ANLY535 Machine Learning II

Assignment 6 - New horizons of Deep Neural Networks

Team 1 : Chirag Agarwal, Kun Hu, Snigdha Jain and Smitha Shivakumar

Highlight

David Duvenaud, an AI researcher at the University of Toronto, and his collaborators recently redesigned the neural networks as we know them. In Dec 2018, their paper **‘*Neural Ordinary Differential Equations*’** *was among the four others crowned “best paper” at the Neural Information Processing Systems conference, one of the largest AI research gatherings in the world*. It is a radical new neural network design could overcome big challenges in AI - Researchers borrowed equations from calculus to redesign the core machinery of deep learning so it can model continuous processes like changes in health.

Problem

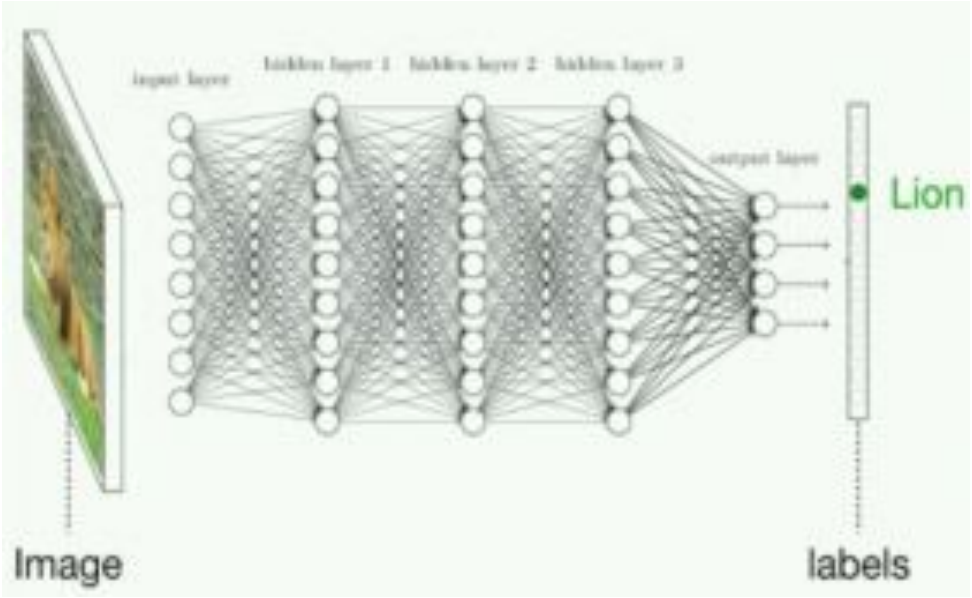
Traditional neural network is not fit for modeling continuous processes such as predicting a patient's health over time. Data from medical records is kind of messy: throughout your life, you might visit the doctor at different times for different reasons, generating a smattering of measurements at arbitrary intervals. The design for traditional neural network requires it to learn from data with clear stages of observation.

Proposed Solution: ODE net (Ordinary Differential Equations)

- No more nodes and connections, just one continuous slab of computation.
- Replace layers with calculus equations: Calculus gives you all the equations for how to calculate a series of changes across infinitesimal steps

If you want to model anything that transforms continuously over time, you also have to chunk it up into discrete steps. Taken to the extreme, this means the best neural network for this job would have an infinite number of layers to model infinitesimal step-changes.

How a traditional neural net transforms an image of a lion into the name “lion”



Performance comparison

	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	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(\tilde{L})$

Limitation of the layers method

The layer approach has served the AI field well—but it also has a drawback. If you want to model anything that transforms continuously over time, you also have to chunk it up into discrete steps. In practice, if we returned to the health example, that would mean grouping your medical records into finite periods like years or months. You could see how this would be inexact. If you went to the doctor on January 11 and again on November 16, the data from both visits would be grouped together under the same year.

Limitation of the layers method

So the best way to model reality as close as possible is to add more layers to increase the granularity. (Why not break your records up into days or even hours? You could have gone to the doctor twice in one day!) Taken to the extreme, this means the best neural network for this job would have an infinite number of layers to model infinitesimal step-changes. The question is whether this idea is even practical.

From Sequences of Transformations to Neural Differential Equations

Quoting directly from our reference :

Today, multiple neural network architectures such as RNNs or Residual Networks contain repeating blocks of layers that have the ability to retain sequential information as well as change it in every step through a learned function. Such networks may in general be described by the equation:

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

Hereby, h_t is the "hidden" information at timestep t, and $f(h_t, \theta_t)$ is the learned function of the current hidden information and parameters θ_t. The central question posed in the paper is whether we can improve on our current state-of-the-art results with these networks by gradually reducing the stepsize *[t, t+1]*.

From Sequences of Transformations to Neural Differential Equations

Quoting directly from our reference :

We can imagine this as gradually increasing the number of evaluations in a RNN or increasing the number of residual layers in a Residual network. If we do this we will eventually arrive at an infinitesimal (differential) version of the previous equation:

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

Such an equation is called an **Ordinary Differential Equation** (ODE), since the solution is a function, namely the function **h(t)**. In other words, by solving the equation, we arrive at the desired sequence of hidden states. We will have to solve the equation during each evaluation, beginning with an initial state h_0. Such a problem is also called the **initial value problem.**

ODE Differentiating features

In addition to being able to model continuous change, an ODE net also changes certain aspects of training. With a traditional neural net, you have to specify the number of layers you want in your net at the start of training, then wait until the training is done to find out how accurate the model is. The new method allows you to specify your desired accuracy first, and it will find the most efficient way to train itself within that margin of error.

Advantages of defining and evaluating models using ODE

- **Memory efficiency:** Gradients of a scalar-valued loss can be computed with respect to all inputs of any ODE solver, without backpropagating through the operations of the solver. Not storing any intermediate quantities of the forward pass allows us to train our models with constant memory cost as a function of depth, a major bottleneck of training deep models.
- **Adaptive computation:** Euler's method is perhaps the simplest method for solving ODEs. Modern ODE solvers provide guarantees about the growth of approximation error, monitor the level of error, and adapt their evaluation strategy on the fly to achieve the requested level of accuracy. This allows the cost of evaluating a model to scale with problem complexity. After training, accuracy can be reduced for real-time or low-power applications

Advantages of defining and evaluating models using ODE

- **Parameter efficiency:** When the hidden unit dynamics are parameterized as a continuous function of time, the parameters of nearby “layers” are automatically tied together. This reduces the number of parameters required on a supervised learning task
- **Scalable and invertible normalizing flows:** An unexpected side-benefit of continuous transformations is that the change of variables formula becomes easier to compute. In Section 4, we derive this result and use it to construct a new class of invertible density models that avoids the single-unit bottleneck of normalizing flows, and can be trained directly by maximum likelihood.

Advantages of defining and evaluating models using ODE

- **Continuous time-series models:** Unlike recurrent neural networks, which require discretizing observation and emission intervals, continuously-defined dynamics can naturally incorporate data which arrives at arbitrary times. Such a model has been constructed and demonstrated in the research paper.

Scope and Limitations of ODE

- **Minibatching:** The use of mini-batches is less straightforward than for standard neural networks.
- **Uniqueness :** When do continuous dynamics have a unique solution? Picard's existence theorem (Coddington and Levinson, 1955) states that the solution to an initial value problem exists and is unique if the differential equation is uniformly Lipschitz continuous in z and continuous in t. This theorem holds for our model if the neural network has finite weights and uses Lipshitz nonlinearities, such as tanh or relu

An Interesting application

- **MNIST written digit classification(Supervised learning):** ODESolve method can achieve comparable performance to a residual network with much less parameters. The network used for evaluation in the paper downsamples the input image twice and then applies 6 residual blocks. The important result is that with roughly 1/3rd of the parameters the RK-Network and the ODE-Net achieve roughly the same performance as the residual network.

Furthermore, the accuracy of the ODE solution can be tuned to maximize computational performance. For instance, one could perform training with high accuracy and lower the accuracy during evaluation.

Scope and Limitations of ODE contd.

- **Setting tolerances:** Allows the user to trade off speed for precision, but requires the user to choose an error tolerance on both the forward and reverse passes during training.
- **Reconstructing forward trajectories:** Reconstructing the state trajectory by running the dynamics backwards can introduce extra numerical error if the reconstructed trajectory diverges from the original. This problem can be addressed by checkpointing: storing intermediate values of z on the 8 forward pass, and reconstructing the exact forward trajectory by re-integrating from those points. The researchers did not find this to be a practical problem, and we informally checked that reversing many layers of continuous normalizing flows with default tolerances recovered the initial states.

References

1. https://www.technologyreview.com/s/612561/a-radical-new-neural-network-design-could-overcome-big-challenges-in-ai/
2. Duvenaud D., Bettencourt J., Chen Ricky .T.Q. & Rubanova Y. (2018). Neural Ordinary Differential Equations. Retrieved from https://arxiv.org/pdf/1806.07366.pdf
3. https://towardsdatascience.com/paper-summary-neural-ordinary-differential-equations-37c4e52df128