Augmented Neural ODEs

Kerri Lu

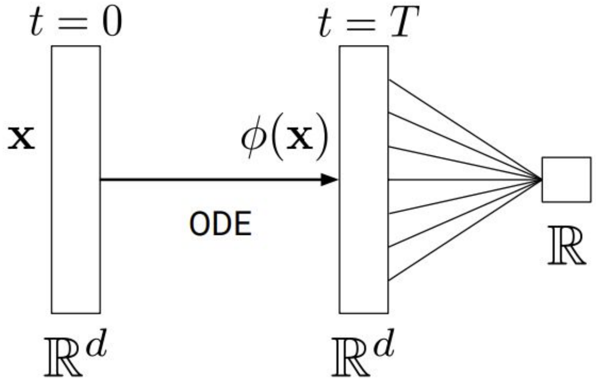
May 3, 2023

Background: Neural ODEs

- NODEs learn a mapping from input **x** to output Φ(**x**) = **h**(T) by solving an IVP:

$$\frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{h}(t), t), \qquad \mathbf{h}(0) = \mathbf{x}$$

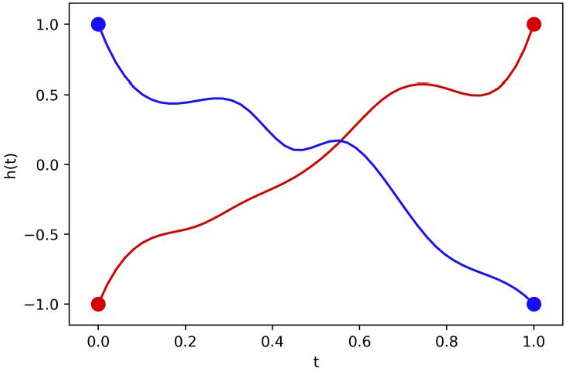
  where **f** is a differentiable, dimension-preserving neural net function, and we adjust **f**'s weights to fit true output labels.



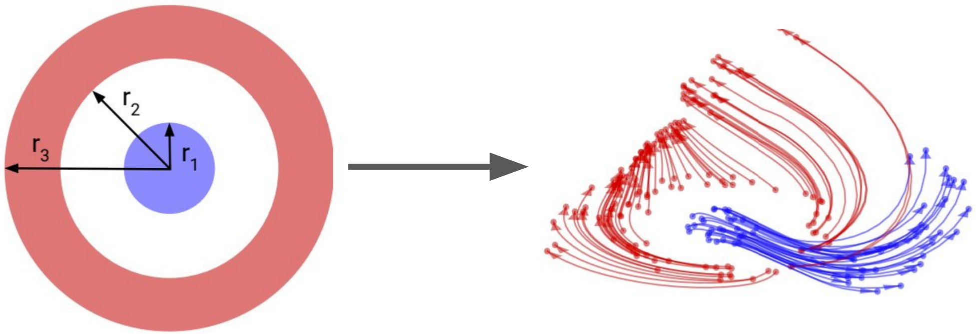
- Continuous, dynamics-based analog to residual networks
- Can do regression or classification by adding linear layer

Motivation: limitations of Neural ODEs

- **Problem**: NODE representations preserve topology of input space, so there are many functions it cannot represent
  - ODE flows (trajectories of **h**(t) over time) for different inputs **x** can't intersect
- Example 1: g(-1) = 1 and g(1) = -1



- Example 2: concentric circles, mapping blue to 1 and red to -1



Augmented Neural ODEs (Dupont et al 2019)

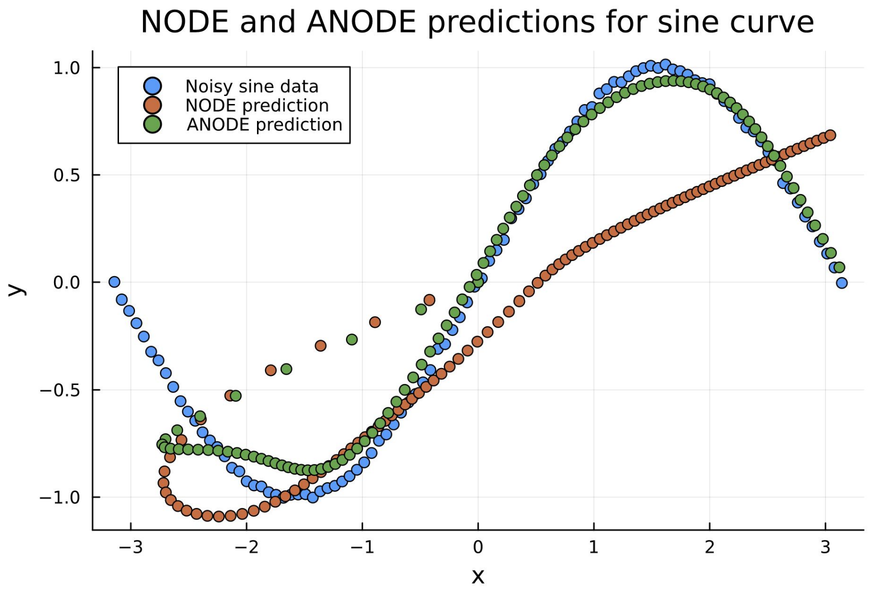
- **Key idea**: augment input space from *d* to *(d+p)* dimensions via zero-padding, so ODE flows are “lifted” to higher dimensions and don’t intersect

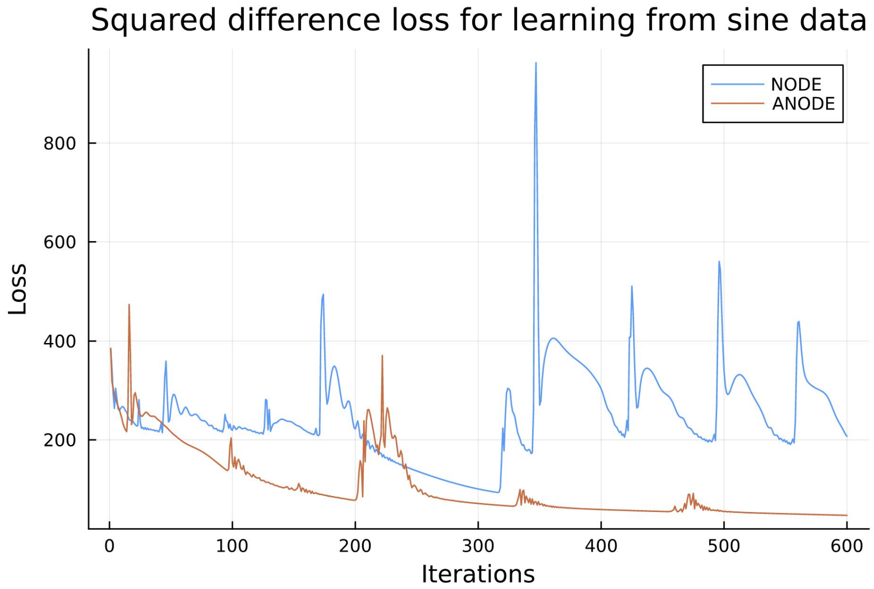
$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{h}(t)\\ \mathbf{a}(t)\end{bmatrix} = \mathbf{f}(\begin{bmatrix}\mathbf{h}(t)\\ \mathbf{a}(t)\end{bmatrix}, t), \qquad \begin{bmatrix}\mathbf{h}(0)\\ \mathbf{a}(0)\end{bmatrix} = \begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix}$$

- ANODEs have smoother and simpler flows, “empirically more stable, generalize better and have a lower computational cost” than NODEs
- Performance metrics: loss, number of function evaluations, convergence

Experiments: learning the sine curve

- ANODE fits sine data, while NODE doesn't converge (flows would intersect)
- Params: augmentation dim *p*=3; two-layer neural net, hidden dim 20

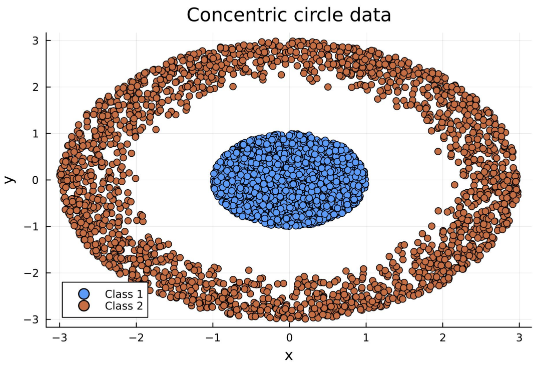


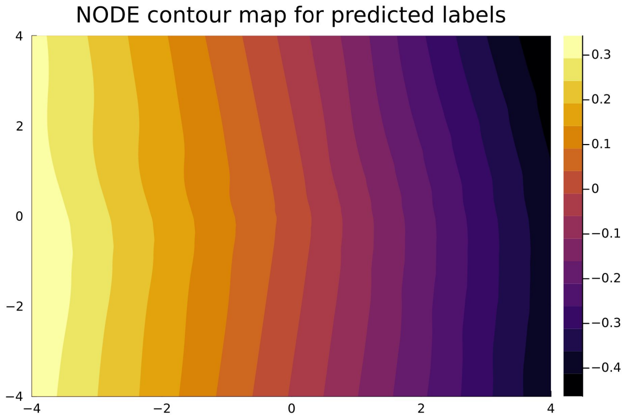


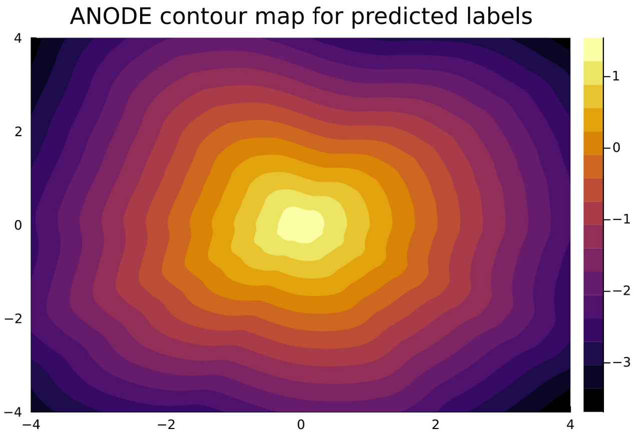
Experiments: 2D classification of concentric circles

- Goal: classify inner circle points with 1 and outer ring with -1
- Params: augmentation dim *p*=5; three-layer neural net with hidden dim 64

- ANODE predicted labels converge to expected contours, while NODE doesn’t

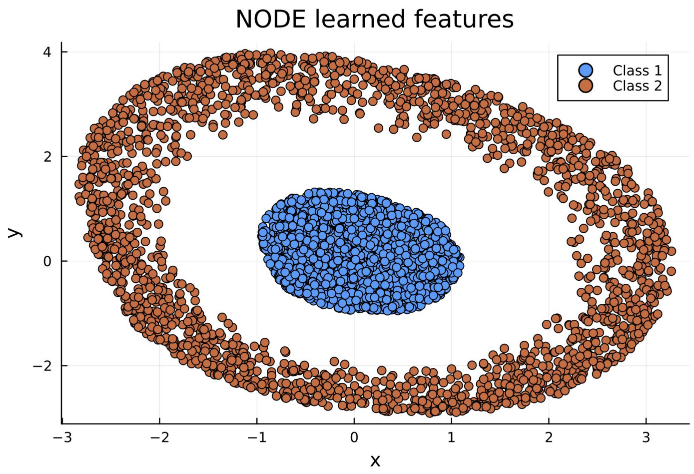


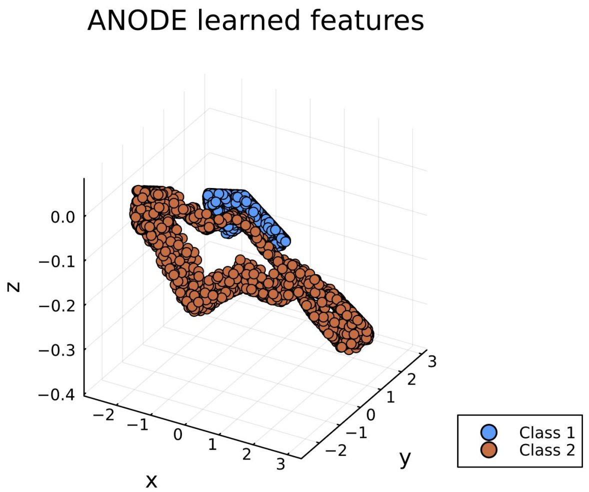




Experiments: 2D classification of concentric circles

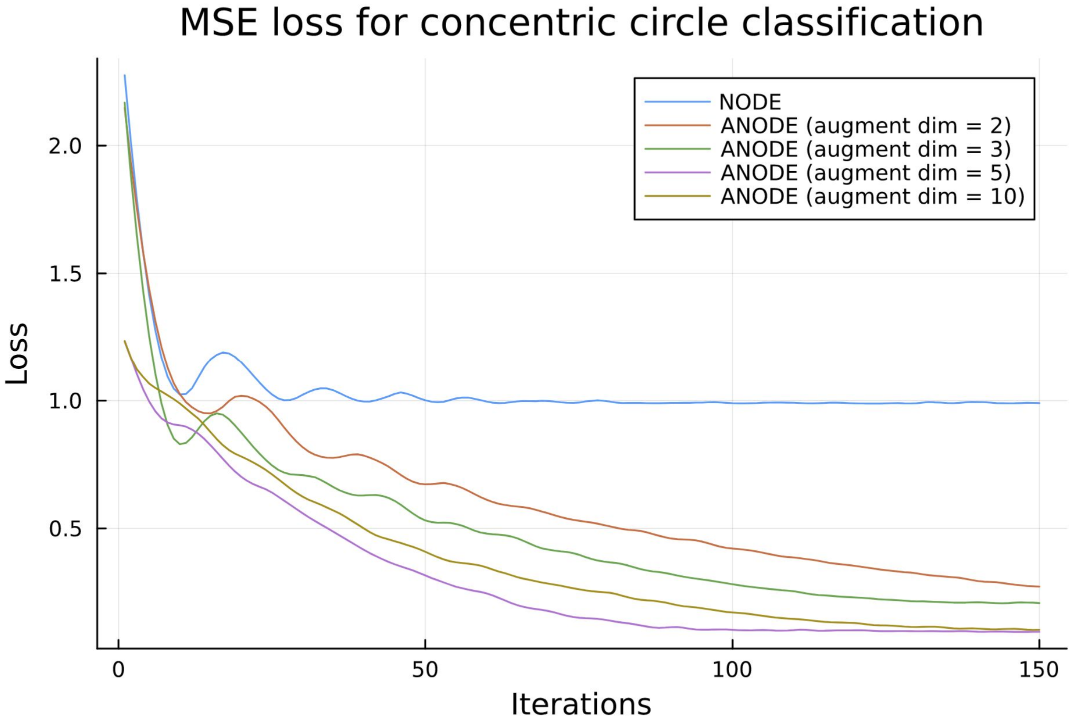
- Visualize features Φ(**x**), which are final location of ODE flows **h**(T)
- In ANODE, higher dimensionality allows classes to “lift out” and become linearly separable without intersecting flows





Experiments: 2D classification of concentric circles

- Increasing augmentation dimension allows convergence in fewer iterations



Next steps

- Experiment with ANODEs on image classification (e.g. MNIST), padding an extra channel of zeros
- Potential performance improvements:
  - “Input layer augmentation” with initial condition **h**(0) trained as an neural net that maps input **x** from d to (d+p) dimensions [Massaroli et al, 2020]
  - Regularization via randomly sampling ODE end time T [Ghosh et al, 2020]
  - Second order ANODEs [Norcliffe et al, 2020]

Takeaway

- Augmenting the space on which we solve neural ODE increases expressivity of the model, resulting in simpler flows for more complicated learning problems

References

[1] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. “Augmented Neural ODEs”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[2] Stefano Massaroli et al. “Dissecting Neural ODEs”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3952–3963.

[3] Arnab Ghosh et al. “Steer: Simple Temporal Regularization For Neural ODE”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14831–14843.

[4] Alexander Norcliffe et al. “On Second Order Behaviour In Augmented Neural ODEs”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5911–5921.