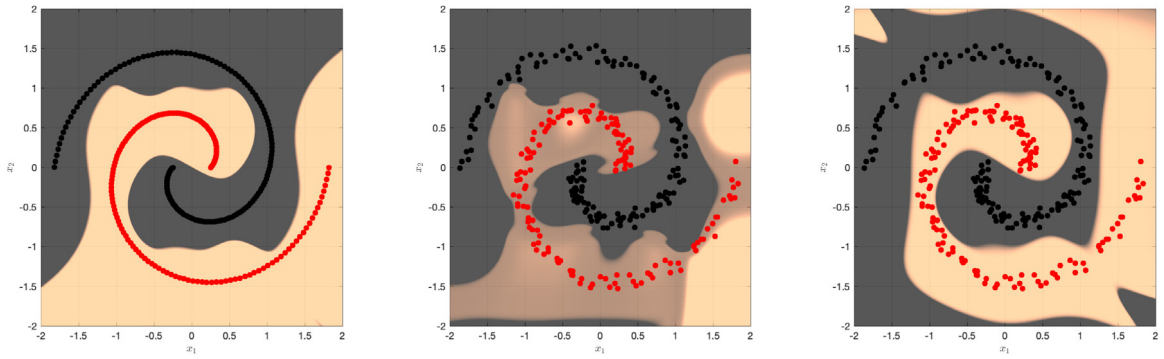
Remark 1 (*Lack of Global Optimality?*). Our formal analysis presented in Sections 3 and 5 works with some global minimizer $\bar{\mathbf{x}}$ of the loss function ℓ_f. Hence the question arises what if the pre-computed $\bar{\mathbf{x}}$ is only a local minimizer of ℓ_f. In this case, provided that Assumptions 1 and 4 hold, all the presented results remain valid. The reason is that the design of the regularization term in the training OCP (3) ensures its strict dissipativity, cf. Lemma 3 if the controllability property from Assumption 4 holds. Note, however, that the performance of the trained network will depend on the considered $\bar{\mathbf{x}}$. The question of the interplay between $\bar{\mathbf{x}}$ and the generalization properties of the trained network remains an open problem at this point. □

Remark 2 (*Extension to Other Architectures?*). Our results presented above raise the question of whether the considered approach towards DNN training can be extended to other architectures beyond ResNets. From the perspective of the information propagation in ResNets are structurally similar to recurrent networks. Indeed, ResNets are a special case of an unfolded recurrent neural network for an input sequence of constant length. A deep ResNet can thus store information across many layers, especially if the residuals are small or zero. This is a direct consequence of the architecture with residual connections that create a kind of "memory" for previous layers, cf. Liao and Poggio (2016) and Sun et al. (2016b). Hence, conceptually, the presented results can be transferred to other DNN designs such recurrent networks and variations thereof with more complicated activation functions such as LTSM or GRU. However, there are two properties of ResNets which simplify the analysis.

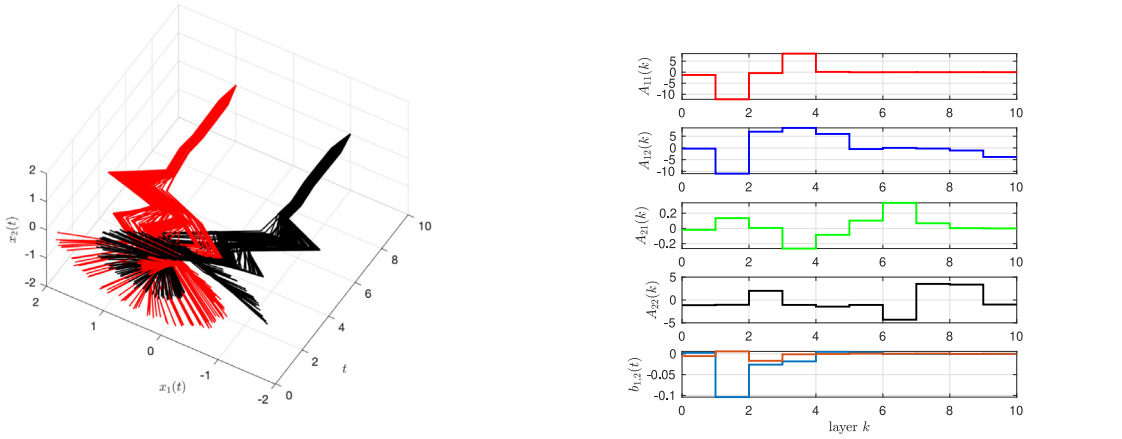
(a) The ResNet dynamics (1) involve a skip connection (the x_k on the right hand side). Considering A_k and b_k as control inputs, one can observe that the entire state-space of (1) is covered by equilibria. This property simplifies the proofs of some of the presented results.
(b) The fact that ResNets can be understood as Euler-forward discretizations of neural ODEs can be used to analyze the controllability properties of ResNets. For first results in this direction we refer to Ruiz-Balet and Zuazua (2023).

Moreover, one may ask if the approach can be further refined for specific loss functions. For first results considering the soft cross-entropy loss function we refer to Püttschneider and Faulwasser (2024). □

Remark 3 (*Links to Other Approaches*). The question of how many neurons are necessary to learn an unknown function with a certain accuracy is of major interest in deep learning. The fact that



(a) $D = 300$, no noise, and $N = 5$. (b) $D = 300$, additive noise, $N = 5$. (c) $D = 300$, additive noise, $N = 10$.



(d) Evolution of data samples for $D = 300$ with noise. (e) Evolution of the network weights for $D = 300$ with noise.

Fig. 1. Results for TST example.

any continuous function defined on a compact set can be practically learned is known as the universal approximation property, which states that a DNN with one hidden layer of sufficiently large width suffices (Cybenko, 1989; Hornik, 1991). For ReLU networks there exist tailored approximation results for depth and width, which often appear to be quite conservative (Petersen & Voigtlaender, 2018; Shen et al., 2022).

In essence, the OCP approach taken in our paper derives depth bounds from controllability assumptions. Our numerical results of Section 4 indicate that this is not overly conservative. One can regard the required controllability property as an assumption that requires that a finitely deep network can solve the learning task at hand. The bottleneck is the verification of the controllability property. Section 4 presents numerical results which use an a-posteriori estimate to this effect. Moreover, the analysis of Ruiz-Balet and Zuazua (2023) indicate the beneficial controllability properties of ResNets.

Moreover, notice that the estimation of bounds on the required network width via the proposed optimal control approach appears to be conceptually difficult. The reason is that the optimal control usually assume a known state dimension, while, from the optimal control perspective width estimation for DNNs means to bound the state dimension. □

4. Numerical example – Two spiral task

For the sake of illustration, we consider the two-dimensional Two Spiral Task (TST) data set. The TST problem refers to a binary classification of data points. As depicted in Figs. 1(a)–1(c), the TST is build upon two intertwined spirals, which are not easy to classify in the given feature space. This allows to check whether the OCP-trained DNN is capable to represent complex decision boundaries (classification) or to learn underlying manifolds (regression). Since its first suggestion (Lang & Witbrock, 1988) this problem has become a standard ML test case (Chalup & Wiklendt, 2007; Wah & Qian, 2000).

The underlying manifold for the 2D learning data generates data samples via

$$x_0^i = \begin{bmatrix} j^i \nu \cos\left(j^i + \varphi_{0c}\right) \\ j^i \nu \sin\left(j^i + \varphi_{0c}\right) \end{bmatrix} + x_{\text{noise}}^i, \tag{17}$$

where j increments angle and radius, $\nu = \frac{1}{4}$ is the TST start radius, and φ_{0c} determines the initial angles. The increments are given by $j^i = \frac{2\pi}{D} i, i = 1, \ldots, D$. The start angle φ_{0c} is used for class assignment, i.e., $\varphi_{01} = 0$ for the first class (red) and $\varphi_{02} = \pi$ for the second class (black), see Fig. 1(a). Except the benchmark, each element of the learning data set $x^i \in \mathbb{D}$ is superposed with uniformly distributed noise $x_{\text{noise}}^i \in \begin{bmatrix} -0.2 & 0.2 \end{bmatrix} \times \begin{bmatrix} -0.2 & 0.2 \end{bmatrix}$. The goal of the classification is to map class one to $y^i = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top$, and class two to $y^i = \begin{bmatrix} -1 & 0 \end{bmatrix}^\top$, known as one hot classification.

4.1. OCP formulation and solution

We consider OCP (3) to train the DNN. We use ℓ from (10) with $\| \cdot \|_2^2$ for $\mathbf{x} - \bar{\mathbf{x}}$ and u, as well as $q = 10^{-2}$ and $r = 10^{-3}$.[4] Simulations are done using CasADi in Matlab (Version

[4] Actually, for the sake of numerical stability, we rescale the inputs by factor 10^2 such that in re-scaled input variables we have $r = 10^{-7}$.

3.55) (Andersson et al., 2019) and IPOPT as NLP solver. The considered loss function is of mean-squared structure (16), where the reference state for regularization coincides with the labels, i.e., $\bar{x}^i = y^i \in \{(1,0)^\top, (-1,0)^\top\}$. Observe that for this loss function the set of minimizers is given by a singleton $\mathbf{X}(\mathbf{y}) = \{\bar{\mathbf{x}} = \mathbf{y}\}$. We set $\gamma = 10^2 D$. The activation function is chosen as $\sigma(s) = \tanh(s)$. Solving the TST example for $N = 10$ and 300 noisy data points from Fig. 1(c), we obtain the solutions depicted in Fig. 1(d) and Fig. 1(e). As one can see, the solution of OCP (3) attains near zero loss. Moreover, the generalization properties of the trained DNN are illustrated in Figs. 1(a), 1(b), and 1(c). For sample data $\mathbb{D}$ has been perturbed by noise. Figs. 1(a), 1(b), and 1(c) depict the decision-boundary as well the classification over a set within the boundaries -2 to 2 for the case where the number of data points $D = 300$ is kept constant. For comparison the unperturbed data and its classification are given as a benchmark in Figs. 1(a). In all examples the decision-boundary remains qualitatively similar, networks with more layers tend to give a better generalization, see the areas, which are unsupported with learning data in Fig. 1(c) or compare to ideal data Fig. 1(a).

4.2. Depth bounds from a-posteriori estimates

As the solution of OCP (3) delivers (near) zero loss on the training data, we now compare three different approaches to a-posteriori depth bounds:

(i) We estimate the exponential constants $\beta, \rho > 0$ appearing in Assumption 4 using the computed optimal trajectories and then we evaluate (11a).
(ii) We use the bound from Proposition 6 in (14) with the two norm.
(iii) We consider (14) for trajectories computed via the two norm but the bound is evaluated in the ∞ norm.

The depth bound from (i) is denoted as $\hat{N}_2(\beta, \rho)$, the one from (ii) as $\hat{N}_2$, and the one from (iii) as $\hat{N}_\infty$. Observe that in the TST example classification is achieved if a data point propagated through the DNN reaches the interior of $\mathcal{N}_{\delta=1}(\bar{\mathbf{x}})$, cf. Proposition 7. Hence we set $\varepsilon = 1$ in evaluating $\hat{N}$ in (14) and (11a). Also note that due to the choice $q = 10^{-2}$ in (10) we have $\alpha_\ell(\varepsilon) = 10^{-2}$ for $\hat{N}_2(\beta, \rho)$ and $\hat{N}_2$.

Approach (i): In a first step we solve the training problem given in OCP (3) as described in Section 4.1. To the end of estimating β and ρ, we then solve the NLP:

$$\min_{\beta,\rho\in\mathbb{R}_0^+} \ \frac{\beta}{1-\rho} \text{ subject to } \ell(\mathbf{x}_k^\star, u_k^\star) \leq \beta\rho^k, \quad k \in \mathbb{N}_{[0,N-1]}.$$

These estimated values of β and ρ are plugged into (11a) to obtain the estimate $\hat{N}_2(\beta, \rho)$.

Approach (ii): We use the same optimal trajectories from above and evaluate the bound from Proposition 6 in (14) to compute $\hat{N}_2$.

Approach (iii): We leverage the result of Proposition 6 to estimate the depth based on the trajectories computed for $\pi = (2, 2, 2, 2, 10^{-2}, 10^{-3})$ in the metric implied by $\pi = (1, 2, \infty, 2, 1, 10^{-3})$. Recall that the ∞-norm penalization of the state deviation is motivated by the ∞ norm not growing with the cardinality of the data set $\mathbb{D}$. To obtain $\hat{N}_\infty$, we evaluate (14) with the computed optimal trajectories.

Table 1 summarizes the constants β and ρ computed via the NLP above as well as the estimated depth bound $\hat{N}_2$ for variations of the network depth (used to solve (3)) and the number of considered data points without noise.

Table 1
Estimated net depths via (11a) for noise-free data.

N	D	β	ρ	$\hat{N}_2(\beta,\rho)$	$\hat{N}_2$	$\hat{N}_\infty$
5	20	0.75	0.61	$2.42 \cdot 10^2$	$1.01 \cdot 10^2$	6.31
5	50	1.85	0.83	$1.11 \cdot 10^3$	$3.21 \cdot 10^2$	6.74
5	100	3.57	0.83	$2.14 \cdot 10^3$	$6.11 \cdot 10^2$	6.98
5	250	8.43	0.67	$2.56 \cdot 10^3$	$1.23 \cdot 10^3$	6.56
5	500	14.4	0.79	$6.83 \cdot 10^3$	$3.36 \cdot 10^3$	7.53
10	20	0.75	0.60	$2.42 \cdot 10^2$	$1.09 \cdot 10^2$	7.00
10	50	2.37	0.68	$7.46 \cdot 10^2$	$3.69 \cdot 10^2$	11.06
10	100	3.08	0.73	$1.14 \cdot 10^3$	$5.34 \cdot 10^2$	8.39
10	250	7.34	0.77	$3.20 \cdot 10^3$	$1.44 \cdot 10^3$	9.83
10	500	19.5	0.58	$4.67 \cdot 10^3$	$2.05 \cdot 10^3$	7.01

Table 2
Estimated net depths via (11a) for noisy data.

N	D	β	ρ	$\hat{N}_2(\beta,\rho)$	$\hat{N}_2$	$\hat{N}_\infty$
5	20	0.87	0.83	$5.24 \cdot 10^2$	$1.45 \cdot 10^2$	7.42
5	50	1.48	0.83	$8.88 \cdot 10^2$	$2.88 \cdot 10^2$	7.20
5	100	3.67	0.83	$2.20 \cdot 10^3$	$6.27 \cdot 10^2$	7.15
5	250	8.83	0.83	$5.30 \cdot 10^3$	$1.58 \cdot 10^3$	8.57
5	500	15.9	0.83	$9.53 \cdot 10^3$	$3.56 \cdot 10^3$	8.40
10	20	1.01	0.75	$4.02 \cdot 10^2$	$1.82 \cdot 10^2$	11.50
10	50	1.65	0.68	$5.07 \cdot 10^2$	$2.35 \cdot 10^2$	8.02
10	100	2.86	0.79	$1.34 \cdot 10^3$	$5.89 \cdot 10^2$	11.38
10	250	7.91	0.72	$2.80 \cdot 10^3$	$1.11 \cdot 10^3$	8.50
10	500	13.8	0.83	$7.94 \cdot 10^3$	$3.30 \cdot 10^3$	18.75

As one can see, the obtained a-posteriori depth estimates $\hat{N}_2(\beta, \rho)$ range from 242 to 6830 if the two norm is used. Observe that the bound $\hat{N}_2(\beta, \rho)$ grows with increasing cardinality of the data set $\#\mathbb{D} = D$. The bound $\hat{N}_2$ which relies on (14) is only marginally better ranging from 101 to 3360. In contrast the bound $\hat{N}_\infty$, which is based on the ∞-norm state penalization in ℓ, delivers much smaller estimates ranging from 6.31 to 11.06. The trends on $\hat{N}_2(\beta, \rho)$ and $\hat{N}_2$ remain unchanged in Table 2 wherein noisy data sets $\mathbb{D}$ are considered. The range of $\hat{N}_2(\beta, \rho)$ spans 524 to 7940 and $\hat{N}_2$ is between 145 and 3560. In contrast, $\hat{N}_\infty$ spans from 7.20 to 18.8. Again the ∞-norm state penalization in ℓ delivers much smaller estimates. Observe that the trend of $\hat{N}$ growing with D is much less pronounced for $\hat{N}_\infty$. The average of values is centered around $7 - 8$. The outliers > 11 and 18.75 are likely due to the training problem (3) being solved to local optimality. Finally, the trend that for increasing number of samples D the estimated bounds $\hat{N}_2$, but also $\hat{N}_\infty$, increase, indicates that the reachability properties of the ensemble dynamics (2) are affected by the dimensionality of the stacked system state $\dim(\mathbf{x}) = D$. Yet, in view of the generalization plot in Fig. 1(c), we conclude that the reachability of $\bar{\mathbf{x}} \in \mathbf{X}^\star(\mathbf{y})$ is sufficient but not necessary for classification. This is also in line with the insights of Proposition 7.

5. Discussion and conclusions

This paper has taken steps towards deriving constructive depth bounds for deep neural networks via turnpike and dissipativity theory. This work is focused on the estimation of depth bounds for DNNs, not the development of a particular learning algorithm. The example of the previous section has primarily assessed the quality of the depth bound derived in Theorem 4. The a-posteriori results of Tables 1 and 2 indicate that the structured design of DNNs is accessible through analysis techniques derived in context of turnpike properties of optimal control problems. Our results do not exploit particular structures of the considered loss function. Instead our approach combines reachability properties with the design of tailored regularization terms in the objective.

The design of these regularization terms requires to solve an unconstrained optimization problem over all available data to obtain a suitable target point. This problem is, however, numerically straightforward compared to training an entire neural network. Indeed, for ResNets the stationary minimization (5) is an unconstrained optimization problem of manageable dimension. Moreover, we have shown how one can use numerically favorable squared two norms in the regularization stage cost, while building the depth estimates via the ∞ norm to avoid scaling with the number of data points. However, several issues require further and future research.

Arguably one of the most interesting questions is how to extend the approach *from the a-posteriori computation of β and ρ* based on a trained DNN *to the a-priori prediction* based on a given loss function ℓ_{f}, the activation function σ, and available data? Indeed, the results from Tables 1 and 2 indicate that estimates could be obtained by considering only parts of the data $\mathbb{D}$, similarly to stochastic gradient techniques, which also rely only on partial data sets.

One key feature driving the success of DNNs in ML applications are their generalization properties, i.e., DNNs provide reasonable classification/regression capabilities for data points not contained in the training data set $\mathbb{D}$. Yet, it is not fully clear how to design the loss function ℓ_{f} and the stage cost regularization ℓ to foster generalization. This also requires further investigation. As mentioned at the end of Section 4, reachability of $\bar{\mathbf{x}}$ is sufficient for classification but not necessary. Hence there is evident need for further analysis on the choice of $\bar{\mathbf{x}}$ or on the choice of more general formulations of ℓ. Intuitively, the OCP formulation of the training problem also suggests the analysis of robustness properties of solutions to (1).

Finally, the presented numerical results underpin the potential of systems and control approaches towards the analysis of DNN training via OCPs. Evidently, further numerical examples with other loss functions, larger data sets, larger data dimension, and considering established stochastic optimization methods (e.g. stochastic gradient methods which only consider a subset of the training data in each optimization step) are of interest and thus subject of future work. For first extensions towards other loss functions we refer to Püttschneider and Faulwasser (2024).

CRediT authorship contribution statement

Timm Faulwasser: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Arne-Jens Hempel:** Writing – review & editing, Visualization, Software, Methodology, Conceptualization. **Stefan Streif:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Second-order sufficient conditions

Proposition 1 requires second-order sufficient condition to hold for (8). For the sake of self-containedness, we recall this standard concept from nonlinear optimization below.

Consider an equality-constrained nonlinear program

$$\min_{z} \ \ell(z) \quad \text{subject to} \quad h(z) = 0 \tag{A.1}$$

where $z \in \mathbb{R}^{n_z}$ is the decision vector. The functions $\ell : \mathbb{R}^{n_z} \to \mathbb{R}$ and $h : \mathbb{R}^{n_z} \to \mathbb{R}^{n_h}$ are at least twice continuously differentiable. Define the Lagrangian

$$\mathcal{L}(z, \nu) = \ell(z) + \nu^\top h(z)$$

and let $\mathcal{H}(z)$ be its Hessian at a point z. Furthermore, let $z^\star$ and the corresponding multiplier vector of the equality constraints, $\nu \in \mathbb{R}^{n_h}$, satisfy

$$\nabla_z L(z^\star) = 0 \qquad \text{and} \qquad h(z^\star) = 0. \tag{A.2}$$

Consider the index set $\mathcal{J} = \{1, \dots, n_h\}$ and the set of directions

$$G^\star(z) = \left\{ s \in \mathbb{R}^{n_z} \,|\, s \neq 0, \nabla h_j(z)^\top s = 0, j \in \mathcal{J} \right\}.$$

The following result recalls well-known sufficient conditions certifying a local minimizer of the equality constrained NLP (A.1), see, e.g., Fletcher (2013) or Chachuat (2009).

Theorem 8 (*Second-order Sufficient Conditions*). *Let $z^\star$ and $\nu \in \mathbb{R}^{n_h}$ with $\nu \neq 0$ satisfy (A.2). If*

$$s^\top \mathcal{H}(z^\star) s > 0 \text{ for all } \quad s \in G^\star(z^\star) \subseteq \mathbb{R}^{n_z}$$

holds, then $z^\star$ is a strict local minimizer of (A.1). □

References

Andersson, J., Gillis, J., Horn, G., Rawlings, J., & Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), 1–36.

Angeli, D., Amrit, R., & Rawlings, J. (2012). On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, *57*(7), 1615–1626.

Antoran, J., Allingham, J., & Miguel, H.-L. J. (2020). Depth uncertainty in neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems*: *vol. 33*, (pp. 10620–10634). Curran Associates, Inc..

Ba, J., & Caruana, R. (2014). Do deep nets really need to be deep?. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*: *vol. 27*, (pp. 2654–2662). Curran Associates, Inc.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, *18*(1), 5595–5637.

Bryson, A., & Denham, W. (1962). A steepest-ascent method for solving optimum programming problems. *Journal of Applied Mechanics*, *29*, 247–257.

Carlson, D., Haurie, A., & Leizarowitz, A. (1991). *Infinite horizon optimal control*. Springer Verlag: Deterministic and Stochastic Systems.

Chachuat, B. (2009). *Nonlinear and dynamic optimization*. EPFL: From Theory to Practice.

Chalup, S. K., & Wiklendt, L. (2007). Variations of the two-spiral task. *Connection Science*, *19*(2), 183–199.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314.

Dorfman, R., Samuelson, P., & Solow, R. (1958). *Linear programming and economic analysis*. McGraw-Hill, New York.

Esteve, C., Geshkovski, B., Pighin, D., & Zuazua, E. (2020). Large-time asymptotics in deep learning. arXiv preprint arXiv:2008.02491.

Faulwasser, T., & Grüne, L. (2022). Turnpike properties in optimal control: An overview of discrete-time and continuous-time results. In E. Zuazua, & E. Trelat (Eds.), *vol. 23, Handbook of numerical analysis* (pp. 367–400). Elsevier: Chap. 11, arxiv:2011.13670.

Faulwasser, T., Grüne, L., & Müller, M. (2018). Economic nonlinear model predictive control: Stability, optimality and performance. *Foundations and Trends in Systems and Control*, *5*(1), 1–98.

Faulwasser, T., Korda, M., Jones, C., & Bonvin, D. (2017). On turnpike and dissipativity properties of continuous-time optimal control problems. *Automatica*, *81*, 297–304.

Faulwasser, T., & Murray, A. (2020). Turnpike properties in discrete-time mixed integer optimal control. *IEEE Control Systems Letters*, *4*, 704–709.

Fletcher, R. (2013). *Practical methods of optimization*. John Wiley & Sons.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Grüne, L. (2022). Dissipativity and optimal control: Examining the turnpike phenomenon. *IEEE Control Systems Magazine*, *42*(2), 74–87.

Grüne, L., & Müller, M. (2016). On the relation between strict dissipativity and turnpike properties. *Systems & Control Letters*, *90*, 45–53.

Gugat, M., Trélat, E., & Zuazua, E. (2016). Optimal Neumann control for the 1D wave equation: Finite horizon, infinite horizon, boundary tracking terms and the turnpike property. *Systems & Control Letters*, *90*, 61–70.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, [ISSN: 0893-6080] *4*(2), 251–257.

Kellett, C. (2014). A compendium of comparison function results. *Mathematics of Control, Signals, and Systems*, *26*(3), 339–374.

Lang, K. J., & Witbrock, M. J. (1988). Learning to tell two spirals apart. In *Proceedings of the 1988 connectionist models summer school* (1989), (pp. 52–59). San Mateo.

Li, Q., Chen, L., Tai, C., & Weinan, E. (2017). Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, *18*(1), 5998–6026.

Liao, Q., & Poggio, T. A. (2016). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv, arXiv:1604.03640.

Maclaurin, D., Duvenaud, D., & Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. In F. Bach, & D. Blei (Eds.), *Proceedings of machine learning research*: *vol. 37*, *Proceedings of the 32nd international conference on machine learning* (pp. 2113–2122). PMLR.

McKenzie, L. (1976). Turnpike theory. *Econometrica*, *44*(5), 841–865.

Petersen, P., & Voigtlaender, F. (2018). Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, [ISSN: 0893-6080] *108*, 296–330.

Püttschneider, J., & Faulwasser, T. (2024). On dissipativity of cross-entropy loss in training ResNets. arXiv preprint arXiv:2405.19013.

Ruiz-Balet, D., Affili, E., & Zuazua, E. (2022). Interpolation and approximation via momentum resnets and neural ODEs. *Systems & Control Letters*, *162*, Article 105182.

Ruiz-Balet, D., & Zuazua, E. (2023). Neural ODE control for classification, approximation, and transport. *SIAM Review*, *65*(3), 735–773.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning*. Cambridge University Press: From Theory to Algorithms.

Shen, Z., Yang, H., & Zhang, S. (2022). Optimal approximation rate of ReLU networks in terms of width and depth. *Journal de Mathematiques Pures et Appliquees*, [ISSN: 0021-7824] *157*, 101–135.

Sontag, E., & Sussmann, H. (1997). Complete controllability of continuous-time recurrent neural networks. *Systems & Control Letters*, *30*(4), 177–183.

Steinberger, T., & Zinner, L. (2000). *Complete controllability of discrete-time recurrent neural networks*: *Tech. rep*, TU Vienna.

Sun, S., Chen, W., Wang, L., Liu, X., & Liu, T.-Y. (2016). On the depth of deep neural networks: A theoretical view. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Sun, S., Chen, W., Wang, L., Liu, X., & Liu, T.-Y. (2016). On the depth of deep neural networks: A theoretical view. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Wah, B. W., & Qian, M. (2000). Constrained formulations for neural network training and their applications to solve the two-spiral problem. In *Proceedings of the 5th international conference on computer science and informatics and 5th joint conference on information sciences, vol. 5* (p. 598).

Willems, J. (1972). Dissipative dynamical systems part I: General theory. *Archive for Rational Mechanics and Analysis*, *45*(5), 321–351.