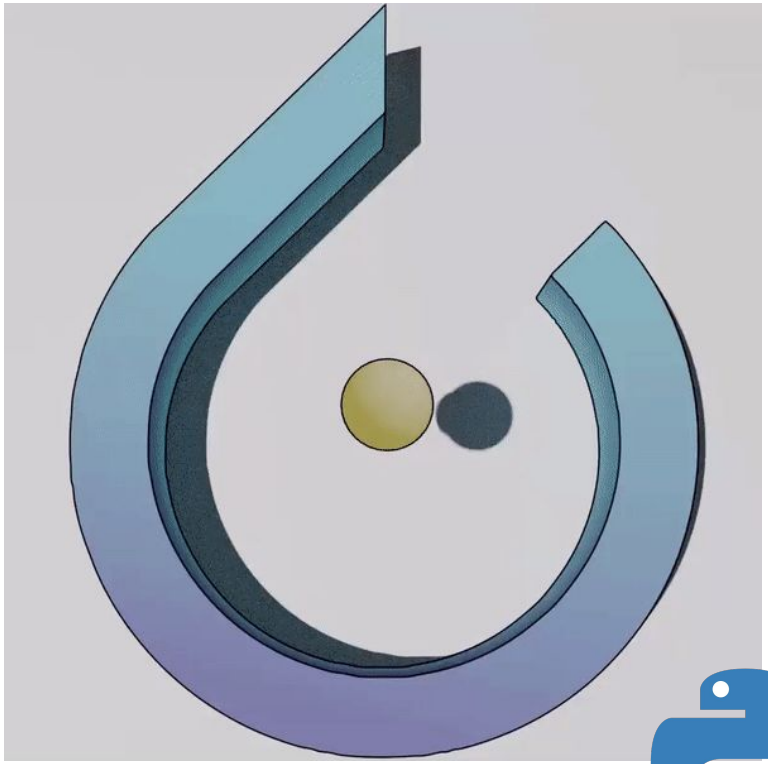


NeuralODEs

A journey

Computational Statistics | 01/03/2022

Luca Caivano
Paulina Moskwa
Manfred Nesti

Roadmap

Model

From ResNets
to Neural Ordinary
Differential Equations
(NeuralODEs)

2D Dataset

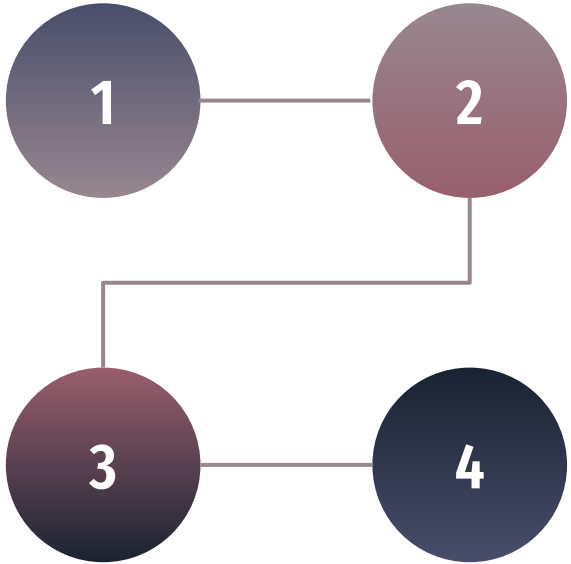
NeuralODEs for Binary Classification
of problematic dataset:
half-moons, concentric annuli,
concentric spirals

MNIST Dataset

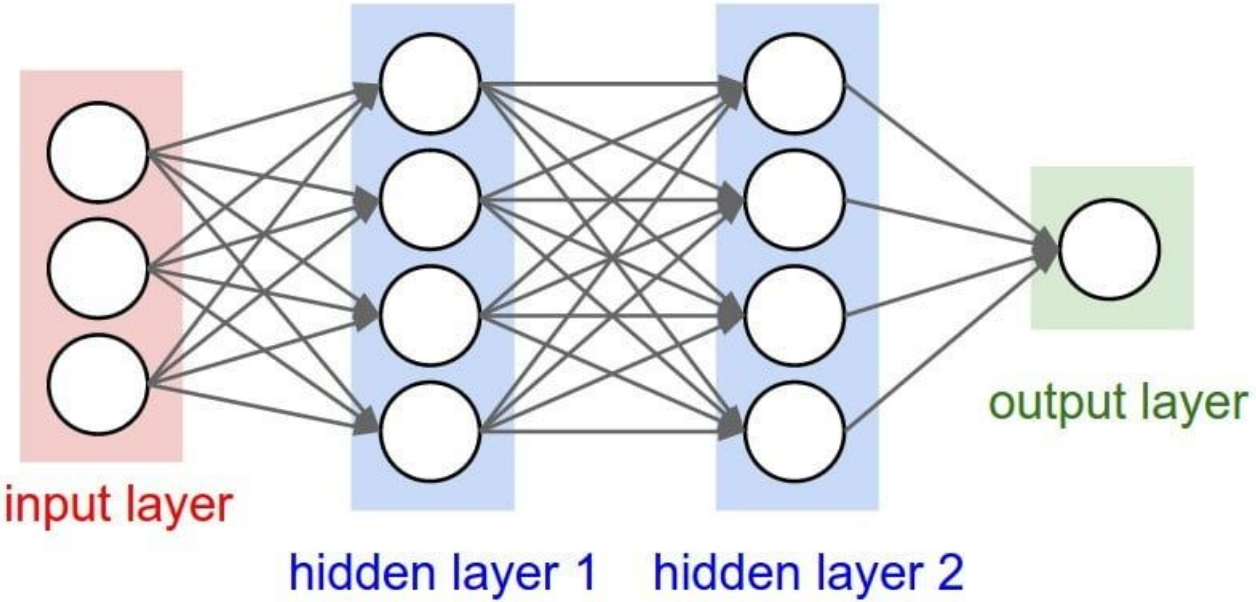
NeuralODEs for
Image Classification:
MNIST dataset

Conclusions

Advantages of NeuralODEs

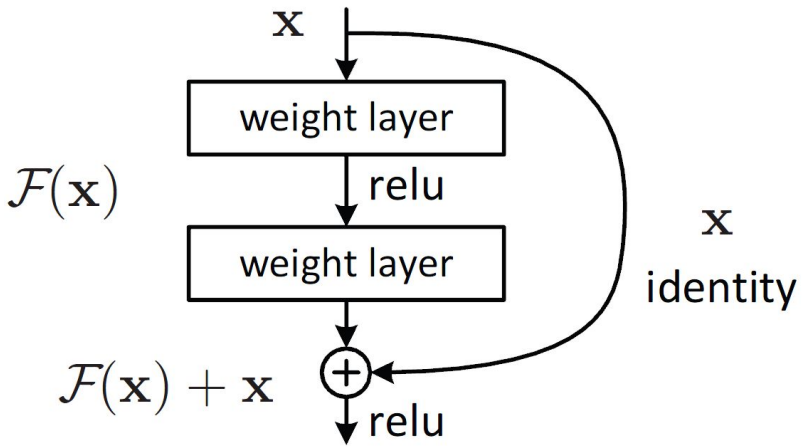


Neural Networks in a Nutshell



From ResNet to NeuralODE

$$\mathbf{z}_{t+1} = \mathbf{z}_t + f\left(\mathbf{z}_t, \theta_t\right)$$ ResNet



From ResNet to NeuralODE

$$\mathbf{z}_{t+1} = \mathbf{z}_t + f\left(\mathbf{z}_t, \theta_t\right)$$ ResNet

$$\downarrow$$

$$\mathbf{z}\left(t + \Delta t\right) = \mathbf{z}\left(t\right) + \Delta t f\left(\mathbf{z}\left(t\right)\right)$$ Explicit Euler

From ResNet to NeuralODE

$$\mathbf{z}_{t+1} = \mathbf{z}_t + f\left(\mathbf{z}_t, \theta_t\right)$$ ResNet

$$\downarrow$$

$$\mathbf{z}\left(t + \Delta t\right) = \mathbf{z}\left(t\right) + \Delta t f\left(\mathbf{z}\left(t\right)\right)$$ Explicit Euler

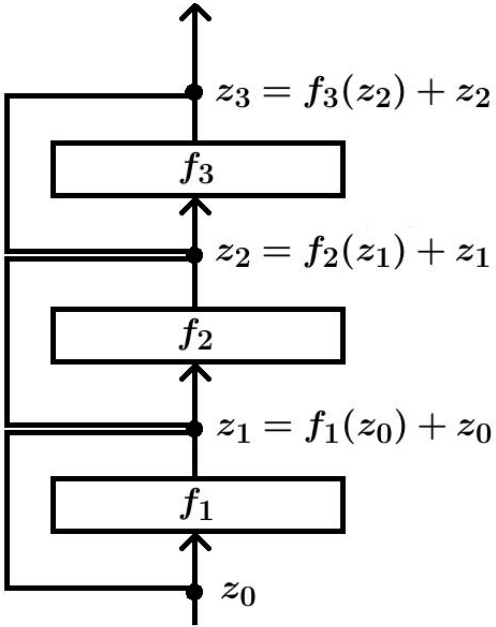
$$\downarrow$$

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

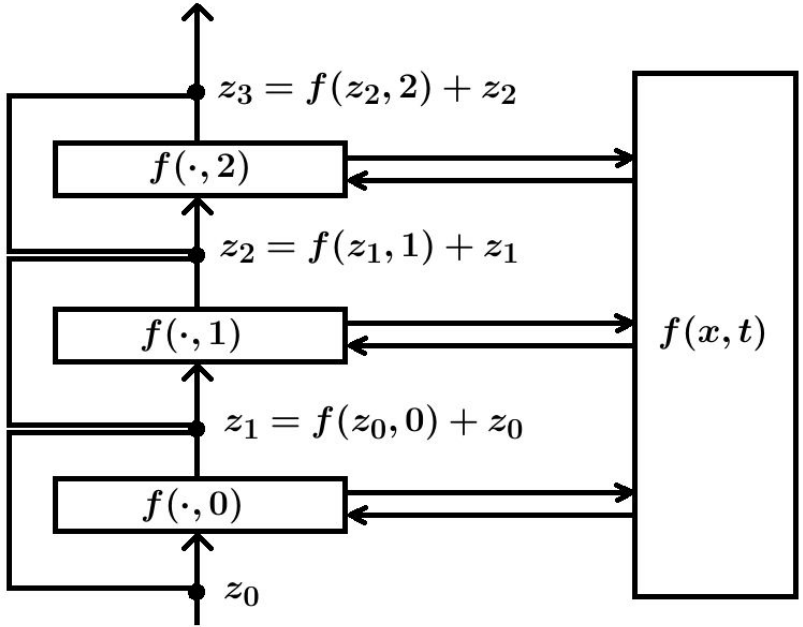
$$\begin{cases} \dot{\mathbf{z}}(t) = f_{\theta(t)}(t, \mathbf{x}, \mathbf{z}(t)) & t \in [0, T_f] \\ \mathbf{z}(0) = h_x(\mathbf{x}) \\ \hat{\mathbf{y}}(t) = h_y(\mathbf{z}(t)) \end{cases}$$

ResNet vs NeuralODE

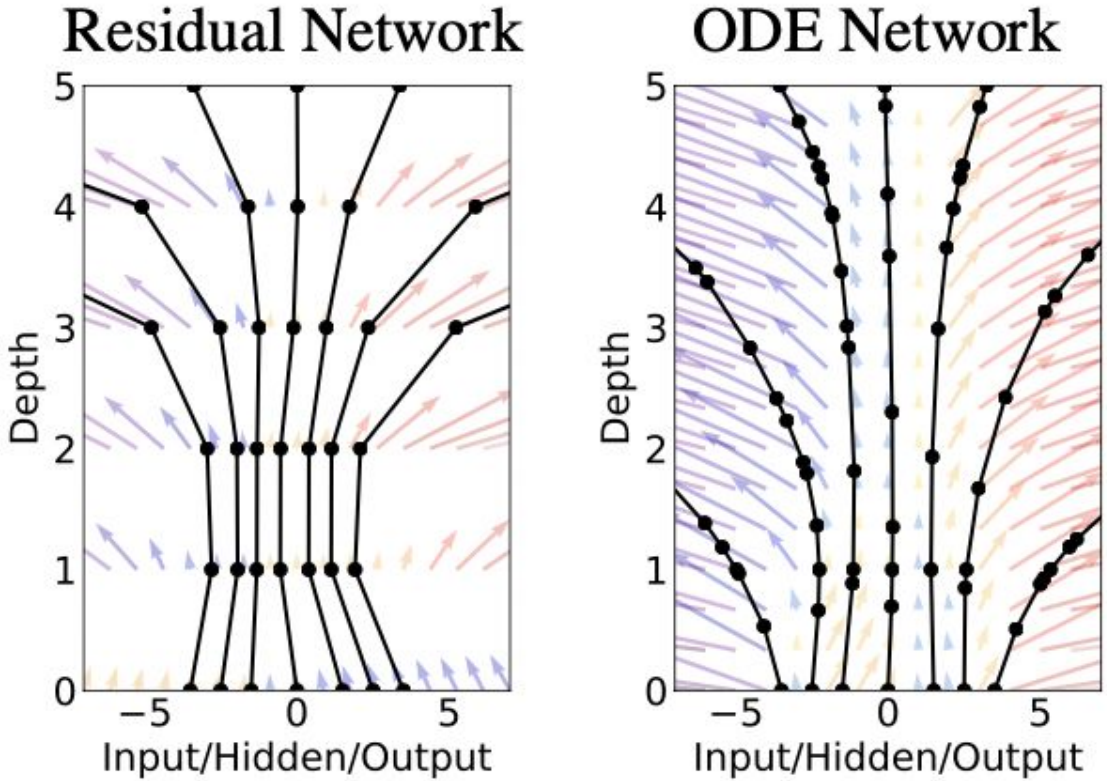
ResNet



ODENet

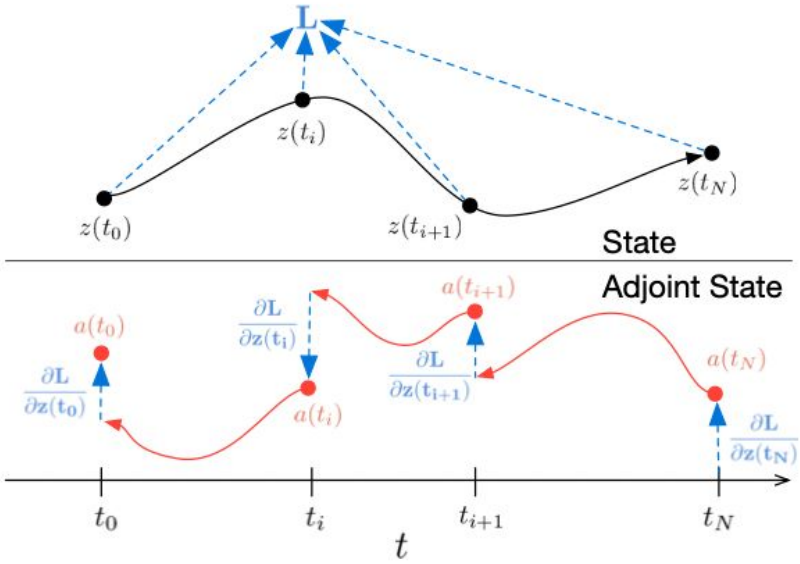


ResNet vs NeuralODE



Adjoint method

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

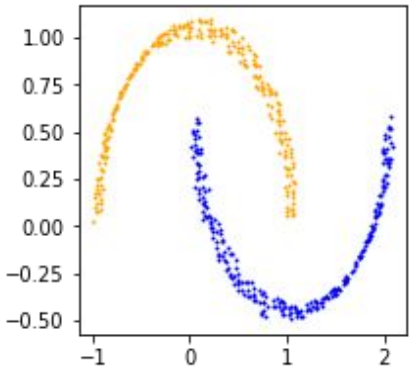


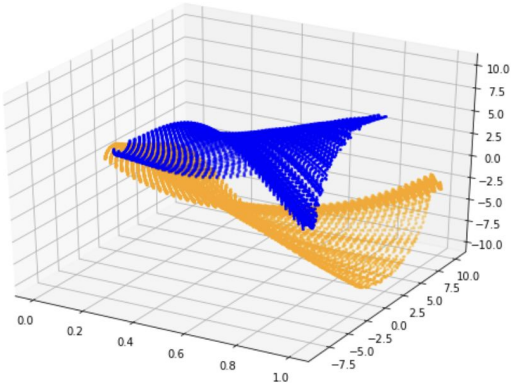
$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^\mathsf{T} \frac{\partial f(\mathbf{z}(t), t, \theta)}{\partial \mathbf{z}}$$

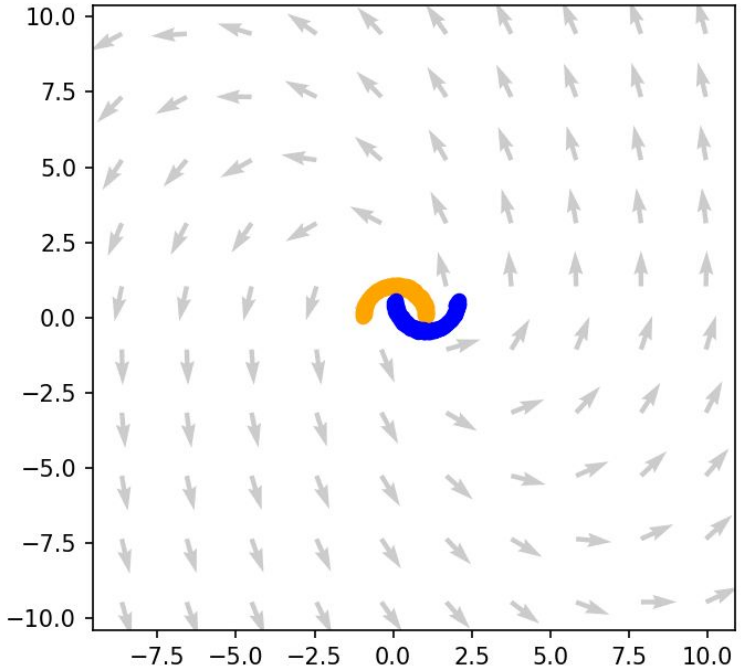
$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} \mathbf{a}(t)^\mathsf{T} \frac{\partial f(\mathbf{z}(t), t, \theta)}{\partial \theta} dt$$

2D Classification

Half Moons Problem

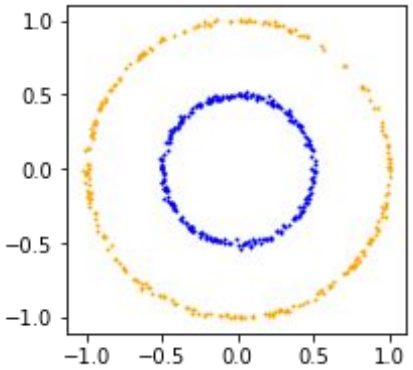


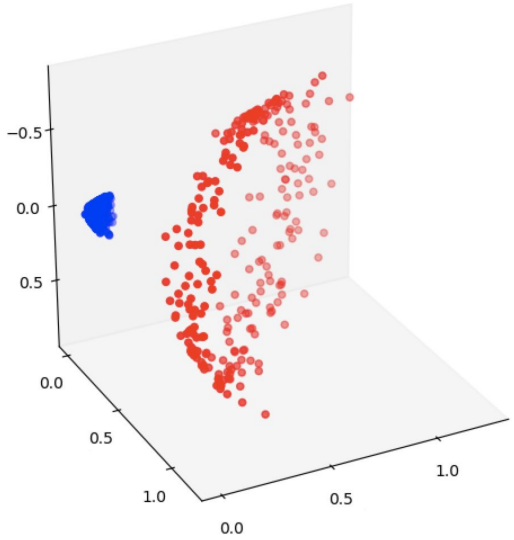


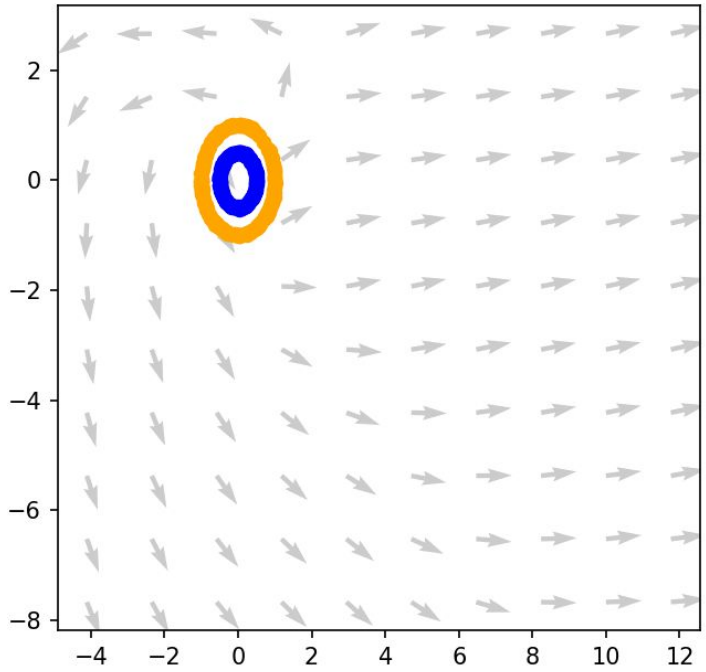


2D Classification

Concentric Annuli Problem

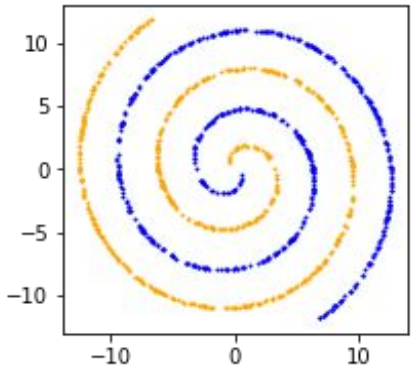


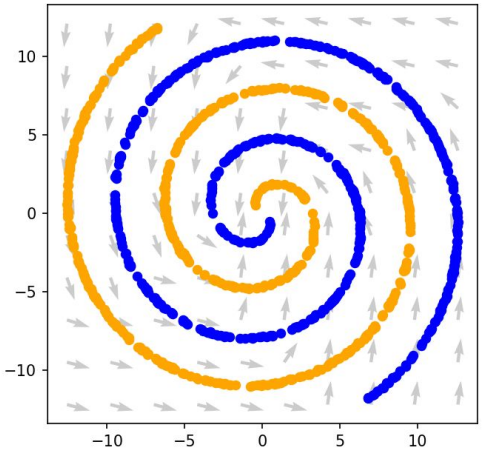


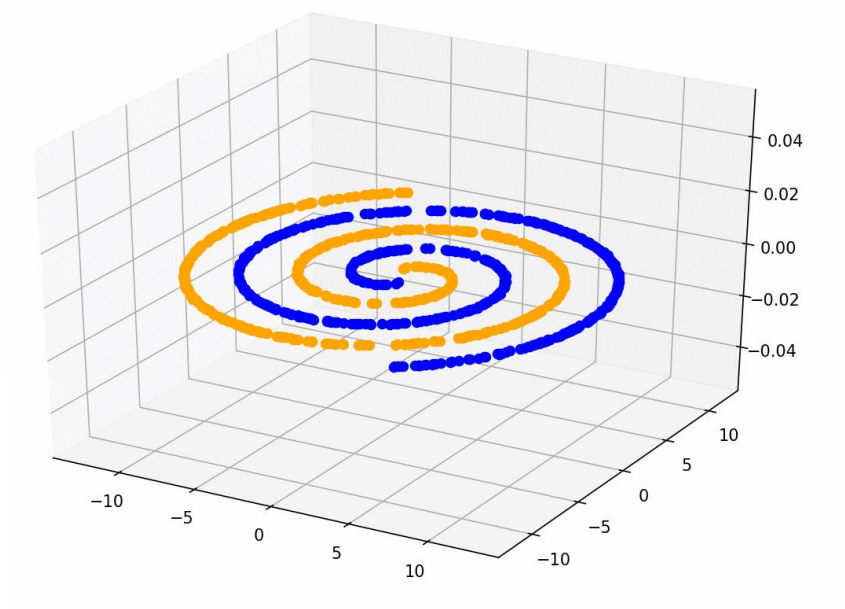


Augmented Classification

Concentric Spirals Problem

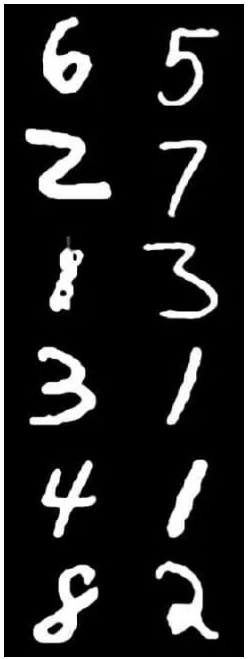




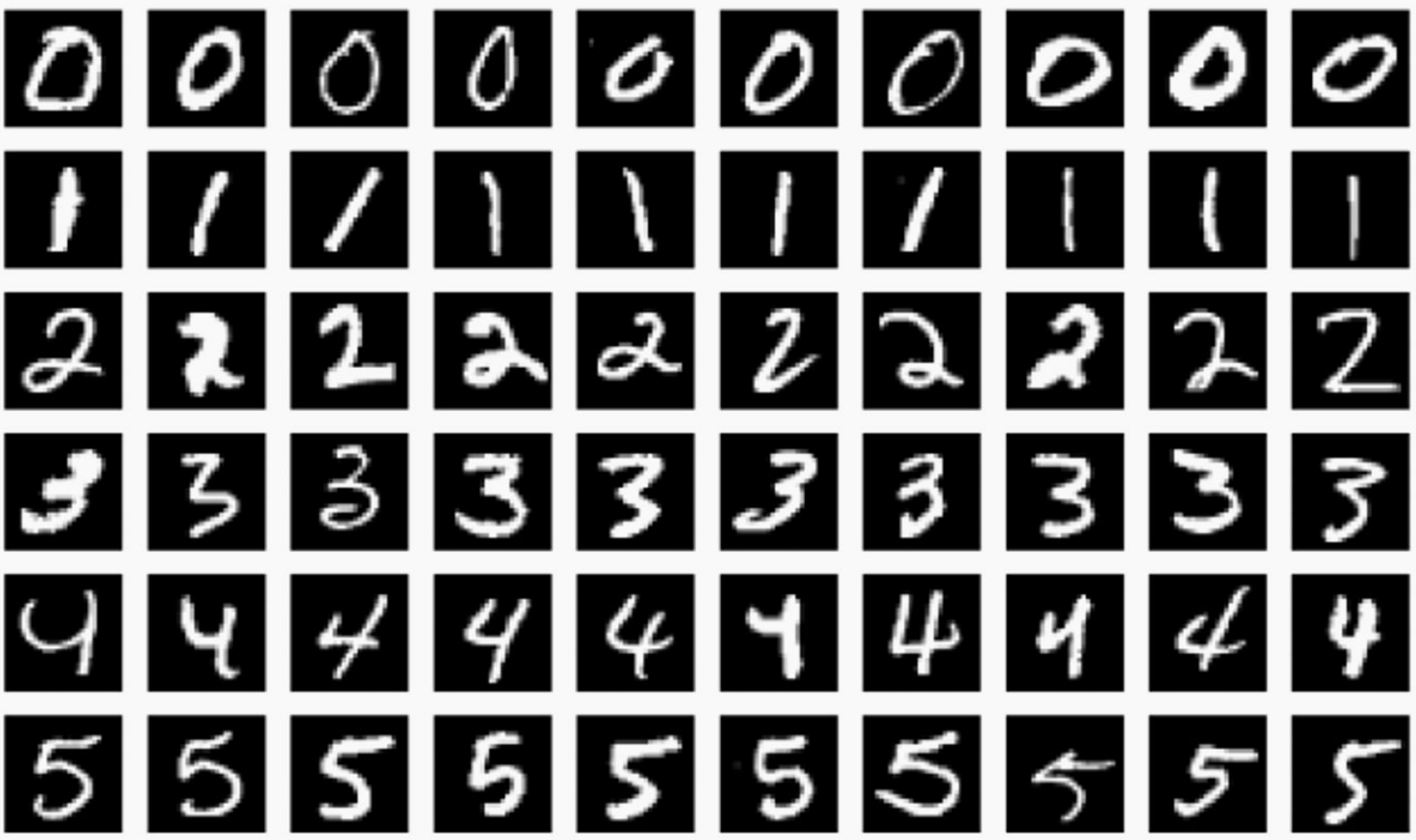


Augmented version

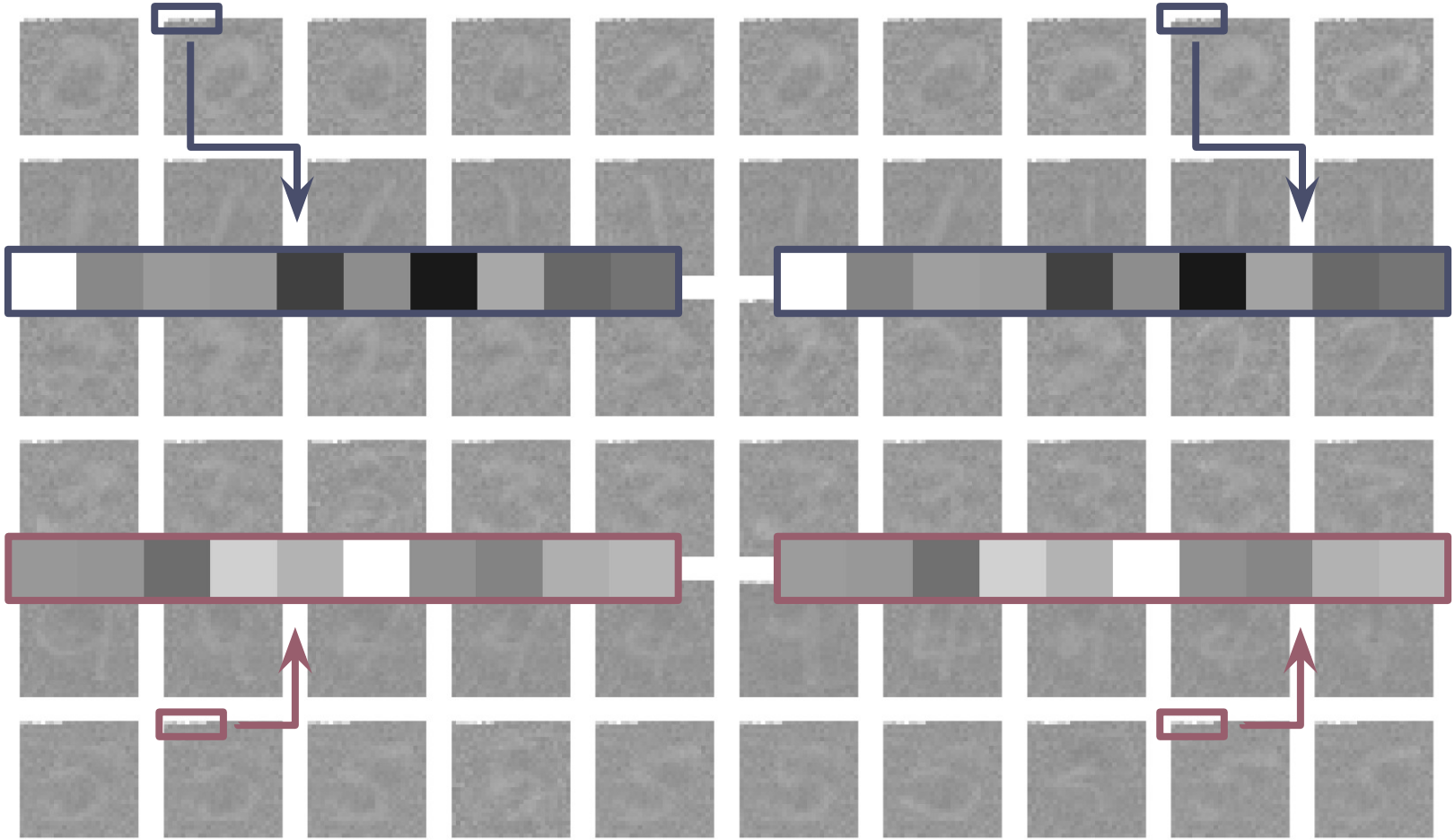
MNIST Dataset Classification



MNIST Dataset Classification



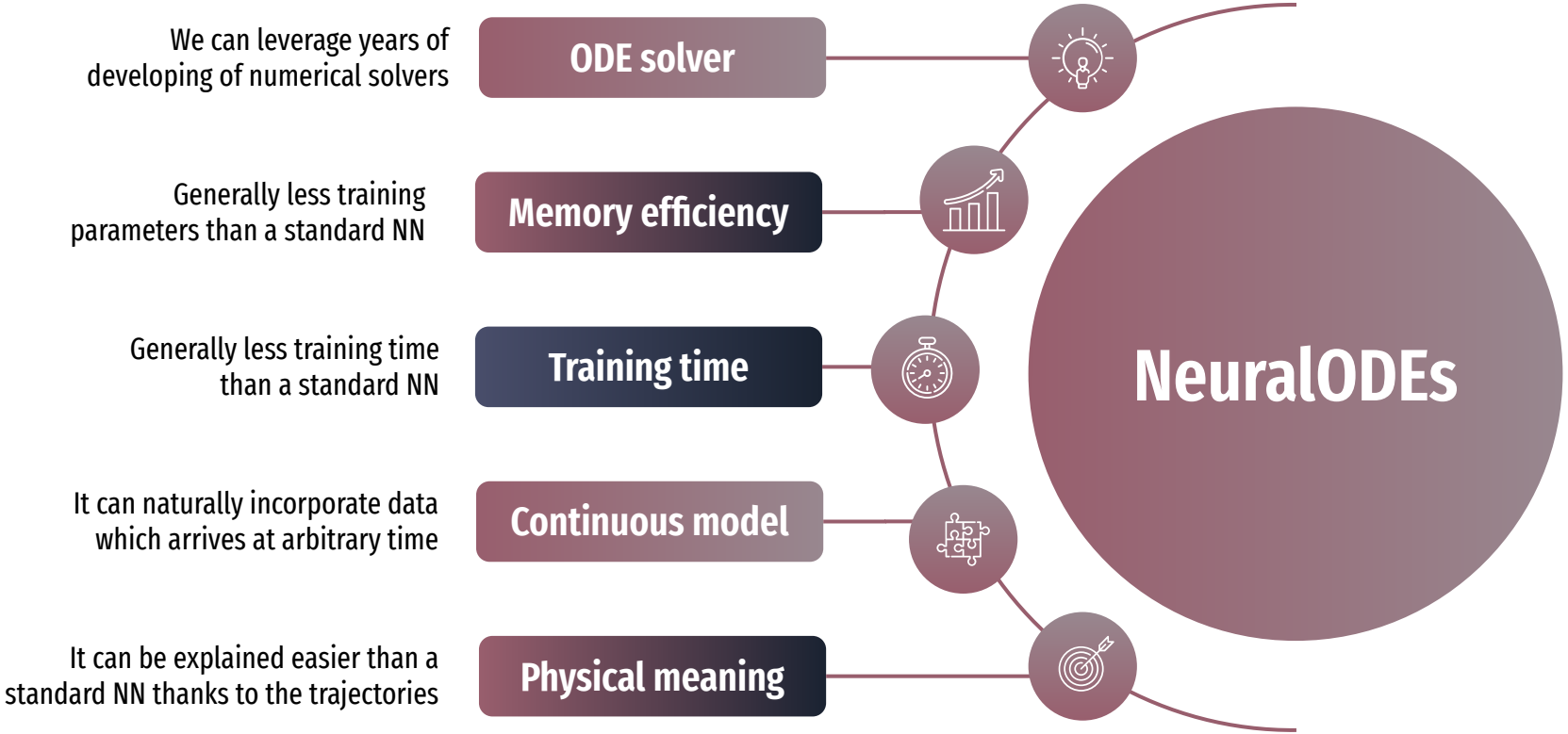
MNIST Dataset Classification



Accuracy on MNIST Dataset

	ODENet - Simple	ODENet - Expert	Best (CNN)
Accuracy on training set	87.41%	-	-
Accuracy on test set	86.57%	99%	99%
Number of parameters	25.888	220.000	1.514.187

Conclusions



References

Ricky T. Q. Chen, et al. | “Neural Ordinary Differential Equations” |
32nd Conference on NeurIPS, 2019

S. Massaroli, M. Poli, et al. | “Dissecting NeuralODEs” |
34th Conference on NeurIPS, 2021

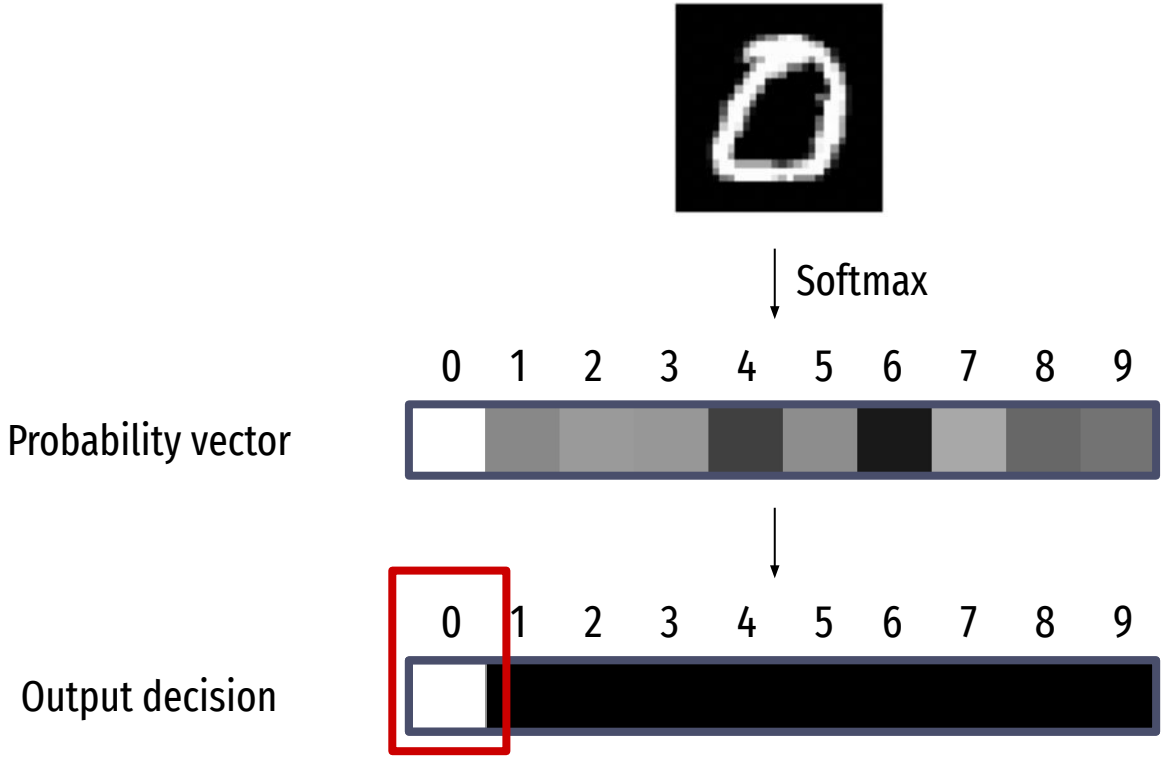
Software: *TorchDyn: Implicit Models and Neural Numerical Methods in PyTorch* |
S.Massaroli, M. Poli, et. al | https://torchdyn.readthedocs.io/en/stable/tutorials/quickstart.html

Repository: *Dissecting NeuralODEs* |
S. Massaroli, M. Poli, et. al |
https://github.com/DiffEqML/diffeqml-research/tree/master/dissecting-neural-odes

Repository: *torchdyn/tutorials/* |
S. Massaroli, M. Poli, et. al | https://github.com/DiffEqML/torchdyn/tree/master/tutorials

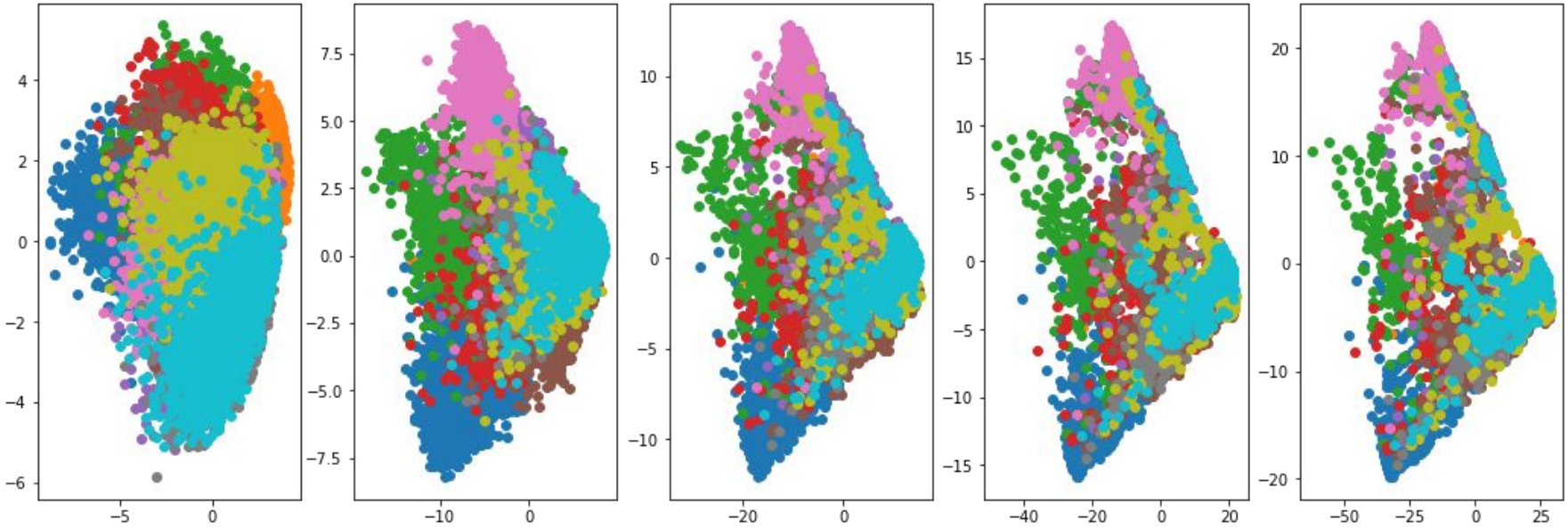
Additional material

Accuracy on MNIST Dataset



PCA on MNIST's Trajectories

Time independent PCA



PCA on MNIST's Trajectories

Time dependent PCA