"Improvements" on Neural Ordinary Differential Equations

Laurie Luo, Jimmy Chen[1]

[1]Harvey Mudd College

January 18, 2023

Table of Contents

Table of Contents

Limitations of Neural ODEs

There are certain functions that cannot be represented by the Neural ODE in an arbitrary dimension d.

Example

$$\begin{cases} g(x) = -1 & \text{if } \|x\| \leq r_1 \\ g(x) = 1 & \text{if } r_2 \leq \|x\| \leq r_3, \end{cases}$$

in which $0 < r_1 < r_2 < r_3$.

Example (Continued)

The figure shows $g(x)$ in two-dimension. The blue region maps to -1 and the red region points to 1.

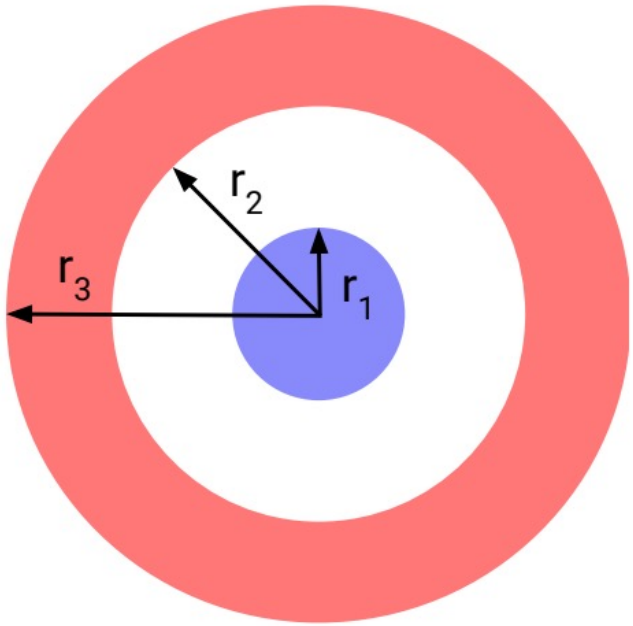

Figure: Diagram of $g(x)$ in two dimension ($\mathbb{R}^2$)

Augmented Neural ODEs

The *Augmented Neural ODE* introduces additional dimensions to the space to produce a simpler solution tot he problem.
In standard practice, the coordinates of the extra dimensions are initially set to zero, that is,

$$\mathbf{x}_0' = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{0}^p \in \mathbb{R}^p \end{bmatrix} \tag{1}$$

If the ANODE block has initial network $\mathrm{NN}_{\mathrm{ANODE}}$ then the state evolution is described as

$$\mathbf{x}'(t) = \int \mathrm{NN}_{\mathrm{ANODE}}(\mathbf{x}', t)dt \tag{2}$$

the same as vanilla NODEs.

Examples of ANODE

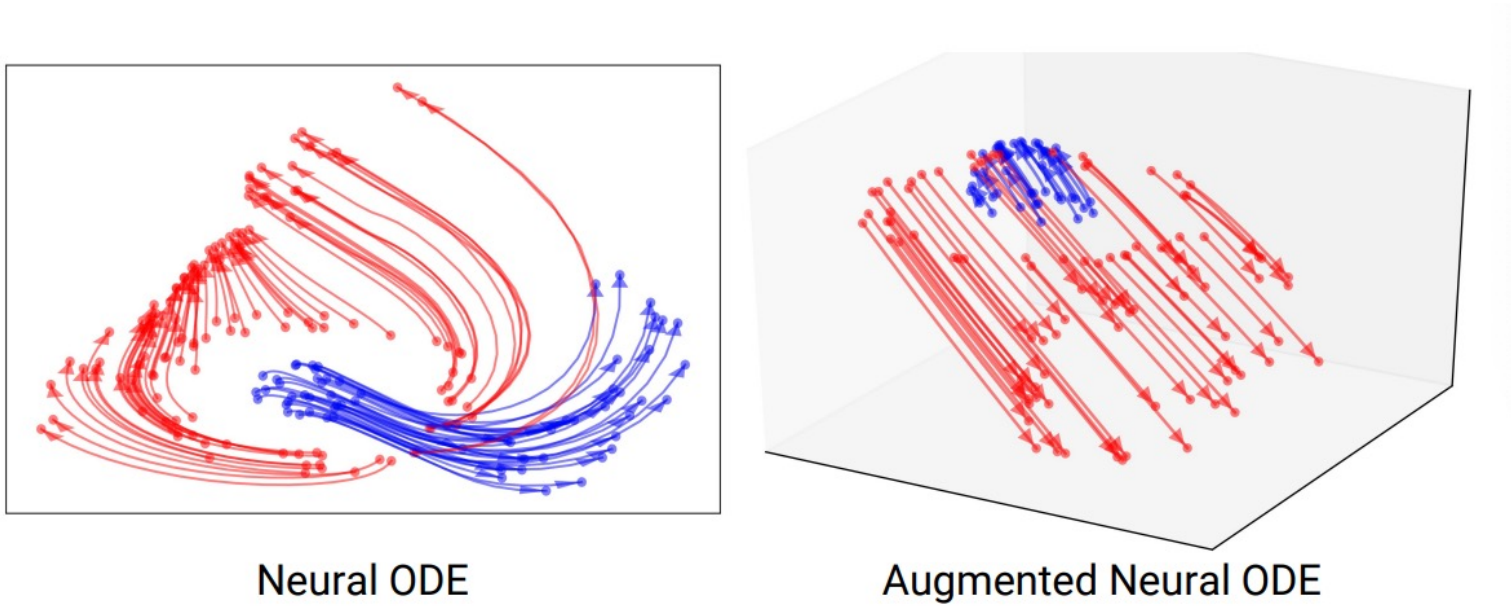

Figure: Neural ODE and Augmented Neural ODE

Examples of ANODE

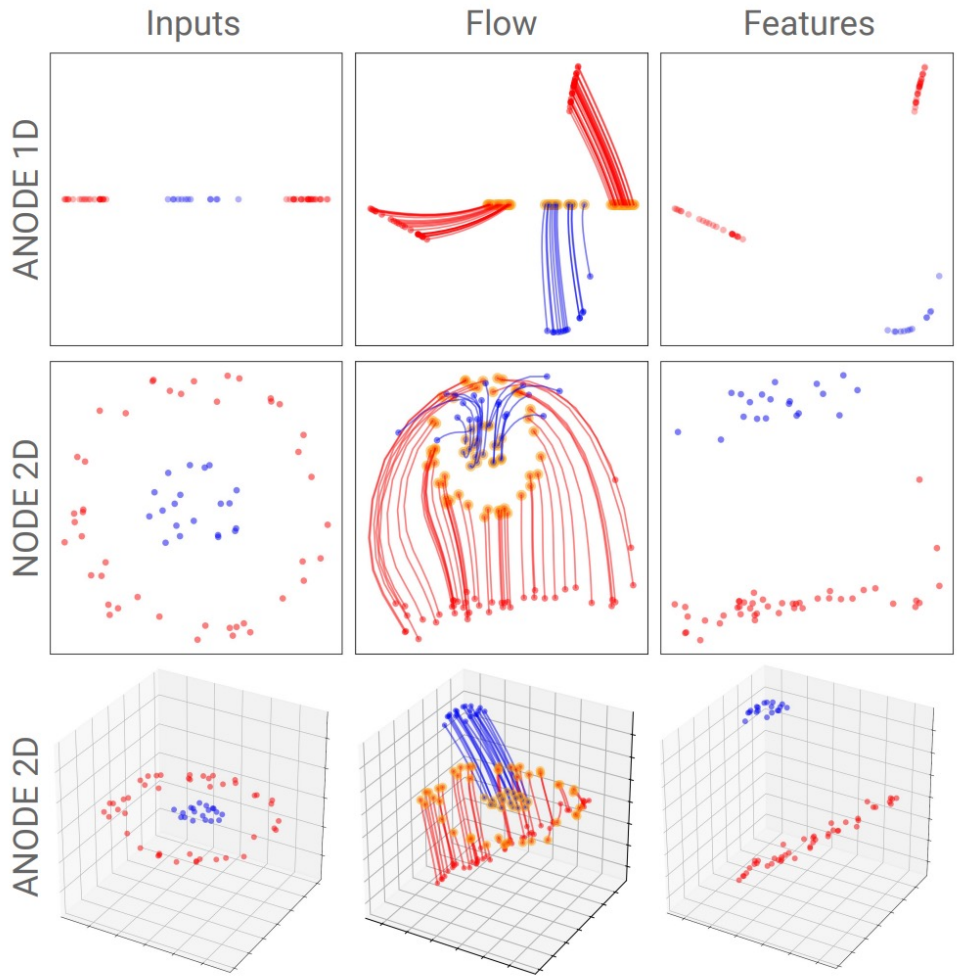

Figure: Flows learned by NODEs and ANODEs

Original Vanilla Node Architechture

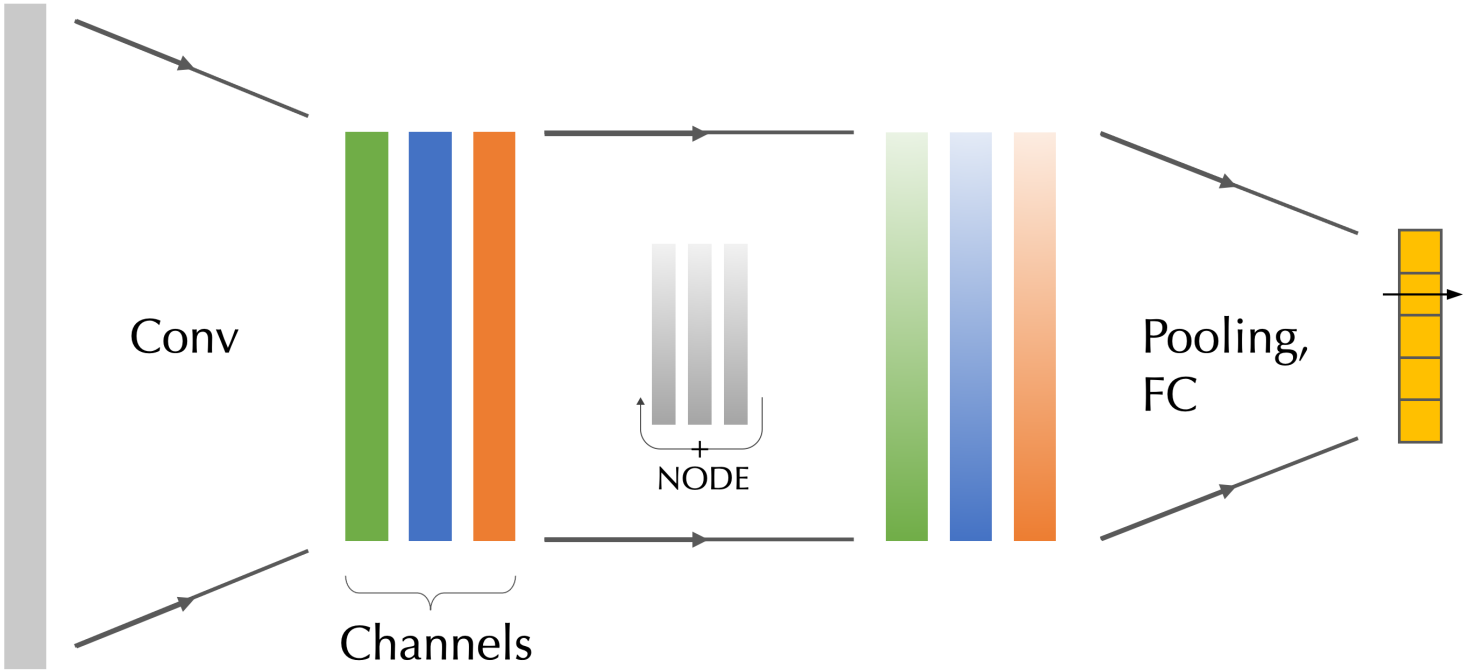

Figure: Illustration of the used vanilla NODE architechture

ANODE Architechture

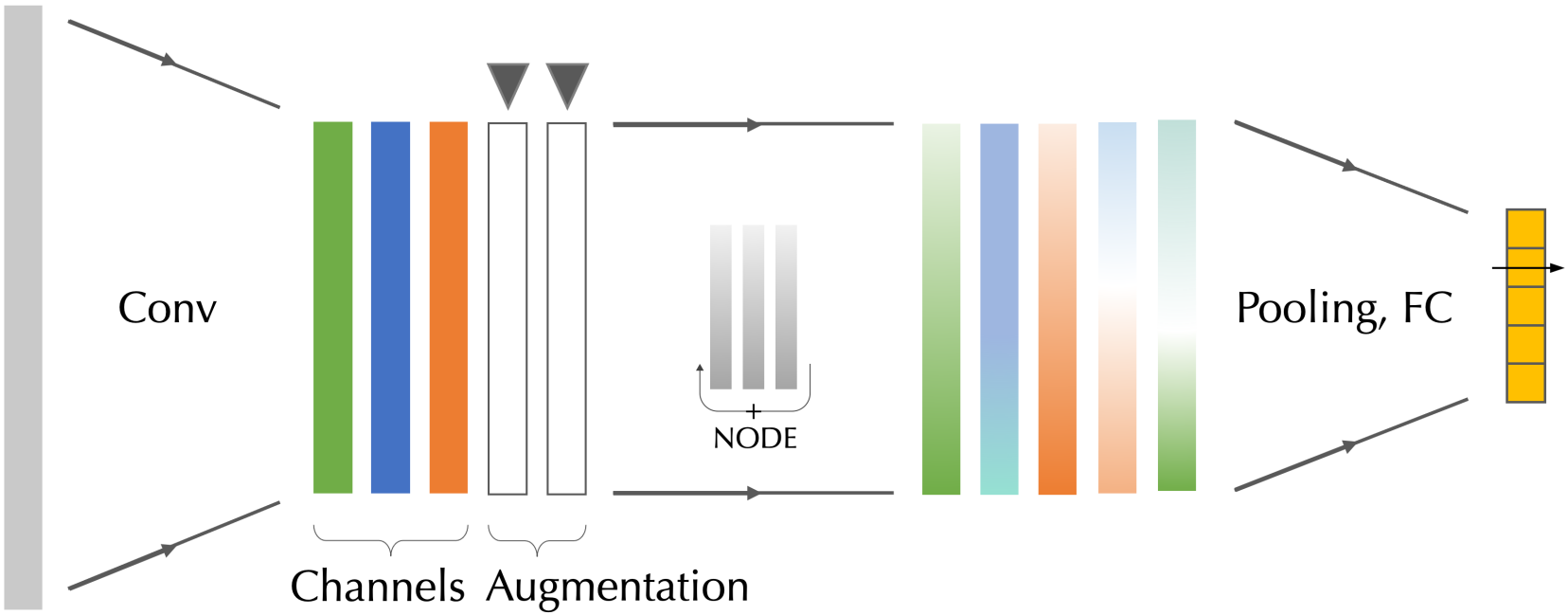

Figure: Illustration of the used Augmented NODE architecture

Results for ANODE on ECG Dataset

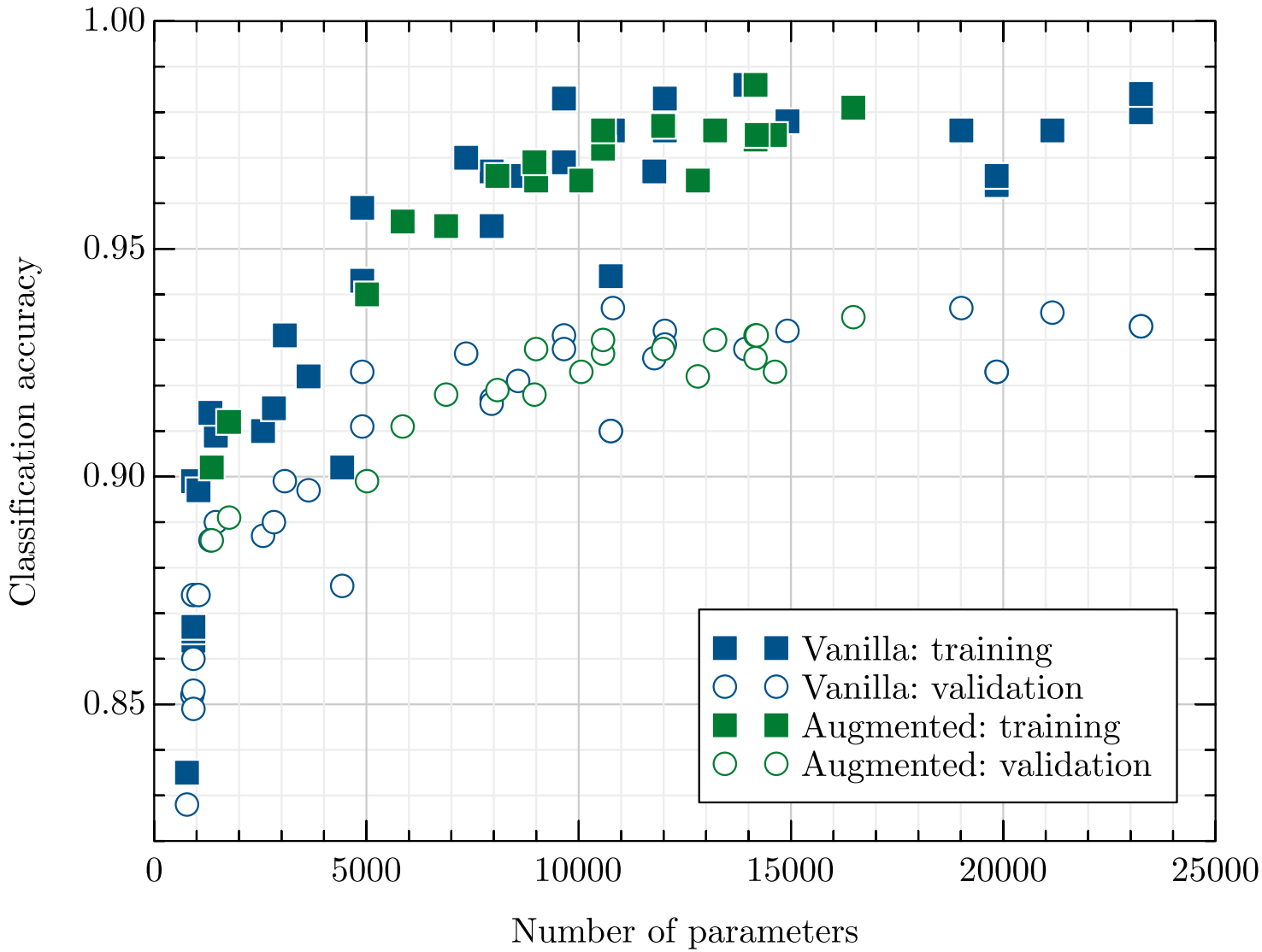

Figure: Plot of classification accuracy versus number of parameters for vanilla and

Table of Contents

Adaptive-Depth Neural ODEs

As mentioned in last section, it is impossible for a Neural ODE to solve a function that involves crossing trajectory. Besides augmented neural ODEs, we are going to introduce *Adaptive-Depth Neural ODEs* to tackle the problem.

Adaptive-Depth Neural ODEs

The integration time span is no long constant and instead decided based on the specific input using a small NN.

The integration time span is $[0, \text{NN}_{\text{depth}}(\mathbf{x}_0)]$. The output of the entire AD NODE block is thus
which leads to

$$\mathbf{x}_{\text{out}} = \int_0^{\text{NN}_{\text{depth}}(\mathbf{x}_0)} \text{NN}_{\text{ADNODE}}(\mathbf{x}, t)dt. \tag{3}$$

Example of adaptive-depth NODE

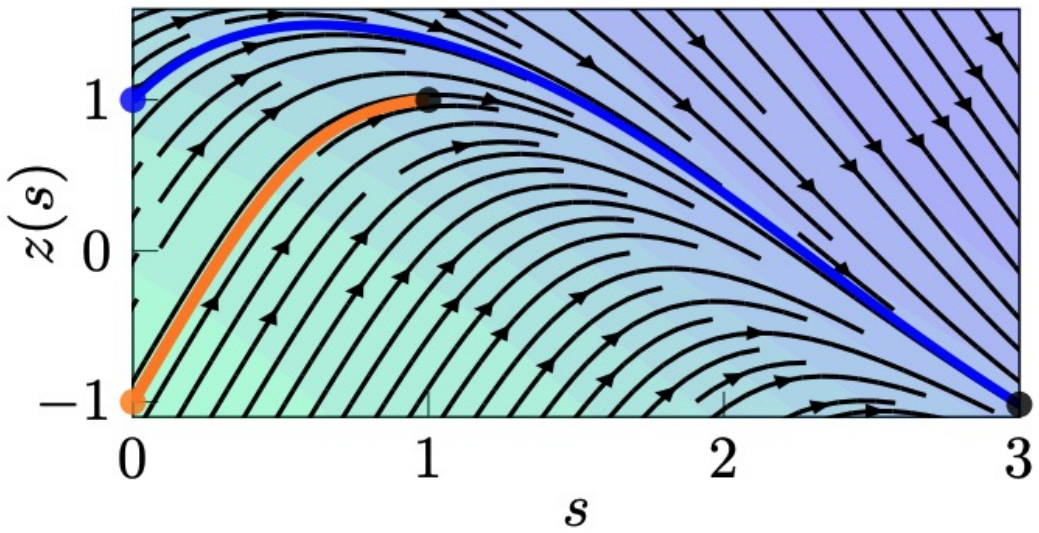

Figure: Adaptive-depth Neural ODEs that avoid the crossing trajectory

Results of Adaptive-Depth NODE

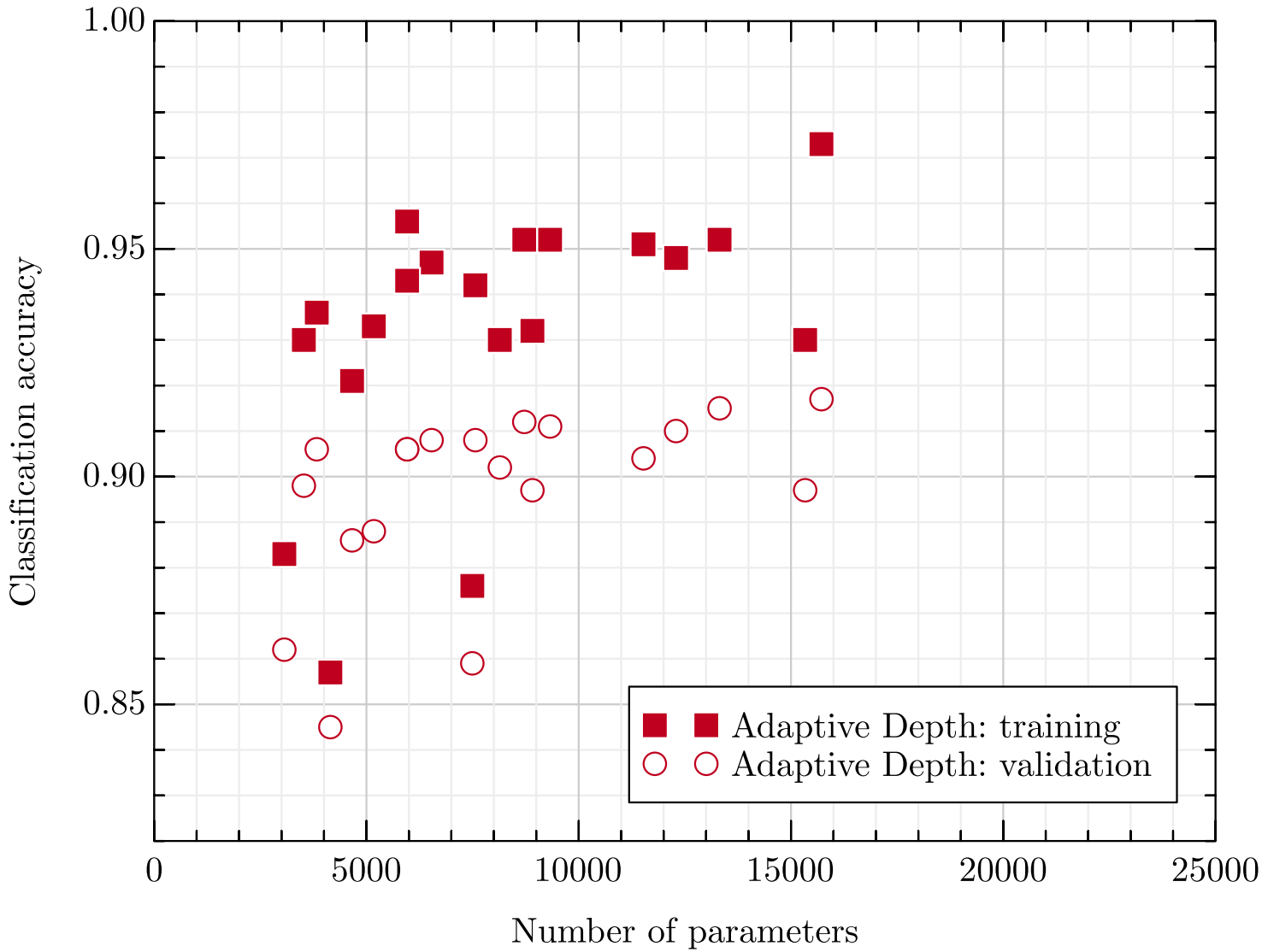

Figure: Plot of classification accuracy versus number of parameters for adaptive

Results of Adaptive-Depth NODE

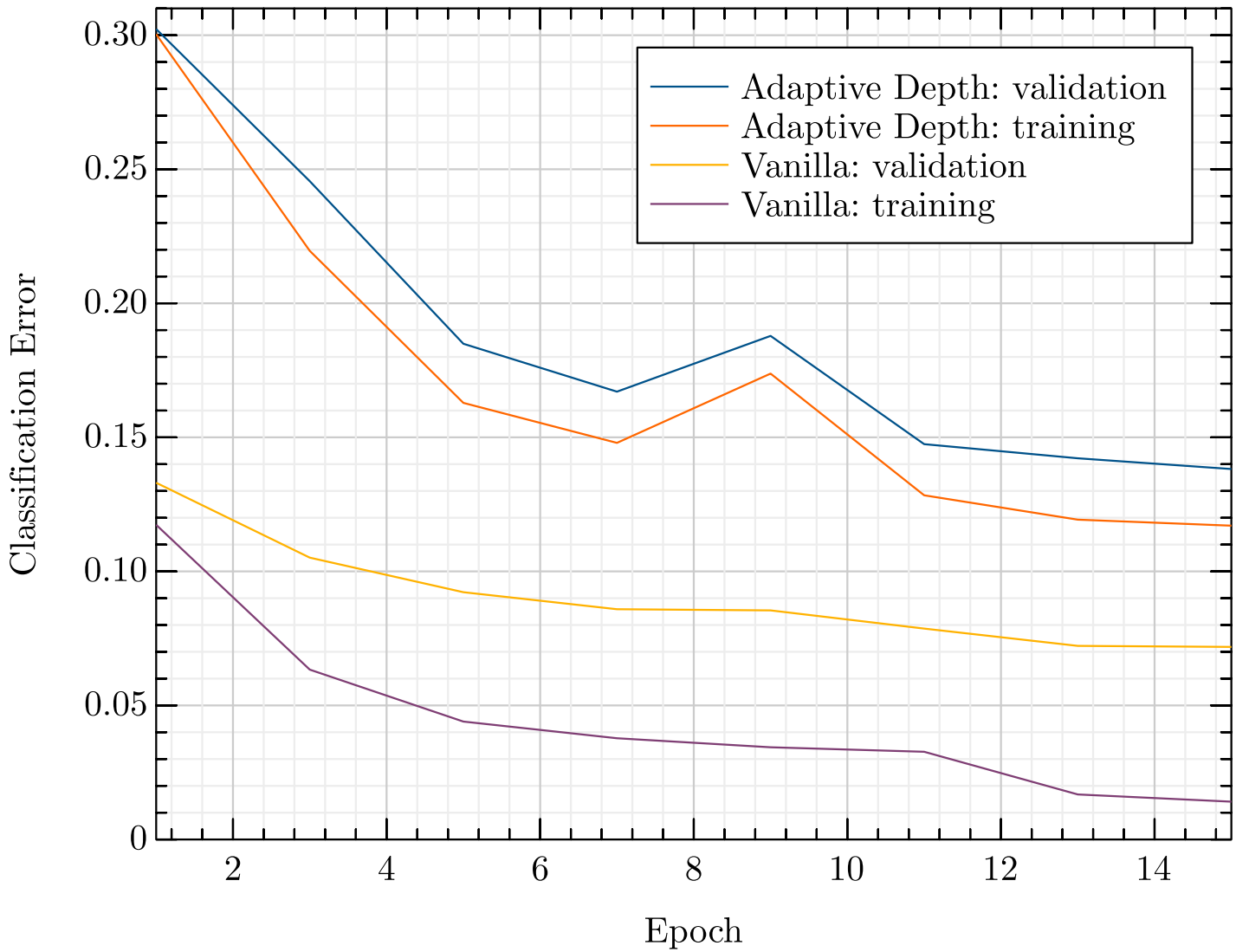

Figure: Plot of typical learning curves between vanilla and adaptive depth NODE

Table of Contents

Conclusions

References

[1] https://www.kaggle.com/datasets/shayanfazeli/heartbeat
[2] https://github.com/abaietto/neural_ode_classification
[3] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural Ordinary Differential Equations. arXiv. https://doi.org/10.48550/ARXIV.1806.07366

Thank you!