Deep Learning, Dynamics and Control

I: Deep Learning from the Control Viewpoint

Qianxiao Li
Department of Mathematics
Institute for Functional Intelligent Materials
blog.nus.edu.sg/qianxiaoli

Dynamical systems and Semi-algebraic geometry
Interactions with Optimization and Deep Learning
Dalat, Vietnam
17–21 Jul 2023

Overview

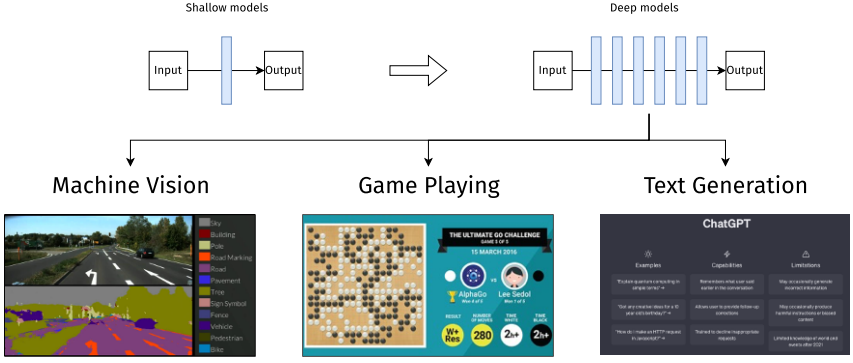
1. Introduction
    - Deep learning
    - Control theory
2. Approximation and controllability
3. Optimisation and optimal control
    - Theory
    - Applications
4. Summary and outlook

Introduction

Introduction

Deep learning

Deep learning: theory vs practice



Practical success vs theoretical mystery

What's new in deep learning?

- Classical models (Polynomials, Fourier series, kernel SVM, shallow NNs) build complexity by increasing number of linear combinations of simple functions
- Deep neural networks build complexity through compositions of simple functions

Supervised learning

x

F^*

$y = F(x)$

Cat
Dog

Goal: learn/approximate the target relationship F^*

Approach: Define a model hypothesis space $\mathcal{H}$, and find a "closest" model $\tilde{F} \in \mathcal{H}$

C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. springer New York, 2006, vol. 4

Example: shallow and deep neural network hypothesis spaces

Shallow neural networks (width m, input dimension d)

$$\mathcal{H} = \left\{ \widehat{F}(x) = v^\top \sigma(Wx + b) : v, b \in \mathbb{R}^m, W \in \mathbb{R}^{m \times d} \right\}$$

The scalar function σ is called the activation function, and is applied element-wise

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016

Example: shallow and deep neural network hypothesis spaces

Shallow neural networks (width m, input dimension d)

$$\mathcal{H} = \left\{ \widehat{F}(x) = v^\top \sigma(Wx + b) : v, b \in \mathbb{R}^m, W \in \mathbb{R}^{m \times d} \right\}$$

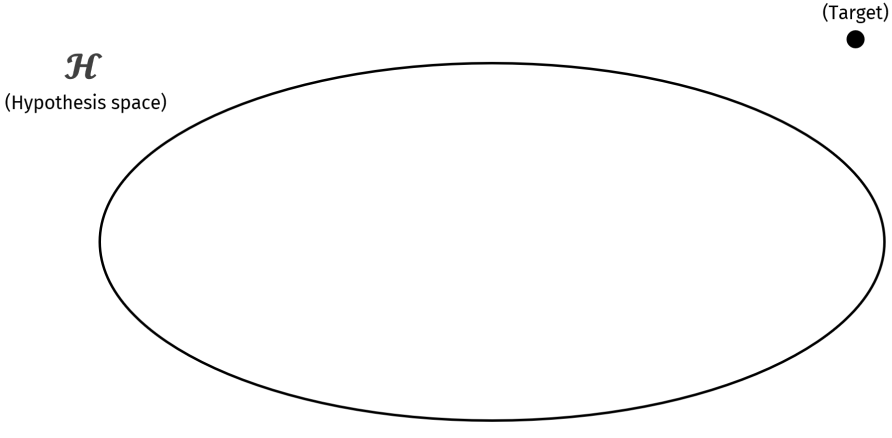
The scalar function σ is called the activation function, and is applied element-wise

Deep (residual) neural networks (widths m_k, depth K, input dimension d)

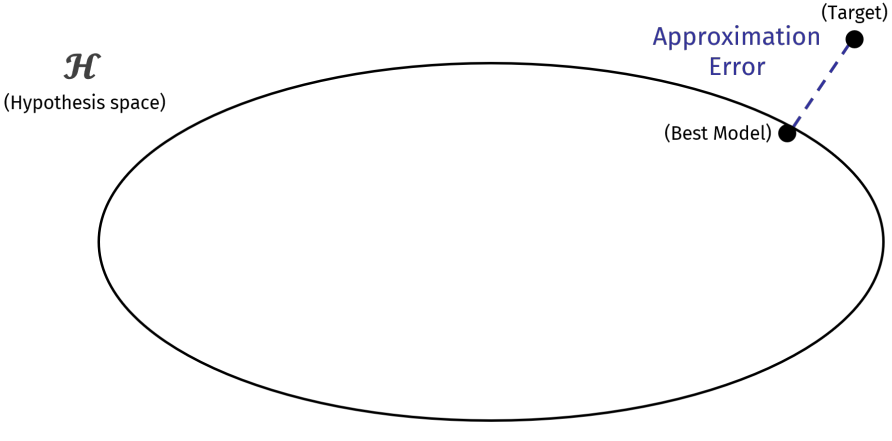
$$\mathcal{H} = \left\{ \begin{array}{c} \widehat{F}(x) = v^\top x_K : x_{k+1} = x_k + V_k \sigma(W_k x_k + b_k), k = 1, \ldots, K-1, x_0 = x \\ W_k \in \mathbb{R}^{d_k \times d_k}, V_k \in \mathbb{R}^{d_{k+1} \times d_k}, b_k \in \mathbb{R}^{d_k}, v \in \mathbb{R}^{d_K} \end{array} \right\}$$

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016

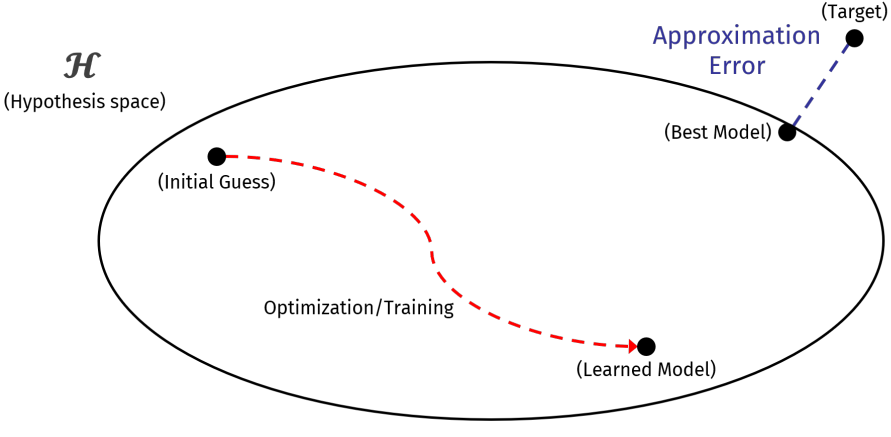
Approximation, optimisation and generalisation



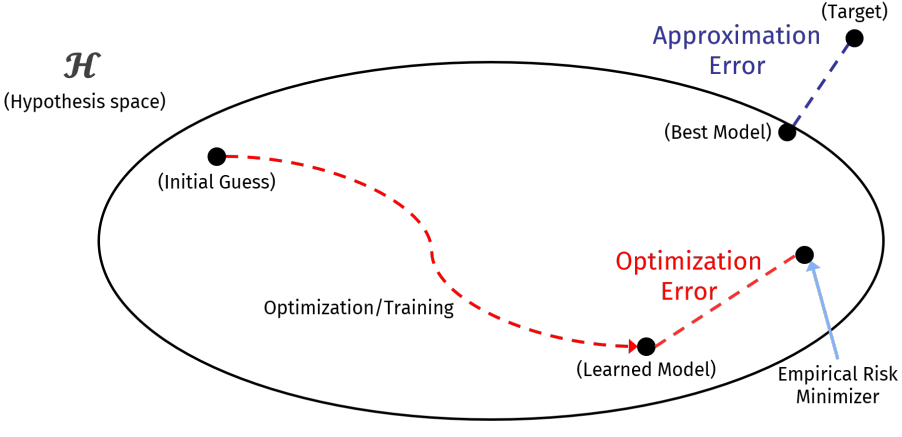
Approximation, optimisation and generalisation



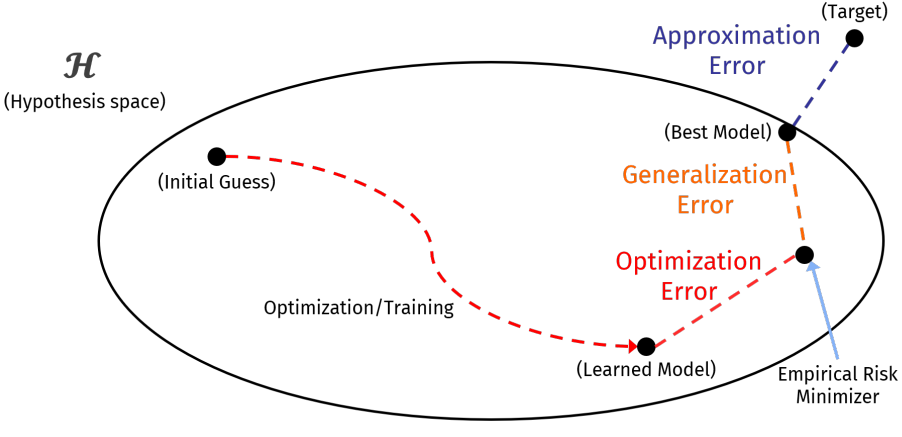
Approximation, optimisation and generalisation



Approximation, optimisation and generalisation

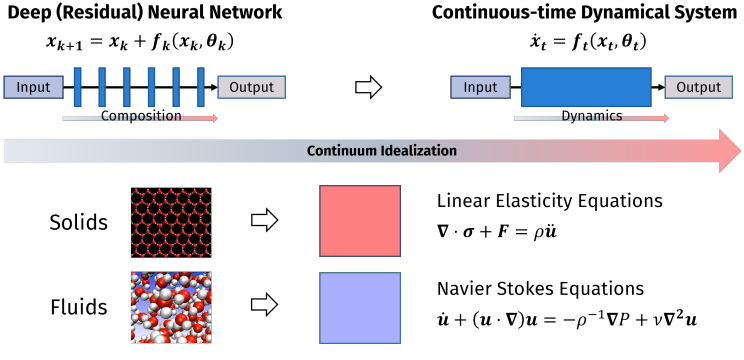


Approximation, optimisation and generalisation



How does the compositional structure affect approximation, optimisation and generalisation?

The dynamics viewpoint of deep learning



W. E, "A Proposal on Machine Learning via Dynamical Systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, 2017

E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, 2017

Q. Li, L. Chen, C. Tai, and W. E, "Maximum principle based algorithms for deep learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, 2017

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, 2018

Introduction

Control theory

Controlling differential equations

Consider the following ODE control system

$$\dot{x}_t = f(x_t, \theta_t), \qquad x_0 \in \mathbb{R}^d, t \in [0, T]$$

where

- State $x_t \in \mathbb{R}^d$
- Dynamics $f : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$
- Control $\theta_t \in \Theta \subset \mathbb{R}^m$ (control set)

Two main mathematical questions for control

Given a control system

$$\dot{x}_t = f(x_t, \theta_t), \qquad x_0 \in \mathbb{R}^d, t \in [0, T],$$

two interesting questions can be posed:

- Controllability: For any $z \in \mathbb{R}^d$, does there exist $T > 0$ and controls $\boldsymbol{\theta} = \{\theta_t \in \Theta : t \in [0, T]\}$ such that $x_T = z$? If this holds for any $x_0, z \in \Omega$, we say that the system is controllable in Ω
- Optimal control: minimise some cost functional

$$\inf_{\boldsymbol{\theta}} \underbrace{\Phi(x_T)}_{\text{Terminal cost}} + \int_0^T \underbrace{L(x_t, \theta_t)}_{\text{Running cost}} \, dt$$

The control family

It is useful to adopt a more geometric view

To this end, define the family of vector fields, called the control family

$$\mathcal{F} = \{f(\cdot, \theta) : \theta \in \Theta\}$$

The control family

It is useful to adopt a more geometric view

To this end, define the family of vector fields, called the control family

$$\mathcal{F} = \{f(\cdot, \theta) : \theta \in \Theta\}$$

One the recasts

$$\dot{x}_t = f(x_t, \theta_t), \quad \theta_t \in \Theta \qquad \rightarrow \qquad \dot{x}_t = f_t(x_t), \quad f_t \in \mathcal{F}$$

Flows maps generated by $f \in \mathcal{F}$ is denoted by $x_0 \mapsto e^{tf}(x_0)$

Controllability from the geometric view

What points can be reached by flows driven by $\mathcal{F}$?

Controllability from the geometric view

What points can be reached by flows driven by $\mathcal{F}$?

Let us assume from now on that $\mathcal{F}$ is symmetric, i.e. $f \in \mathcal{F} \implies -f \in \mathcal{F}$

A local analysis shows for any $f, g \in \mathcal{F}$

$$e^{tf}(x_0) = x_0 + tf(x_0) + o(t)$$
$$e^{a_1 tf} \circ e^{a_2 tg}(x_0) = x_0 + t\left(a_1 f + a_2 g\right)(x_0) + o(t)$$

Thus, points in $\operatorname{span}(\mathcal{F})$ are reachable

Controllability from the geometric view

What points can be reached by flows driven by $\mathcal{F}$?

Let us assume from now on that $\mathcal{F}$ is symmetric, i.e. $f \in \mathcal{F} \implies -f \in \mathcal{F}$

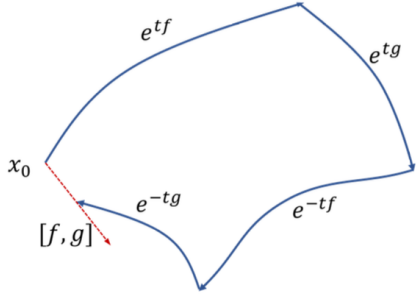
A local analysis shows for any $f, g \in \mathcal{F}$

$$e^{tf}(x_0) = x_0 + tf(x_0) + o(t)$$
$$e^{a_1 tf} \circ e^{a_2 tg}(x_0) = x_0 + t\left(a_1 f + a_2 g\right)(x_0) + o(t)$$

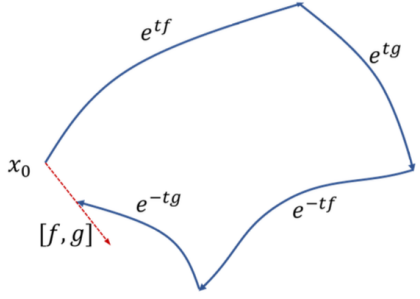
Thus, points in $\operatorname{span}(\mathcal{F})$ are reachable

Is this all that are reachable?

The Lie bracket



The Lie bracket



Local analysis shows

$$e^{-tg} \circ e^{-tf} \circ e^{tg} \circ e^{tf}(x_0) = x_0 + t^2[f,g](x_0) + o\left(t^2\right)$$

where $[f,g] = \nabla g \cdot f - \nabla f \cdot g$ is the Lie bracket

Then, $\mathrm{Span}\{\mathcal{F}, \{[f,g]\}\}$ is again reachable

A controllability result

Repeating the previous procedure we obtain

$$\text{Lie } \mathcal{F} := \text{Span}\left\{[f_1, [\ldots [f_{k-1}, f_k]\ldots]] \mid f_i \in \mathcal{F}, k \in \mathbb{N}\right\}$$

and it's not hard to see that all points in Lie $\mathcal{F}$ are reachable!

W.-L. Chow, "Über systeme von liearren partiellen differentialgleichungen erster ordnung," *Mathematische Annalen*, vol. 117, no. 1, 1940

P. K. Rashevsky, "About connecting two points of a completely nonholonomic space by admissible curve," *Uch. Zapiski Ped. Inst. Libknechta*, vol. 2, 1938

A controllability result

Repeating the previous procedure we obtain

$$\text{Lie } \mathcal{F} := \text{Span}\,\{[f_1, [\ldots [f_{k-1}, f_k] \ldots]] \mid f_i \in \mathcal{F}, k \in \mathbb{N}\}$$

and it's not hard to see that all points in Lie $\mathcal{F}$ are reachable!

Note: Lie $\mathcal{F}$ can be much larger than Span $\mathcal{F}$!

Example: $\mathcal{F} = \{1, x, x^2, x^3\}$, and Lie $\mathcal{F}$ contains all polynomials!

W.-L. Chow, "Über systeme von liearren partiellen differentialgleichungen erster ordnung," *Mathematische Annalen*, vol. 117, no. 1, 1940

P. K. Rashevsky, "About connecting two points of a completely nonholonomic space by admissible curve," *Uch. Zapiski Ped. Inst. Libknechta*, vol. 2, 1938

A controllability result

Repeating the previous procedure we obtain

$$\text{Lie } \mathcal{F} := \text{Span}\{[f_1, [\dots [f_{k-1}, f_k]\dots]] \mid f_i \in \mathcal{F}, k \in \mathbb{N}\}$$

and it's not hard to see that all points in Lie $\mathcal{F}$ are reachable!

Note: Lie $\mathcal{F}$ can be much larger than Span $\mathcal{F}$!

Example: $\mathcal{F} = \{1, x, x^2, x^3\}$, and Lie $\mathcal{F}$ contains all polynomials!

Theorem [Chow 40, Rashevsky 38]

Let $\Omega \subset \mathbb{R}^d$ be open. Let $\mathcal{F}$ be a symmetric family of smooth vector fields and dim Lie $\mathcal{F} = d$, then the system $\dot{x}_t = f_t(x_t)$, $f_t \in \mathcal{F}$ is controllable in Ω.

W.-L. Chow, "Über systeme von liearren partiellen differentialgleichungen erster ordnung," *Mathematische Annalen*, vol. 117, no. 1, 1940

P. K. Rashevsky, "About connecting two points of a completely nonholonomic space by admissible curve," *Uch. Zapiski Ped. Inst. Libknechta*, vol. 2, 1938

The Bolza problem of optimal control

Consider the following optimal control problem, also called a Bolza problem

$$\min_{\boldsymbol{\theta}} J[\boldsymbol{\theta}] \equiv \Phi(x_T) + \int_0^T L(x_t, \theta_t) dt$$

subject to

$$\dot{x}_t = f(x_t, \theta_t), \qquad x_0 \in \mathbb{R}^d \text{ is given}$$

We seek

- Necessary conditions: if $\boldsymbol{\theta}$ minimises J, what properties must it have?
- Sufficient conditions: what properties ensures that $\boldsymbol{\theta}$ minimises J?

Necessary conditions: Pontryagin's maximum principle

Let us define the Hamiltonian $H(x, p, \theta) = p^\top f(x, \theta) - L(x, \theta)$

Theorem [Pontryagin's maximum principle]

Let $\boldsymbol{\theta}^*$ be a bounded measurable optimal control and $\boldsymbol{x}^*$ be its corresponding state trajectory. Then, there exists an absolutely continuous process $\boldsymbol{p} = \{p_t \in \mathbb{R}^d : t \in [0, T]\}$ such that

$$
\begin{aligned}
\dot{x}_t^* &= \nabla_p H(x_t^*, p_t^*, \theta_t^*), & x_0^* &= x_0 \\
\dot{p}_t^* &= -\nabla_x H(x_t^*, p_t^*, \theta_t^*), & p_T^* &= -\nabla \Phi(x_T^*) \\
H(t, x_t^*, p_t^*, \theta_t^*) &\geq H(t, x_t^*, p_t^*, \theta), & &\forall \theta \in \Theta \text{ and } a.e.\, t \in [0, T]
\end{aligned}
$$

V. G. Boltyanskii, R. V. Gamkrelidze, and L. S. Pontryagin, "The theory of optimal processes. {I}. {T}he maximum principle," TRW Space Technology Labs, Los Angeles, California, 1960

The value function

To derive more general conditions, we define the value function

$$V(s,z) := \inf_{\boldsymbol{\theta}} \int_s^T L(x_t,\theta_t)dt + \Phi(x_T)$$

subject to

$$\dot{x}_t = f(x_t,\theta_t), \qquad t \in [s,T], \qquad x_s = z.$$

- $V(0,x_0)$ is the optimal cost of the Bolza problem
- This expands the Bolza problem to a family of problems, but we can derive a recursion!

Hamilton-Jacobi-Bellman equation

Let $V : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ be the value function. Then, V is the unique viscosity solution of the Hamilton-Jacobi-Bellman equation

$$\partial_t V(t,x) + \inf_{\theta \in \Theta} \left\{ L(x,\theta) + [\nabla_x V(t,x)]^\top f(x,\theta) \right\} \qquad (t,x) \in (0,T) \times \mathbb{R}^d$$
$$V(T,x) = \Phi(x)$$

Moreover, any minimizer of $\min_{\theta \in \Theta} \left\{ L(x_t,\theta) + [\nabla_x V(t,x_t)]^\top f(x_t,\theta) \right\}$ is an optimal control.

R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, 1966

M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, 1983

Approximation and controllability

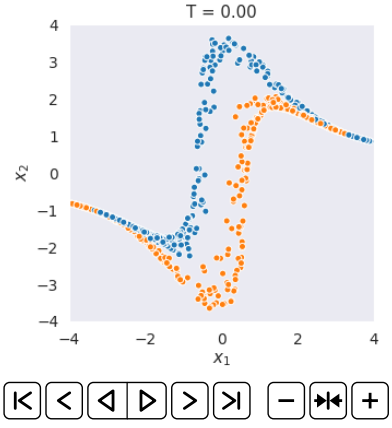
Controllability and approximation theory for deep NNs

Let us now return to the problem of approximation for continuum deep ResNets

We will see that it is intimately connected with controllability, but with key differences

- We need to handle an (infinite) ensemble of points together
- Is arbitrary point matching enough?

How do dynamics approximate functions?



Binary Classification Problem

Not linearly separable!

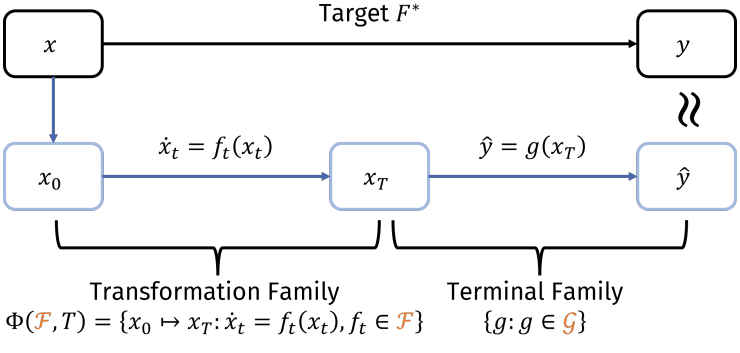
Evolve with the dynamics

$$\dot{x}_{t,1} = -x_{t,2}\sin(t)$$
$$\dot{x}_{t,2} = -\frac{1}{2}(1 - x_{t,1}^2)x_{t,2} + x_{t,1}\cos(t)$$

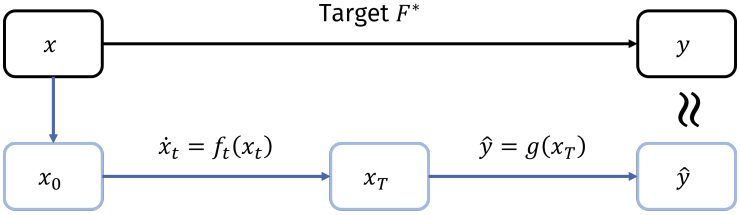
Classify using linear classifier at the end:

$$g(x_T) = \mathbf{1}_{x_{T,1}>0}$$

How do dynamics approximate functions?



How do dynamics approximate functions?



Dynamical Hypothesis Space

$$\mathcal{H}(\mathcal{F},\mathcal{G}) = \cup_{T \geq 0}\{g \circ \varphi : g \in \mathcal{G}, \varphi \in \Phi(\mathcal{F}, T)\}$$

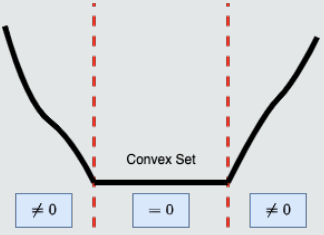
Universal approximation by dynamics

Theorem [LLS, 22]

Let $d \geq 2$ and Suppose that $\mathcal{F}, \mathcal{G}$ satisfy

1. $\mathcal{G}$ contains a surjective Lipschitz function
2. $\mathcal{F}$ is restricted affine invariant
3. $\overline{\mathrm{conv}(\mathcal{F})}$ contains a well function

Then, $\mathcal{H}(\mathcal{F}, \mathcal{G})$ is dense in L^p_{loc}, $p \in [1, \infty)$.



- Applies to most dense ResNet type architectures with width$\geq d$
- $d = 1$ case requires target to be increasing

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," *J. Eur. Math. Soc.*, 13, 2022

Extension to symmetric functions

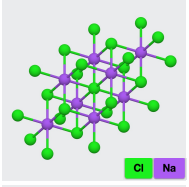
Functions invariant to (some) permutations of its indices

$$F^*(x) = F^*(s(x)) \quad \text{where} \quad s(x)_i = x_{s(i)}, \quad s \in G \text{ (subgroup of } S_d)$$

Examples

- Convolutional NN:
  $G = T(\text{Group of Translations})$
- DeepSets: $G = S_d$
- Material Property Prediction from CIF data: $G = S_{d_1} \times S_{d_2}$

Similar sufficient conditions for approximation [LLS 22b, c]



	X-coord	Y-coord	Z-coord
Na	0	0	0
Cl	0.5	0.5	0.5

	X-coord	Y-coord	Z-coord
Ca	0	0	0
Ca	0.5	0.5	0.5
C	0.25	0.25	0.25
C	0.75	0.75	0.75
O	0.0073	0.4927	0.75
O	0.25	0.9927	0.5073
O	0.4927	0.75	0.0073
O	0.5073	0.25	0.9927
O	0.75	0.0073	0.4927
O	0.9927	0.5073	0.25

Q. Li, T. Lin, and Z. Shen, "Deep Neural Network Approximation of Invariant Functions through Dynamical Systems," 18, 2022

Q. Li, T. Lin, and Z. Shen, "On the Universal Approximation Property of Deep Fully Convolutional Neural Networks," 25, 2022

Approximation results via controllability

It is shown in [LLS, 22] that for universal approximation it is enough to show that flow maps of

$$\dot{x}_t = f_t(x_t), \qquad f_t \in \mathcal{F}$$

is dense in $L^p(\mathbb{R}^d, \mathbb{R}^d)$

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," *J. Eur. Math. Soc.*, 13, 2022

Approximation results via controllability

It is shown in [LLS, 22] that for universal approximation it is enough to show that flow maps of

$$\dot{x}_t = f_t(x_t), \qquad f_t \in \mathcal{F}$$

is dense in $L^p(\mathbb{R}^d, \mathbb{R}^d)$

This is close to requiring that any N distinct initial points can be matched to N distinct target points

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," *J. Eur. Math. Soc.*, 13, 2022

Approximation results via controllability

It is shown in [LLS, 22] that for universal approximation it is enough to show that flow maps of

$$\dot{x}_t = f_t(x_t), \qquad f_t \in \mathcal{F}$$

is dense in $L^p(\mathbb{R}^d, \mathbb{R}^d)$

This is close to requiring that any N distinct initial points can be matched to N distinct target points

This leads to the definition of universal interpolation: ability to match two arbitrary sets of source-target points

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," *J. Eur. Math. Soc.*, 13, 2022

Universal interpolation

Variety of ways to achieve the universal interpolation property (UIP)

[CLT 20] constructs a $\mathcal{F}$ containing 5 polynomial vector fields with Lie $\mathcal{F}$ containing all polynomials, achieving UIP

[TG 22] shows that if the activation function of continuum ResNets are chosen appropriately (satisfying an ODE), UIP is achieved. This includes tanh and sigmoid activations

[RZ 21] shows that ReLU networks can achieve UIP by construction

---

C. Cuchiero, M. Larsson, and J. Teichmann, "Deep Neural Networks, Generic Universal Interpolation, and Controlled ODEs," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 3, 2020

P. Tabuada and B. Gharesifard, "Universal Approximation Power of Deep Residual Neural Networks Through the Lens of Control," *IEEE Transactions on Automatic Control*, 2022

D. Ruiz-Balet and E. Zuazua, "Neural ODE control for classification, approximation and transport," 2021

Two questions

How difficult is it to achieve UIP?

Is achieving UIP enough to get UAP (density)?

UIP vs UAP

Characterising UIP (ensemble controllability) in general is difficult, but often we have affine invariance

$$f \in \mathcal{F} \implies Af(B \cdot -b) \in \mathcal{F}, \qquad \forall \quad A, B \in \mathbb{R}^{d\times d}, c \in \mathbb{R}^d$$

which is satisfied by many NN architectures

J. Cheng, Q. Li, T. Lin, and Z. Shen, "Interpolation, approximation and controllabilit of deep neural networks," *In Preparation*, 2023

UIP vs UAP

Characterising UIP (ensemble controllability) in general is difficult, but often we have affine invariance

$$f \in \mathcal{F} \implies Af(B \cdot -b) \in \mathcal{F}, \qquad \forall \quad A, B \in \mathbb{R}^{d \times d}, c \in \mathbb{R}^d$$

which is satisfied by many NN architectures

Under the assumption of affine invariance, we obtain [CLLS 23]

- A characterisation of UIP: UIP holds if and only if $\mathcal{F}$ contains a non-linear function
- UIP and UAP are in general independent, but are equivalent under special classes of target mappings

J. Cheng, Q. Li, T. Lin, and Z. Shen, "Interpolation, approximation and controllabilit of deep neural networks," *In Preparation*, 2023

Approximation Rates?

So far, the discussion is on density-type results

What about approximation rates, e.g. minimum T required to achieve a specified approximation error?

- In 1D, rates can be obtained [LLS, 22]
- In general, problem is much more delicate
  - Requires identification of right function spaces, complexity measures, etc.
  - Connections to switching controls [RZ 21], compositional features [KG 22], etc.

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," *J. Eur. Math. Soc.*, 13, 2022

D. Ruiz-Balet and E. Zuazua, "Neural ODE control for classification, approximation and transport," 2021

W. Kang and Q. Gong, "Feedforward Neural Networks and Compositional Functions with Applications to Dynamical Systems," *SIAM J. Control Optim.*, vol. 60, no. 2, 30, 2022

Optimisation and optimal control

Optimisation and optimal control

Theory

Bolza problem revisited

Recall the Bolza problem

$$\min_{\boldsymbol{\theta}} J[\boldsymbol{\theta}] \equiv \Phi(x_T) + \int_0^T L(x_t, \theta_t) dt$$

subject to

$$\dot{x}_t = f(x_t, \theta_t), \qquad x_0 \in \mathbb{R}^d \text{ is given}$$

In deep continuum ResNets, we are using the dynamics to steer an ensemble/distribution of points!

Deep learning as mean-field optimal control

Learning on dynamical hypothesis spaces is a variant of the Bolza problem

$$\inf_{\boldsymbol{\theta}} J[\boldsymbol{\theta}] := \mathbb{E}_{\mu^*}\left[\underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{L(x_t, \theta_t)}_{\text{Regularizer}}\, dt\right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \le t \le T \qquad \underbrace{(x_0, y) \sim \mu^*}_{\text{input-output distribution}}$$

Deep learning as mean-field optimal control

Learning on dynamical hypothesis spaces is a variant of the Bolza problem

$$\inf_{\boldsymbol{\theta}} J[\boldsymbol{\theta}] := \mathbb{E}_{\mu^*}\left[\underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{L(x_t, \theta_t)}_{\text{Regularizer}}\, dt\right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \le t \le T \qquad \underbrace{(x_0, y) \sim \mu^*}_{\text{input-output distribution}}$$

This is a mean-field optimal control problem, because we need to select $\boldsymbol{\theta}$ that controls not one, but an entire distribution of inputs and outputs

Deep learning as mean-field optimal control

Learning on dynamical hypothesis spaces is a variant of the Bolza problem

$$\inf_{\boldsymbol{\theta}} J[\boldsymbol{\theta}] := \mathbb{E}_{\mu^*}\left[\underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{L(x_t, \theta_t)}_{\text{Regularizer}}\, dt\right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \le t \le T \qquad \underbrace{(x_0, y) \sim \mu^*}_{\text{input-output distribution}}$$

This is a mean-field optimal control problem, because we need to select $\boldsymbol{\theta}$ that controls not one, but an entire distribution of inputs and outputs

- Theoretical questions: Necessary and sufficient conditions for optimality
- Practical questions: Understanding, improving learning algorithms

Mean-field Pontryagin's maximum principle

Let $\boldsymbol{\theta}^* \in L^\infty([0,T],\Theta)$ be an optimal control, and $\boldsymbol{x}^*$ the corresponding controlled trajectory. Then, there exists an absolutely continuous stochastic process $\boldsymbol{p}^*$ such that

$$\dot{x}_t^* = \nabla_p H(x_t^*, p_t^*, \theta_t^*) \qquad x_t^* = x,$$
$$\dot{p}_t^* = -\nabla_x H(x_t^*, p_t^*, \theta_t^*), \qquad p_T^* = -\nabla_x \Phi(x_T^*, y),$$
$$\mathbb{E}_\mu H(x_t^*, p_t^*, \theta_t^*) \geq \mathbb{E}_\mu H(x_t^*, p_t^*, \theta), \qquad \forall\, \theta \in \Theta, \quad a.e.\, t \in [0,T],$$
$$(x,y) \sim \mu$$

W. E, J. Han, and Q. Li, "A mean-field optimal control formulation of deep learning," *Research in the Mathematical Sciences*, vol. 6, no. 1, 2019

The value function

We can define an analogous value function

$$V : [0, T] \times \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$$

as

$$V(s, \rho) = \inf_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\rho} \left[\int_t^T L(x(t), \theta(t)) dt + \Phi(x(T), y) \right]$$

subject to

$$\dot{x}(t) = f(x(t), \theta(t)), \qquad t \in [s, T], \qquad x(s) = x.$$

Key difference: the state is now a distribution over $\mathbb{R}^d$, instead of a vector in $\mathbb{R}^d$

Mean-field Hamilton-Jacobi-Bellman equation

The value function is the unique viscosity solution to the following mean-field Hamilton-Jacobi-Bellman equation

$$\partial_t V(t,\rho) + \inf_{\theta\in\Theta} \int_{\mathbb{R}^{d+1}} L(x,\theta) + [\partial_\rho V(t,\rho)(x,y)]^\top [f(x,\theta),0] d\rho(x,y) = 0,$$
$$V(T,\rho) = \int_{\mathbb{R}^{d+1}} \Phi(x,y) d\rho(x,y).$$

Note: $\partial_\rho V$ is defined via the "lifting" technique commonly used in mean-field games

W. E, J. Han, and Q. Li, "A mean-field optimal control formulation of deep learning," *Research in the Mathematical Sciences*, vol. 6, no. 1, 2019

P. Cardaliaguet, "Notes on mean field games," Technical report, 2010

Optimisation and optimal control

Applications

Backpropagation and the MSA algorithm

The mean-field Pontryagin's maximum principle gives us a method to find an optimal control candidate via the method of successive approximations (MSA)

$$\dot{x}_t^n = f(x_t^n, \theta_t^n) \qquad x^n(0) = x$$
$$\dot{p}_t^n = -\nabla_x H(x_t^n, p_t^n, \theta_t^n) \qquad p_T^n = -\nabla_x \Phi(x_T^n, y)$$
$$\theta_t^{n+1} = \arg\max_{\theta \in \Theta} \mathbb{E}_{(x,y)} H(x_t^n, p_t^n, \theta).$$

Q. Li, L. Chen, C. Tai, and W. E, "Maximum principle based algorithms for deep learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, 2017

Q. Li and S. Hao, "An Optimal Control Approach to Deep Learning and Applications to Discrete-Weight Neural Networks," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., vol. 80, 2018

Backpropagation and the MSA algorithm

The mean-field Pontryagin's maximum principle gives us a method to find an optimal control candidate via the method of successive approximations (MSA)

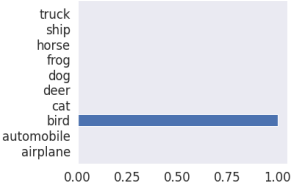
$$\dot{x}_t^n = f(x_t^n, \theta_t^n) \qquad x^n(0) = x$$
$$\dot{p}_t^n = -\nabla_x H(x_t^n, p_t^n, \theta_t^n) \qquad p_T^n = -\nabla_x \Phi(x_T^n, y)$$
$$\theta_t^{n+1} = \operatorname*{arg\,max}_{\theta \in \Theta} \mathbb{E}_{(x,y)} H(x_t^n, p_t^n, \theta).$$

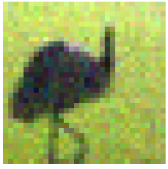
- If the arg max step is replaced by gradient ascent, this is just the back-propagation algorithm
- Replacing it with other methods leads to alternatives [LTE 17]
- E.g. can handle case where Θ is finite, e.g. binary/ternary networks [LH 18]

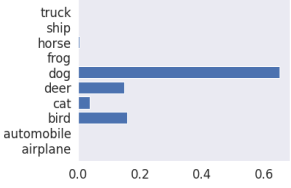
Q. Li, L. Chen, C. Tai, and W. E, "Maximum principle based algorithms for deep learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, 2017

Q. Li and S. Hao, "An Optimal Control Approach to Deep Learning and Applications to Discrete-Weight Neural Networks," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., vol. 80, 2018

Adversarial examples



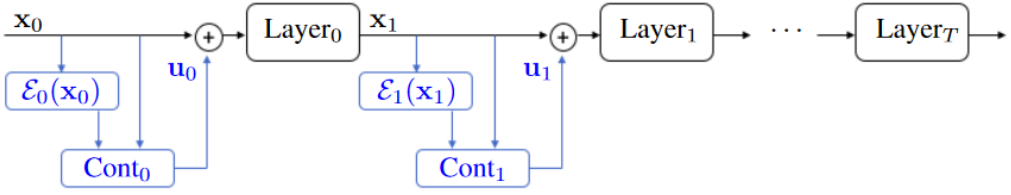




Reason: perturbations get magnified by propagation through layers

The control viewpoint of adversarial defense

We can use closed loop control to stabilize propagation of features through a deep network [CLZ 20, 22]



This leads to increased adversarial robustness without the need to retrain!

Z. Chen, Q. Li, and Z. Zhang, "Towards Robust Neural Networks via Close-loop Control," in *International Conference on Learning Representations*, 28, 2020

Z. Chen, Q. Li, and Z. Zhang, "Self-Healing Robust Neural Networks via Closed-Loop Control," *Journal of Machine Learning Research*, vol. 23, no. 319, 1, 2022

Summary and outlook

Summary and outlook

Take-home messages

- Composition=dynamics
- Approximation≈controllability
- Optimisation≈optimal control

Many unanswered questions:

- Approximation rates and approximation spaces
- The optimisation landscape of deep models
- Generalisation≈?
  (Connections: Γ-convergence, propagation of chaos, mean-field games)