ResNets vs. Neural ODEs

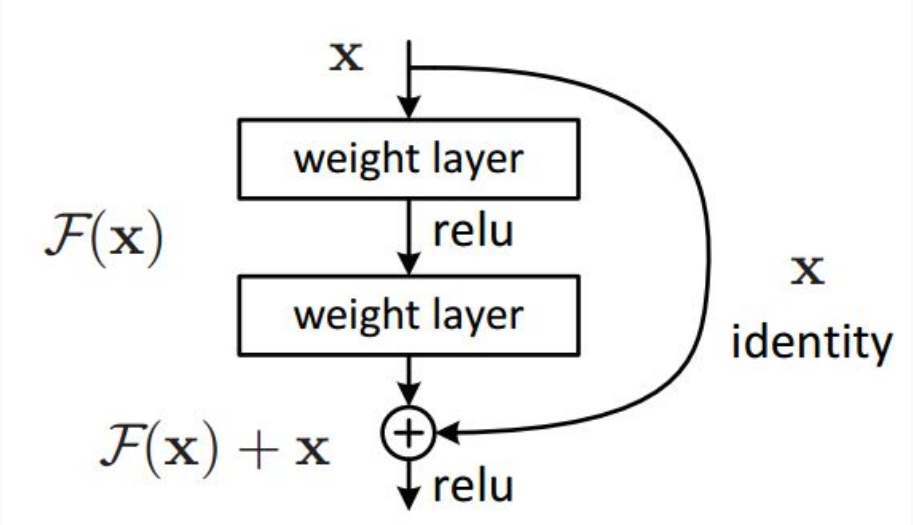
March 17th, 2020

Overview

- What are…
  - ResNets?
  - Neural ODEs?
- ECG Classification
- Model Comparison

ResNet

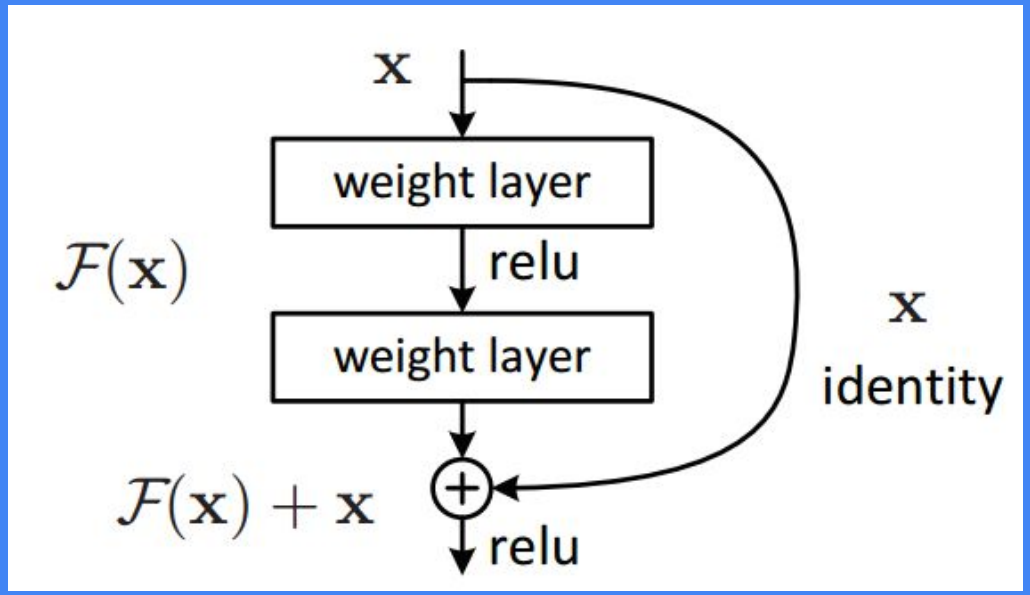
- What is it?
  - Residual neural network
  - Consists of residual blocks
- Why is it useful?
  - Deep neural networks were performing worse not better
  - Shortcut connections help!
  - ResNets have state-of-the-art accuracy on image classification tasks



Bridge to Neural ODEs

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

Residual Block

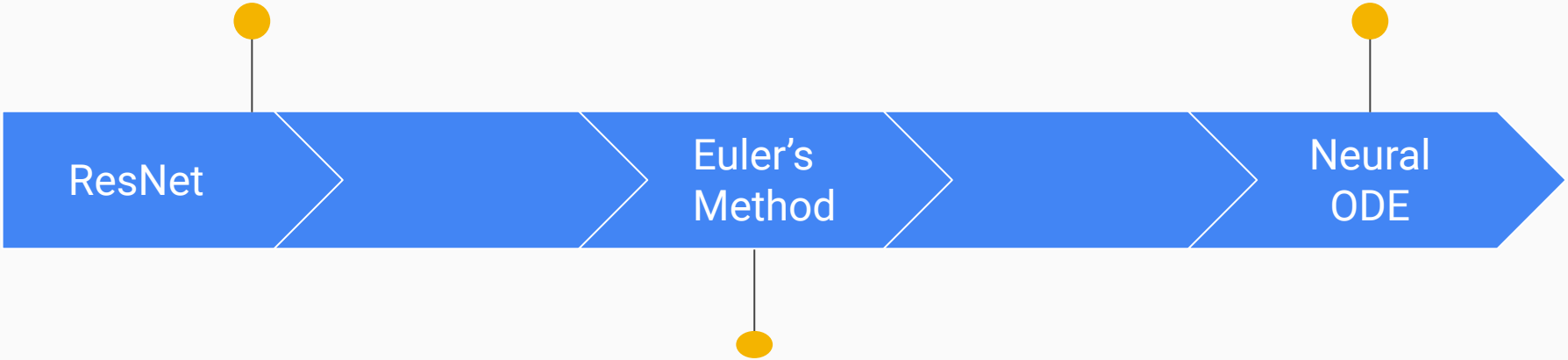


ResNet Equation

- Looks like Euler's method!

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

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



$$y_{n+1} = y_n + hf(t_n, y_n).$$

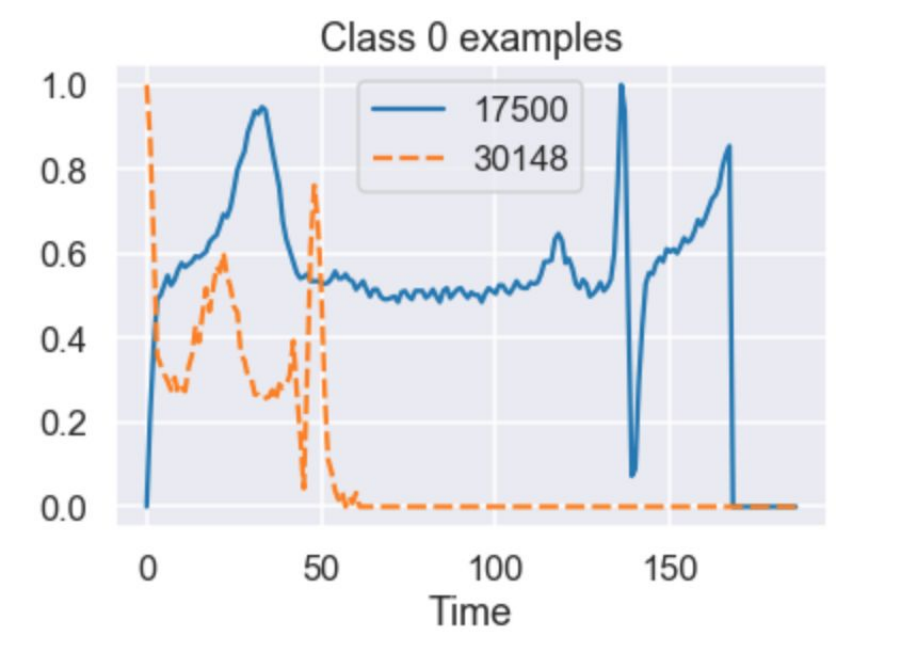
$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0.$$

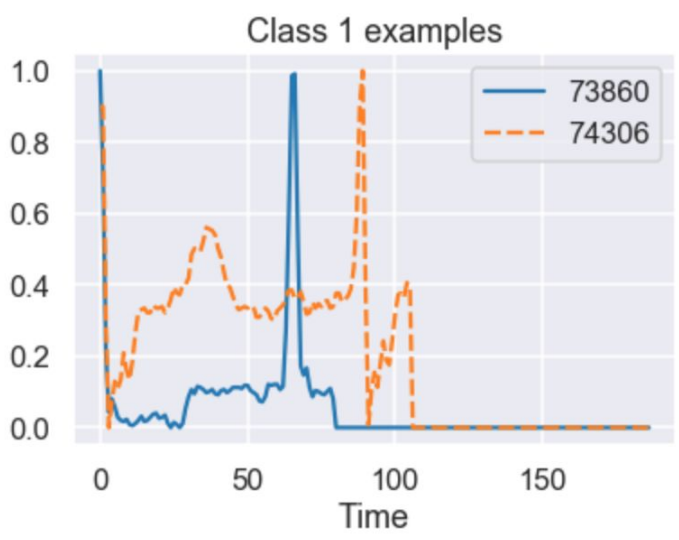
Neural ODEs

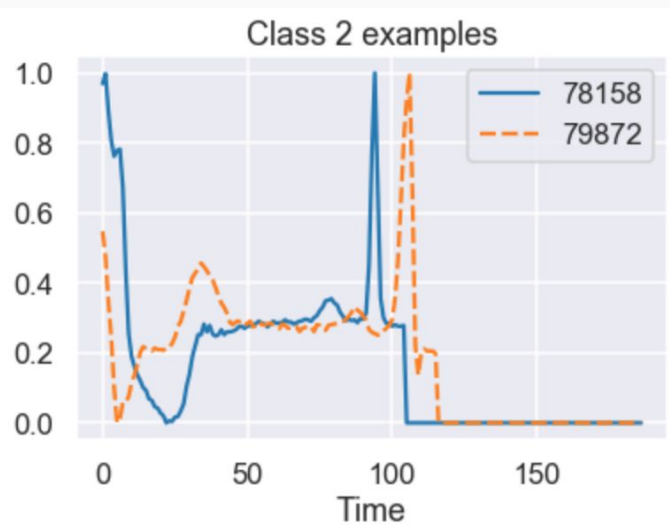
- What?
  - Neural ordinary differential equations
- How?
  - Use ODE solver to get predictions from model input
  - Backpropagate with adjoint method
- Why?
  - Rely on 300 years of ODE research
  - Memory/accuracy
  - Can adjust errors
  - Continuous dynamics

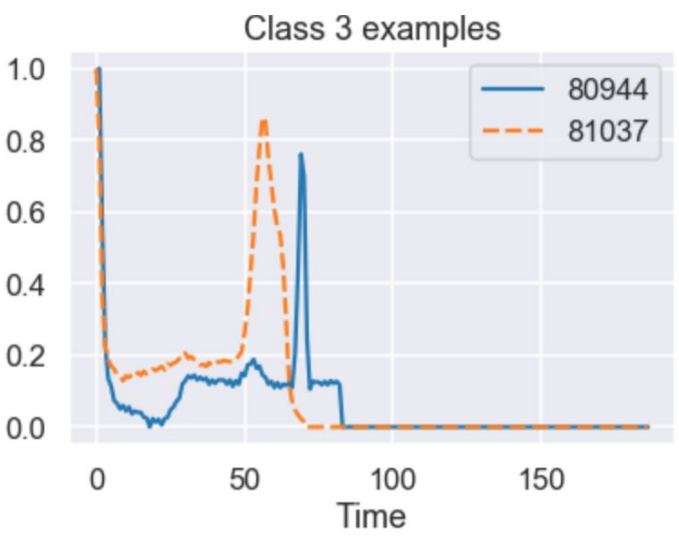
ECG Classification

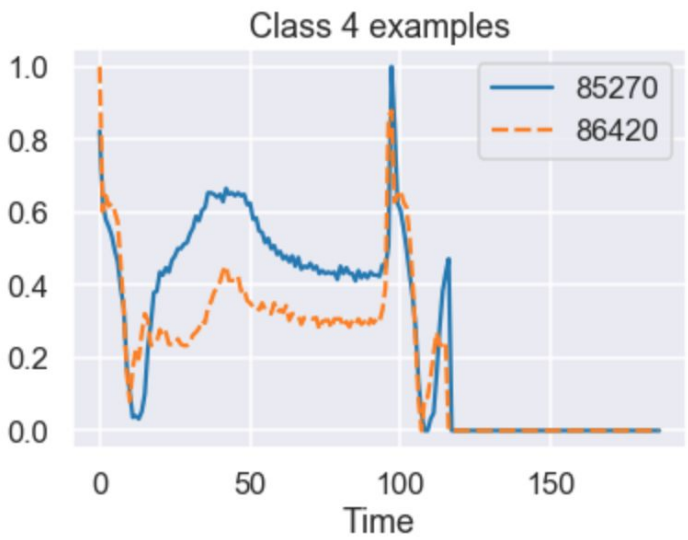
- MIT-BIH ECG dataset
  - 110,000 annotated samples
- 5 classes
  - 0: Normal beat
  - 1: Supraventricular premature beat
  - 2: Premature ventricular contraction
  - 3: Fusion of ventricular and normal beat
  - 4: Unclassified beat




Class 1 examples
73860
74306
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Time


Class 2 examples
78158
79872
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Time


Class 3 examples
80944
81037
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Time


Class 4 examples
85270
86420
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
Time

Model Building (feature layers)

- ResNet:
  - Six residual blocks stacked
  - Each residual block consisted of two convolutions, normalizations and ReLU activations
- NeuralODE:
  - Same structure as a single residual block
  - Called odeint_adjoint function from torchdiffeq
    - Forward: dopri5 solver
    - Backward: adjoint method

Model Comparison

ResNet

```
Training... epoch 1
    Percent trained: 100.0%  Time elapsed: 11.0 min
    val loss: 0.36

Training... epoch 2
    Percent trained: 100.0%  Time elapsed: 10.8 min
    val loss: 0.79

Training... epoch 3
    Percent trained: 100.0%  Time elapsed: 10.4 min
    val loss: 0.23

Training... epoch 4
    Percent trained: 100.0%  Time elapsed: 10.1 min
    val loss: 0.13

Training... epoch 5
    Percent trained: 100.0%  Time elapsed: 10.1 min
    val loss: 0.1
```

NeuralODE

```
Training... epoch 1
    Percent trained: 100.0%  Time elapsed: 61.9 min
    val loss: 0.23

Training... epoch 2
    Percent trained: 100.0%  Time elapsed: 71.2 min
    val loss: 0.14

Training... epoch 3
    Percent trained: 100.0%  Time elapsed: 73.6 min
    val loss: 0.12

Training... epoch 4
    Percent trained: 100.0%  Time elapsed: 80.5 min
    val loss: 0.09

Training... epoch 5
    Percent trained: 100.0%  Time elapsed: 97.4 min
    val loss: 0.09
```

ResNet vs. NeuralODE

```
ResNet accuracy: 0.974
ODENet accuracy: 0.976
```

Accuracy

- Models perform comparably (baseline 0.83)
- NeuralODE slightly better
  - Can be tuned further with ODEsolver errors

```
Number of tunable parameters in...
    ResNet: 182853
    ODENet: 59333
```

Memory

- NeuralODE has ⅓ of the parameters
- Comes at cost of longer training times

Conclusions

- NeuralODEs offer interesting take on neural networks with a lot of active research
  - Augmented NeuralODEs
  - Stochastic NeuralODEs
- Tradeoff between memory and speed
- Next steps:
  - Investigate applications, such as continuous normalizing flows
  - Make generative latent time series model