EECS 6322 Week 5, Paper 1
Deep Equilibrium Models

Shivani Sheth (Student #: 218011783)
shivs29@yorku.ca

I. Summary and contributions

The paper proposes a new approach to model sequential data, known as the Deep Equilibrium model (DEQ). The model aims to directly calculate the equilibrium point that the deep-layered models reach via a sequential series of layers. Thus, the model is equivalent to an infinite depth weight-tied feedforward neural network with the advantage that it can analytically backpropagate through the equilibrium point using implicit differentiation. With the given architecture, the model requires constant memory which is equal to a single layer's activations. It thus reduces the memory requirements drastically, while performing either at par or slightly better than the current state of the art models in handling sequential data.

In the forward pass of most feedforward deep learning models, the models calculate and store the transformations of L layers, typically required for computations by backpropagation. Thus, as the depth of the models increase, the memory requirements of the feed forward network increases. The number of layers L is usually chosen by the model designers as a hyperparameter. The iteration of the feedforward sequential model can be represented by:

$$z_{1:T}^{[i+1]} = f_\theta^{[i]}(z_{1:T}^{[i]}\ ;\ x_{1:T}) \quad \textit{for } i = 0, 1, 2, \cdots, L-1$$

where i the number of layer, $x_{1:T}$ is the input sequence , $z_{1:T}^{[i]}$ is the hidden sequence of length T at layer i, and $f_\theta^{[i]}$ is a non-linear transformation that enforces causality, that is, a current point cannot depend on a future point.

The proposed architecture employs weight tying across the network, that is, it applies the same transformation across all layers of the network. Weight tying gives a number of benefits to the model namely, generalization, regularization for stabilizing training, reducing model size, and enabling the model to represent any deep learning network. It directly computes the solution to a nonlinear system given by:

$$z_{1:T}^{*} = f_\theta(z_{1:T}^{*}\ ;\ x_{1:T}).$$

This model solves the equilibrium via any black box root method and can directly differentiate through the fixed point equilibrium using implicit differentiation. This does not require storing the intermediate values of the activation functions and hence requires a constant memory.

In the forward pass, the DEQ model calculates the equilibrium point, that could involve any procedure leading to this equilibrium point. It can use any black-box root finding algorithm, such as the quasi-Newton algorithm, to find the equilibrium point in the forward pass given by:

$$z_{1:T}^{*} = RootFind(g_\theta\ ;\ x_{1:T})$$

where the initial estimate $z_{1:T}^{[0]}$ is set to a very small value such as 0. In the backward pass, the proposed model utilizes the gradient of the equilibrium model and by approximating the inverse of the Jacobian matrix, it accelerates the computation of the backpropagation without depending on any knowledge of the black-box 'RootFind' method.

The DEQs have been tested on various large scale applications by instantiating them on two very different architectures, namely TrellisNet and Self-attention architectures, representing different families of the deep sequential networks. Both instantiations use Broyden's method to approximate the calculation of the inverse Jacobian matrix, and introduce a new hyperparameter to stop the Broyden iterations at a given point.

In the copy memory task, the DEQ model was tested against LSTM and GRU to memorise elements across a long period of time. The results proved the DEQ-transformer had a much lower error than its given counterparts, thus giving a better performance. The architecture was also tested on large-scale datasets that measured its performance on the Penn Treebank and WikiText-103 data sets. On the Penn Treebank data set, the DEQ-TrellisNet architecture was tested against the deeply supervised TrellisNet, where both the models had a similar performance but the DEQ model greatly reduced the memory by about 7GB. On the WikiText-103 dataset, the DEQ-TrellisNet architecture achieves a lower test perplexity, and hence a better performance while reducing memory constraints by $80\% - 85\%$ in most cases, which provides a huge advantage to the DEQ model in terms of resource efficiency. The DEQs can also almost always converge to a sequence-level fixed point.

Thus, the DEQ model is memory efficient, providing a constant memory requirement for the architecture. It is also independent of the choice of the non-linear transformations applied on each layer of the network, given that the transformations are stable and constrained. Another property of DEQ includes no change in representational power when several DEQs are stacked together. Through the experiments given in the paper, we can also conclude that the models have a good temporal memory retention, while performing competitively or better with large scale datasets as compared to the state of the art results.

II. Strengths

The model analytically backpropagates through the equilibrium point using implicit differentiation which does not require storing any intermediate values and thus the model takes up the memory equal to a single layer's activations. The network requires only constant memory for training and prediction, regardless of the "depth" of the network. Hence, it greatly reduces the memory consumption of the model which is usually the bottleneck feature for most large models. The model also does not depend on the choice of the RootFind method used in the forward pass and hence can support many different types of models.

III. Weaknesses

The DEQ models are more time consuming as compared to layer-based deep networks. This is because the runtime of the models are based on the number of Broyden steps that increase with the number of training epochs, hence taking longer time to compute the equilibrium point of the system. Another possible weakness of the method is that if the system has no fixed equilibrium point, then the method might not work.

IV. Correctness

The claims and empirical methodology of the paper are correct under the given constraints and are backed by theoretical proofs and elaborated derivation on the forward and backward pass of the system. With the experiments tested in the paper, the model has also proved to be compatible with different types of architectures such as TrellisNet and Self-attention systems which perform competitively or better with their state of the art counterparts.

V. Clarity

The paper consists of adequate theoretical proofs, graphical illustrations, and comparative tables that support the explanation given by the authors, hence making it easy to understand. The proofs are also well formatted in the paper, where they are briefly explained along with the concepts, and are later elaborated in the appendix. Overall, the paper is well written.

VI. Relation to prior work

Previous work include Deep Sequence Models that take an input sequence $x_{1:T} = [x_1, ..., x_T]$ and produces an output $y_{1:T}$ that satisfies the causality constraint. Three major families in these deep sequence models include Recurrent Neural Networks (RNNs) and its variants such as Long Short Term Memory (LSTM), Temporal Convolutions, and the Self-attention transformer architecture. Other types of models designed for memory efficiency include methods such as gradient checkpointing that reduce the memory requirements to $O(\sqrt{L})$ at the cost of extra computation taken by extra forward passes. Some prior works have also developed an equilibrium propagation framework that predicts fixed-point energy dynamics at its local minimum.

The Deep Equilibrium models (DEQs) differ from the previous models as they reduce the memory constraints to a constant memory requirement, unlike any other model, and prevent the storage of layer values by directly differentiating through the equilibrium point. Unlike the other models, it also does not predict the inter-layer transformations of the network, and can be used on very different types of sequence learning architectures. Unlike the ODE-based methods, the DEQ model also directly calculates the equilibrium point using a quasi-Newton method which does not depend on the sequence of layers that lead to the fixed point. Finally, the DEQs were also able to perform significantly well on higher dimensional data while reaping the benefits of lower memory constraints.

VII. Reproducibility

The work proposed by the authors can be implemented in Deep Learning libraries such as PyTorch. The paper also consists of links for the implementation of the model which makes it easy to reproduce the major details of the proposed work. It also has adequate theoretical grounding which makes it easy to implement in the existing architectures.

VIII. Additional comments

While comparing the results of the experiments, the evaluation metrics and their optimal values for each metric would supplement the results of the given experiment. For example, for a given experiment, the authors could mention the model was evaluated on metrics 'a, b, c', where lower values for 'a', higher values for 'b', and lower values for 'c' gives an optimal performance, which could be followed by different comparative tables.