Neural Ordinary Differential Equations

Reem Elmahdi

28th June 2020

Manobi Africa

Contents

Background

What is Neural Ordinary Differential Equations?

- Neural ODE or ODENet is a type of the machine learning models that uses ODE to find the approximation function.
- No need to specify the number of layers.
- Slower in training but faster in testing.
- Neural ODE uses of the new mathematical approach.

Neural ODEs Solve Problems Continuously

Figure 1 shows a comparison of RNN and Neural ODE solutions.

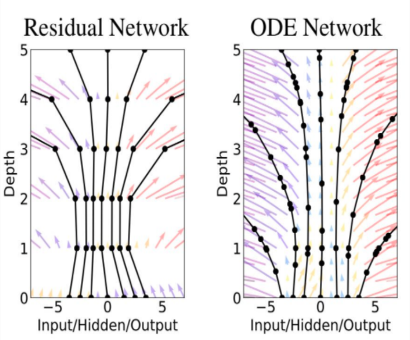


Figure 1: Discrete sequence of steps in ResNets (left) and an ODE network as a continuous transformation of a states.

Artificial Neural Networks

Artificial Neural Networks Architecture

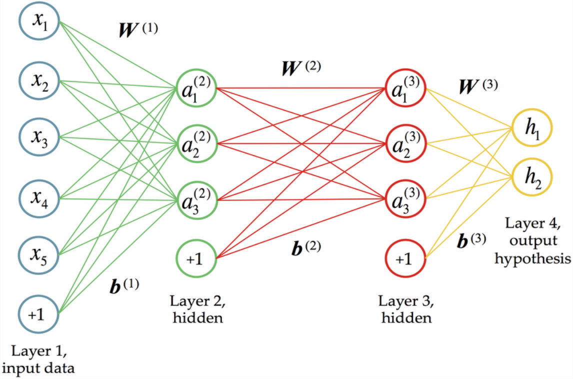


Figure 2: An Artificial Neural Network architecture.

$a^{(2)} = f(x * W^{(1)} + b(1))$
$a^{(3)} = f(a^{(2)} * W^{(2)} + b(2))$
$h = f(a^{(3)} * W^{(3)} + b(3))$

Where:

- x: x_1, x_2, x_3, x_4, x_5,
- $a^{(2)}$: $a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$,
- $a^{(3)}$: $a_1^{(3)}, a_2^{(3)}, a_3^{32)}$,
- h: h_1, h_2.

Can we solve any problem by stacking more layers?

Keep in mind!

- The output of ANN is influenced by the number of neurons and the layers.
- Underfitting and Overfitting.
- Accuracy and complexity trade off.

Residual Neural Networks

- ResNet is one of the machine learning models that use skip connections between layers.
- ResNet was introduced in 2015 and increases the accuracy.
- Skip connections help in reducing the number of layers.
- $x_{t+1} = x_t + f(x_t)$

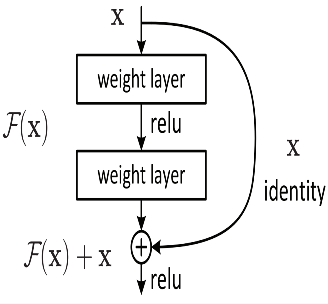


Figure 3: A residual block.

ANNs vs ResNets

$$
\begin{aligned}
a^{(2)} &= f(x * W^{(1)} + b^{(1)}) \\
a^{(3)} &= f(a^{(2)} * W^{(2)} + b^{(2)}) \\
h &= f(a^{(3)} * W^{(3)} + b^{(3)})
\end{aligned}
$$

$$
\begin{aligned}
a^{(2)} &= f(x * W^{(1)} + b^{(1)}) + x \\
a^{(3)} &= f(a^{(2)} * W^{(2)} + b^{(2)}) + a^{(2)} \\
h &= f(a^{(3)} * W^{(3)} + b^{(3)}) + a^{(3)}
\end{aligned}
$$

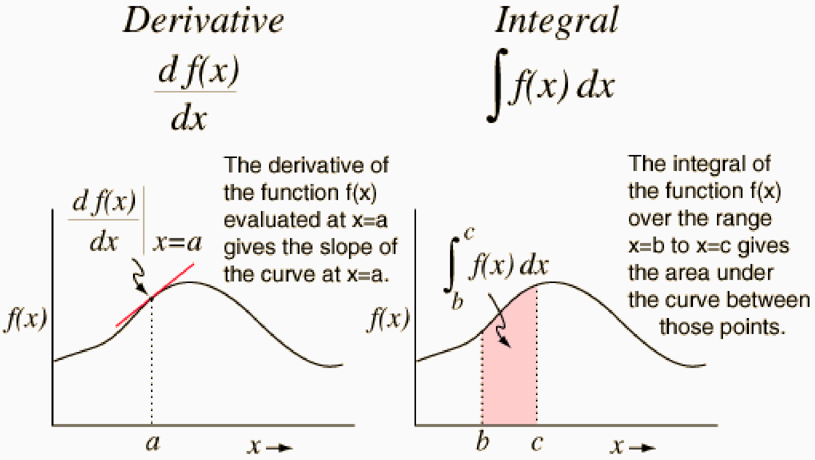
How does ResNets motivate Neural ODE?

It turns out that:

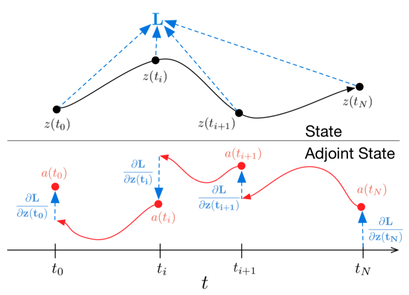
- $x_{t+1} = x_t + Wf(x_t)$ is Eulerś method formula for solving ODEs when $W = 1$.

Ordinary Differential Equations

Derivation vs Integration



- Vector-valued x changes in time.
- Time-derivative: $\frac{dx}{dt} = f(x(t), t)$.
- Initial-value problem: given $x(t_0)$, find $x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t), t, W)dt$.
- Euler approximation: $x(t+1) = x(t) + Wf(x, t)$.



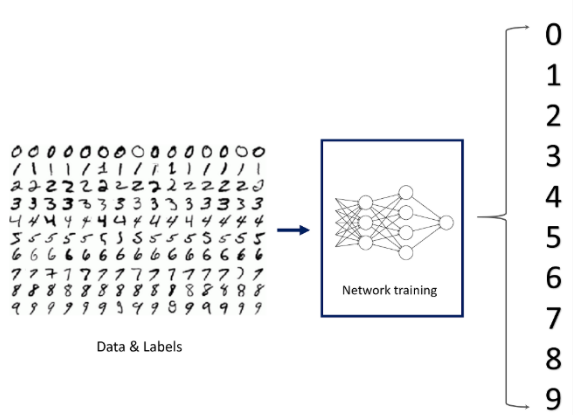
Don’t see the trick yet?

```
def f(z, t, θ):
  return nnet([z, t], θ)

def resnet(z, θ):
  for t in [1:T]:
    z = z + f(z, t, θ)
  return z
```

```
def f(z, t, θ):
  return nnet([z, t], θ)

def ODEnet(z, θ):
  return ODESolve(f, z, 0, 1, θ)
```

Do Neural ODE really work?

Performance on MNIST dataset



	Test Error	# Params	Memory	Time
1-Layer MLP†	1.60%	0.24 M	-	-
ResNet	0.41%	0.60 M	$\mathcal{O}(L)$	$\mathcal{O}(L)$
RK-Net	0.47%	0.22 M	$\mathcal{O}(\tilde{L})$	$\mathcal{O}(\tilde{L})$
ODE-Net	0.42%	0.22 M	$\mathcal{O}(\mathbf{1})$	$\mathcal{O}(\tilde{L})$

The End

Thank you!

Bibliography

- Baccianella, S., Esuli, A. & Sebastiani, F. (2010). SentiWordNet 3.0: Neural ordinary differential equations.. In *LREC*.
- Bond, F., Isahara, H., Fujita, S., Uchimoto, K., Kuribayashi, T. & Kanzaki, K. (2009). Enhancing the Japanese Wordnet. In *Neural Ordinary Differential Equations*. Association for Computational Linguistics.