# Approximation Theory of Deep Learning from the Dynamical Viewpoint

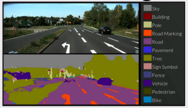
Qianxiao Li
Department of Mathematics, National University of Singapore
https://blog.nus.edu.sg/qianxiaoli

3rd Symposium on Machine Learning and Dynamical Systems
Fields Institute, Toronto, Canada
29 Sep 2022

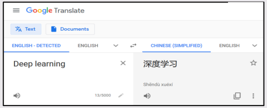
# Background

# Deep Learning: Theory vs Practice

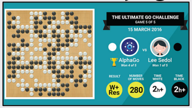
Self-Driving




Machine Translation



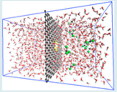

Game-Playing




**Practical Success**
**vs.**
**Theoretical Mystery**

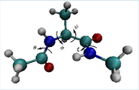
Science and Engineering

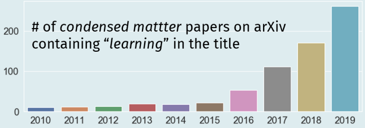
*Molecular Dynamics*



*Phase Transitions*


*Conformational Changes*





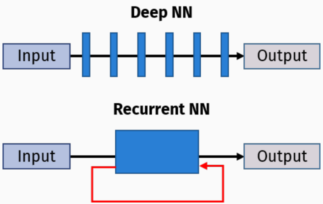

**Emerging Applications**

# What's new?

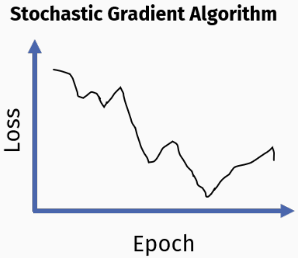
# What’s new?

Compositional/dynamical structures

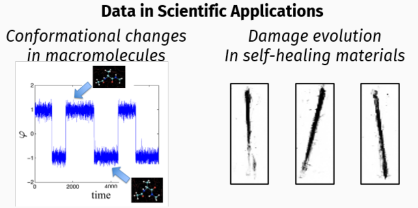
## Models




## Algorithms




## Data



# Composition is Dynamics

## Composition

$$y = F_T \circ F_{T-1} \circ \cdots \circ F_0(x)$$

## Dynamics

$$\begin{aligned} y &= x_T, & x &= x_0 \\ x_{t+1} &= F_t(x_t) & t &= 0, 1, \dots, T-1 \end{aligned}$$

# Composition is Dynamics

## Composition

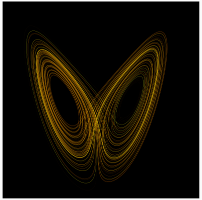
$$y = F_T \circ F_{T-1} \circ \cdots \circ F_0(x)$$

## Dynamics

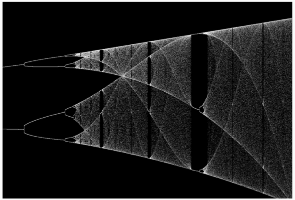
$$\begin{aligned} y &= x_T, & x &= x_0 \\ x_{t+1} &= F_t(x_t) & t &= 0, 1, \dots, T-1 \end{aligned}$$

Such connections underlies the study of dynamical systems

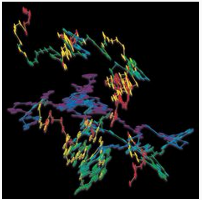
Continuous



Discrete



Stochastic

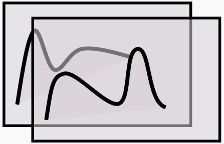


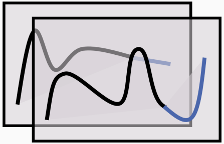
# Supervised Learning

Cat

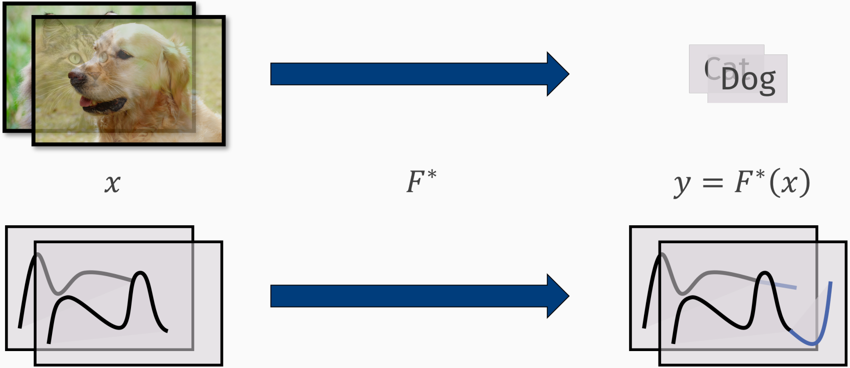
Dog

$x$ $\quad$ $F^*$ $\quad$ $y = F^*(x)$



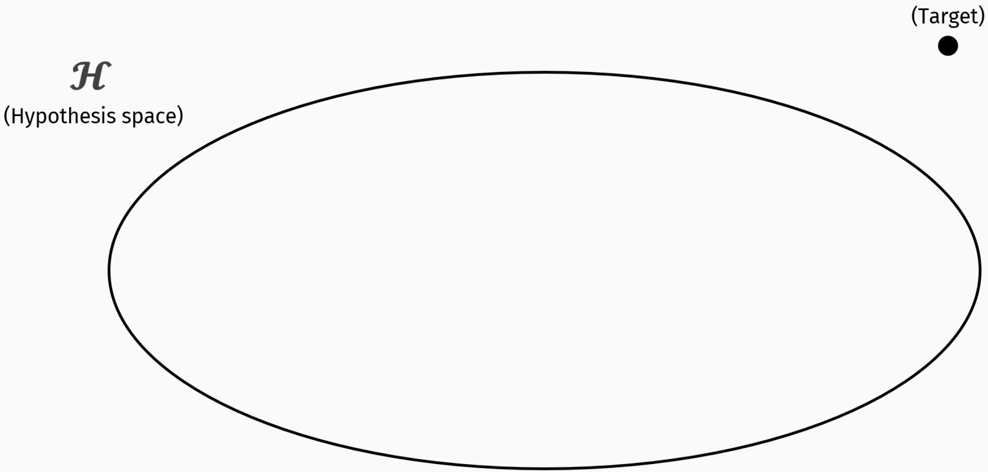


# Supervised Learning

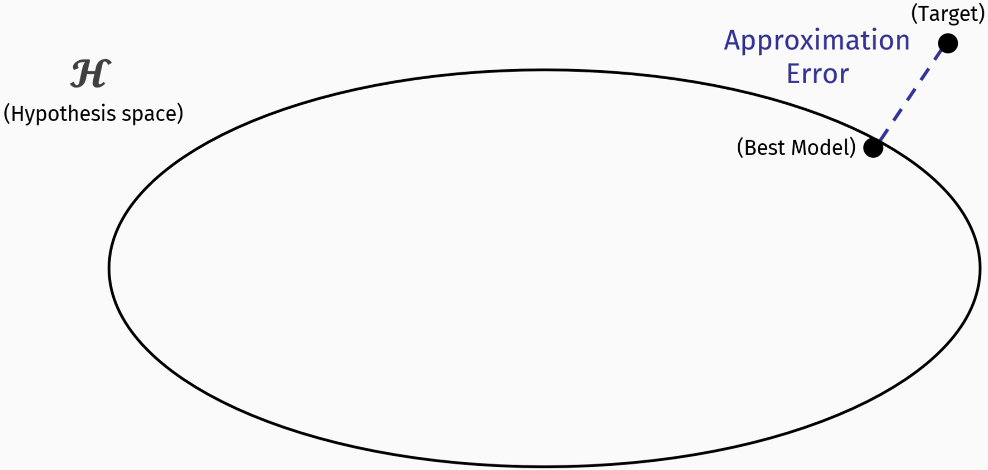



Goal: Learn/approximate target $F^*$

# The Problem of Approximation



# The Problem of Approximation



# The Problem of Approximation

Given a hypothesis space $\mathcal{H}$ and a target (concept) space $\mathcal{C}$, we seek two types of approximation results

- Universal Approximation (Density)

> For each $F^* \in \mathcal{C}$ and $\epsilon > 0$, there exist $\hat{F} \in \mathcal{H}$ such that $\|F^* - \hat{F}\| \leq \epsilon$

# The Problem of Approximation

Given a hypothesis space $\mathcal{H}$ and a target (concept) space $\mathcal{C}$, we seek two types of approximation results

- Universal Approximation (Density)

$$\boxed{\text{For each } F^* \in \mathcal{C} \text{ and } \epsilon > 0, \text{ there exist } \hat{F} \in \mathcal{H} \text{ such that } \|F^* - \hat{F}\| \le \epsilon}$$

- Approximation Rates. Let $\mathcal{H} = \cup_m \mathcal{H}_m$, where $\mathcal{H}_m \subset \mathcal{H}_{m+1}$, $m$ measures size of hypothesis space (approximation budget)

$$\boxed{\inf_{\hat{F} \in \mathcal{H}_m} \|F^* - \hat{F}\| \le \text{Complexity}(F^*)\text{rate}(m), \qquad \text{rate}(m) \to 0}$$

## Example: Approximation by Trigonometric Polynomials

Consider

- $\mathcal{C} = C^{\alpha}_{\text{per}}([0, 2\pi], \mathbb{R})$ (periodic $C^{\alpha}$ functions)
- $\mathcal{H}_m = \left\{ \sum_{i=0}^{m-1} a_i \cos(ix) + b_i \sin(ix) : a_i, b_i \in \mathbb{R} \right\}$ (trigonometric polynomials)

# Example: Approximation by Trigonometric Polynomials

Consider

- $\mathcal{C} = C^{\alpha}_{\text{per}}([0, 2\pi], \mathbb{R})$ (periodic $C^{\alpha}$ functions)
- $\mathcal{H}_m = \left\{ \sum_{i=0}^{m-1} a_i \cos(ix) + b_i \sin(ix) : a_i, b_i \in \mathbb{R} \right\}$ (trigonometric polynomials)

Then,

- Density: (Stone) Weierstrass Theorem (gives sufficient conditions)
- Approximation Rate: Jackson's Theorem

$$\inf_{\hat{F} \in \mathcal{H}_m} \|F^* - \hat{F}\|_{\mathcal{C}} \leq \frac{C(\alpha) \max_{i \leq \alpha} \|F^{*(i)}\|_{\mathcal{C}}}{m^{\alpha}}$$

## Example: Approximation by Trigonometric Polynomials

Consider

- $\mathcal{C} = C^{\alpha}_{\text{per}}([0, 2\pi], \mathbb{R})$ (periodic $C^{\alpha}$ functions)
- $\mathcal{H}_m = \left\{ \sum_{i=0}^{m-1} a_i \cos(ix) + b_i \sin(ix) : a_i, b_i \in \mathbb{R} \right\}$ (trigonometric polynomials)

Then,

- Density: (Stone) Weierstrass Theorem (gives sufficient conditions)
- Approximation Rate: Jackson's Theorem

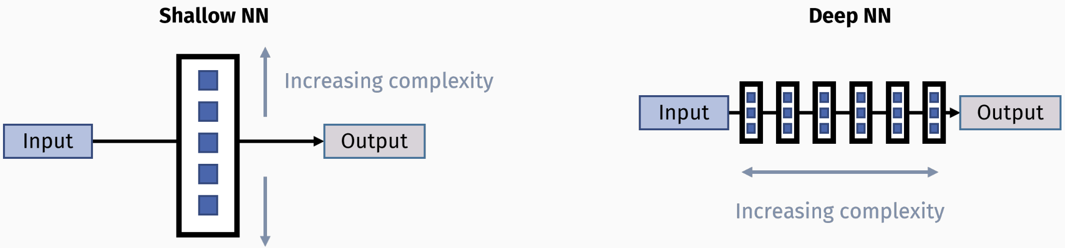
$$\inf_{\hat{F} \in \mathcal{H}_m} \|F^* - \hat{F}\|_{\mathcal{C}} \leq \frac{C(\alpha) \max_{i \leq \alpha} \|F^{*(i)}\|_{\mathcal{C}}}{m^{\alpha}}$$

**Insight:** Efficient approximation if $F^*$ is smooth (small gradient norm)

# Approximation Theory of Deep Learning: Stone-Weierstrass and Jackson type results when $\mathcal{H}$ and $\mathcal{C}$ have compositional/dynamical structures

# Approximation Theory of DL: Function Approximation

# Dynamical Structures in Deep Learning



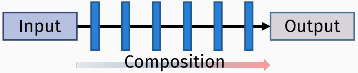

DL builds complexity through composition/dynamics

How to achieve universal approximation this way?

# The Continuum Idealization of Residual Networks

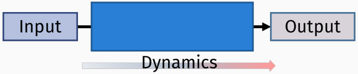
**Deep (Residual) Neural Network**

$$x_{k+1} = x_k + f_k(x_k, \theta_k)$$



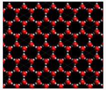

**Continuous-time Dynamical System**

$$\dot{x}_t = f_t(x_t, \theta_t)$$




**Continuum Idealization**

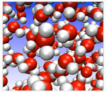
Solids



Linear Elasticity Equations

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F} = \rho \ddot{\boldsymbol{u}}$$

Fluids



Navier Stokes Equations

$$\dot{\boldsymbol{u}} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\rho^{-1}\nabla P + \nu \nabla^2 \boldsymbol{u}$$

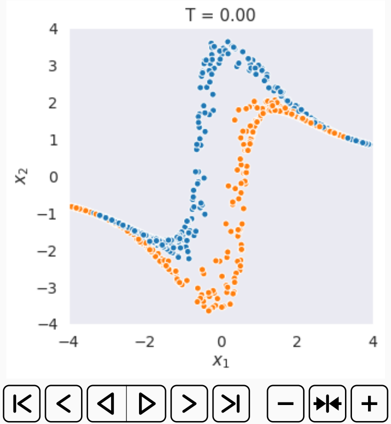

W. E, "A Proposal on Machine Learning via Dynamical Systems," Communications in Mathematics and Statistics, vol. 5, no. 1, 2017

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

# How do dynamics approximate functions?




**Binary Classification Problem**

Not linearly separable!

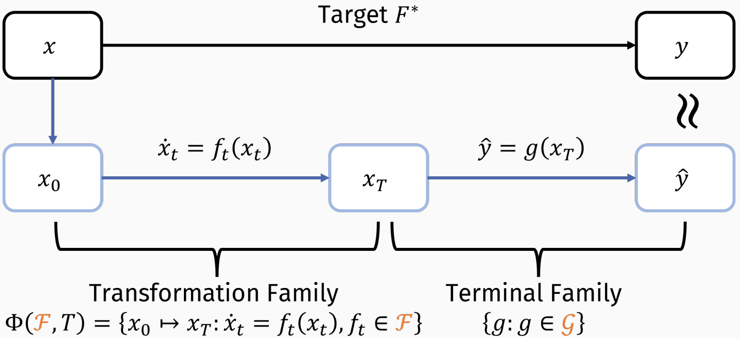
Evolve with the dynamics

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

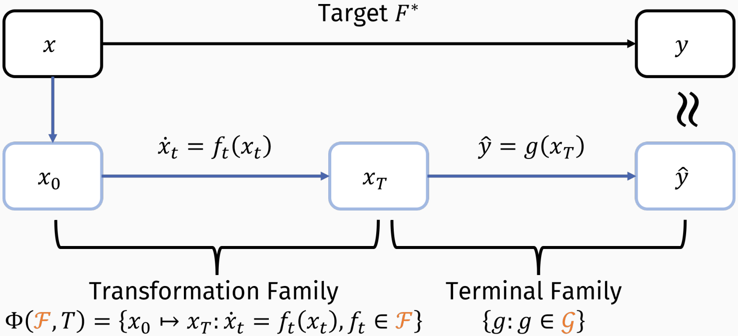
Classify using linear classifier at the end:

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

# How do dynamics approximate functions?



# How do dynamics approximate functions?



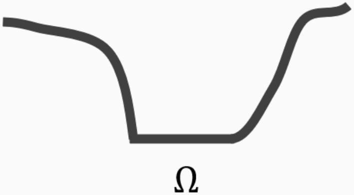

**Dynamical Hypothesis Space**

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

# Universal Approximation by Dynamics

- Sufficient conditions for universal approximation by dynamics [LLS, 22]
- In dimension $\geq 2$, always possible under mild conditions

1. $\mathcal{G}$ covers range of $F^*$
2. $\mathcal{F}$ is restricted affine invariant
3. $\overline{\mathrm{Conv}}(\mathcal{F})$ contains a well function




- In dimension 1, only increasing functions if $\mathcal{G} = \{\mathrm{id}\}$
- Connections with controllability [Cuchiero et al, 20; Tabuda & Gharesifard, 22]

---

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," Journal of the European Mathematical Society, 2022

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

# Approximation of Symmetric Functions by Dynamical Hypothesis Spaces

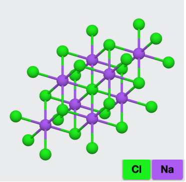
Functions invariant to (some) permutations of its indices

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

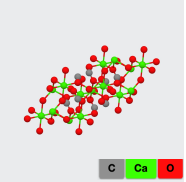
Examples

- Convolutional NN: $G = T$(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 of $G$-invariant functions for any transitive $G$ [LLS, 22b]




| | 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," arXiv, no. arXiv:2208.08707, 2022. arXiv: 2208.08707

# Approximation Rates?

We have a Stone-Weierstrass type result for dynamical/compositional hypothesis spaces

What about Jackson type results?

- In 1D, some crude 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, Barron spaces, compositional features . . .

Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," Journal of the European Mathematical Society, 2022

# Deep Learning as Mean-field Optimal Control

Learning/optimization on dynamical hypothesis spaces:

$$\inf_{\boldsymbol{\theta} \in L^\infty([0,T],\Theta)} J(\boldsymbol{\theta}) := \mathbb{E}_{\mu^*}\left[\underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{R(x_t, \theta_t)}_{\text{Regularizer}} dt\right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \leq t \leq T \qquad (x_0, y) \sim \mu^*$$

## Deep Learning as Mean-field Optimal Control

Learning/optimization on dynamical hypothesis spaces:

$$\inf_{\boldsymbol{\theta} \in L^\infty([0,T],\Theta)} J(\boldsymbol{\theta}) := \mathbb{E}_{\mu^*} \left[ \underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{R(x_t, \theta_t)}_{\text{Regularizer}} \, dt \right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \leq t \leq T \qquad (x_0, y) \sim \mu^*$$

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/optimization on dynamical hypothesis spaces:

$$\inf_{\boldsymbol{\theta} \in L^\infty([0,T],\Theta)} J(\boldsymbol{\theta}) := \mathbb{E}_{\mu^*} \left[ \underbrace{\Phi(x_T, y)}_{\text{Loss}} + \int_0^T \underbrace{R(x_t, \theta_t)}_{\text{Regularizer}} \, dt \right]$$

$$\dot{x}_t = f(x_t, \theta_t) \qquad 0 \leq t \leq T \qquad (x_0, y) \sim \mu^*$$

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

Key questions:

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

# Mathematical Results and Numerical Algorithms

- Necessary and sufficient conditions for optimality
  - Mean-field Pontryagin's maximum principle (PMP) [EHL, 19]
  - Mean-field Hamilton Jacobi Bellman equations (HJB) [EHL, 19]
- Algorithms
  - Training algorithms based on PMP [LCTE, 17], for quantized networks [LH, 18]
  - Close-loop control method to improve adversarial robustness [CLZ, 21]

---


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

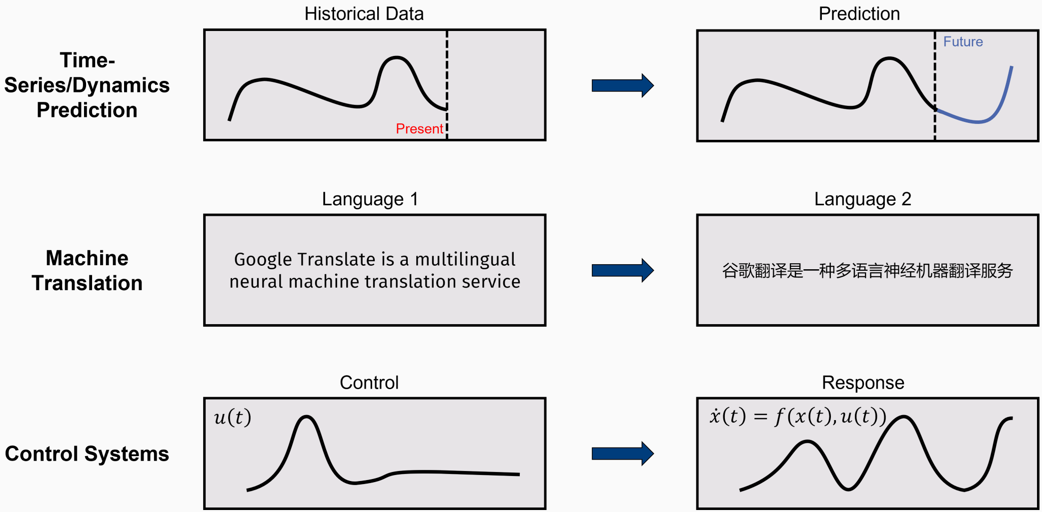
Q. Li, L. Chen, C. Tai, and W. E, "Maximum principle based algorithms for deep learning," The Journal of Machine Learning Research, vol. 18, no. 1, 2017

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 (ICML), vol. 80, 2018

Z. Chen, Q. Li, and Z. Zhang, "Towards robust neural networks via close-loop control," in International Conference on Learning Representations (ICLR), 2021

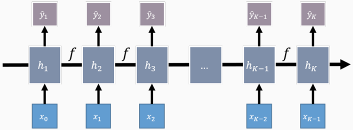
# Approximation Theory of DL: Sequence Modelling

# Sequence Modelling Applications

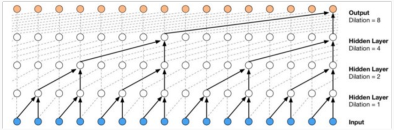


# DL Architectures for Sequence Modelling

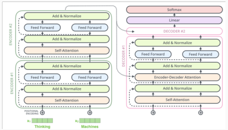
RNN




WaveNet (CNN)




Transformers



**General question:** How are they different? When should we use which?

# Modelling Static vs Dynamic Relationships

## Static setting

(input) $x \in \mathcal{X} = \mathbb{R}^d$

(output) $y \in \mathcal{Y} = \mathbb{R}^n$

(target) $y = F^*(x)$

# Modelling Static vs Dynamic Relationships

## Static setting

(input) $x \in \mathcal{X} = \mathbb{R}^d$

(output) $y \in \mathcal{Y} = \mathbb{R}^n$

(target) $y = F^*(x)$

## Dynamic setting

(input) $\boldsymbol{x} = \{x_k \in \mathbb{R}^d\} \in \mathcal{X}$

(output) $\boldsymbol{y} = \{y_k \in \mathbb{R}^n\} \in \mathcal{Y}$

(target) $y_k = H_k^*(\boldsymbol{x}) \quad \forall \quad k$

# Modelling Static vs Dynamic Relationships

## Static setting

(input) $x \in \mathcal{X} = \mathbb{R}^d$

(output) $y \in \mathcal{Y} = \mathbb{R}^n$

(target) $y = F^*(x)$

## Dynamic setting

(input) $\boldsymbol{x} = \{x_k \in \mathbb{R}^d\} \in \mathcal{X}$

(output) $\boldsymbol{y} = \{y_k \in \mathbb{R}^n\} \in \mathcal{Y}$

(target) $y_k = H_k^*(\boldsymbol{x}) \quad \forall \quad k$

Goal of supervised learning

- Static: learn/approximate the target $F^*$
- Dynamic: learn/approximate the target $\{H_k^*\}$

# An Approximation Theory for Sequence Modelling

Our goal is to derive similar statements like Jackson's Theorem, but for

- $\mathcal{C} \rightarrow$ suitable classes of sequence relationships (functionals, operators)
- $\mathcal{H} \rightarrow$ RNNs, CNNs/WaveNets, Encoder-Decoders, Transformers

# An Approximation Theory for Sequence Modelling

Our goal is to derive similar statements like Jackson's Theorem, but for

- $\mathcal{C} \to$ suitable classes of sequence relationships (functionals, operators)
- $\mathcal{H} \to$ RNNs, CNNs/WaveNets, Encoder-Decoders, Transformers

For each case, we aim to characterize

- What $\mathcal{C}$ can be approximated (efficiently)?
- How does the complexity measure and rate estimate depend on different $\mathcal{H}$?
- How to choose which $\mathcal{H}$ to use in practice?

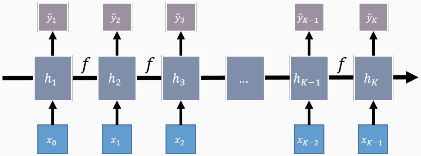
# Recurrent Neural Networks

# The Recurrent Neural Network Hypothesis Space

The recurrent neural network (RNN) architecture

$$h_{k+1} = \sigma(Wh_k + Ux_k), \qquad h_k \in \mathbb{R}^m$$

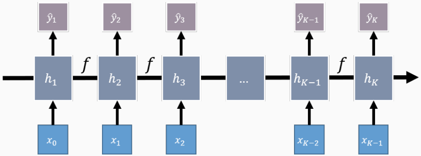
$$h_0 = 0, \qquad \hat{y}_k = c^\top h_k$$



## The Recurrent Neural Network Hypothesis Space

The recurrent neural network (RNN) architecture

$$h_{k+1} = \sigma(W h_k + U x_k), \qquad h_k \in \mathbb{R}^m$$
$$h_0 = 0, \qquad \hat{y}_k = c^\top h_k$$



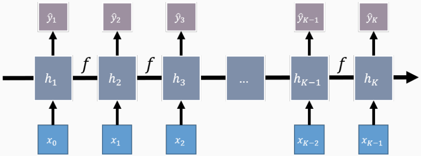

- The RNN parametrizes a sequence of functions $\{\hat{H}_k = \{x_0, \ldots, x_{k-1}\} \mapsto \hat{y}_k\}$.

# The Recurrent Neural Network Hypothesis Space

The recurrent neural network (RNN) architecture

$$h_{k+1} = \sigma(Wh_k + Ux_k), \qquad h_k \in \mathbb{R}^m$$

$$h_0 = 0, \qquad \hat{y}_k = c^\top h_k$$




- The RNN parametrizes a sequence of functions $\{\hat{H}_k = \{x_0, \ldots, x_{k-1}\} \mapsto \hat{y}_k\}$.
- A continuous-time idealization parametrizes functionals $\{\boldsymbol{x} \equiv \{x_t\} \mapsto \hat{y}_t\}$

$$\dot{h}_t = \sigma(Wh_t + Ux_t), \quad h_{-\infty} = 0, \quad \hat{y}_t = c^\top h_t, \quad t \in \mathbb{R}$$

**Empirically, it is found RNN performs poorly when modelling "long-term memory"**

**Can we investigate this phenomena precisely?**

# The Linear RNN Hypothesis Space

We analyze the linear case where $\sigma(h) = h$, we have the dynamics

$$\begin{aligned} \hat{y}_t &= c^\top h_t, \\ \dot{h}_t &= W h_t + U x_t. \end{aligned} \quad \text{where} \quad \begin{array}{ll} h_t \in \mathbb{R}^m & \text{(hidden state)} \\ W \in \mathbb{R}^{m \times m} & \text{(Recurrent Kernel)} \\ U \in \mathbb{R}^{m \times d} & \text{(Input Kernel)} \\ c \in \mathbb{R}^m & \text{(Output layer weights)} \end{array}$$

# The Linear RNN Hypothesis Space

We analyze the linear case where $\sigma(h) = h$, we have the dynamics

$$\begin{aligned} \hat{y}_t &= c^\top h_t, \\ \dot{h}_t &= W h_t + U x_t. \end{aligned} \qquad \text{where} \qquad \begin{aligned} &h_t \in \mathbb{R}^m && \text{(hidden state)} \\ &W \in \mathbb{R}^{m \times m} && \text{(Recurrent Kernel)} \\ &U \in \mathbb{R}^{m \times d} && \text{(Input Kernel)} \\ &c \in \mathbb{R}^m && \text{(Output layer weights)} \end{aligned}$$

This gives rise to the (stable) linear RNN hypothesis space

$$\mathcal{H}_{\mathrm{RNN}} = \cup_{m \geq 1} \underbrace{\left\{ \{ \hat{H}_t(\boldsymbol{x}) = \int_0^\infty c^\top e^{Ws} U x_{t-s} ds \}, W \in \mathcal{W}_m, U \in \mathbb{R}^{m \times d}, c \in \mathbb{R}^m \right\}}_{\mathcal{H}_{\mathrm{RNN}}^{(m)}}$$

$$\mathcal{W}_m = \{ W \in \mathbb{R}^{m \times m} : \text{eigenvalues of } W \text{ have negative real parts (Hurwitz)} \}$$

# Properties of Linear RNN Hypothesis Space

$$\mathcal{H}_{\text{RNN}}^{(m)} = \left\{ \{\hat{H}_t(\boldsymbol{x}) = \int_0^\infty c^\top e^{Ws} U x_{t-s} ds\} : W \in \mathcal{W}_m, U \in \mathbb{R}^{m \times d}, c \in \mathbb{R}^m \right\}$$

**Proposition**

Let $\{\hat{H}_t : t \in \mathbb{R}\}$ be any family of functionals in $\mathcal{H}_{\text{RNN}}$. Then for each $t \in \mathbb{R}$,

- $\hat{H}_t$ is a continuous, linear functional.
- $\hat{H}_t$ is a causal functional.
- $\hat{H}_t$ is a regular functional.
- The family $\{\hat{H}_t : t \in \mathbb{R}\}$ is time-homogeneous.

# Approximation Guarantee (Density)

**Theorem [LHEL, 2021]**

Let $\{H_t^* : t \in \mathbb{R}\}$ be a family of continuous, linear, causal, regular and time-homogeneous functionals on $C_0(\mathbb{R}, \mathbb{R}^d)$. Then, for any $\epsilon > 0$ there exists $\{\hat{H}_t : t \in \mathbb{R}\} \in \mathcal{H}_{\mathrm{RNN}}$ such that

$$\|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \equiv \sup_{t\in\mathbb{R}} \sup_{\|\boldsymbol{x}\|_{\mathcal{C}}\leq 1} |H_t^*(\boldsymbol{x}) - \hat{H}_t(\boldsymbol{x})| \leq \epsilon.$$

Main idea: Prove a general Riesz representation

$$H_t^*(\boldsymbol{x}) = \int_0^\infty \rho(s)^\top x_{t-s} ds \qquad \left[\text{Recall: } \hat{H}_t(\boldsymbol{x}) = \int_0^\infty c^\top e^{Ws} U x_{t-s} ds\right]$$

Then, RNN approximation reduces to the $L^1$ approximation of $\rho(t)$ by $[c^\top e^{Wt} U]^\top$.

Z. Li, J. Han, W. E, and Q. Li, "On the curse of memory in recurrent neural networks: Approximation and optimization analysis," in International Conference on Learning Representations (ICLR), 2021

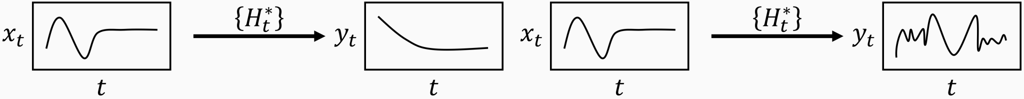
## Smoothness and Memory

Approximation rates depend on appropriate complexity measures

# Smoothness and Memory

Approximation rates depend on appropriate complexity measures

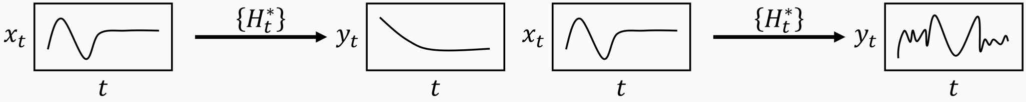
Key concepts: smoothness and memory



# Smoothness and Memory

Approximation rates depend on appropriate complexity measures

Key concepts: smoothness and memory



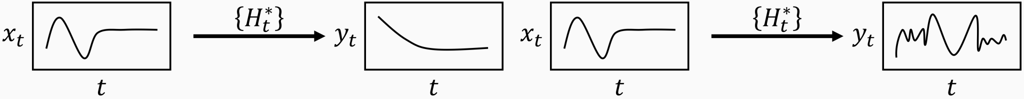

Define

- $e_i$, $i = 1, \ldots, d$ as the standard basis vectors in $\mathbb{R}^d$
- $\boldsymbol{e}_i$ as the constant signal $e_{i,t} = e_i \mathbb{1}_{\{t \geq 0\}}$

## Smoothness and Memory

Approximation rates depend on appropriate complexity measures

Key concepts: smoothness and memory




Define

- $e_i$, $i = 1, \ldots, d$ as the standard basis vectors in $\mathbb{R}^d$
- $\boldsymbol{e}_i$ as the constant signal $e_{i,t} = e_i \mathbb{1}_{\{t \geq 0\}}$

Given a family of functionals $\{H_t^* : t \in \mathbb{R}\}$

- Denote the output of constant signal $y_i(t) := H_t^*(\boldsymbol{e}_i)$
- smoothness is characterized by the smoothness of $t \mapsto y_i(t)$
- memory is characterized by the decay rate of the $t \mapsto y_i^{(k)}(t)$

# Approximation Rate

**Theorem [LHEL, 2021]**

Set $y_i = H_t^*(\boldsymbol{e}_i)$. Suppose there exist constants $\alpha \in \mathbb{Z}^+, \beta, \gamma \in \mathbb{R}^+$ such that for $i = 1, \ldots, d$, $y_i(t) \in C^{(\alpha+1)}(\mathbb{R})$ and for $k = 1, \ldots, \alpha + 1$,

$$e^{\beta t} y_i^{(k)}(t) = o(1) \text{ as } t \to +\infty \qquad \text{and} \qquad \sup_{t \geq 0} \frac{|e^{\beta t} y_i^{(k)}(t)|}{\beta^k} \leq \gamma.$$

Then there exists a universal constant $C(\alpha)$ such that for each $m \geq 1$,

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}_{\mathrm{RNN}}^{(m)}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^\alpha}.$$

Z. Li, J. Han, W. E, and Q. Li, "On the curse of memory in recurrent neural networks: Approximation and optimization analysis," in International Conference on Learning Representations (ICLR), 2021

## Curse of Memory in Approximation

Rate estimate

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}^{(m)}_{\mathrm{RNN}}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^{\alpha}}.$$

# Curse of Memory in Approximation

Rate estimate

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}_{\mathrm{RNN}}^{(m)}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^{\alpha}}.$$

Observations

- The smoothness dependence ($\alpha$) is familiar

# Curse of Memory in Approximation

Rate estimate

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}^{(m)}_{\text{RNN}}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^\alpha}.$$

Observations

- The smoothness dependence ($\alpha$) is familiar
- The memory dependence ($\beta$) is new: we need

$$y_i(t) \equiv H^*_t(\boldsymbol{e}_i) \sim e^{-\beta t}, \qquad \beta > 0$$

# Curse of Memory in Approximation

Rate estimate

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}_{\mathrm{RNN}}^{(m)}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^{\alpha}}.$$

Observations

- The smoothness dependence ($\alpha$) is familiar
- The memory dependence ($\beta$) is new: we need

$$y_i(t) \equiv H_t^*(\boldsymbol{e}_i) \sim e^{-\beta t}, \qquad \beta > 0$$

- There is no curse of dimensionality due to linearity

# Curse of Memory in Approximation

Rate estimate

$$\inf_{\hat{\boldsymbol{H}} \in \mathcal{H}^{(m)}_{\text{RNN}}} \|\boldsymbol{H}^* - \hat{\boldsymbol{H}}\| \leq \frac{C(\alpha)\gamma d}{\beta m^\alpha}.$$

Observations

- The smoothness dependence ($\alpha$) is familiar
- The memory dependence ($\beta$) is new: we need

$$y_i(t) \equiv H^*_t(\boldsymbol{e}_i) \sim e^{-\beta t}, \qquad \beta > 0$$

- There is no curse of dimensionality due to linearity
- However, hidden in these results is a curse of memory:

> If $H^*_t(\boldsymbol{e}_i) \sim t^{-\omega}$, then to get error $\epsilon$, need $m \sim \mathcal{O}\left(\omega \varepsilon^{-\frac{1}{\omega}}\right)$

# Insights on the (Linear) RNN Hypothesis Space

**Insight:** Efficient approximation if $\boldsymbol{H}^*$ is smooth and has exponential decaying memory

Futhermore

- The "only if" part is also true [LHEL, 2022]

$$\text{efficient approximation} \quad \Longrightarrow \quad \text{exponential decaying memory}$$

- A related curse of memory holds for optimizing RNNs [LHEL, 2021; 2022]
- Nonlinear recurrent activation does not alleviate this [WLL, 2022]

---


Z. Li, J. Han, W. E, and Q. Li, "On the curse of memory in recurrent neural networks: Approximation and optimization analysis," in International Conference on Learning Representations (ICLR), 2021

Z. Li, J. Han, W. E, and Q. Li, "Approximation and Optimization Theory for Linear Continuous-Time Recurrent Neural Networks," Journal of Machine Learning Research, vol. 23, no. 42, 2022

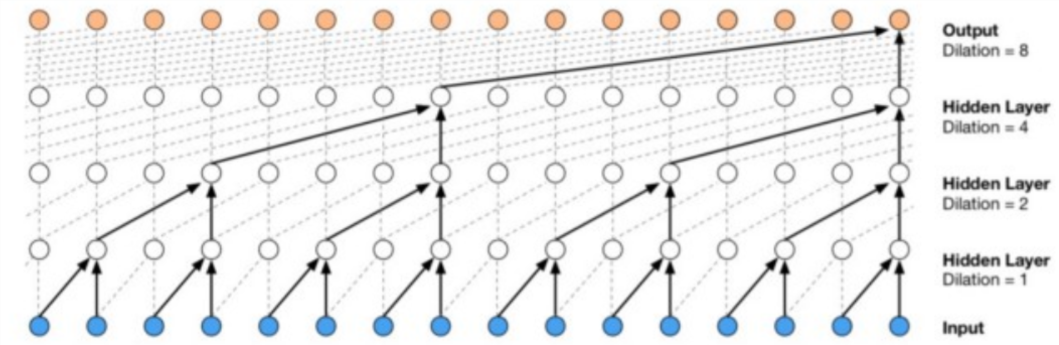
S. Wang, Z. Li, and Q. Li, "The effects of nonlinearity on approximation capacity of recurrent neural networks," Submitted, 2022

# Extension to Other Architectures

# Convolutional Architectures

A popular alternative to recurrent architectures is convolutional based architectures for sequence modelling

**Example: WaveNet**

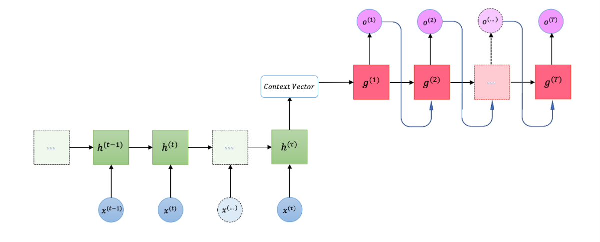



A. v. d. Oord, S. Dieleman, H. Zen, et al., "WaveNet: A Generative Model for Raw Audio," arXiv:1609.03499 [cs], 2016, arXiv: 1609.03499. (visited on 02/09/2022)

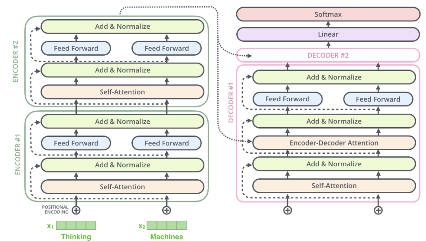
# Encoder-Decoder Architectures

Yet another alternative are encoder-decoder class of architectures (e.g. RNN encoder-decoder, transformer)

Recurrent Encoder-Decoder




Transformer




How are they different, when to use which?

A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017

# Extending the RNN Analysis

These architectures can be analyzed in the same setting of functional approximation.

Key insights:

- They can all achieve density in appropriate functional spaces
- Efficient approximation depends on different notions of complexity
  - RNN: Exponential memory decay
  - CNN: Sparse dependence on inputs (low tensorization rank)
  - Recurrent Encoder-Decoder: Dependence on global features of the input (low rank under temporal products)

Need **structural compatibility** between the model and the target

H. Jiang, Z. Li, and Q. Li, "Approximation theory of convolutional architectures for time series modelling," International Conferences on Machine Learning (ICML), 2021

Z. Li, H. Jiang, and Q. Li, "On the approximation properties of recurrent encoder-decoder architectures," in International Conference on Learning Representations (ICLR), 2022

# References i


1. Q. Li, T. Lin, and Z. Shen, "Deep learning via dynamical systems: An approximation perspective," Journal of the European Mathematical Society, 2022
2. Q. Li, T. Lin, and Z. Shen, "Deep Neural Network Approximation of Invariant Functions through Dynamical Systems," arXiv, no. arXiv:2208.08707, 2022. arXiv: 2208.08707
3. Z. Li, J. Han, W. E, and Q. Li, "On the curse of memory in recurrent neural networks: Approximation and optimization analysis," in International Conference on Learning Representations (ICLR), 2021
4. H. Jiang, Z. Li, and Q. Li, "Approximation theory of convolutional architectures for time series modelling," International Conferences on Machine Learning (ICML), 2021
5. Z. Li, J. Han, W. E, and Q. Li, "Approximation and Optimization Theory for Linear Continuous-Time Recurrent Neural Networks," Journal of Machine Learning Research, vol. 23, no. 42, 2022
6. Z. Li, H. Jiang, and Q. Li, "On the approximation properties of recurrent encoder-decoder architectures," in International Conference on Learning Representations (ICLR), 2022

# References ii


7. 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
8. 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
9. 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 (ICML), vol. 80, 2018
10. Z. Chen, Q. Li, and Z. Zhang, "Towards robust neural networks via close-loop control," in International Conference on Learning Representations (ICLR), 2021