Neural Ordinary Differential Equations

Laurie Luo, Jimmy Chen[1]

[1]Harvey Mudd College

June 21, 2022

Table of Contents

Table of Contents

Neural Network

Neural Networks (NN) are a family of algorithms that models over datasets and provides predictions. They play a central role in modern machine learning process, particularly deep learning.

Example

$X = (\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_N, \mathbf{x}_N))$, where $\mathbf{x}_i \in X$ is the input domain and $\mathbf{y}_i \in Y$ is the output domain, and some sort of relationship exists between the two variables:

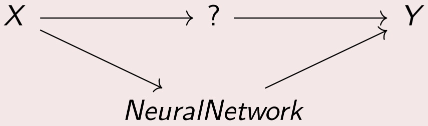
$$X \longrightarrow ? \longrightarrow Y$$

We would like to utilize the existing dataset and make prediction of the output $\mathbf{y}^*$ giving a new data point $\mathbf{x}^*$.

Example of NN (Continued)

Example

The highly flexible NN can be iteratively trained on the dataset, ultimately describing the dataset with accuracy. Using an NN turns the classical situation into the situation below:



Neural Network

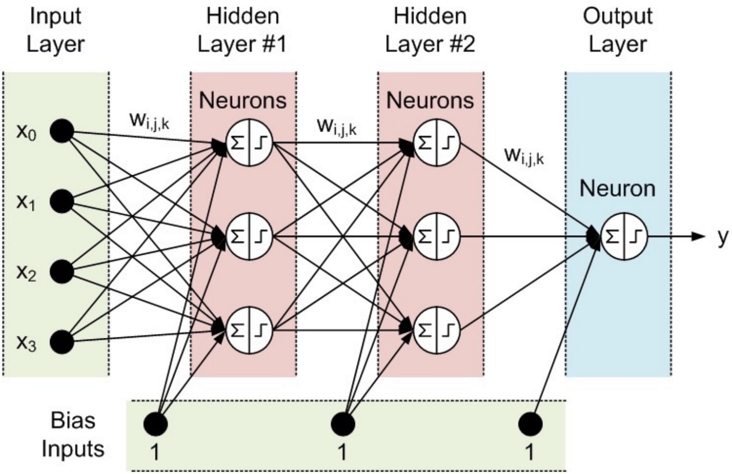


Figure: Detailed Illustration of Neural Network

Number of Layers in NN

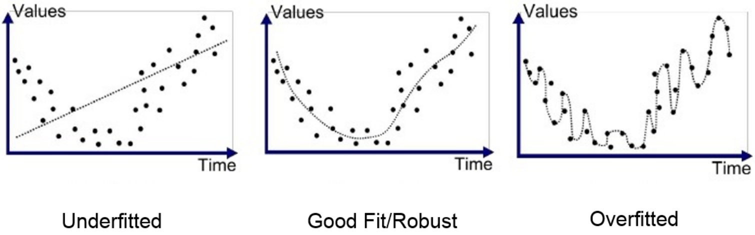


Figure: Underfitting and Overfitting

ResNet

One class of NNs effectively allows the number of layers to be adaptively optimized for the dataset: they are the *Residual Neural Networks (ResNet)*.

ResNets introduce shortcuts that skip over groups of layers by adding intermediate results from a few layers prior to the current output.

ResNet

One can consider a group of layers that has such a shortcut over it as a single residual block (or group). For the kth residual block, the output is the addition $\text{Res}_k(\mathbf{x}_k) + \mathbf{x}_k$, between the processed output of the internal layers and the original input.

$$\mathbf{x}_{k+1} = \text{ResBlock}_k(\mathbf{x}_k) = \text{Res}_k(\mathbf{x}_k) + \mathbf{x}_k, \quad (1)$$

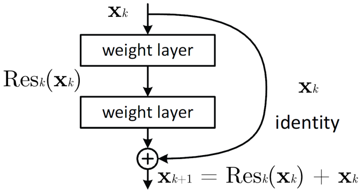


Figure: ResNet Block

ResNet: Transition to Neural ODE

If we treat the internal process $\text{Res}_k(\mathbf{x}_k)$ as giving the derivative of a unified "variable" $\mathbf{x}$ at "time" k, then the equation effectively represents the Euler method for solving the ODE (with unit time step)

$$\left.\frac{d\mathbf{x}}{dt}\right|_{t=k} = \text{Res}(\mathbf{x}_k, k) \tag{2}$$

where $\text{Res}(\mathbf{x}_k, k) = \text{Res}_k(\mathbf{x}_k)$ represents Res_k at all ks.

ResNet and Neural ODE

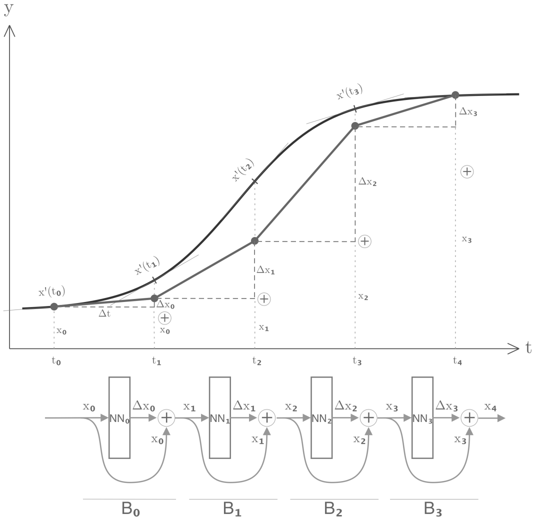


Table of Contents

Neural ODE

Neural ODEs (NODE) are a continuous re-imagination of ResNet. They model curves in arbitrary dimensions as solutions to first order ODEs, and are thus inherently applicable to time series.

NODE

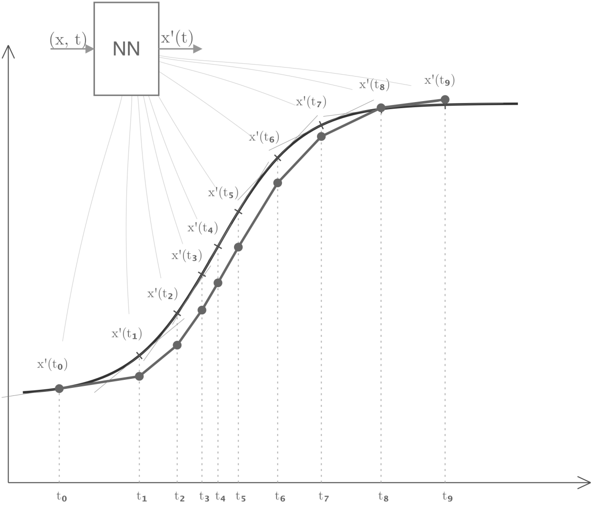
Note that each residual group is only responsible for a single time point. Neural ODE instead uses a single NN to produce all derivatives by taking the time point as explicit input:

$$\frac{d\mathbf{x}}{dt} = \mathrm{NN}_{\mathrm{NODE}}(\mathbf{x}, t). \quad (3)$$

It follows that the solution to this ODE is the integral over time:

$$\mathbf{x}(t) = \int \mathrm{NN}_{\mathrm{NODE}}(\mathbf{x}, t)dt \quad (4)$$

NODE



(x, t)
NN
x'(t)
x'(t9)
x'(t8)
x'(t7)
x'(t6)
x'(t5)
x'(t4)
x'(t3)
x'(t2)
x'(t1)
x'(t0)
t0
t1
t2
t3
t4
t5
t6
t7
t8
t9

Advantages of NODE

- Flexible in integration methods: reflect in adaptive strategies in choosing integration points
- Flexibility of choosing algorithms and of balancing between accuracy and speed: number of steps is variable and controllable, which effectively establishes flexible depth

Table of Contents

ECG Example

We are going to replicate the comparison proposed by the authors to verify and compare the accuracy of ResNet and NODE approaches in the context of a supervised learning task, namely the classification of an ECG signal.

ECG Dataset Intro

- MIT Beth Israel Hospital (BIH) electrocardiogram (ECG) dataset
- The data is obtained from Kaggle [1]
- The data contains about 110,000 labeled data points about heartbeat classification

ECG Dataset Intro

Each sample is annotated into the following five categories:

- normal (0)
- supraventricular premature beat (1)
- premature ventricular contraction (2)
- fusion of ventricular and normal beat (3)
- unclassifiable beat (4)

Sample for Each Heartbeat Category

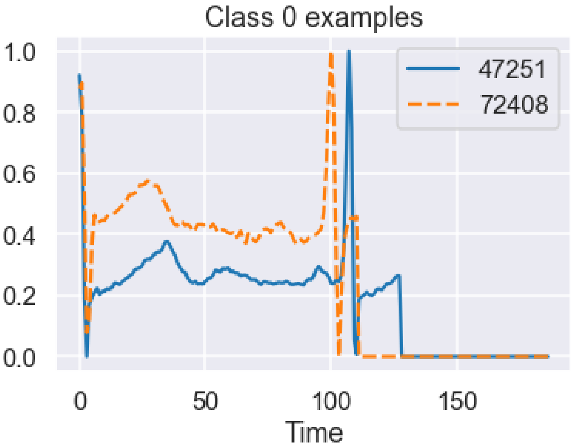


Figure: Heartbeat Example of Category 0

Sample for Each Heartbeat Category

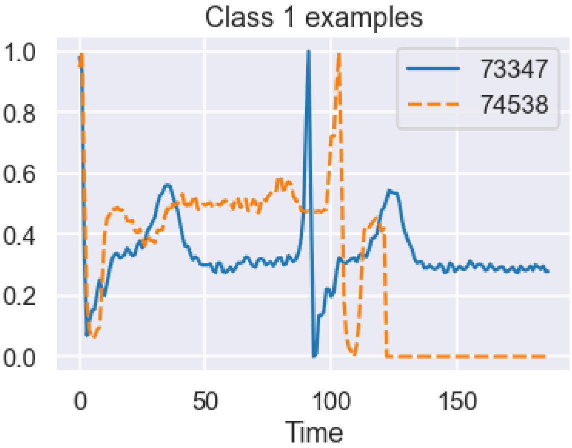


Figure: Heartbeat Example of Category 1

Sample for Each Heartbeat Category

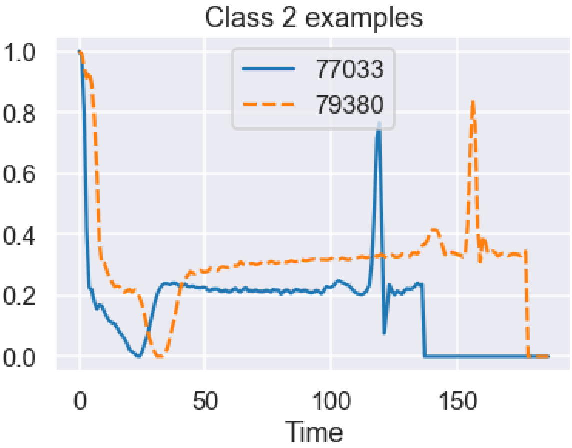


Figure: Heartbeat Example of Category 2

Sample for Each Heartbeat Category

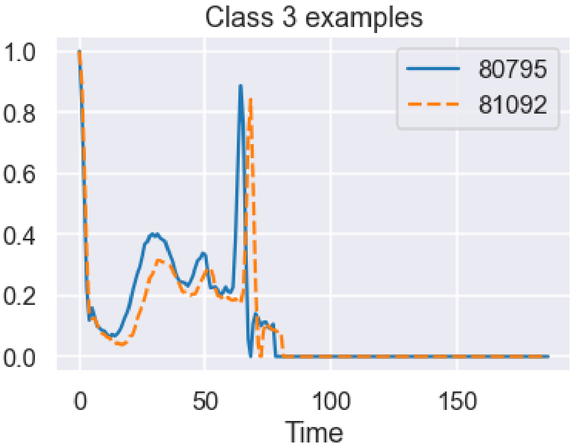


Figure: Heartbeat Example of Category 3

Sample for Each Heartbeat Category

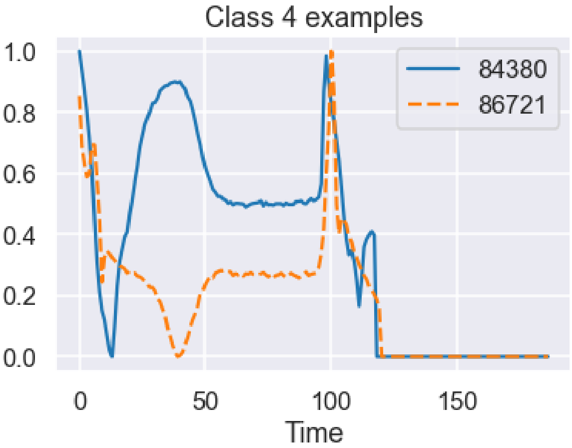


Figure: Heartbeat Example of Category 4

Network structure (all the layers)

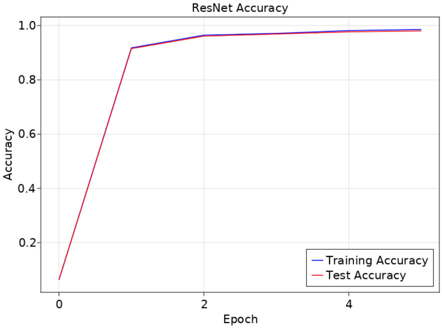


Figure: ResNet Accuracy

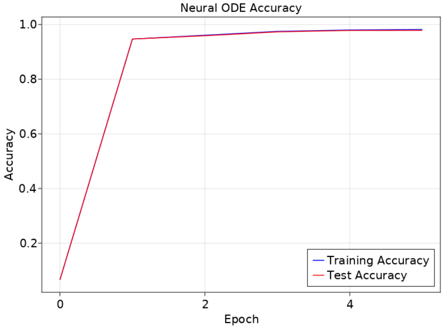


Figure: NODE Accuracy

ResNet vs. NODE

We are able to fully replicate the results of the original authors. Training each network for 5 epochs, the ResNet produced a final accuracy of 0.985 on the training dataset and 0.980 on the test set, while the Neural ODE produced 0.982 and 0.979 for training and test respectively.

Advantages of NODE

- ResNet contains 182853 parameters v.s. the NODE has only 59333
- The ResNet notably uses 6 residual blocks each containing 24832 parameters while the NODE uses one Neural ODE block with 25472 parameters.

Disadvantages of NODE

- Training and evaluation time: For evaluation, the ResNet takes around 9 seconds on our hardware with the test set as input while the NODE takes around 70 seconds; for training with the selected optimizer (stochastic gradient descent with momentum optimization), the ResNet takes around 1.6 to 1.8 minutes per epoch while the NODE spends around 11.7 to 16.6 minutes per.

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!