Optimal Control and Deep Learning – Turnpikes, Dissipativity and Early Exits

Timm Faulwasser

Institute of Control Systems, Hamburg University of Technology

Joint work with

Jens Püttschneider (TUHH), Simon Heilig & Asja Fischer (RU Bochum),

Stefan Streif (TU Chemnitz) and Jens-Arne Hempel (BA Sachsen)

Workshop on Structured Learning @ ALU Freiburg

timm.faulwasser@ieee.org

FOR 5785 ALeSCo Active Learning for Systems and Control

Transregio 391

Machine learning → Control ?

Machine learning

- Regression
- Support vector machines
- Gaussian Processes & kernel methods
- Neural networks
- …

Systems & control

- System identification
- Controller approximation
- Data-driven and adaptive control
- Fault detection
- State estimation
- …

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 1, NO. 1, MARCH 1990

Identification and Control of Dynamical Systems Using Neural Networks

KUMPATI S. NARENDRA FELLOW, IEEE, AND KANNAN PARTHASARATHY

Pergamon

0005–1098(95)00044–5

Automatica, Vol. 31, No. 10, pp. 1443–1451, 1995

0005-1098/95 $9.50 + 0.00

Brief Paper

A Receding-horizon Regulator for Nonlinear Systems and a Neural Approximation*

T. PARISINI† and R. ZOPPOLI†

ELSEVIER

Automatica

Volume 172, February 2025, 112006

Meta-learning for model-reference data-driven control ☆

Riccardo Busetto [a], Valentina Breschi [b], Simone Formentin [a]

Control → Machine learning ?

Complete Controllability of
Continuous-Time Recurrent Neural Networks*

Eduardo Sontag
Dept. of Mathematics, Rutgers University
New Brunswick, NJ 08903
sontag@control.rutgers.edu

Héctor Sussmann
Dept. of Mathematics, Rutgers University
New Brunswick, NJ 08903
sussmann@hamilton.rutgers.edu

Systems & Control Letters 22 (1994) 235–244
North-Holland

State observability in recurrent neural networks*

Francesca Albertini** and Eduardo D. Sontag

Math. Control Signals Systems (1989) 2: 303–314

Mathematics of Control, Signals, and Systems

Approximation by Superpositions of a Sigmoidal Function*

G. Cybenko†

IEEE CSS IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 68, NO. 5, MAY 2023

Universal Approximation Power of Deep Residual Neural Networks Through the Lens of Control

Paulo Tabuada, *Fellow, IEEE*, and Bahman Gharesifard, *Member, IEEE*

Systems & control

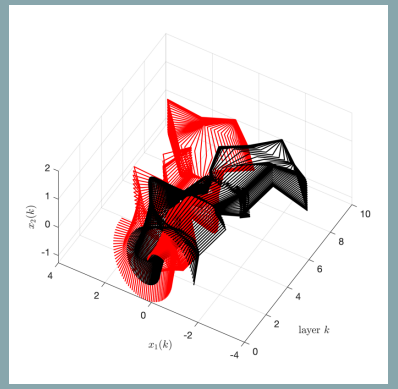
- Analysis
  - Stability, controllability, …
- Synthesis
  - Feedback & feedforward controls
- Computational concepts
- …

Machine learning

?

Overview

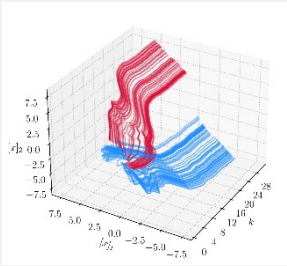
Optimal Control & Deep Learning?



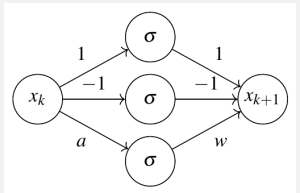
Dissipativity & ResNet Training?

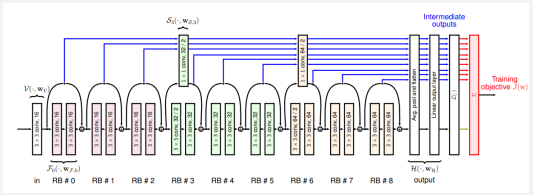
$$S(f(\mathbf{x},u)) - S(\mathbf{x}) \leq \tilde{\ell}_{\mathrm{f}}(\mathbf{x},u) + r\|u\|_2^2 - \mathrm{dist}\left((\mathbf{x},u),\mathbb{Z}^\star\right)$$

Turnpikes in ResNet Training & Experiments

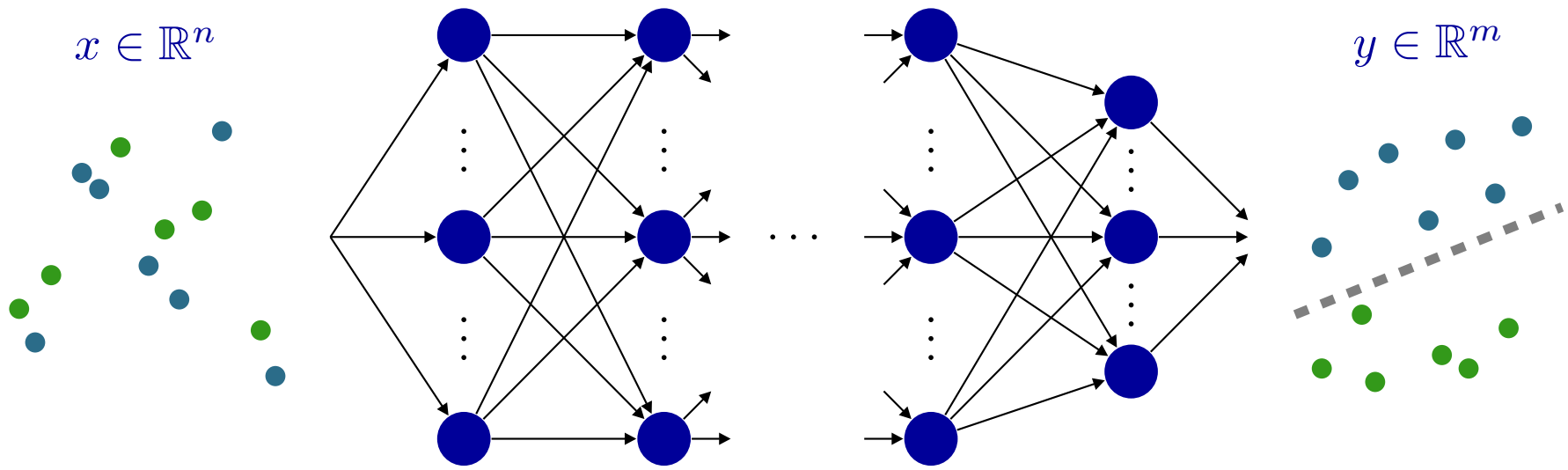


Extension to Other Architectures





Here – Supervised learning with ResNets



Task – Supervised learning

- Data: $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, M\}$
- Learn a model $f : x \mapsto y$
- Classification: $y^i \in \mathbb{N}^m$

Architecture – ResNet

$$v(k+1) = v(k) + \sigma(A(k)v(k) + b(k))$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \sigma(A(t)v(t) + b(t))$$

He et al. (2015); …

Optimal control for deep learning?

Data propagation:

$$
\begin{aligned}
\forall i \in \{1, \dots, M\}, k \in \{0, \dots, N-1\} \\
v_i(k+1) &= v_i(k) + \sigma(A(k)v_i(k) + b(k)) \\
v_i(0) &= x_i \\
\hat{y}_i &= h(v_i(N; x_i))
\end{aligned}
$$

x_i data points = initial condition y_i labels = targets $A(k)$ weights & $b(k)$ biases = $u(k)$ inputs

Optimal control for deep learning?

$$\min_{A(\cdot),b(\cdot)} \frac{1}{M} \sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(N; x_i)), y_i)$$

} training objective

$$\begin{aligned}
\text{subject to} \quad & \forall i \in \{1, \dots, M\}, k \in \{0, \dots, N-1\} \\
v_i(k+1) &= v_i(k) + \sigma(A(k)v_i(k) + b(k)) \\
v_i(0) &= x_i \\
\hat{y}_i &= h(v_i(N; x_i))
\end{aligned}$$

} ensemble dynamics

x_i data points = initial condition y_i labels = targets $A(k)$ weights & $b(k)$ biases = $u(k)$ inputs

Optimal control for deep learning?

$$\min_{A(\cdot),b(\cdot)} \frac{1}{M}\sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(N;x_i)),y_i)$$

subject to $\quad \forall i \in \{1,\dots,M\}, k \in \{0,\dots,N-1\}$

$$v_i(k+1) = v_i(k) + \sigma(A(k)v_i(k) + b(k))$$
$$v_i(0) = x_i$$
$$\hat{y}_i = h(v_i(N;x_i))$$

ResNet training = optimal control

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \rho\cdot\|\mathrm{vect}(A(k),b(k))\|^2 + \mathbf{l}_{\mathrm{f}}(\mathbf{x}(N),\mathbf{y})$$

subject to

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I\otimes A(k))\mathbf{x}(k) + \mathbf{1}\otimes b(k)\right)$$
$$\mathbf{x}(0) = \begin{bmatrix} x_1 & \dots & x_M \end{bmatrix}^\top$$

x_i data points = initial condition $\quad$ y_i labels = targets $\quad$ $A(k)$ weights & $b(k)$ biases = $u(k)$ inputs

Optimal control for deep learning?

$$\min_{A(\cdot),b(\cdot)} \frac{1}{M} \sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(N; x_i)), y_i)$$

subject to $\quad \forall i \in \{1, \dots, M\}, k \in \{0, \dots, N-1\}$

$$v_i(k+1) = v_i(k) + \sigma(A(k)v_i(k) + b(k))$$
$$v_i(0) = x_i$$
$$z_i = h(v_i(N; x_i))$$

Common practise:
regularizing stage cost / weight decay

ResNet training = optimal control

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2 + \mathbf{l}_{\mathrm{f}}(\mathbf{x}(N), \mathbf{y})$$

subject to

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$$
$$\mathbf{x}(0) = \begin{bmatrix} x_1 & \dots & x_M \end{bmatrix}^\top$$

x_i data points = initial condition $\quad$ y_i labels = targets $\quad$ $A(k)$ weights & $b(k)$ biases = $u(k)$ inputs

- Training of deep neural networks $\Leftrightarrow$ Optimal control! (LeCun 1988; Li et al. 2017; . . .)
- **Now**: Design of training problem? Analysis?

Training from the dynamic systems perspective

Recursion over network layers

$$v(k+1) = v(k) + \sigma(A(k)v(k) + b(k))$$

Euler-forward integration
$h = 1$ and $t = h \cdot k$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \sigma(A(t)v(t) + b(t))$$

NN topology & numerical integration?

Continuous-time training formulation

$$V_T(\mathbf{x}) = \min_{A(\cdot),b(\cdot)} \sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(T; x_i)), y_i)$$

subject to $\forall i \in \{1, \ldots, M\}$

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sigma(A(t)v_i(t) + b(t)), \quad t \in [0, T]$$
$$v_i(0) = x_i$$
$$y = h(v_i(T; x_i))$$

- Advantages?
- Insights?

Training from the dynamic systems perspective

Continuous-time training formulation

$$V_T(\mathbf{x}) = \min_{A(\cdot),b(\cdot)} \sum_{i=1}^{M} \ell_f(h(v_i(T; x_i)), y_i)$$

subject to $\forall i \in \{1, \dots, M\}$

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sigma(A(t)v_i(t) + b(t)), \quad t \in [0, T]$$
$$v_i(0) = x_i$$
$$y = h(v_i(T; x_i))$$

Optimality conditions

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sigma(A(t)v_i(t) + b(t)),$$
$$v_i(0) = x_i, \ \forall i \in \{1, \dots, M\}$$
$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}t} = -\left(\frac{\partial}{\partial v_i}\sigma(A(t)v_i(t) + b(t))\right)^\top \lambda_i,$$
$$\lambda_i(T) = \frac{\partial}{\partial v_i}\left(\ell_f(h(v_i(T; x_i))), y_i\right)$$
$$0 = \nabla_{A,b}\left(\lambda^\top \sigma(Av + b)\right)$$

Gradient of optimal value function

$$\nabla_{\mathbf{x}} V_T(\mathbf{x}) = \lambda(0) = [\lambda_i(0)] \qquad \frac{\partial V_T}{\partial b} = \frac{\partial V_T}{\partial x}\frac{\partial x}{\partial b}$$

back propagation ⇕ adjoint sensitivity computation

Compute by backward integration

$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}t} = -\left(\frac{\partial}{\partial v_i}\sigma(A(t)v_i(t) + b(t))\right)^\top \lambda_i, \quad \lambda_i(T) = \frac{\partial}{\partial v_i}\left(\ell_f(h(v_i(T; x_i))), y_i\right)$$

LeCun (1988) A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*
Bryson, A. and Denham, W. (1962). A steepest ascent method for solving optimum programming problems. Journal of Applied Mechanics, 29(2):247

Training from the dynamic systems perspective

Continuous-time training formulation

$$V_T(\mathbf{x}) = \min_{A(\cdot),b(\cdot)} \sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(T; x_i)), y_i)$$

subject to $\forall i \in \{1, \dots, M\}$

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sigma(A(t)v_i(t) + b(t)), \quad t \in [0, T]$$
$$v_i(0) = x_i$$
$$y = h(v_i(T; x_i))$$

Optimality conditions

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sigma(A(t)v_i(t) + b(t)),$$
$$v_i(0) = x_i, \ \forall i \in \{1, \dots, M\}$$
$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}t} = -\left(\frac{\partial}{\partial v_i}\sigma(A(t)v_i(t) + b(t))\right)^\top \lambda_i,$$
$$\lambda_i(T) = \frac{\partial}{\partial v_i}\left(\ell_{\mathrm{f}}(h(v_i(T; x_i))), y_i\right)$$
$$0 = \nabla_{A,b}\left(\lambda^\top \sigma(Av + b)\right)$$

Machine Learning

- Regularization with $\|(A, b)\|^2$?
- Back propagation
- Approximation error?
- Network topology?

Optimal Control

- Avoid singular arcs via $\|(A, b)\|^2$
- Adjoint equations
- Reachability
- Numerical integration

Optimal control for deep learning?

$$\min_{A(\cdot),b(\cdot)} \frac{1}{M} \sum_{i=1}^{M} \ell_{\mathrm{f}}(h(v_i(N; x_i)), y_i)$$

$$\begin{aligned}
\text{subject to} \quad & \forall i \in \{1, \dots, M\}, k \in \{0, \dots, N-1\} \\
v_i(k+1) &= v_i(k) + \sigma(A(k)v_i(k) + b(k)) \\
v_i(0) &= x_i \\
z_i &= h(v_i(N; x_i))
\end{aligned}$$

ResNet training = optimal control

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2 + \mathbf{l}_{\mathrm{f}}(\mathbf{x}(N), \mathbf{y})$$

subject to

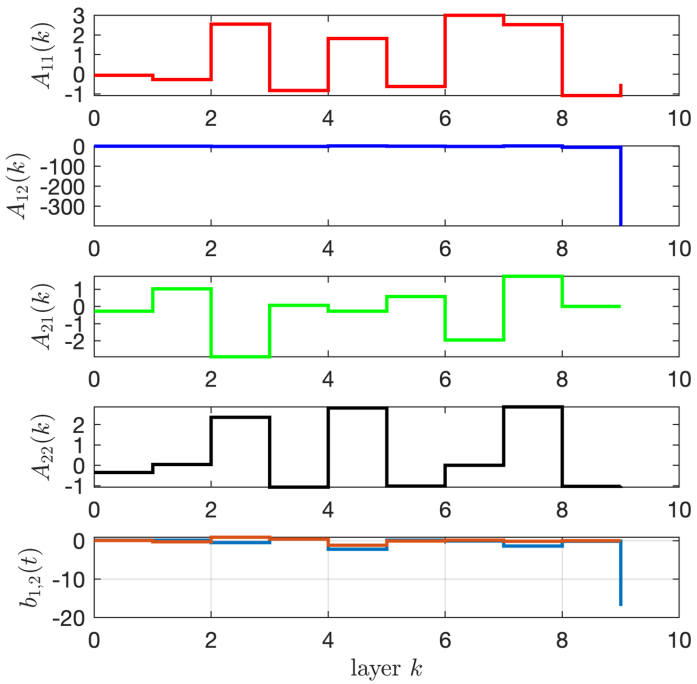
$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right) \\
\mathbf{x}(0) &= \begin{bmatrix} x_1 & \dots & x_M \end{bmatrix}^\top
\end{aligned}$$

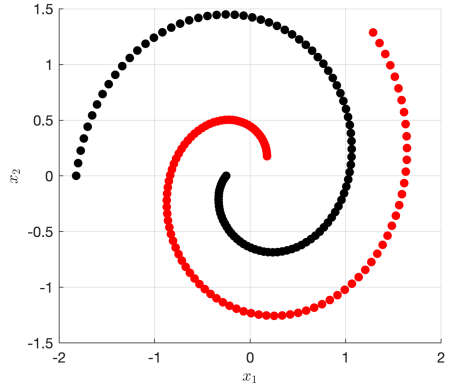
x_i data points = initial condition $\quad$ y_i labels = targets $\quad$ $A(k)$ weights & $b(k)$ biases = $u(k)$ inputs

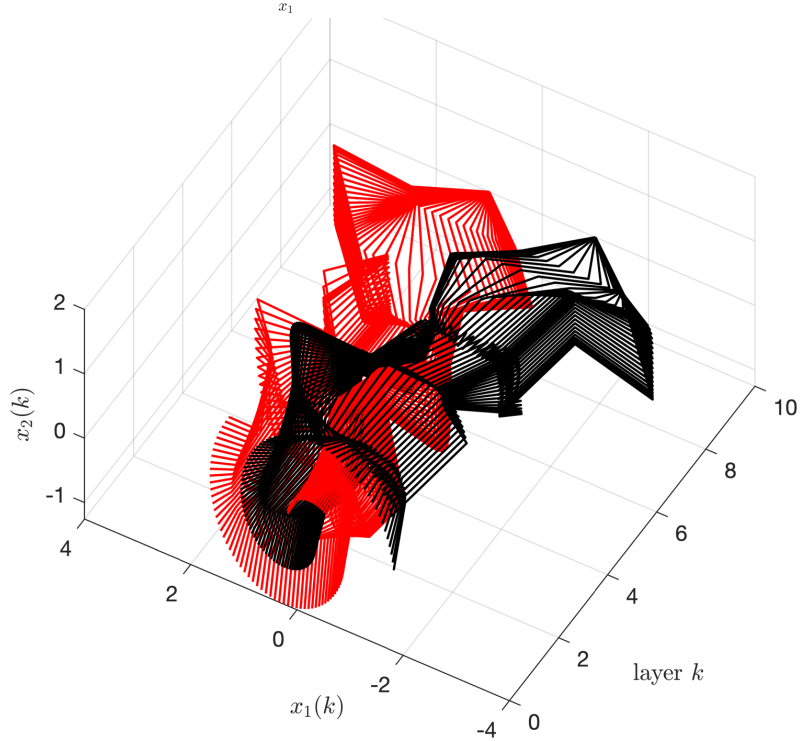
- Training of deep neural networks $\Leftrightarrow$ Optimal control! (LeCun 1988; Li et al. 2017; . . .)
- **Now**: Design of training problem? Analysis?

Let's start with a toy problem …

- Spiral data set with 200 samples
- Binary classification of $2D$ data
- NN with depth 10
- Loss $\ell_{\mathrm{f}}(v(N;x_i);y_i) = \|x_i - y_i\|^2$, $y_i \in \{(-1,0),\,(1,0)\}$







… and turn it into a teaser

ResNet training

design?

given

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \underbrace{\mathbf{l}(\mathbf{x}(k)) + \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2}_{\doteq \mathbf{l}(\mathbf{x}(k),\mathbf{u}(k))} + \gamma \cdot \mathbf{l}_\mathrm{f}(\mathbf{x}(N), \mathbf{y})$$

subject to $\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$

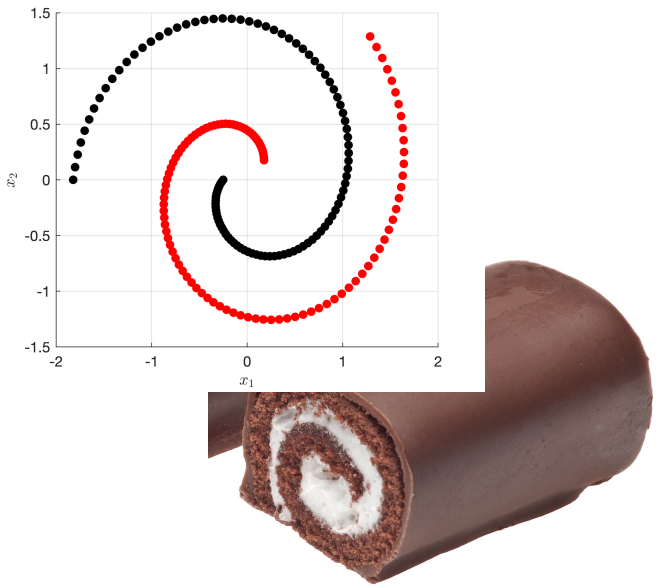
Compute set of optimal *steady states*

$$\bar{\mathbf{x}} \in \mathbb{X}^\star(\mathbf{y}) \doteq \operatorname*{argmin}_{\mathbf{x}} \mathbf{l}_\mathrm{f}(\mathbf{x}, \mathbf{y}) \neq \emptyset$$

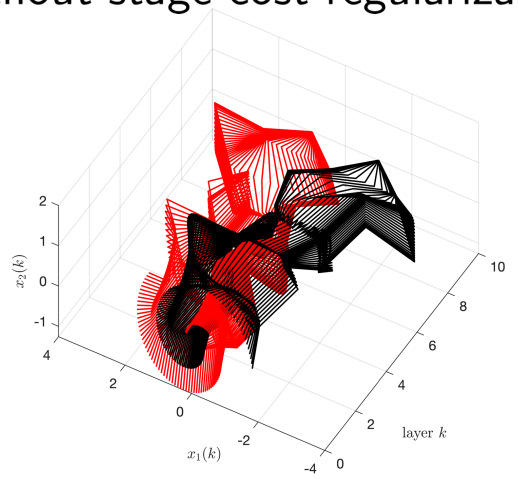
ℓ^2 stage cost

$$\mathbf{l}(\mathbf{x}, \mathbf{u}) = \|\mathbf{x} - \bar{\mathbf{x}}\|_Q^2 + \|\mathbf{u}\|_R^2$$

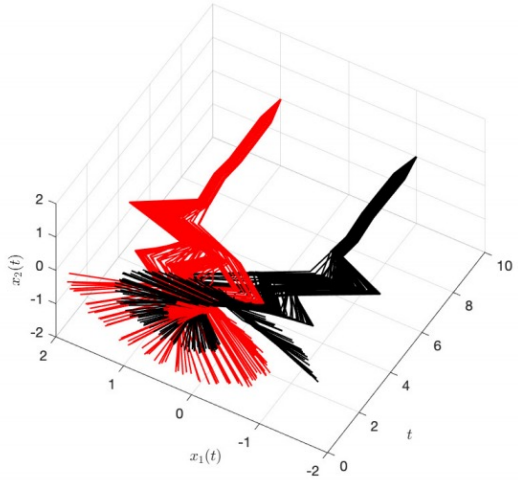
Swiss role



Without stage cost regularization



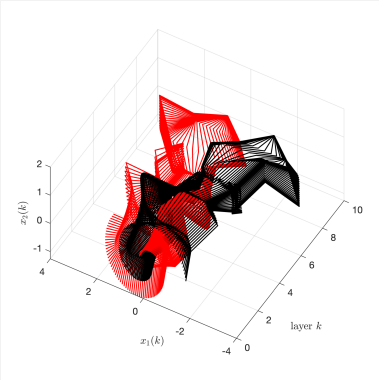
With stage cost regularization



Interplay of stage cost and dynamics on data propagation?

Overview – Dissipativity and Turnpike Properties in NN Training

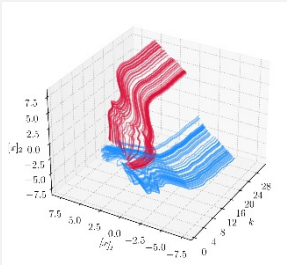
Optimal Control & Deep Learning?



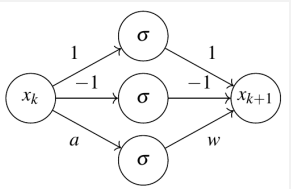
Dissipativity & ResNet Training?

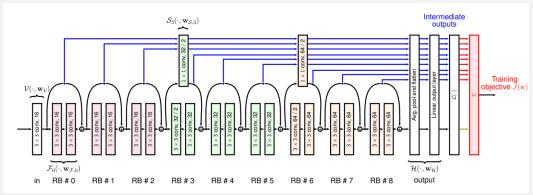
$$S(f(\mathbf{x},u)) - S(\mathbf{x}) \leq \tilde{\ell}_{\mathrm{f}}(\mathbf{x},u) + r\|u\|_2^2 - \mathsf{dist}\left((\mathbf{x},u),\mathbb{Z}^\star\right)$$

Turnpikes in ResNet Training & Experiments



Extension to Other Architectures





A short detour – Lyapunov-like functions for open systems?

Lyapunov (1892) – Analysis of closed systems

$$\forall x \in \mathbb{R}^n : \quad \frac{\mathrm{d}V}{\mathrm{d}t} = \nabla V^\top f(x) \leq -\alpha(\|x - \bar{x}\|)$$

change of "energy" along trajectories

decay

$$\Sigma : \quad \dot{x} = f(x), \quad x(0) = x_0$$

Willems (1972, 2007):

"A generalization of Lyapunov functions to open systems, to systems with inputs and outputs"

Willems (1971, 1972); Willems (2007). In control, almost from the beginning until the day after tomorrow. *EJC*

A short detour – Dissipativity

Lyapunov (1892) – Analysis of closed systems

$$\forall x \in \mathbb{R}^n : \quad \nabla V^\top f(x) \leq -\alpha(\|x - \bar{x}\|)$$

$$\Sigma : \quad \dot{x} = f(x), \quad x(0) = x_0$$

Willems (1972) – Control and analysis of open systems

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \nabla S^\top f(x,u) \leq w(y,u)$$

change of "energy" along trajectories — "power" supply

dissipation inequality

$$\Sigma : \begin{cases} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{cases}$$

$\mathbf{u}$ ← , → $\mathbf{y}$

Willems (1971, 1972)

A short detour – Dissipativity

Lyapunov (1892) – Analysis of closed systems

$$\forall x \in \mathbb{R}^n : \quad \nabla V^\top f(x) \leq -\alpha(\|x - \bar{x}\|)$$

$$\Sigma : \quad \dot{x} = f(x), \quad x(0) = x_0$$

Willems (1972) – Dissipation inequality for open systems

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \nabla S^\top f(x,u) \leq w(y,u)$$

$$\Sigma : \begin{cases} \dot{x} &= f(x,u) \\ y &= h(x,u) \end{cases} \quad \leftarrow \mathbf{u}, \quad \rightarrow \mathbf{y}$$

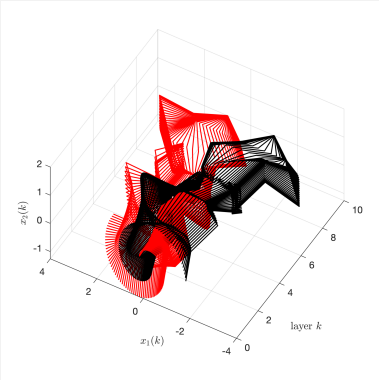
Discrete-time dissipation inequality

$$S(f(x,u)) - S(x) \leq w(x,u) := w(h(x,u),u)$$

$$\Sigma : \begin{cases} x^+ &= f(x,u) \\ y &= h(x,u) \end{cases} \quad \leftarrow \mathbf{u}, \quad \rightarrow \mathbf{y}$$

Overview

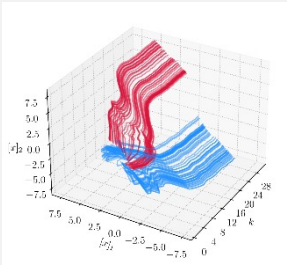
Optimal Control & Deep Learning?



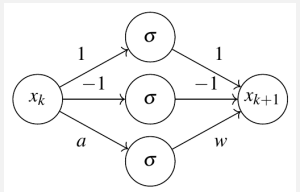
Dissipativity & **ResNet Training?**

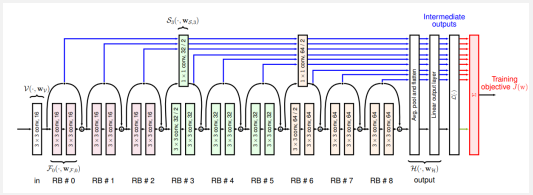
$$S(f(\mathbf{x},u)) - S(\mathbf{x}) \leq \tilde{\ell}_{\mathrm{f}}(\mathbf{x},u) + r\|u\|_2^2 - \mathsf{dist}\left((\mathbf{x},u),\mathbb{Z}^\star\right)$$

Turnpikes in ResNet Training & Experiments



Extension to Other Architectures





Dissipativity for ResNet training?

ResNet training with stage cost loss

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \mathbf{l}(\mathbf{x}(k),\mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(k),b(k))\|^2 + \gamma \cdot \mathbf{l}_\mathrm{f}(\mathbf{x}(N),\mathbf{y})$$

$$\text{subject to} \quad \mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right), \ \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^{nM}$$

Let $\mathbb{X}^\star(\mathbf{y}) \doteq \operatorname{argmin}_{\mathbf{x}} \mathbf{l}(\mathbf{x},\mathbf{y})$

Definition: If there exists a bounded storage function $S : \mathbb{R}^{nM} \to \mathbb{R}_0^+$ such that

$$\underbrace{S(\mathbf{x}(k+1)) - S(\mathbf{x}(k))}_{\text{change of storage}} \leq \underbrace{+\mathbf{l}(\mathbf{x}(k),\mathbf{y}) + \rho \cdot \overbrace{\|\mathrm{vect}(A(k),b(k))\|^2}^{\|u\|^2}}_{\text{supply rate}}$$

then the ResNet dynamics are w.r.t. the loss function $\mathbf{l}$.

Simplifying assumption: $\mathbf{l}(\mathbf{x},\mathbf{y}) = 0$ on $\mathbb{X}^\star(\mathbf{y})$

Observation – Dissipativity is closely related to training objective

$$
\begin{aligned}
S(\mathbf{x}(1)) - S(\mathbf{x}(0)) &\leq -\mathrm{dist}(\mathbf{x}(0), \mathbb{X}^\star(\mathbf{y})) + \mathbf{l}(\mathbf{x}(0), \mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(0), b(0))\|^2 \\
S(\mathbf{x}(2)) - S(\mathbf{x}(1)) &\leq -\mathrm{dist}(\mathbf{x}(1), \mathbb{X}^\star(\mathbf{y})) + \mathbf{l}(\mathbf{x}(1), \mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(1), b(1))\|^2 \\
&\vdots \\
S(\mathbf{x}(N-1)) - S(\mathbf{x}(N-2)) &\leq -\mathrm{dist}(\mathbf{x}(N-2), \mathbb{X}^\star(\mathbf{y})) + \dots \\
+ \quad S(\mathbf{x}(N)) - S(\mathbf{x}(N-1)) &\leq -\mathrm{dist}(\mathbf{x}(N-1), \mathbb{X}^\star(\mathbf{y})) + \dots
\end{aligned}
$$

$$
S(\mathbf{x}(N)) - S(\mathbf{x}_0) \leq \sum_{k=0}^{N-1} -\mathrm{dist}(\mathbf{x}(k), \mathbb{X}^\star(\mathbf{y})) + \sum_{k=0}^{N-1} \mathbf{l}(\mathbf{x}(k), \mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2
$$

Use dissipativity to analyze ResNet training?

Reachability neural ODE → Reachability of ResNets

Reachability (Neural ODEs with ReLU activation (Ruiz-Balet))

For any distinct initial point x^i and target z^i, there exists piecewise constant control functions $w(t), A(t), b(t)$ such that all datasamples

$$\dot{x}^i(t) = w(t)\sigma\left(A(t)x^i(t) + b(t)\right), \quad x^i(0) = x^i$$

reach $x^i(T) = z^i$ for every $T > 0$

Reachability of ResNet

$$x(k+1) = x(k) + \sigma(A(k)x(k) + b(k))$$

Hardt, M., & Ma, T. (2017). Identity Matters in Deep Learning. In International Conference on Learning Representations.
Ruiz-Balet, D., & Zuazua, E. (2023). Neural ode control for classification, approximation, and transport. SIAM Review, 65(3), 735-773.

Dissipativity – A very useful system property!

Reachability of $\mathbb{X}^\star(\mathbf{y})$: Given $\mathbf{x}_0$ there exist $A(\cdot), b(\cdot)$ defined on $\{0, \dots, \tilde{N}-1\}$ and some depth $\tilde{N}$ such that

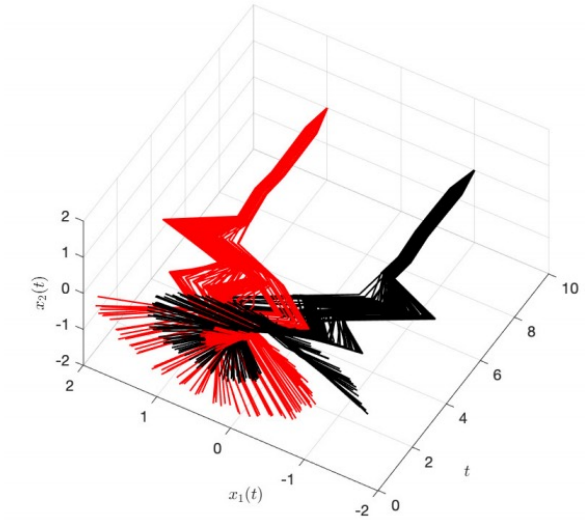
$$\operatorname{dist}(\mathbb{X}^\star(\mathbf{y}), \mathbf{x}(\tilde{N})) = 0$$

$$S(\mathbf{x}^\star(N)) - S(\mathbf{x}_0) \leq \sum_{k=0}^{N-1} -\operatorname{dist}(\mathbf{x}^\star(k), \mathbb{X}^\star(\mathbf{y})) + \mathbf{l}(\mathbf{x}^\star(k), \mathbf{y}) + \rho \cdot \|\operatorname{vect}(A^\star(k), b^\star(k))\|^2 \leq \tilde{C}(\mathbf{x}_0)$$

What is a turnpike? How to exploit it? Does it occur often? Can this be enforced? . . .

Reachability results for ResNets: Ruiz-Balet and Zuazua (2023); Hardt & Ma (2016)

Turnpike properties in optimal control?



Dorfman et al: "It is exactly like a **turnpike paralleled by a network of minor roads**. […]
if the origin and destination are far enough apart, it will always pay **to get on to the turnpike** and **cover distance at the best rate of travel**, even if this means adding a little mileage at either end."

Ramsey (1928); **von Neumann (1938); Dorfman et al. (1954**); Mckenzie (1976); … Wilde & Kokotovic (1972); Anderson & Kokotovic (1987); **Carlson et al. (1991);** Rawlings & Amrit (2009); Grüne (2013); F. et al (2014); Trelat & Zuazua (2014); Damm et al. (2014); …; F. & Grüne (2022)

Turnpikes in optimal control?

$$\min_{u_{0,\dots,N-1}} \quad \sum_{k=0}^{N-1} u_k^2$$

$$\text{s.t.} \quad \forall k = 0, \dots, N-1$$

$$\mathbf{x}_{k+1} = 2\mathbf{x}_k + u_k, \quad \mathbf{x}_0 = 2$$

$$\mathbf{x}_k \in [-2, 2], \quad \mathbf{u}_k \in [-3, 3]$$

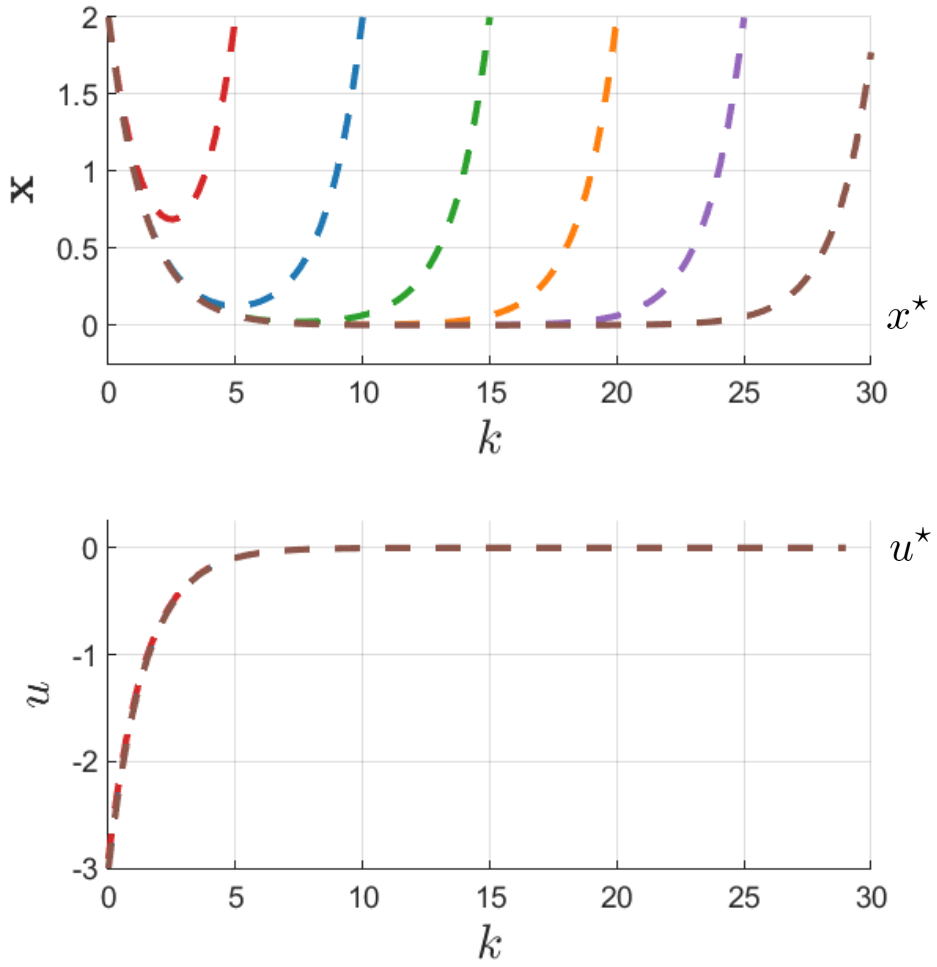
Optimal steady state pair

$$(\bar{\mathbf{x}}^\star, \bar{u}^\star) = \arg\min_{\mathbf{x},u} \quad \ell(\mathbf{x}, u)$$

$$\text{s.t.} \quad \mathbf{x} = \mathbf{f}(\mathbf{x}, u)$$

Example

$$(\bar{\mathbf{x}}^\star, \bar{u}^\star) = (0, 0)$$



Optimal steady state pair = **turnpike** = attractor of infinite-horizon optimal solutions

Faulwasser, T., & Grüne, L. (2022). Turnpike properties in optimal control: An overview of discrete-time and continuous-time results. Handbook of numerical analysis

Damm, T., Grune, L., Stieler, M., & Worthmann, K. (2014). An exponential turnpike theorem for dissipative discrete time optimal control problems. *SIAM Opt*

The turnpike phenomenon occurs often in optimal control

J. Math. Biol. (2015) 70:289–327
DOI 10.1007/s00285-014-0768-9

Mathematical Biology

Optimal Control Problems over Large Time

Acta Numerica (2022), pp. 135–263
doi:10.1017/S0962492922000046

Printed in the United Kingdom

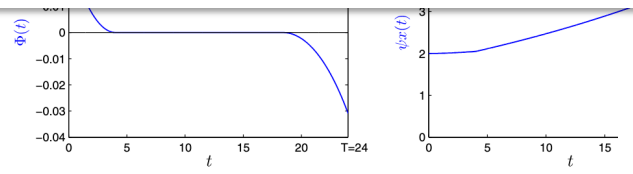
Turnpike in optimal control of PDEs, ResNets, and beyond

Borjan Geshkovski* Enrique Zuazua

in optimal shape design

ce[a], Emmanuel Trélat[a], Enrique Zuazua[b]

ersité, CNRS, Université de Paris, Inria, Laboratoire Jacques-Louis Lions (LJLL)
d Analysis, Alexander von Humboldt-Professorship, Department of Mathematics
any;
tational Mathematics, Fundación Deusto Av. de las Universidades 24, 48007 Bilb
e Matemáticas, Universidad Autónoma de Madrid, 28049 Madrid, Spain.



NONLINEAR SYSTEMS

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 17, No. 4, November 1992
Printed in U.S.A.

TURNPIKE SETS AND THEIR ANALYSIS IN STOCHAS' PRODUCTION PLANNING PROBLEMS*

S. SETHI, H. M. SONER, Q. ZHANG AND J. JIANG

This paper considers optimal infinite horizon stochastic production planning problems capacity and demand to be finite state Markov chains. The existence of the optimal feedback

ELSEVIER

IFAC Journal of Systems and Control
Volume 30, December 2024, 100290

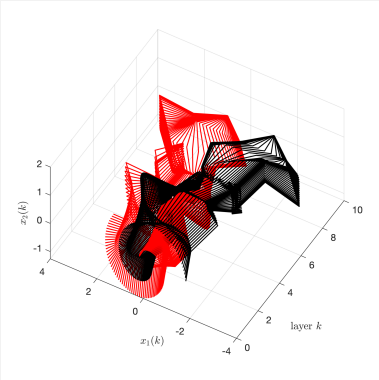
Full length article

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

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

Overview – Dissipativity and Turnpike Properties in NN Training

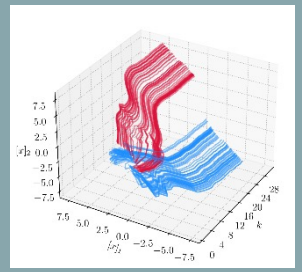
Optimal Control & Deep Learning?



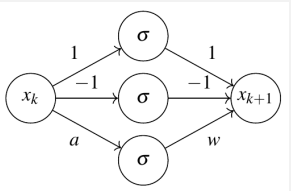
Dissipativity & ResNet Training?

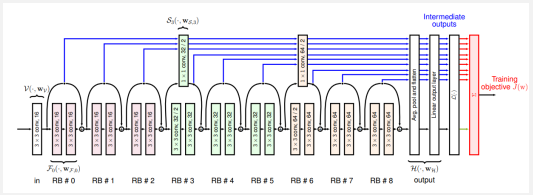
$$S(f(\mathbf{x},u)) - S(\mathbf{x}) \leq \tilde{\ell}_{\mathrm{f}}(\mathbf{x},u) + r\|u\|_2^2 - \mathsf{dist}\left((\mathbf{x},u),\mathbb{Z}^\star\right)$$

Turnpikes in ResNet Training & Experiments



Extension to Other Architectures



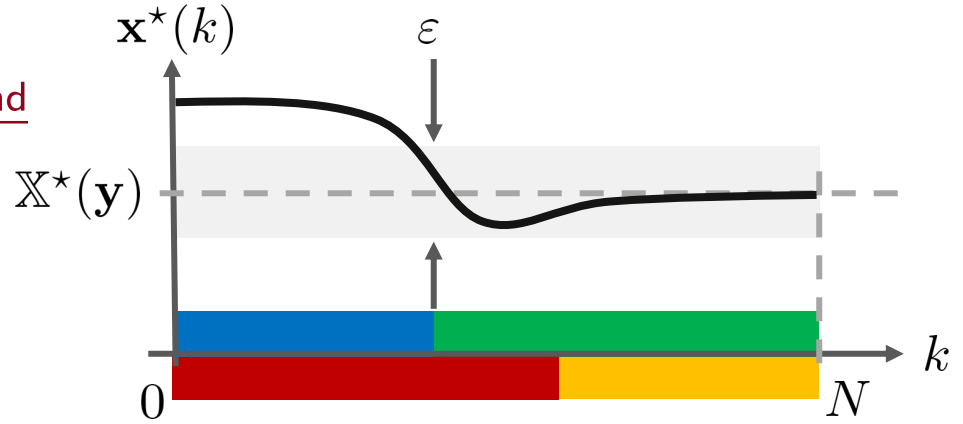


How to exploit turnpikes for deep learning?

$$\mathbb{X}^\star(\mathbf{y}) \doteq \operatorname*{argmin}_{\mathbf{x}} \mathbf{l}_\mathrm{f}(\mathbf{x}, \mathbf{y})$$

Suppose the training OCP has the turnpike property:

$$\#\{k \in \{0, N-1\} \,|\, \mathbf{x}^\star(k) \notin \mathcal{B}_\varepsilon(\mathbb{X}^\star(\mathbf{y}))\} \leq \frac{\text{Perf+Storage Bnd}}{\varepsilon}$$



Split depth horizon $\{0, \ldots, N-1\}$:

$$\#\{k \,|\, \mathbf{x}^\star(k) \notin \mathcal{B}_\varepsilon\} \quad \bigcup \quad \#\{k \,|\, \mathbf{x}^\star(k) \in \mathcal{B}_\varepsilon\}$$

Fix $\varepsilon > 0$, **pick** N such that $\quad 0 \quad < \quad N - \frac{\text{Perf+Storage Bnd}}{\varepsilon}$

Hence, **design** training OCP with turnpike $\quad \Leftarrow \quad$ strict dissipativity w.r.t. $\mathbf{l}_\mathrm{f}$!

Implications of dissipative design of ResNet training

- Strict dissipativity + reachability $\Rightarrow N > \frac{\text{Perf+Storage Bnd}}{\varepsilon} \Rightarrow |\mathbf{l}_\mathrm{f}(\mathbf{x}^\star(N), \mathbf{y}) - \mathbf{l}^\star| \leq C \cdot \varepsilon$
- Problems where ε-close to $\mathbb{X}^\star(\mathbf{y})$ is good enough $\quad \Rightarrow \quad$ empirical risk $= 0$
  (ML: performance on data)

Ruiz-Balet & Zuazua (2023); F., Hempel & Streif (2024)

Dissipative OCP design for deep learning?

ResNet training

design

given

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \underbrace{\mathbf{l}(\mathbf{x}(k)) + \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2}_{\doteq \mathbf{l}(\mathbf{x}(k),\mathbf{u}(k))} + \gamma \cdot \mathbf{l}_\mathrm{f}(\mathbf{x}(N), \mathbf{y})$$

subject to $\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$

Compute set of optimal *steady states*

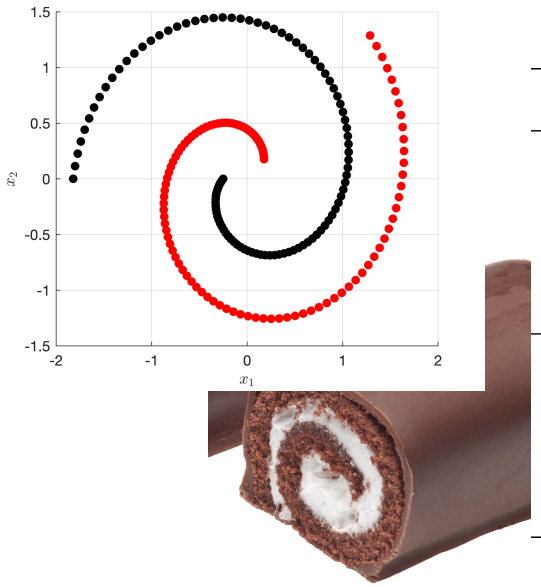
$$\bar{\mathbf{x}} \in \mathbb{X}^\star(\mathbf{y}) \doteq \operatorname*{argmin}_{\mathbf{x}} \mathbf{l}_\mathrm{f}(\mathbf{x}, \mathbf{y}) \neq \emptyset$$

Stage cost

$$\mathbf{l}(\mathbf{x}, \mathbf{u}) = \|\mathbf{x} - \bar{\mathbf{x}}\|_Q^2 + \|\mathbf{u}\|_R^2$$

$Q, R \succ 0 \Rightarrow$ strict dissipativity & $S(x) = 0$

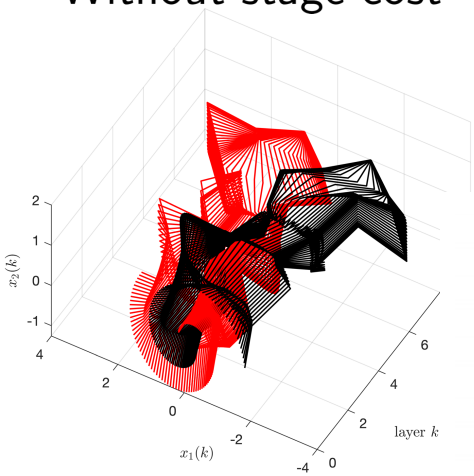
Swiss role



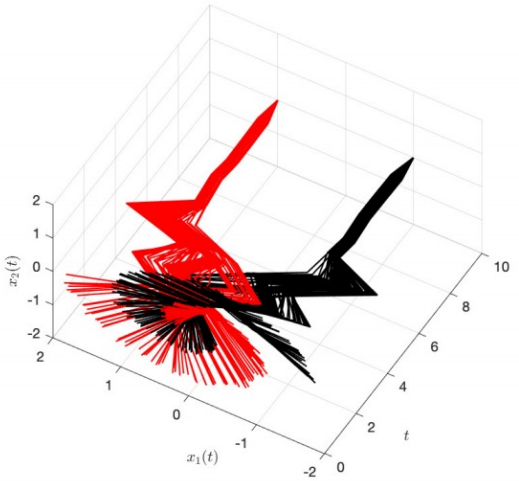
Results with label noise

N	M	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

Without stage cost



With stage cost



F., Hempel & Streif (2024), *IFAC JSC*

But … most ML problems are much more complex – Cross entropy?

Target probabilities induced by labels

$$q(i|y) = \begin{cases} 1 & i = y \\ 0 & i \neq y \end{cases}$$

Predicted softmax probability

$$p(y|x) = \frac{e^{[x]_y}}{\sum_{i=1}^{C} e^{[x]_j}}$$

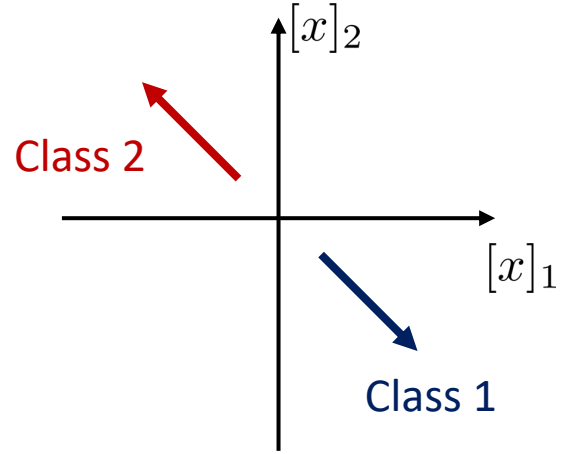
$[x]_i$ = i-th component of x

Cross-entropy between predicted and target probabilities

$$H(p,q) = -\sum_{c=1}^{C} q(c|y) \log\left(p(c|x)\right)$$

Loss function (binary targets)

$$\ell_{\mathrm{f}}(x,y) = -\log(p(y|x))$$



Minimization of the cross entropy?

$$\min_{x \in \mathbb{R}^2} \quad \ell_{\mathrm{f}}(x,1)$$

No finite minimizer in $\mathbb{R}^C$!

How to get finite loss minimizers?

Label smoothing target probabilities
(Szegedy et al. 2016)

$$\tilde{q}(c|y) = \begin{cases} p_d & c = y \\ \frac{1-p_d}{C-1} & c \neq y \end{cases}$$

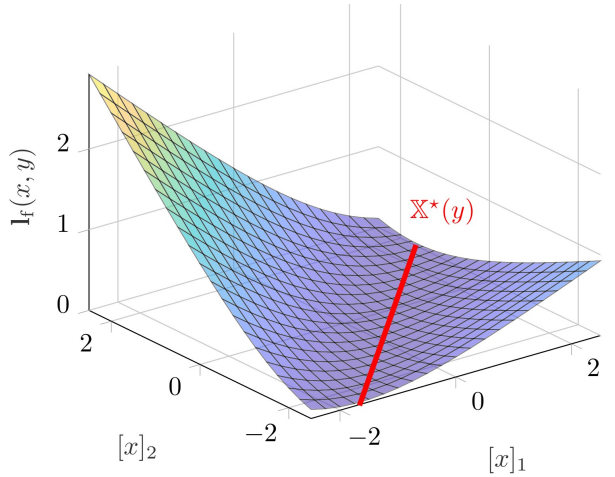
Predicted softmax probability

$$p(y|x) = \frac{e^{[x]_y}}{\sum_{i=1}^{C} e^{[x]_j}}$$

Soft cross entropy

$$H(p,q) = -\sum_{c=1}^{C} q(c|y) \log\left(p(c|x)\right)$$

Soft cross entropy (label smoothing)

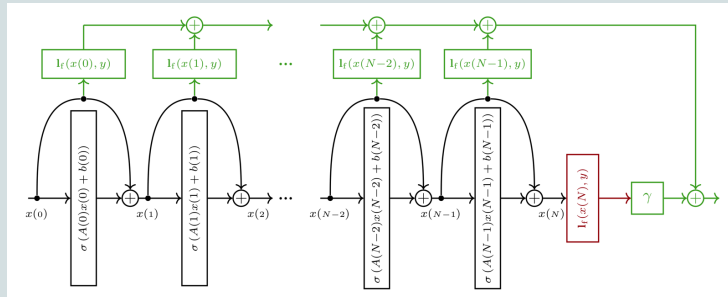


$$\ell_{\mathbf{f}}(\boldsymbol{x},\boldsymbol{y}) \geq \alpha \circ \mathrm{dist}(x, \mathbb{X}^\star(y))$$
$$\mathbf{l}_{\mathbf{f}}(\mathbf{x},\mathbf{y}) \geq \alpha \circ \mathrm{dist}(\mathbf{x}, \mathbb{X}^\star(\mathbf{y}))$$
$\alpha(0) = 0$ & strictly mono. increasing

Soft cross entropy in stage cost & end penalty ⇒ strictly dissipative training OCP

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \mathbf{l}_{\mathrm{f}}(\mathbf{x}(k),\mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(k),b(k))\|^2 + \gamma \cdot \mathbf{l}_{\mathrm{f}}(\mathbf{x}(N),\mathbf{y})$$

subject to $\quad \mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$



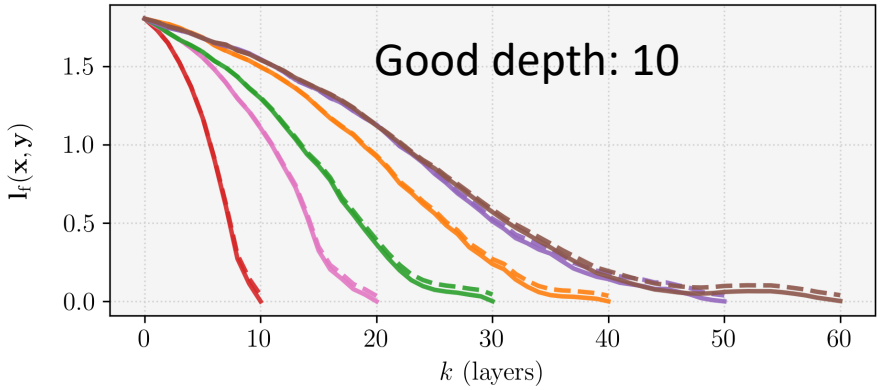
Püttschneider & F. (2025); *Püttschneider et al. (2025); Early exits in* BranchyNet: Teerapittayanon et al. (2015); Lee et al (2015)

Evaluation on MNIST

Standard training

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2 + \mathbf{l}_\mathrm{f}(\mathbf{x}(N), \mathbf{y})$$

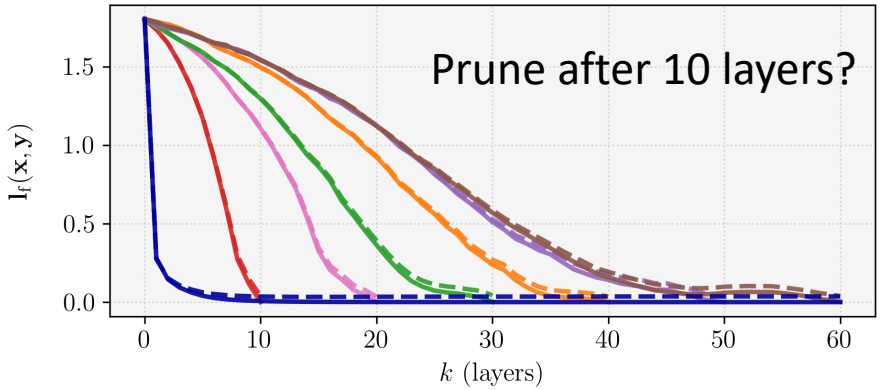
subject to $\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$



Proposed dissipative formulation

$$\min_{A(\cdot),b(\cdot)} \sum_{k=0}^{N-1} \mathbf{l}_\mathrm{f}(\mathbf{x}(k), \mathbf{y}) + \rho \cdot \|\mathrm{vect}(A(k), b(k))\|^2 + \gamma \cdot \mathbf{l}_\mathrm{f}(\mathbf{x}(N), \mathbf{y})$$

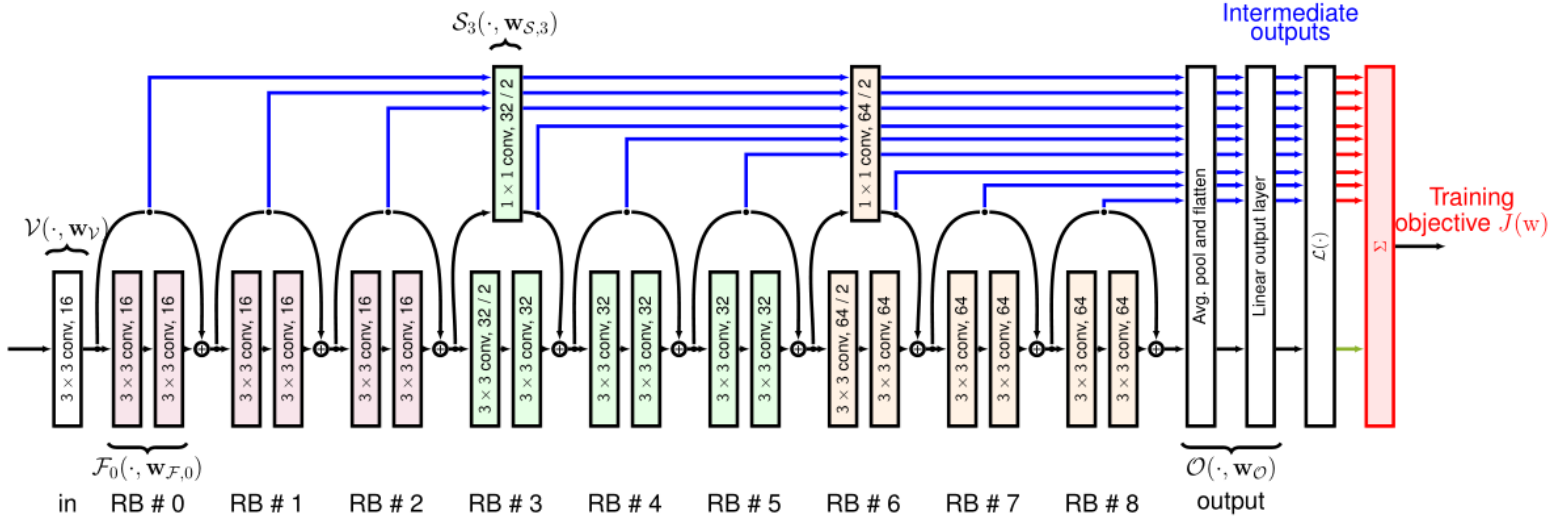
subject to $\mathbf{x}(k+1) = \mathbf{x}(k) + \sigma\left((I \otimes A(k))\mathbf{x}(k) + \mathbf{1} \otimes b(k)\right)$

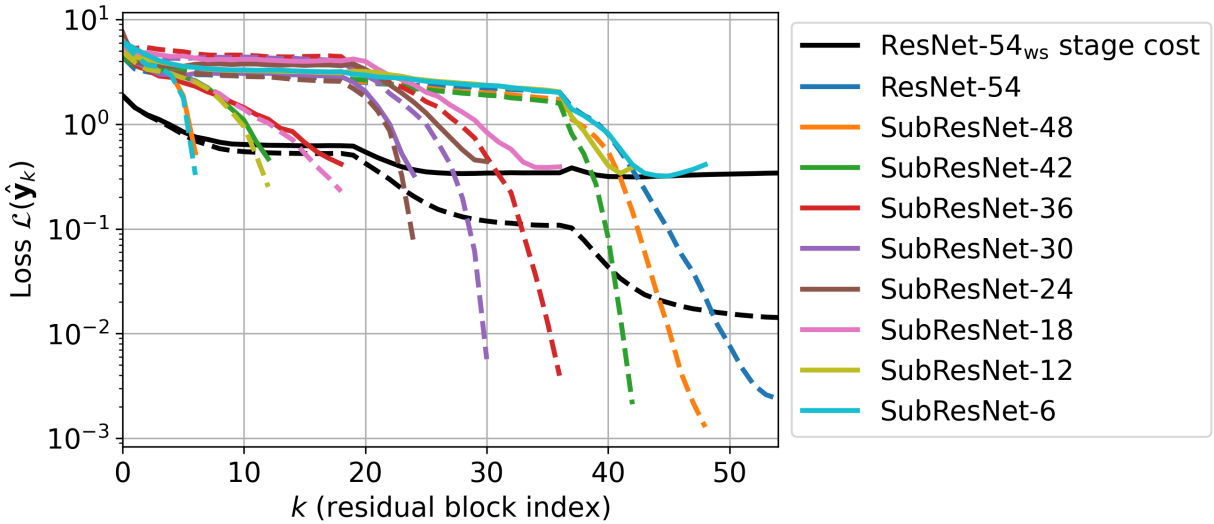


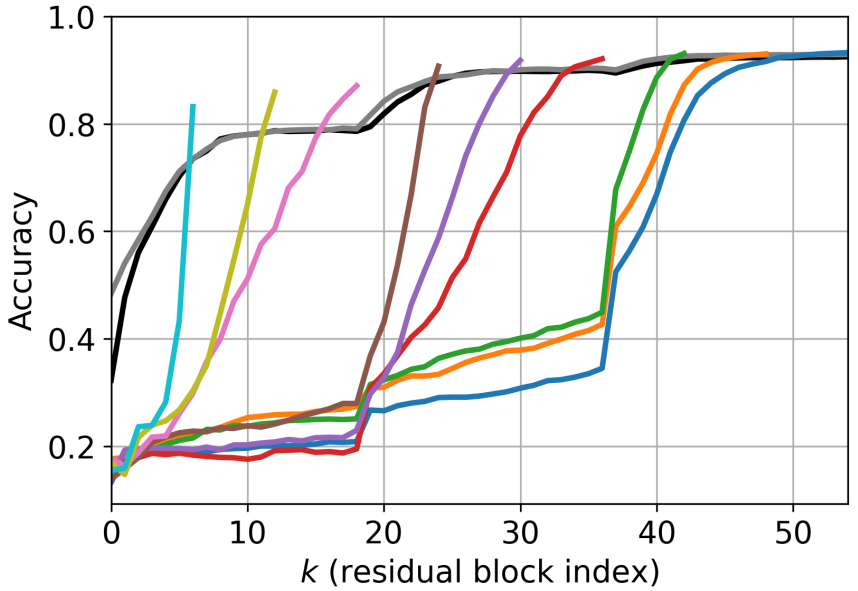
Dissipative training formulation ⇒ Reveals early exits & enables formal analysis

Püttschneider & F. (2025); Püttschneider, Heilig, Fischer & F. (2025)

Results for more complex ResNets and CIFAR-10





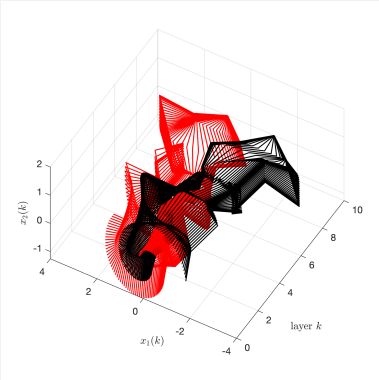


Püttschneider, Heilig, Fischer & F. (2025)

Analysis with multiple stages? → Ongoing work

Overview – Dissipativity and Turnpike Properties in NN Training

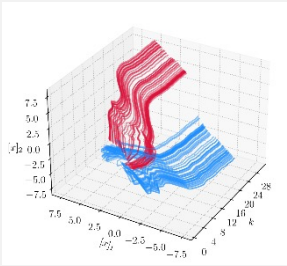
Optimal Control & Deep Learning?



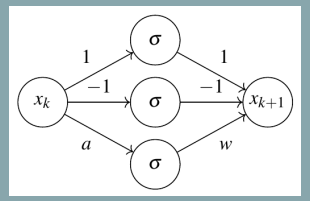
Dissipativity & ResNet Training?

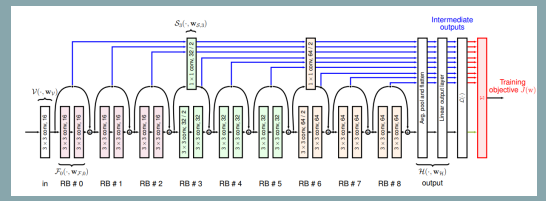
$$S(f(\mathbf{x},u)) - S(\mathbf{x}) \leq$$
$$\tilde{\ell}_{\mathrm{f}}(\mathbf{x},u) + r\|u\|_2^2 - \mathsf{dist}\left((\mathbf{x},u),\mathbb{Z}^\star\right)$$

Turnpikes in ResNet Training & Experiments



Extension to Other Architectures

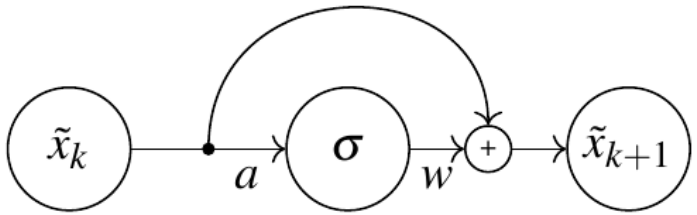


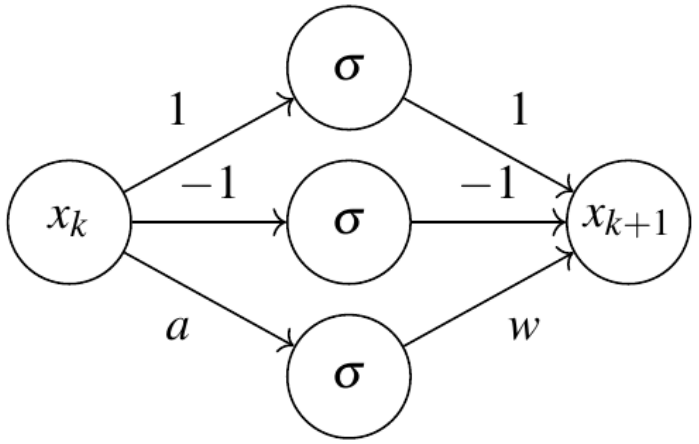


ReLU activation and ResNets

Observation – ReLU identity

$$\sigma(x) - \sigma(-x) = \max\{0, x\} - \max\{0, -x\} = x$$





ReLU residual connection neuron

Three equivalent ReLU neurons

From a neuron to a neural network?

Petersen, P., & Voigtlaender, F. (2018). Optimal approximation of piecewise smooth functions using deep ReLU neural networks. Neural Networks, 108, 296–330.
Liu, C., Liang, E., & Chen, M. (2024). Characterizing ResNet's universal approximation capability. Proceedings of Machine Learning Research, 235, 31477–31515.
Püttschneider, J., Heilig, S., Adilova, L., Fischer, A., & Faulwasser, T. (2026). Turnpikes in deep learning: Beyond ResNets and neural ODEs? In *2026 European Control Conference (ECC)*

Equivalence of ResNets and fully connected networks

ResNet

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + (I^D \otimes \tilde{W}_k)\sigma\left((I^D \otimes \tilde{A}_k)\tilde{\mathbf{x}}_k + \mathbf{1}^D \otimes \tilde{b}_k\right)$$

Two linear layers

-State dimension n to hidden dimension $\tilde{h}$

$$\tilde{A}_k \in \mathbb{R}^{\tilde{h}\times n}, \tilde{b}_k \in \mathbb{R}^{\tilde{h}}$$

-Project back to state dimension

$$\tilde{W}_k \in \mathbb{R}^{n\times \tilde{h}}$$

Fully connected network

$$\mathbf{x}_{k+1} = (I^D \otimes W_k)\sigma\left((I^D \otimes A_k)\mathbf{x}_k + \mathbf{1}^D \otimes b_k\right)$$

Two linear layers

-State dimension n to hidden dimension h

$$A_k \in \mathbb{R}^{h\times n}, b_k \in \mathbb{R}^{h}$$

-Project back to state dimension n

$$W_k \in \mathbb{R}^{n\times h}$$

Equivalence property for $h = \tilde{h}+2n$

ResNet parameters

$\tilde{A}_k$ $\tilde{b}_k$ $\tilde{W}_k$

Fully connected network parameters

$$A_k = \begin{bmatrix} I^n \\ -I^n \\ \tilde{A}_k \end{bmatrix} \quad b_k = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{0}^n \\ \tilde{b}_k \end{bmatrix} \quad W_k = \begin{bmatrix} I^n & -I^n & \tilde{W}_k \end{bmatrix}$$

Lead to the same state trajectory $\mathbf{x}_k = \tilde{\mathbf{x}}_k$, for all $k = 0, ..., N-1$.

Implications?

Püttschneider, J., Heilig, S., Adilova, L., Fischer, A., & Faulwasser, T. (2026). Turnpikes in deep learning: Beyond ResNets and neural ODEs? In *2026 European Control Conference (ECC)*

Optimal steady states of fully connected networks?

Steady state properties

ResNet

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + (I^D \otimes \tilde{W}_k)\sigma\left((I^D \otimes \tilde{A}_k)\tilde{\mathbf{x}}_k + \mathbf{1}^D \otimes \tilde{b}_k\right)$$

Any state $\tilde{\mathbf{x}}$ = steady state for $\tilde{u}_k = (\tilde{A}_k, \tilde{b}_k, \tilde{W}_k) = 0$

Fully connected network

$$\mathbf{x}_{k+1} = (I^D \otimes W_k)\sigma\left((I^D \otimes A_k)\mathbf{x}_k + \mathbf{1}^D \otimes b_k\right)$$

Any state $\mathbf{x}$, there exists a nonempty set of steady-state control inputs $\bar{u} \in \bar{\mathbb{U}}(\bar{\mathbf{x}})$, but in general $0 \notin \bar{\mathbb{U}}(\bar{\mathbf{x}})$

Optimal steady states

ResNet

$$\ell(\tilde{\mathbf{x}}, \tilde{u}) = \ell_{\mathrm{f}}(\tilde{\mathbf{x}}, \mathbf{y}) + r\|u\|_2^2$$

Set of optimal steady states

$$\mathbb{Z}_{\mathbf{y}}^{\star} = \left\{(\bar{\mathbf{x}}, 0) \middle| \bar{\mathbf{x}} \in \mathbb{X}_{\mathbf{y}}^{\star}\right\}$$

Fully connected network

$$\ell(\tilde{\mathbf{x}}, \tilde{u}) = \ell_{\mathrm{f}}(\tilde{\mathbf{x}}, \mathbf{y})$$

Set of optimal steady states

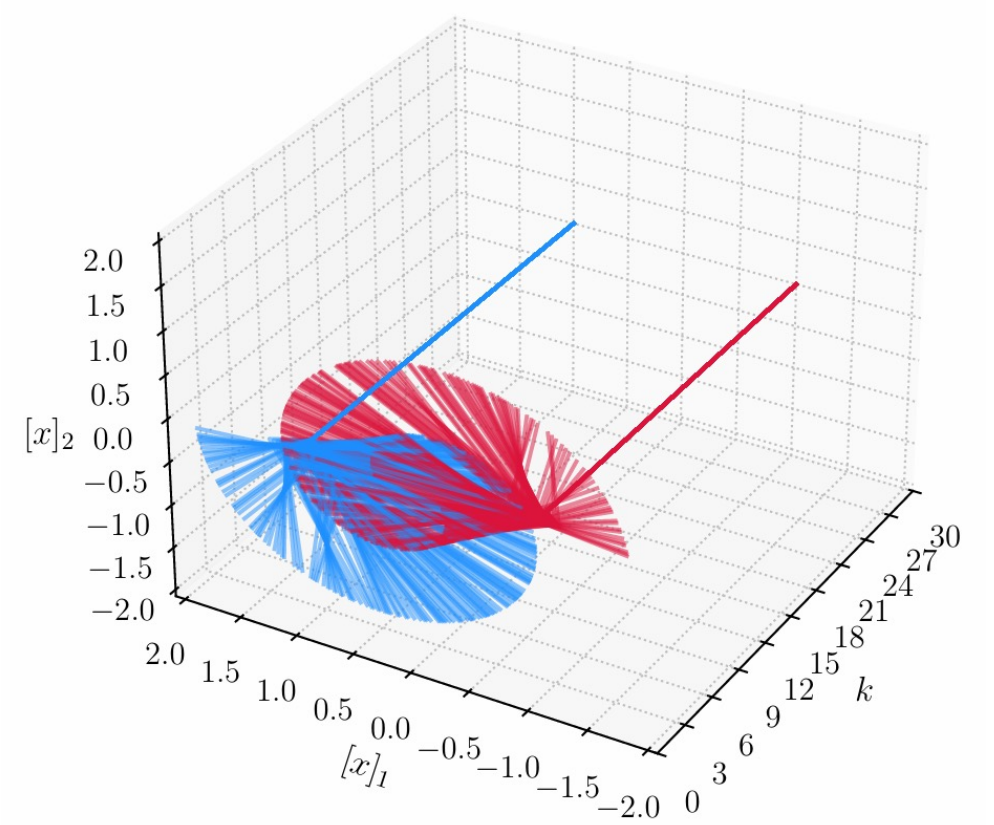
$$\mathbb{Z}_{\mathbf{y}}^{\star} = \left\{(\bar{\mathbf{x}}, \bar{u}) \middle| \bar{\mathbf{x}} \in \mathbb{X}_{\mathbf{y}}^{\star}, \bar{u} \in \bar{\mathbb{U}}(\bar{\mathbf{x}})\right\}$$

Reachability and analysis results carry over!

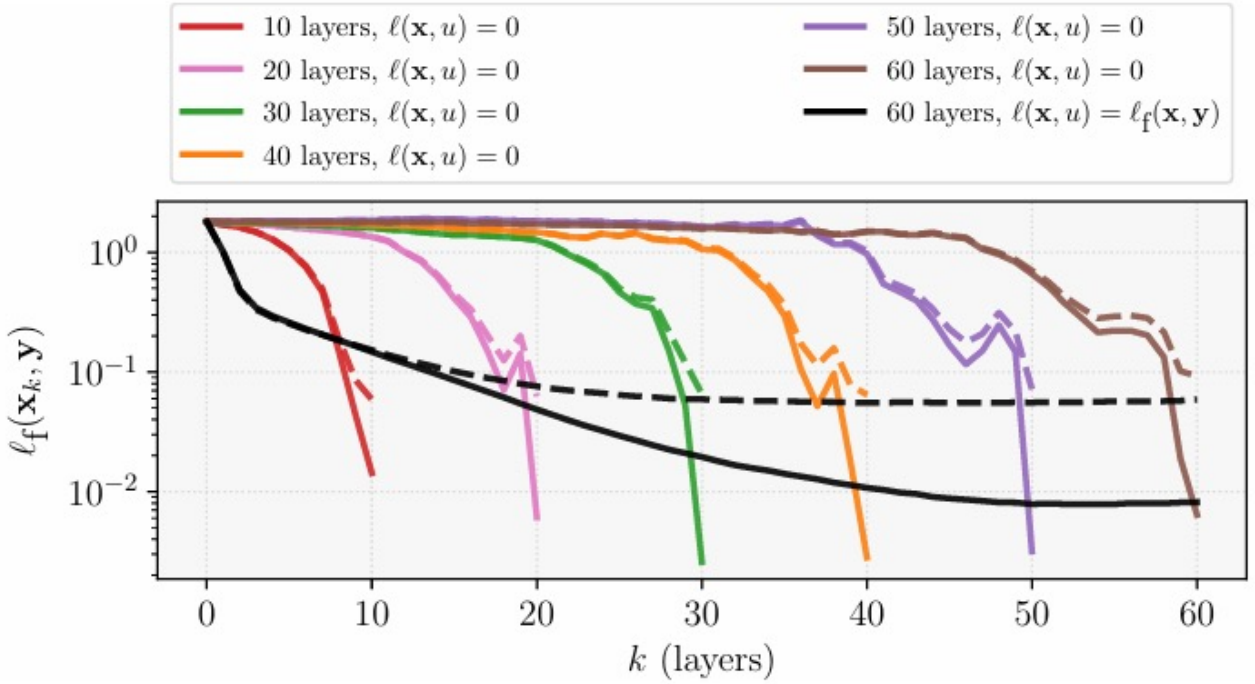
Püttschneider, J., Heilig, S., Adilova, L., Fischer, A., & Faulwasser, T. (2026). Turnpikes in deep learning: Beyond ResNets and neural ODEs? In *2026 European Control Conference (ECC)*

Turnpikes with fully connected networks – Experiments

Two spirals + l_2 loss



MNIST + soft cross-entropy loss



Reachability and analysis results carry over!

Püttschneider, J., Heilig, S., Adilova, L., Fischer, A., & Faulwasser, T. (2026). Turnpikes in deep learning: Beyond ResNets and neural ODEs? In *2026 European Control Conference (ECC)*

Summary

Turnpike properties in optimal control

- Very useful tool for non-local analysis: properties of **OCPs parametric in x_0**
- Dissipativity and turnpikes are closely related
- Results for ODEs, PDEs, stochastic systems, …
- **Problem analysis and OCP synthesis**
- …

Turnpikes & dissipativity in deep learning

- Known – shape loss landscape:
  - BranchyNet: Teerapittayanon et al. (2015); …
- **Dissipativity-based formulation of training uses loss as stage cost**
- **Dissipativity + early exits = avenue for analysis of deep networks**
- ***We are just at the beginning …***

References

He, Zhang, Ren & Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*
LeCun (1988) A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*
Li, Chen, Tai & E (2018). Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*
Esteve, Geshkovski, Pighin & Zuazua (2020). Large-time asymptotics in deep learning. *arXiv*
Ruiz-Balet & Zuazua (2023). Neural ODE control for classification, approximation, and transport. *SIAM Review*
Tabuada & Gharesifard (2023). Universal approximation power of deep residual neural networks through the lens of control. *IEEE Transactions on Automatic Control*
…

Faulwasser, Hempel & Streif (2024). On the turnpike to design of deep neural networks: Explicit depth bounds. *IFAC Journal of Systems and Control*
Pan, Stomberg, Engelmann & Faulwasser (2021). First results on turnpike bounds for stabilizing horizons in NMPC. *IFAC-PapersOnLine*
Püttschneider & Faulwasser (2024). On dissipativity of cross-entropy loss in training ResNets. *Automatica*
Püttschneider, Heilig, Fischer & Faulwasser (2025). Towards an Optimal Control Perspective of ResNet Training. *arXiv & ICML workshop*
Püttschneider, J., Heilig, S., Adilova, L., Fischer, A., & Faulwasser, T. (2026). Turnpikes in deep learning: Beyond ResNets and neural ODEs? In *2026 European Control Conference (ECC)*

Summary

Turnpike properties in optimal control

- Very useful tool for non-local analysis: properties of **OCPs parametric in x_0**
- Dissipativity and turnpikes are closely related
- Results for ODEs, PDEs, stochastic systems, …
- **Problem analysis and OCP synthesis**
- …

Turnpikes & dissipativity in deep learning

- Known – shape loss landscape:
  - BranchyNet: Teerapittayanon et al. (2015); …
- **Dissipativity-based formulation of training uses loss as stage cost**
- **Dissipativity + early exits = avenue for analysis of deep networks**
- ***We are just at the beginning …***

Thank you & and all co-authors…

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Transregio 391

FOR 5785 ALeSCo
Active Learning for Systems and Control