Dynamic View of Deep Learning

Shihua Zhang

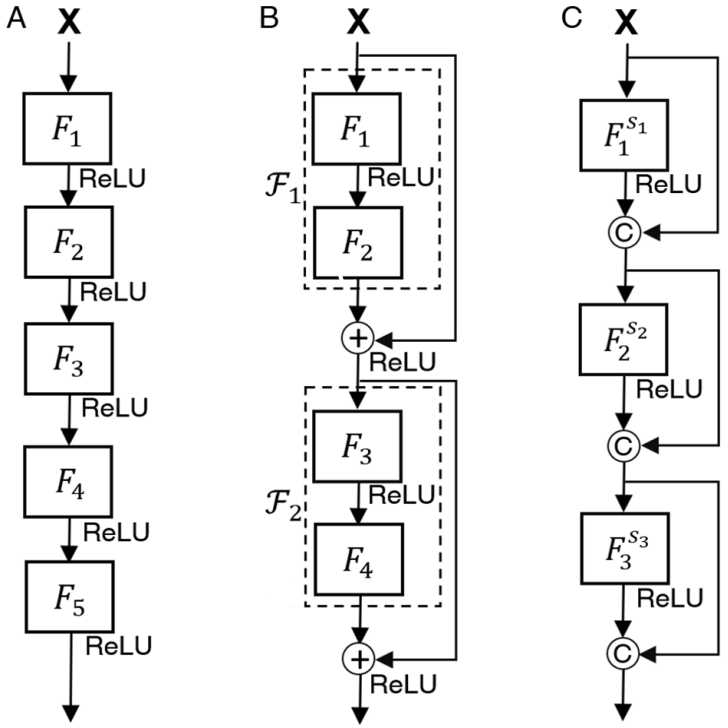
Academy of Mathematics and Systems Science, CAS

Dec 15, 2021

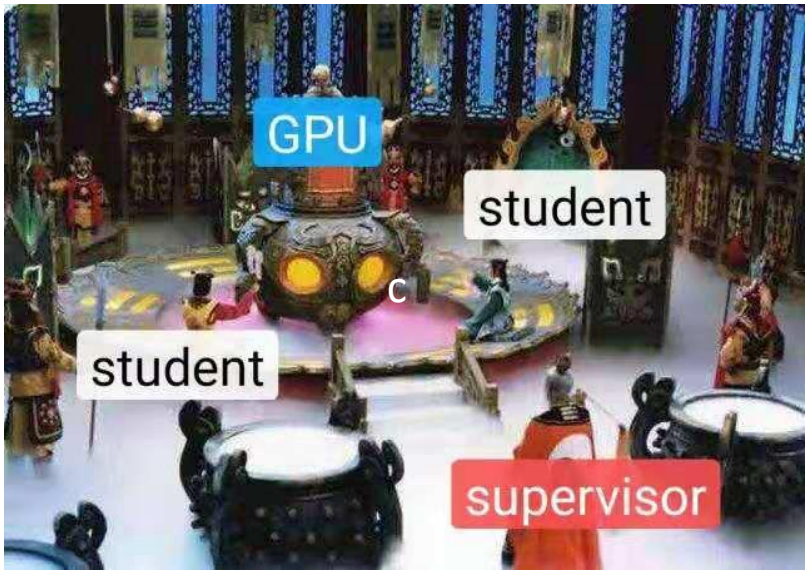
Deep Learning Achieves Great Progresses

Deep Learning is Everywhere!

- Convolutional Neural Network (CNN)
- Residual Neural Network (ResNet)
- Dense Neural Network (DenseNet)



Understanding of its fundamental properties is lacking

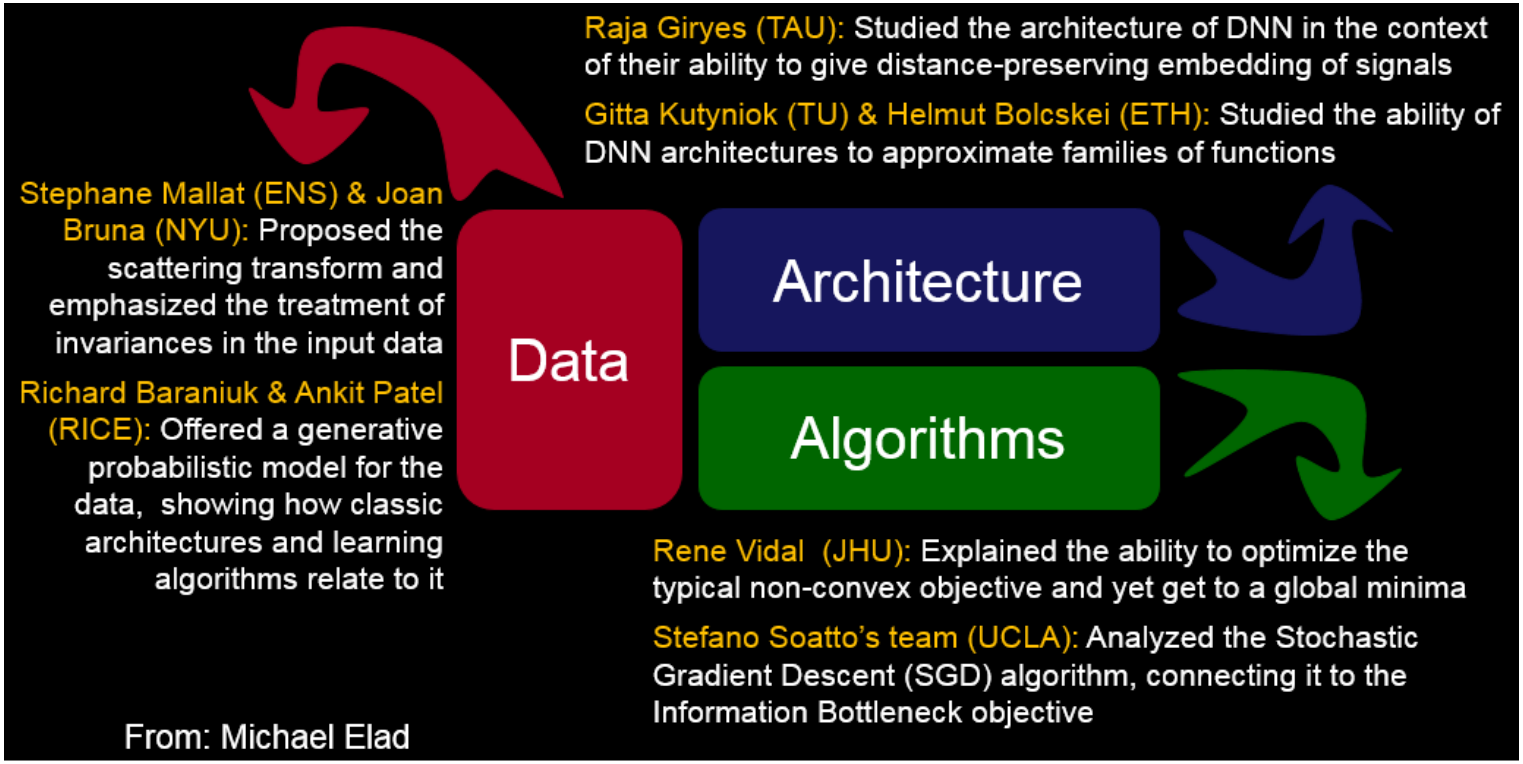


Ali Rahimi: Machine learning has become alchemy (NIPS 2017)

Mathematical Understanding of Deep Learning

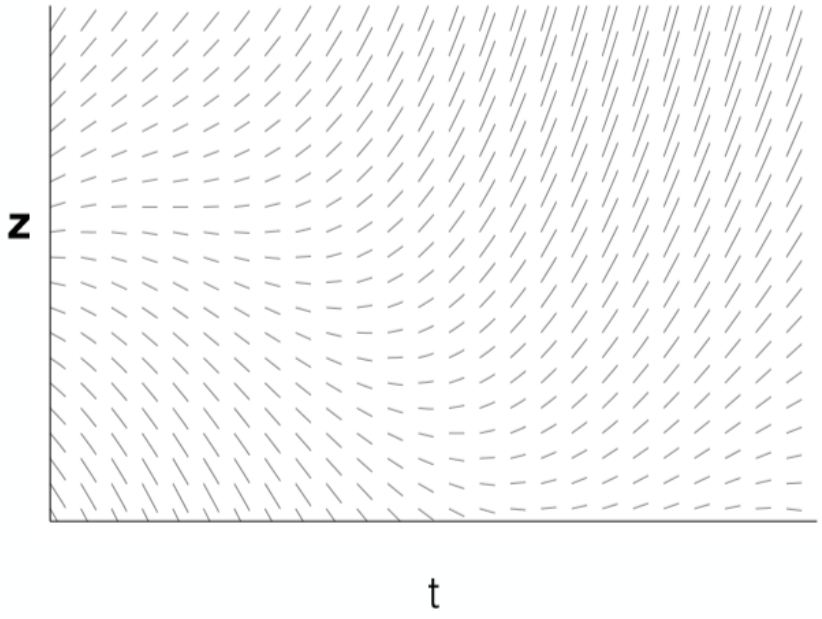
- map the limitations of existing deep learning
- bring the next rounds of ideas
- turn it into a solid scientific discipline

What Kinds of Theory?



Part I: Neural ordinary differential equations

ODE Solvers



- Vector-valued $\mathbf{z}$ changes in time
- Time-derivative: $\frac{d\mathbf{z}}{dt} = f(\mathbf{z}(t), t)$
- Initial-value problem: given $\mathbf{z}(t_0)$, find:

$$\mathbf{z}(t_1) = \mathbf{z}(t_0) + \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \theta) dt$$

- Euler approximates with small steps:

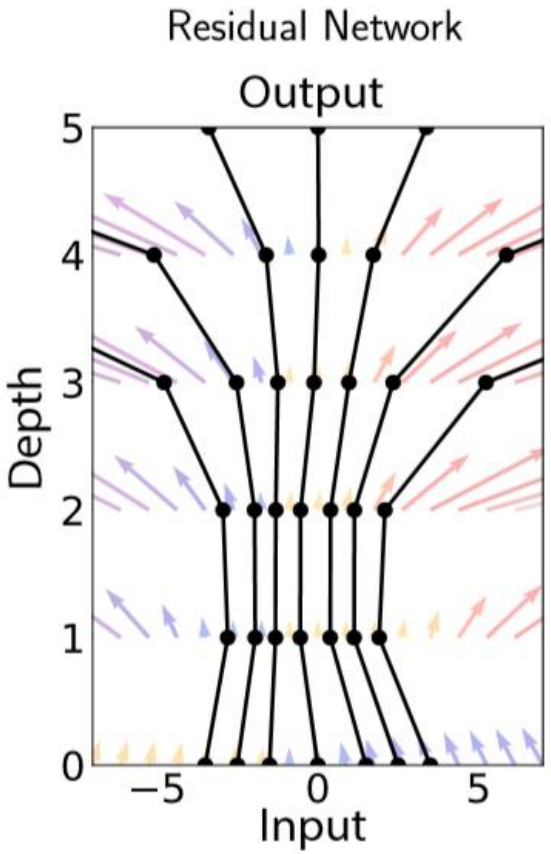
$$\mathbf{z}(t + h) = \mathbf{z}(t) + hf(\mathbf{z}, t)$$

ResNets as Euler Integrators

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

ResNet

```
def resnet(x, θ):
  h1 = x + NeuralNet(x, θ[0])
  h2 = h1 + NeuralNet(h1, θ[1])
  h3 = h2 + NeuralNet(h2, θ[2])
  h4 = h3 + NeuralNet(h3, θ[3])
  return h4
```



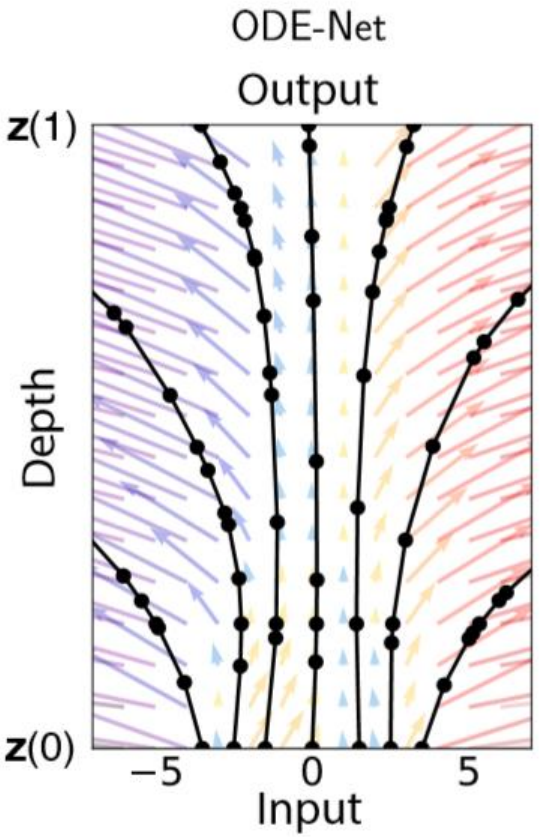
Consider Infinite-Depth Neural Networks

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

ODE-Net

```
def f(z, t, θ):
  return NeuralNet([z, t], θ)

def ODEnet(x, θ):
  return ODESolve(f, x, 0, 1, θ)
```



Chen et al. Neural ordinary differential equations. arXiv preprint arXiv:1806.07366, 2018.

How to Train an ODE Net?

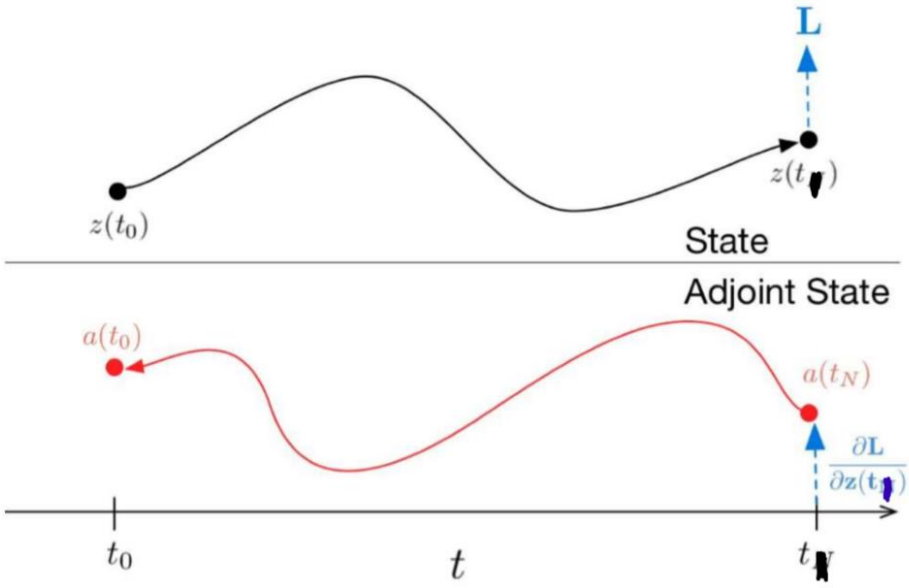
- Consider optimizing a scalar-valued loss function $L(\cdot)$, whose input is the result of an ODE solver:

$$L(\mathbf{z}(t_1)) = L\left(\mathbf{z}(t_0) + \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \theta)dt\right) = L\left(\text{ODESolve}(\mathbf{z}(t_0), f, t_0, t_1, \theta)\right)$$

$$\frac{\partial L}{\partial \theta} = ?$$

Continuous-time Backpropagation

$$L(\mathbf{z}(t_1)) = L\left(\mathbf{z}(t_0) + \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \theta)dt\right) = L\left(\text{ODESolve}(\mathbf{z}(t_0), f, t_0, t_1, \theta)\right)$$



Define adjoint state:

$a(t) = -\partial L / \partial \mathbf{z}(t)$

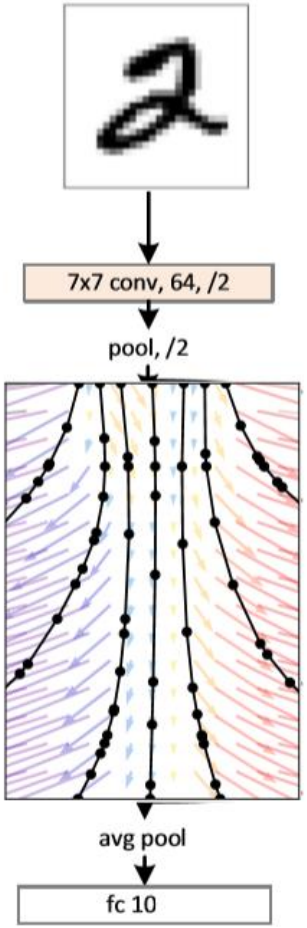
Adjoint state dynamics:

$\frac{a(t)}{dt} = -a(t)\frac{\partial f(\mathbf{z}(t), t, \theta)}{\partial \mathbf{z}}$

Solve ODE backwards in time:

$\frac{dL}{d\theta} = \int_{t_0}^{t_1} a(t)^T \frac{\partial f(\mathbf{z}(t), t, \theta)}{\partial \theta} dt$

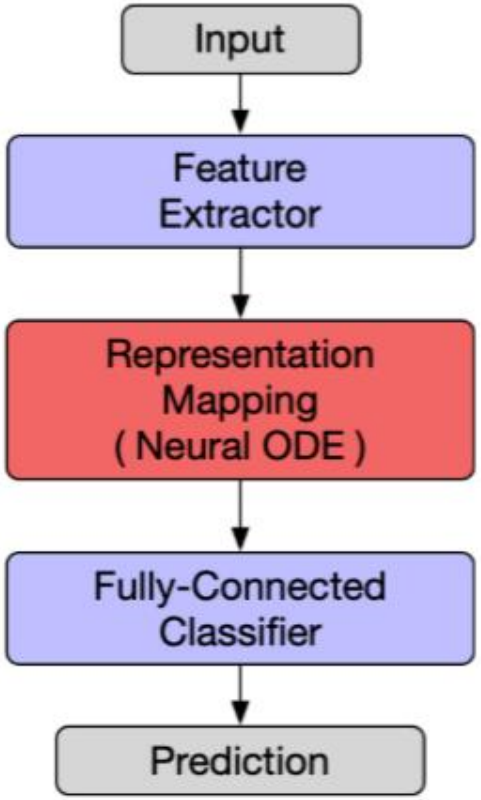
ODE Nets for Supervised Learning



	Test Error	# Params
1-Layer MLP	1.60%	0.24 M
ResNet	0.41%	0.60 M
ODE-Net	0.42%	0.22 M

- Same performance with fewer parameters
- 2-4x the depth of ResNet architectures

The Robustness of ODE-Nets



- Studying the robustness: Neural ODE vs ResNet
- Made sure that the number of parameters of an ODENet is close to that of its counterpart CNN model
- Same training strategy (learning rate, epochs, …)

Yan et al. On robustness of neural ordinary differential equations. arXiv:1910.05513, 2019.

Robustness of ODE-Nets Trained Only on Non-Perturbed Images

	Gaussian noise			Adversarial attack		
MNIST	$\sigma = 50$	$\sigma = 75$	$\sigma = 100$	FGSM-0.15	FGSM-0.3	FGSM-0.5
CNN	98.1±0.7	85.8±4.3	56.4±5.6	63.4±2.3	24.0±8.9	8.3±3.2
ODENet	**98.7**±0.6	**90.6**±5.4	**73.2**±8.6	**83.5**±0.9	**42.1**±2.4	**14.3**±2.1
SVHN	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	FGSM-3/255	FGSM-5/255	FGSM-8/255
CNN	90.0±1.2	76.3±2.7	60.9±3.9	29.2±2.9	13.7±1.9	5.4±1.5
ODENet	**95.7**±0.7	**88.1**±1.5	**78.2**±2.1	**58.2**±2.3	**43.0**±1.3	**30.9**±1.4
ImgNet10	$\sigma = 10$	$\sigma = 15$	$\sigma = 25$	FGSM-5/255	FGSM-8/255	FGSM-16/255
CNN	80.1±1.8	63.3±2.0	40.8±2.7	28.5±0.5	18.1±0.7	9.4±1.2
ODENet	**81.9**±2.0	**67.5**±2.0	**48.7**±2.6	**36.2**±1.0	**27.2**±1.1	**14.4**±1.7

ODE-Nets outperform CNN trained on non-perturbed images!

Robustness of ODE-Nets Trained on Original Images Together with Gaussian Perturbations

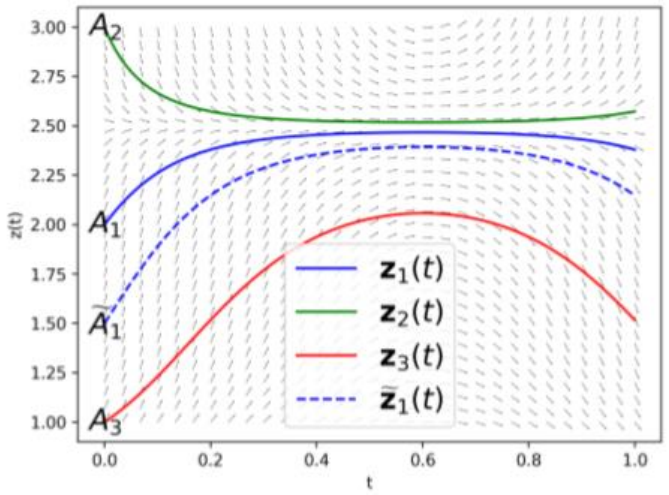
	Gaussian noise	Adversarial attack			
MNIST	$\sigma = 100$	FGSM-0.3	FGSM-0.5	PGD-0.2	PGD-0.3
CNN	98.7±0.1	54.2±1.1	15.8±1.3	32.9±3.7	0.0±0.0
ODENet	**99.4**±0.1	**71.5**±1.1	**19.9**±1.2	**64.7**±1.8	**13.0**±0.2
SVHN	$\sigma = 35$	FGSM-5/255	FGSM-8/255	PGD-3/255	PGD-5/255
CNN	90.6±0.2	25.3±0.6	12.3±0.7	32.4±0.4	14.0±0.5
ODENet	**95.1**±0.1	**49.4**±1.0	**34.7**±0.5	**50.9**±1.3	**27.2**±1.4
ImgNet10	$\sigma = 25$	FGSM-5/255	FGSM-8/255	PGD-3/255	PGD-5/255
CNN	92.6±0.6	40.9±1.8	26.7±1.7	28.6±1.5	11.2±1.2
ODENet	92.6±0.5	**42.0**±0.4	**29.0**±1.0	**29.8**±0.4	**12.3**±0.6

ODE-Nets outperform CNN trained on original images together with gaussian perturbations

Insights on the Robustness of ODE-Nets

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = f_\theta(\mathbf{z}(t), t), \quad \mathbf{z}(0) = \mathbf{z}_{\text{in}}, \quad \mathbf{z}_{\text{out}} = \mathbf{z}(T),$$

Theorem 1 (ODE integral curves do not intersect (Coddington & Levinson, 1955; Younes, 2010; Dupont et al., 2019)). *Let $\mathbf{z}_1(t)$ and $\mathbf{z}_2(t)$ be two solutions of the ODE in (1) with different initial conditions, i.e. $\mathbf{z}_1(0) \neq \mathbf{z}_2(0)$. In (1), f_θ is continuous in t and globally Lipschitz continuous in $\mathbf{z}$. Then, it holds that $\mathbf{z}_1(t) \neq \mathbf{z}_2(t)$ for all $t \in [0, \infty)$.*



Note: a small change on the feature map will not lead to a large deviation from the original output associated with the feature map

Insights on the Robustness of ODE-Nets

- A stronger theorem based on the assumption of f

Theorem 2. *Let $U \subset \mathbb{R}^d$ be an open set. Let $f : U \times [0, T] \to \mathbb{R}^d$ be a continuous function and let $\mathbf{z}_1$, $\mathbf{z}_2$: $[0, T] \to U$ satisfy the initial value problems:*

$$\frac{\mathrm{d}\mathbf{z}_1(t)}{\mathrm{d}t} = f(\mathbf{z}_1(t), t), \quad \mathbf{z}_1(0) = \mathbf{x}_1$$
$$\frac{\mathrm{d}\mathbf{z}_2(t)}{\mathrm{d}t} = f(\mathbf{z}_2(t), t), \quad \mathbf{z}_2(0) = \mathbf{x}_2$$

Assume there is a constant $C \geq 0$ such that, for all $t \in [0, T]$,

$$\|f(\mathbf{z}_2(t), t) - f(\mathbf{z}_1(t), t))\| \leq C\|\mathbf{z}_2(t) - \mathbf{z}_1(t)\|$$

Then, for any $t \in [0, T]$,

$$\|\mathbf{z}_1(t) - \mathbf{z}_2(t)\| \leq \|\mathbf{x}_2 - \mathbf{x}_1\| \cdot e^{Ct}.$$

Part II: Deep Learning (ResNet) Learns the Geodesic Curve of Wasserstein Space

Gai and Zhang, Preprint arXiv:2102.09235, 2021

Continuous View of Deep Learning

- The world is continuous
  - In practice, continuous → discrete
  - In theory, discrete → continuous

- This same logic applies to for deep learning

data points → continuous distribution
layer transformation → continuous transformation

Neural networks learn a continuous transformation from the data distribution μ_0 to the label distribution μ_1

Dynamic View of Deep Learning

Let $x^{(k)}$ denote the representation in the k-th layer. The forward propagation of ResNet is

$$x^{(k+1)} = x^{(k)} + v(x^{(k)}), k \in \{0, 1, \cdots, n-1\}$$

which can be viewed as a discretization of the dynamic system

$$\frac{dx}{dt} = v(x, t), t \in [0, 1]$$

assume $x^{(k)}$ is sampled from a continuous distribution μ_k, then the corresponding dynamic system is

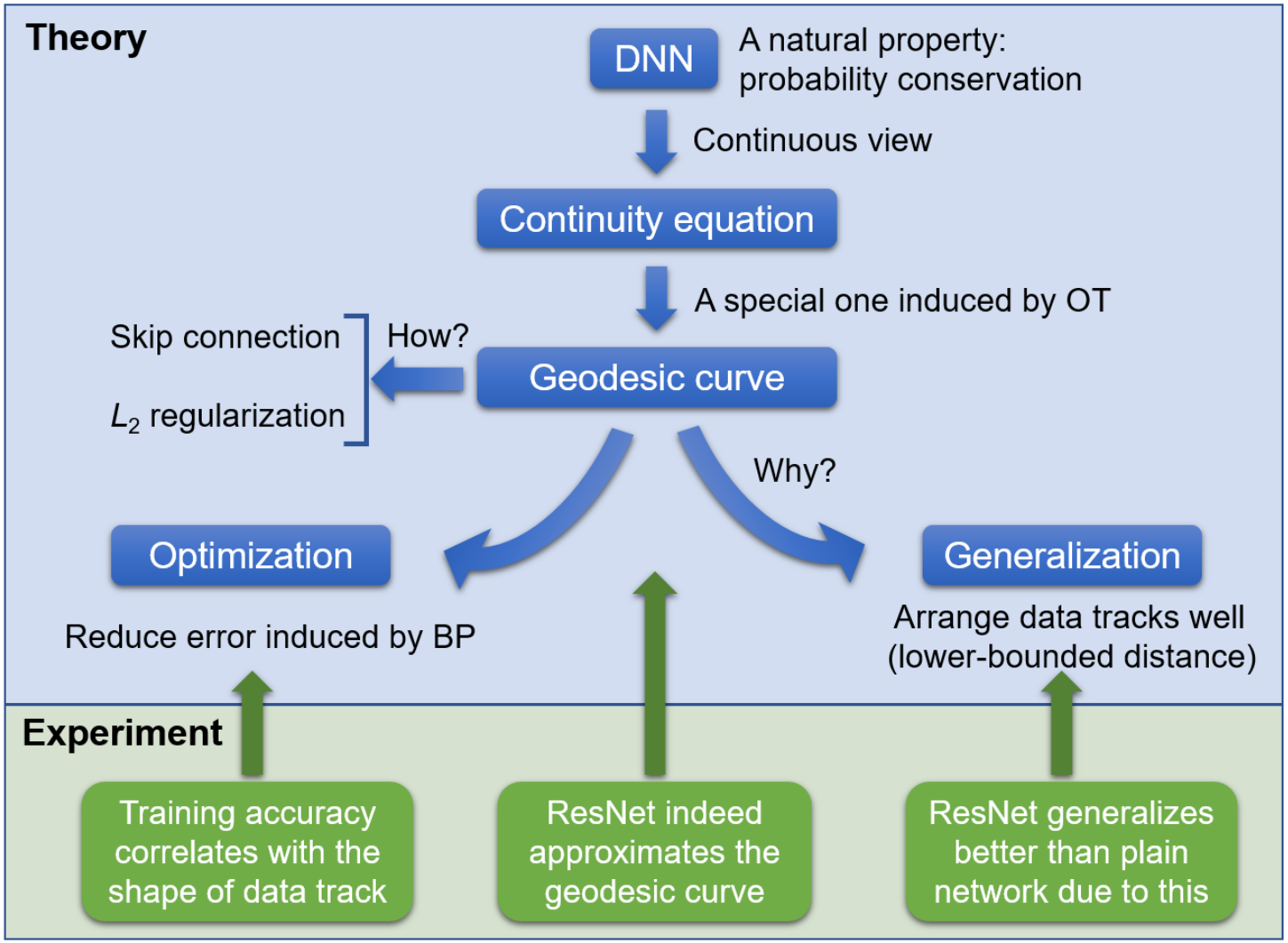
$$\frac{d\mu_t}{dt} = \tilde{v}(\mu_t, t), t \in [0, 1]$$

Dynamic View of Deep Learning

- Interpret neural networks as dynamical system [E, 2017; Haber and Ruthotto, 2017]
- Interpret ResNet as ODE [Chen et al., 2018]
- Build the connection of ResNet and transport equation [Li and Shi, 2017]
- Show the convergence of deep ResNet to ODE in a variational sense [Thorpe and van Gennip, 2018]
- Each local optimal is global for ResNet with infinite width and depth [Lu et al., 2020]

Pro : Build the connection and (partially) explains how it works.
Con: Fail to explain why deep learning works.

Roadmap


Theory
DNN
A natural property:
probability conservation
Continuous view
Continuity equation
A special one induced by OT
Skip connection
L_2 regularization
How?
Geodesic curve
Why?
Optimization
Generalization
Reduce error induced by BP
Arrange data tracks well
(lower-bounded distance)
Experiment
Training accuracy correlates with the shape of data track
ResNet indeed approximates the geodesic curve
ResNet generalizes better than plain network due to this

Probability Conserving Property

For the *k*-th layer f_k, let

$$\mathcal{D}^{(k)} = \left\{x_1^{(k)}, \cdots, x_m^{(k)}\right\} \ (x_i^{(k)} \neq x_j^{(k)}),$$

$$\mathcal{D}^{(k+1)} = \left\{f(x_1^{(k)}), \cdots, f(x_m^{(k)})\right\} \ (f(x_i^{(k)}) \neq f(x_j^{(k)})),$$

let μ_k and μ_{k+1} be the empirical measure of $\mathcal{D}^{(k)}$ and $\mathcal{D}^{(k+1)}$, i.e.,

$$\mu_k = \frac{1}{m}\sum_{i=1}^{m} \delta_{x_i^{(k)}}, \mu_{k+1} = \frac{1}{m}\sum_{i=1}^{m} \delta_{f(x_i^{(k)})}$$

then $\forall S \subset \mathcal{D}^{(k+1)}, \mu_{k+1}(S) = \mu_k\left(f_k^{-1}(S)\right)$, which means f_k is a probability conserving map from μ_k to μ_{k+1}.

- Probability conserving is a intrinsic property of DNN and should be preserved in the continuous view (if DNN generalizes well)

For the dynamic system $\frac{d\mu_t}{dt} = \tilde{v}(\mu_t, t), t \in [0,1]$, $\forall S_0 \subset supp(\mu_0)$, let $S_t \subset supp(\mu_t)$ be the set corresponding to S_0, then

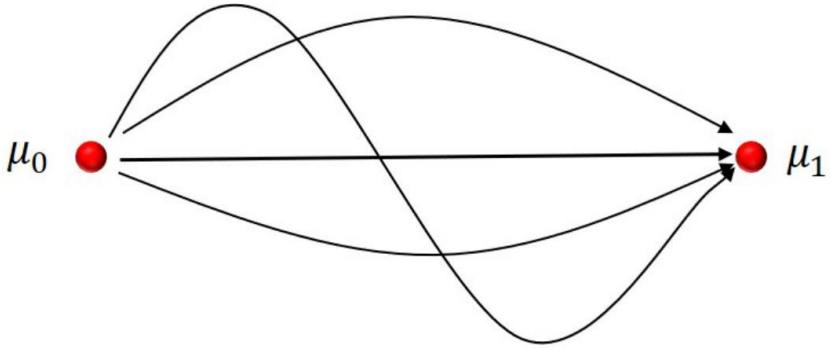
$$\mu_t(S_t) = \mu_0(S_0)$$

Continuity Equation

- Dynamics which conserve mass are termed as continuity equation

$$\frac{d}{dt}\mu_t + \nabla \cdot (v_t \mu_t) = 0,$$

- There are infinite curves connecting two distributions in $\mathcal{P}(R^d)$



- For each absolutely continuous curve in R^d, there exist vector field v_t satisfying continuity equation

- Among all the curves, which one is desired? Geodesic!

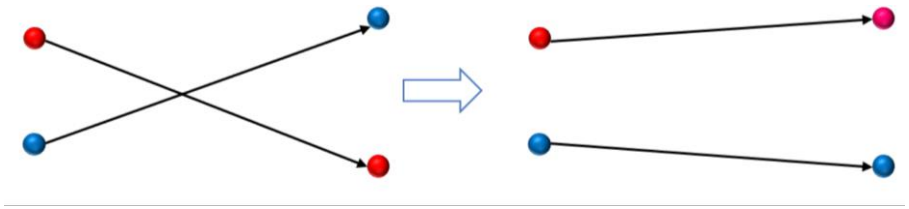
Optimal Transport (OT)

OT: transform a distribution to another with minimum cost

- Monge formulation:

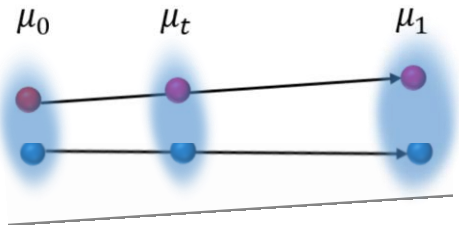
$$W_c(\mu_0, \mu_1) = \inf_{T_{\#}\mu_0 = \mu_1} \int c(x, T(x))\, d\mu_0(x),$$

where T is OT map and conserve probability measure, $c(x, T(x))$ is the cost function (e.g. $c(x, T(x)) = |x - T(x)|_2^2$)



- For arbitrary distributions μ_0 and μ_1, the geodesic curve connecting μ_0 and μ_1 is induced by the OT map

$$\mu_t = ((1-t)\, I_d + tT)_{\#}\, \mu_0.$$



How to Learn the Geodesic Curve?

- Benamou-Brenier formula [Benamou and Brenier, 1999]. Let μ_0 and $\mu_1 \in \mathcal{P}(R^d)$. Then it holds

$$W_2(\mu_0, \mu_1) = \inf \left\{ \int_0^1 \|v_t\|_{L^2(\mu_t)} \, dt \right\}$$

- Under moderate condition, we have

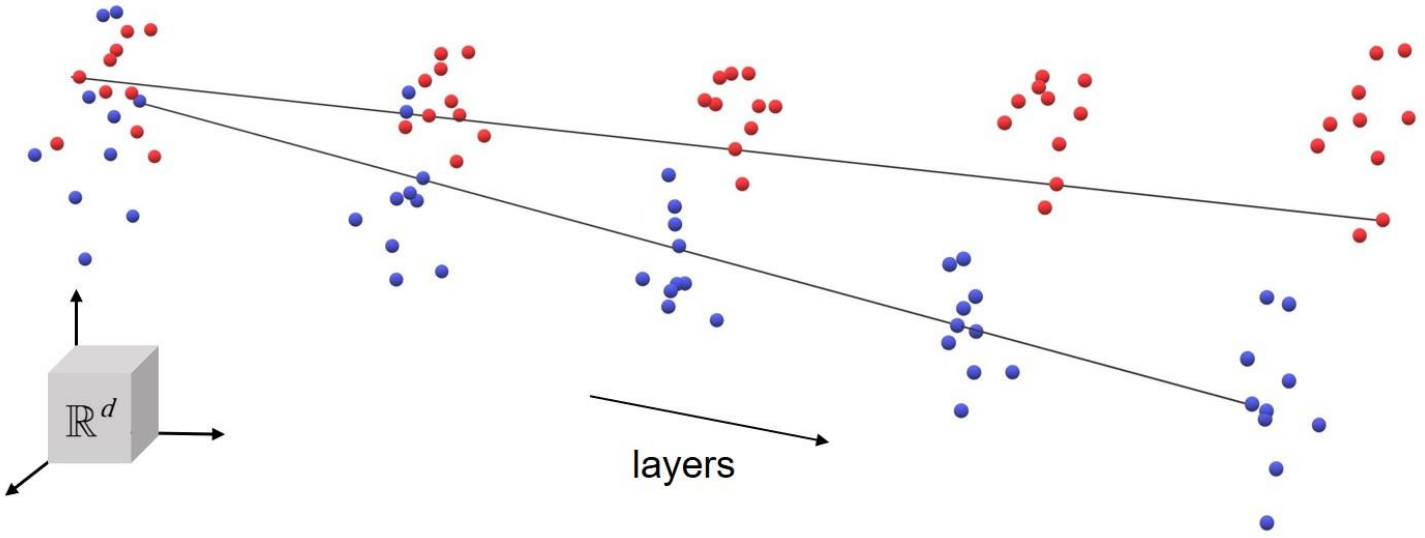
$$\sum_{i=0}^{n-1} \|\tilde{v}_i\|_{L^2(\mu_{ih})} \leqslant C \sum_i \|W^{(i)}\|_2^2$$

where the right side is the L_2 regularizers

A Mathematical Principle of ResNet

Theory

ResNet (with L_2 regularization) is a discrete approximation to the geodesic curve and shares the good properties of optimal transport map.

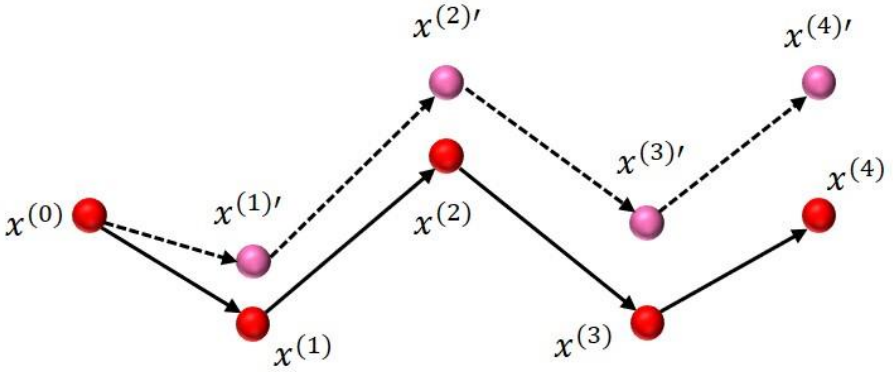


- In this view, we can interpret the superiority of ResNets over plain networks on optimization and generalization.

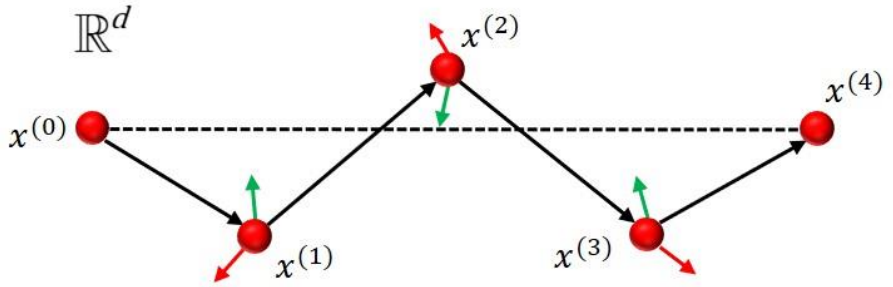
Gai and Zhang, Preprint arXiv:2102.09235, 2021

Optimization

Degradation problem



- If DNN approximates the geodesic curve, the data track $x^{(0)} \rightarrow x^{(1)} \cdots \rightarrow x^{(n)}$ is restricted close to a straight line
- The constraints eliminate the error induced by BP, which solve the degradation problem



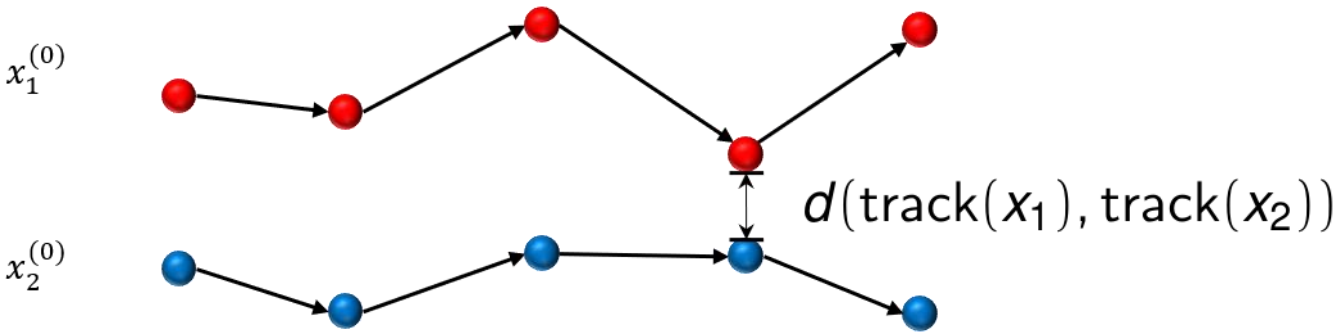
Generalization

- The generalization of a model is correlated to its robustness
- For multilayer case, the distance of tracks is of importance

Distance of two tracks

The distance of two tracks $x_1^{(0)} \to x_1^{(1)} \cdots \to x_1^{(n)}$ and $x_2^{(0)} \to x_2^{(1)} \cdots \to x_2^{(n)}$ is defined as

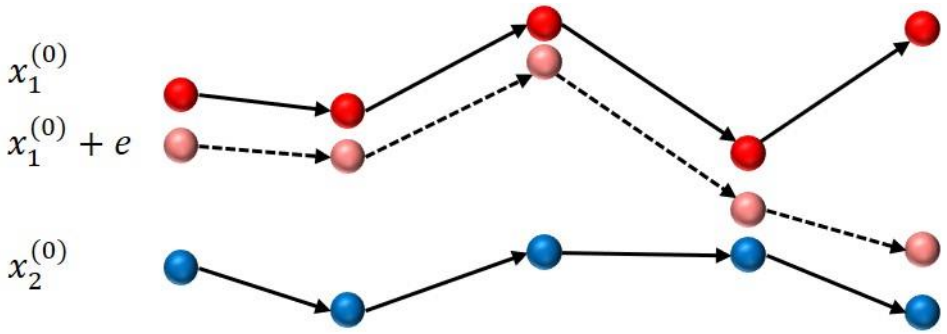
$$d(\text{track}(x_1), \text{track}(x_2)) = \min_{i=0}^{n} \|x_1^{(i)} - x_2^{(i)}\|_2$$



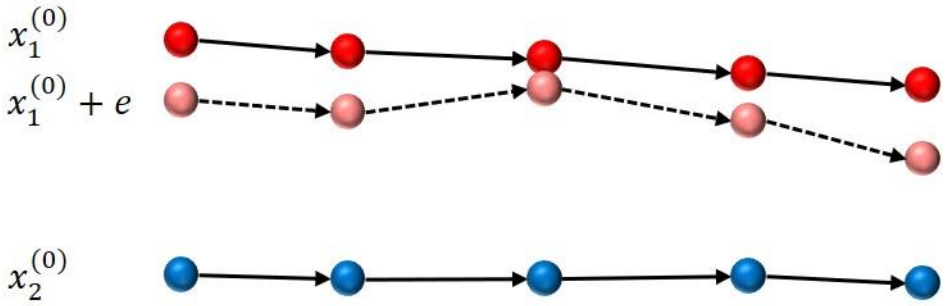
Generalization

The distance of tracks is of importance

- $d\big(\mathrm{track}(x_1), \mathrm{track}(x_2)\big)$ small ⇒ tracks can be easily mixed up.



- $d\big(\mathrm{track}(x_1), \mathrm{track}(x_2)\big)$ large ⇒ numerically robustness.



Generalization

- If DNN approximates geodesic curve exactly, the distance of two arbitrary tracks is lower bounded

Theorem

If a DNN f approximates a geodesic curve, then $\forall\, x_1, x_2 \in D$,

$$d\big(\mathrm{track}(x_1), \mathrm{track}(x_2)\big) \geqslant \frac{\|x_1 - x_2\|_2 \|f(x_1) - f(x_2)\|_2}{\sqrt{\|x_1 - x_2\|_2^2 + \|f(x_1) - f(x_2)\|_2^2}}.$$

Closeness of DNN and Geodesic Curve

A DNN f approximates the geodesic curve:

- f is OT map
- track of f is line-shape

Line-Shape Score (LSS)

$$\tilde{x}^{(0)} = x^{(0)},\ \tilde{x}^{(l)} = \tilde{x}^{(l-1)} + \frac{x^{(l)} - x^{(l-1)}}{\|x^{(l)} - x^{(l-1)}\|_2}$$

$$\mathrm{LSS} = \frac{n}{\|\tilde{x}^{(n)} - \tilde{x}^{(0)}\|_2}$$

Optimal Transport Score (OTS)

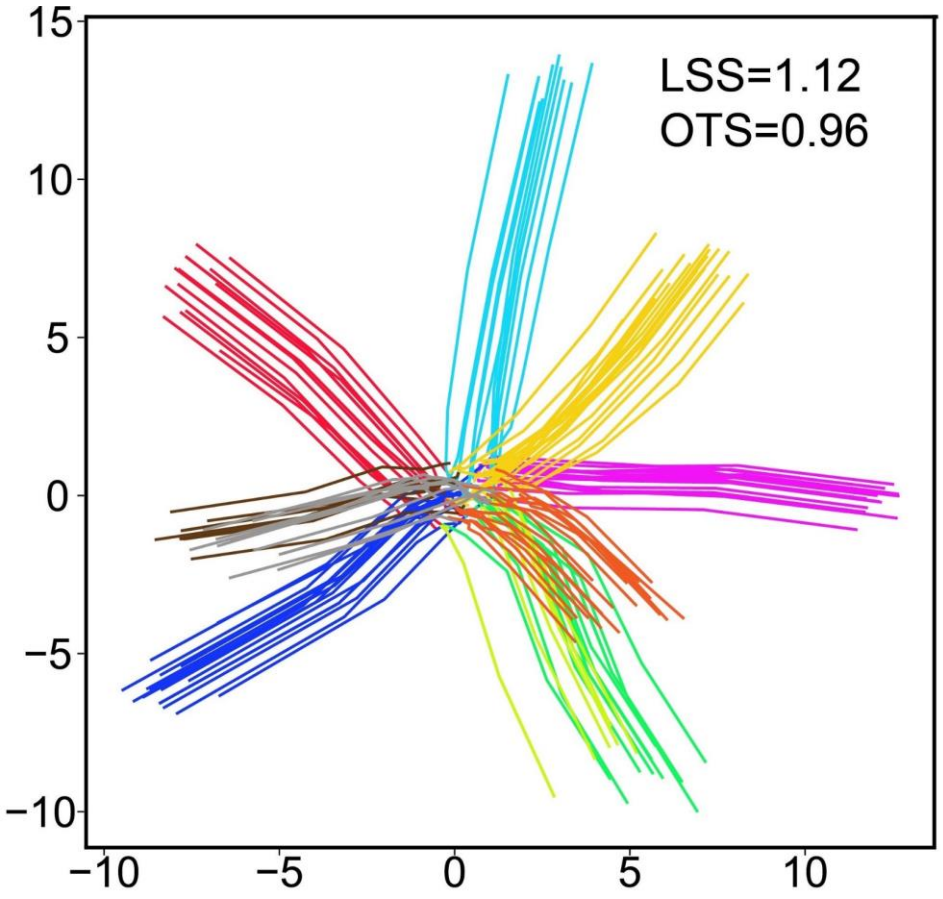
The real discrete OT map T can be computed by solving:

$$\min_{c_{ij}} \sum_{i,j} c_{ij} \|x_i - \tilde{y}_j\|_2^2$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{m} c_{ij} = 1,\ j = 1, 2, \cdots, m, \\ \sum_{j=1}^{m} c_{ij} = 1,\ i = 1, 2, \cdots, m, \\ c_{ij} \in \{0, 1\}, \end{cases}$$

$$\mathrm{OTS} = \frac{\#\{i \in \{1, 2, \cdots, m\} | T(x_i) = f(x_i)\}}{m}$$

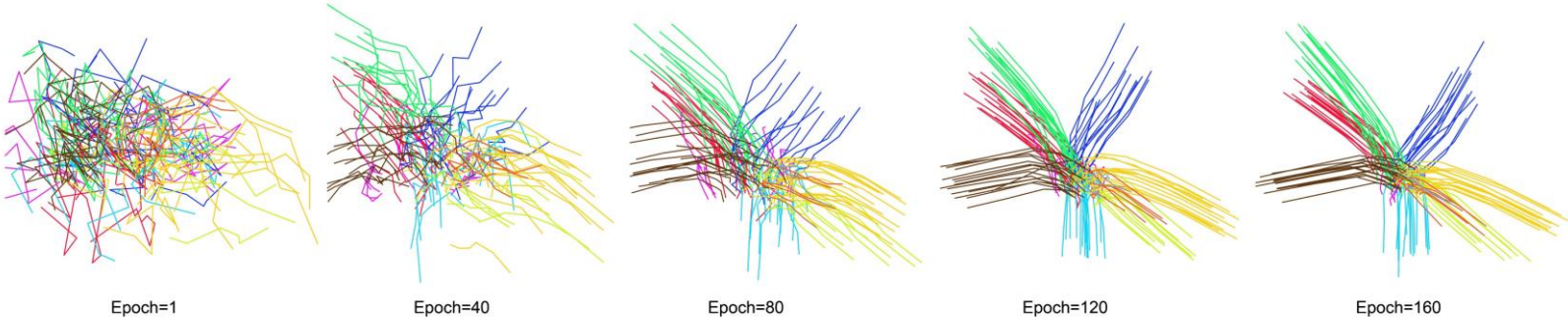
Visualization of Data Tracks on MNIST



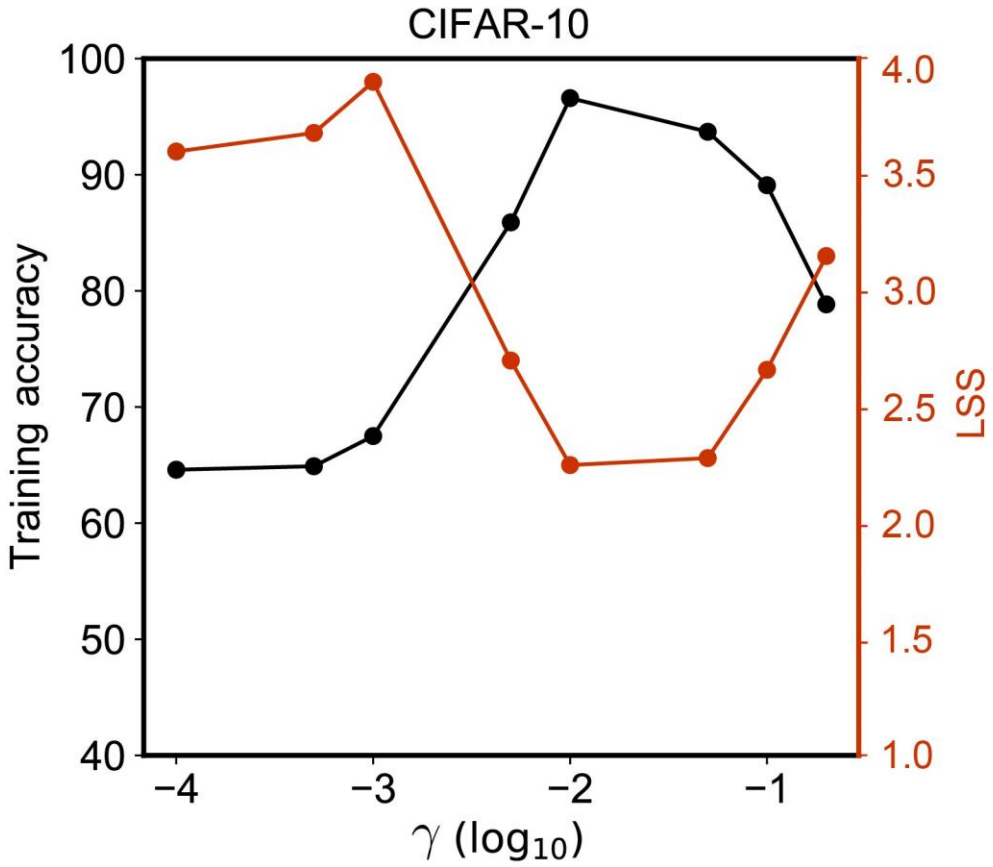
ResNet with L_2 regularizer indeed approximates the geodesic curve induced by OT!

Visualization of Data Tracks on MNIST

Training Dynamics

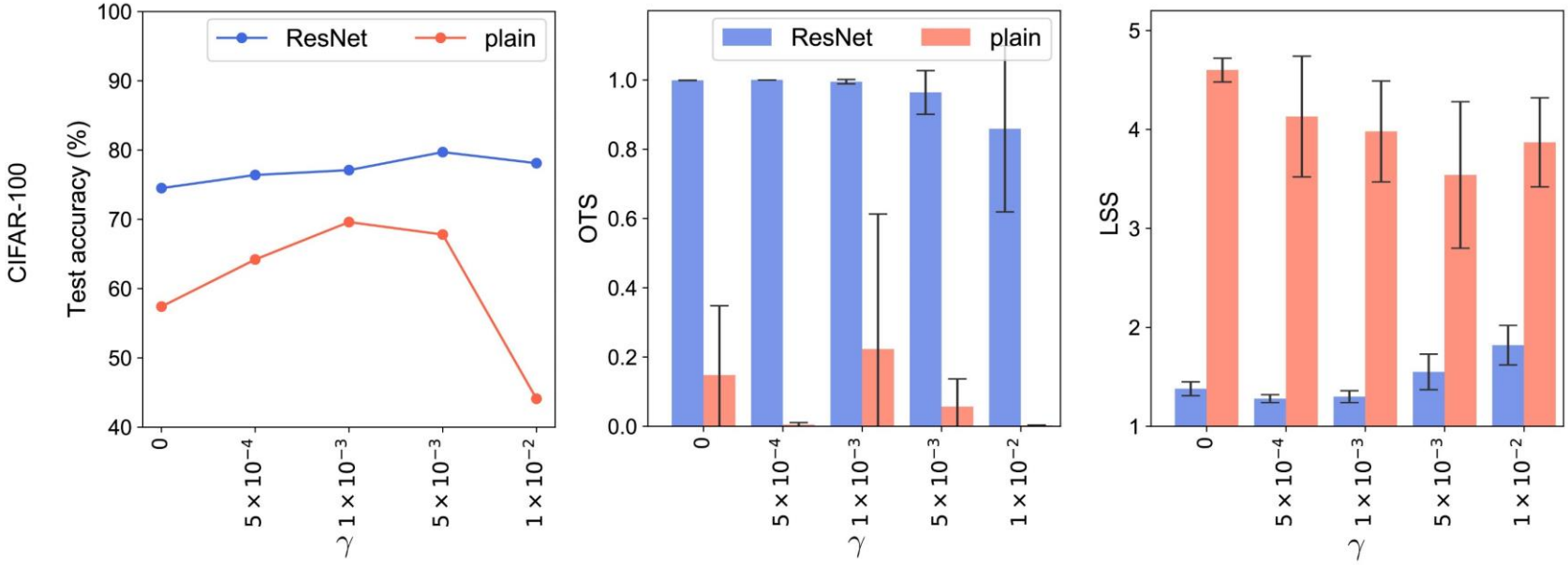


Optimization



Training accuracy increases with LSS decreasing!

Generalization



Both OTS and LSS of ResNets are closer to 1.

ResNet approximates the geodesic curve better

⇓

better generalization than plain network

Summary

- ResNet learns the curve induced by the optimal transport map.
- Explain why ResNet is superior to plain networks in terms of optimization and generalization.
- ResNet is an efficient engineering realization of data transformation.
- Provide insights into designing robust models with stronger learning ability.

Acknowledgements

Thanks for your attention!