Carnegie Mellon University

Deep Equilibrium Models

Shaojie Bai[1], J. Zico Kolter[1,2] and Vladlen Koltun[3]

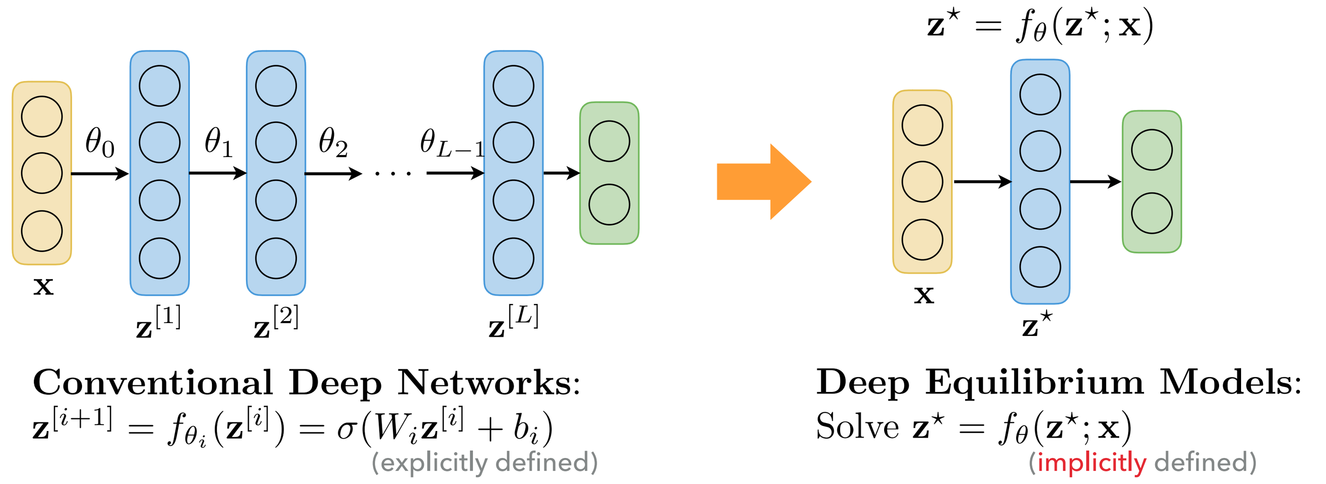
Carnegie Mellon University[1], Bosch Center for AI[2], Intel Labs[3]

MACHINE LEARNING DEPARTMENT

TL;DR: "One layer" is all you need! An implicit-depth (or infinite-depth) model that solves for equilibriums and achieves SOTA, with only constant memory.

Introduction

- Deep networks have long been built on a core concept: *layers*. Network depth are usually hand-picked by model designers: e.g., ResNet-18/101.
- We propose the Deep Equilibrium (DEQ) Models that reduce many classes of deep models with a single layer (defined implicitly) without loss of representational capacity:



- DEQ aims to directly compute the hidden features of a weight-tied network at its "infinite limit", but with memory cost of just one layer (i.e., constant).
- Empirically, we show DEQ instantiated on sequence models reduces SOTA deep networks by almost **90%** of its GPU memory cost, while achieving better results.
- Output(s) of the network are defined **implicitly**.

Explicit Layers	Implicit Layers
$\mathbf{z}^{[i+1]} = f(\mathbf{z}^{[i]})$	find $\mathbf{z}^{[i+1]}$ such that $f(\mathbf{z}^{[i]}, \mathbf{z}^{[i+1]}) = 0$

Q: Is weight-tying a big restriction?
A: Not at all (as we will show later). Empirical evidence: TrellisNet, Universal Transformer, ALBERT, ...

- Code at: `https://github.com/locuslab/deq`

Related Work

- Prior works have used gradient-checkpointing [4] and reversible networks [6] to reduce memory cost.
- Neural-ODE [3]; more work on implicit deep learning can also trace back to some original works on the attractor networks and recurrent backpropagation (RBP) [1].

Deep Equilibrium Models

- To do the reduction from deep networks to a single-layer model, we broadly consider the class of weight-tied, input-injected deep models:

$$\mathbf{z}^{[i+1]} = f_\theta(\mathbf{z}^{[i]}; \mathbf{x}), \quad i = 0, \dots, L-1, \quad \mathbf{z}^{[0]} = \mathbf{0}$$

where f_θ could be complex. Illustration in Figure 1 (left).

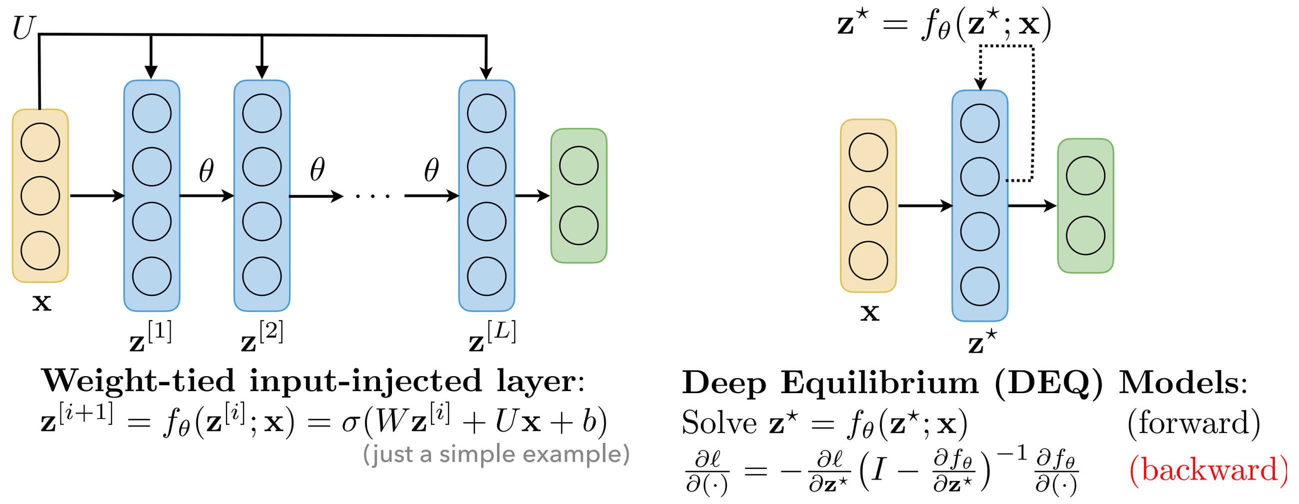


Figure 1: A **weight-tied, input-injected** network and a **deep equilibrium** "network".

- In practice, after these types of models converge to an equilibrium point:

$$f_\theta(\mathbf{z}^\star; \mathbf{x}) = \mathbf{z}^\star$$

Q: When will the equilibrium points exist?
A: They virtually always exist. Intuitively, the layers we use for **deep** networks already need this amount of stability.

Deep Equilibrium (DEQ) Models: directly solve for this equilibrium/stable point via black-box root-finding; directly differentiate *through* the equilibrium state.

To train/predict with a DEQ:

- Define a single layer f_θ, and $g_\theta(\mathbf{z}; \mathbf{x}) := f_\theta(\mathbf{z}; \mathbf{x}) - \mathbf{z}$.

Forward pass: Given input $\mathbf{x}_{1:T}$, compute the equilibrium point $\mathbf{z}^\star$, which is the root of g_θ:

$$\mathbf{z}^\star = \text{(black-box)}\mathrm{RootFind}(g_\theta; \mathbf{x})$$

Backward pass: Implicitly differentiate through this equilibrium state only using the output:

$$\frac{\partial \ell}{\partial (\cdot)} = -\frac{\partial \ell}{\partial \mathbf{z}^\star}\left(J_{g_\theta}^{-1}\Big|_{\mathbf{z}^\star}\right)\frac{\partial f_\theta(\mathbf{z}^\star; \mathbf{x})}{\partial (\cdot)}$$

regardless of the path/trajectory to this equilibrium.

Accelerating DEQ and Properties of DEQ Models

- No need to actually compute the inverse Jacobian $J_{g_\theta}^{-1}$.
- In the forward pass, we propose to use Broyden's method (quasi-Newton) that makes low-rank updates to estimate $J_{g_\theta}^{-1}$:

$$J_{g_\theta}^{-1}\Big|_{\mathbf{z}^{[i+1]}} \approx B_{g_\theta}^{[i+1]} = B_{g_\theta}^{[i]} + \mathbf{u}^{[i]}\mathbf{v}^{[i]^\top}$$

- In the backward pass, we alternatively solve the linear system

$$\left(J_{g_\theta}^\top\big|_{\mathbf{z}^\star}\right)\mathbf{x}^\top + \left(\frac{\partial \ell}{\partial \mathbf{z}^\star}\right)^\top = \mathbf{0}$$

where the vector-Jacobian product can be efficiently computed (time-wise and space-wise).

- Memory cost of DEQ: Forward pass only needs to store $\mathbf{z}^\star$ and $\mathbf{x}$ for the backward updates.
- Universality of DEQ model's capacity:
  Theorem 1[informal]: Any conventional deep network can be recovered by a DEQ.
  Theorem 2[informal]: Stacking multiple DEQs doesn't create extra representational power than a single DEQ.
  (Proof in paper.)

Key idea: One layer (of DEQ) is all you need.

Instantiations of DEQ on Sequences

- We specifically investigate the domain of **sequence modeling**, where $\mathbf{z} = \mathbf{z}_{1:T}$ and $\mathbf{x} = \mathbf{x}_{1:T}$ now have T timesteps, with f_θ autoregressive.
- The formulation of DEQ is not predicated on any particular kind of f_θ. We highlight its two (very different) instantiations using state-of-the-art sequence models (Figure 2).
- DEQ-TrellisNet [2]: (i.e., f_θ is temporal convolutions).
- DEQ-Transformer [5] (i.e., f_θ is multihead self-attention).

(More options possible...)

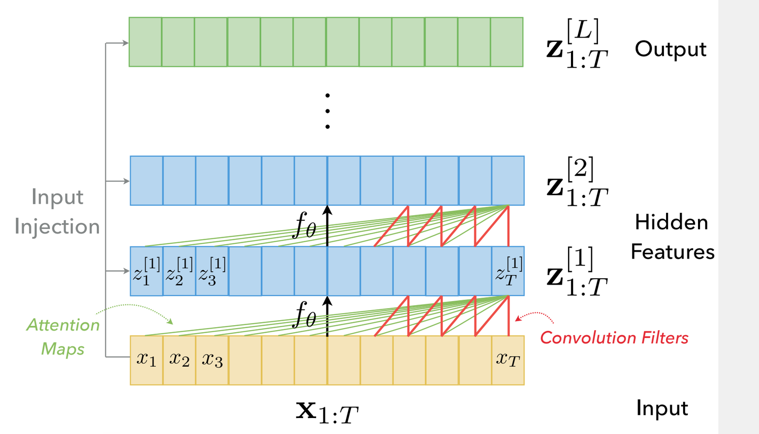


Figure 2: Two SOTA sequence models

Experiments

- We test both instantiations of DEQ on both synthetic and realistic, large-scale and high-dimensional sequence benchmarks.

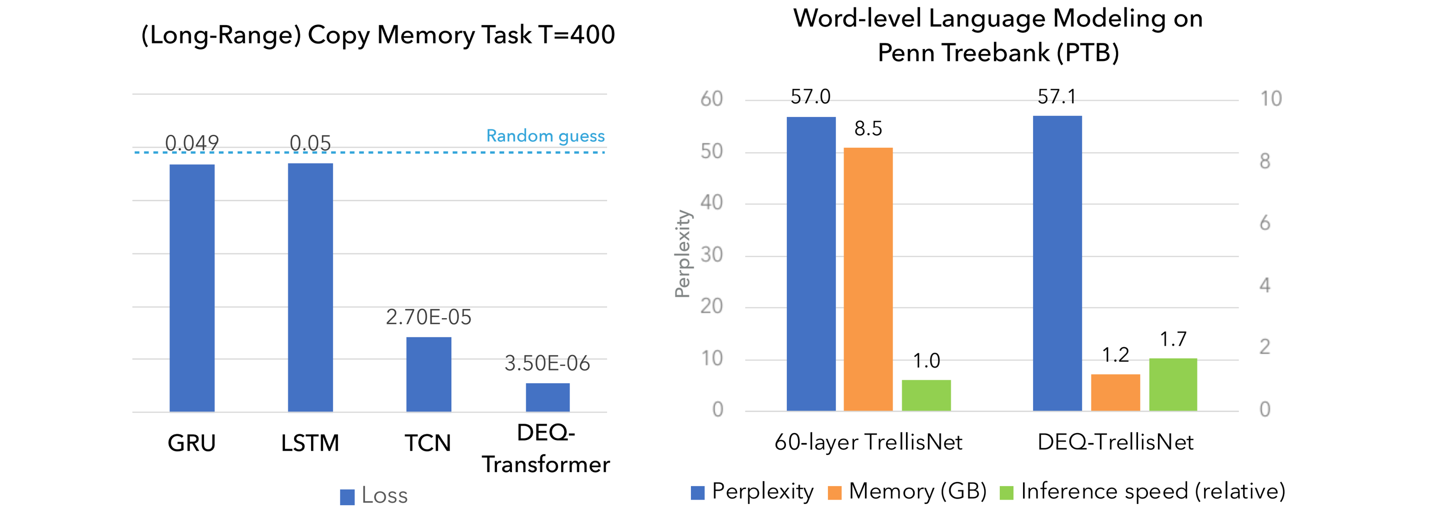


Figure 3: Left: DEQ demonstrates good memory retention over relatively long sequences. Right: DEQ achieves competitive performance on PTB corpus with > 80% reduction in memory cost.

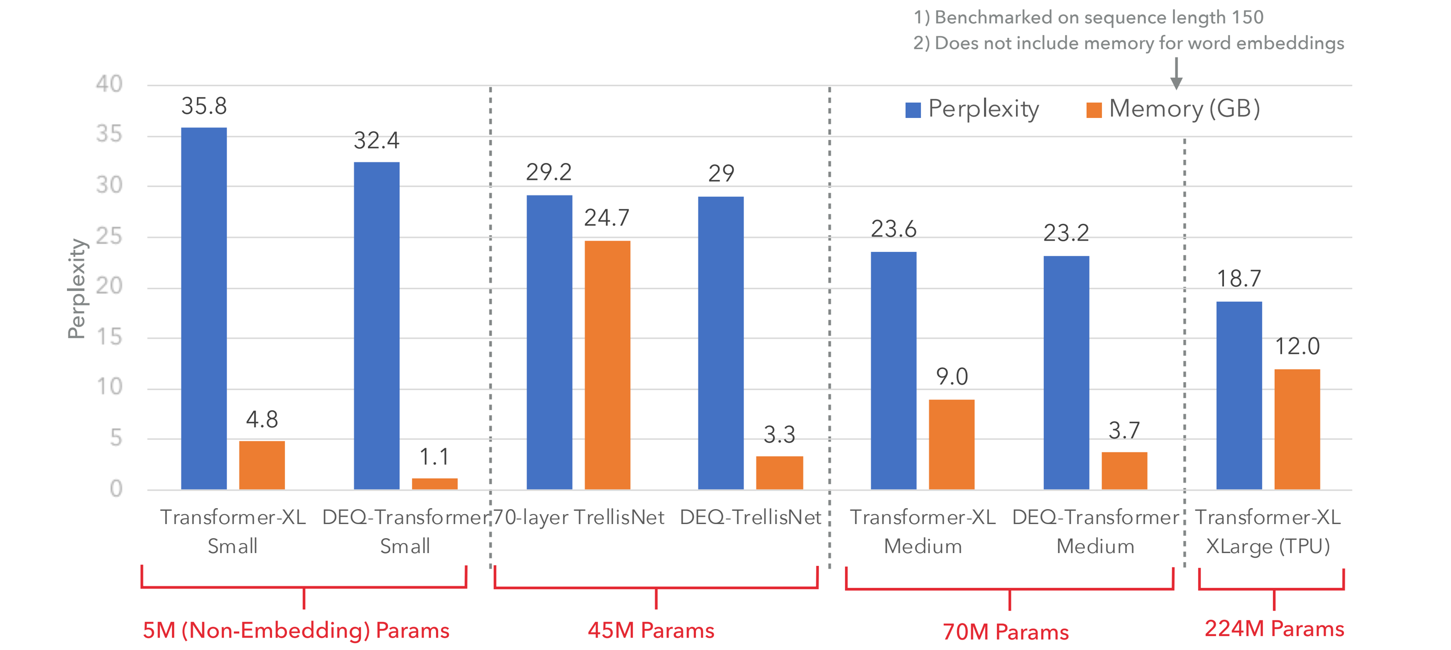


Figure 4: DEQ significantly reduces memory while improving the results on the large-scale WikiText-103 dataset.

- DEQ sequence models achieve results on par with (or better than) the SOTA, while paying 10-30% of the memory cost.
- DEQ runtime: about 1.5-2× slower at prediction.

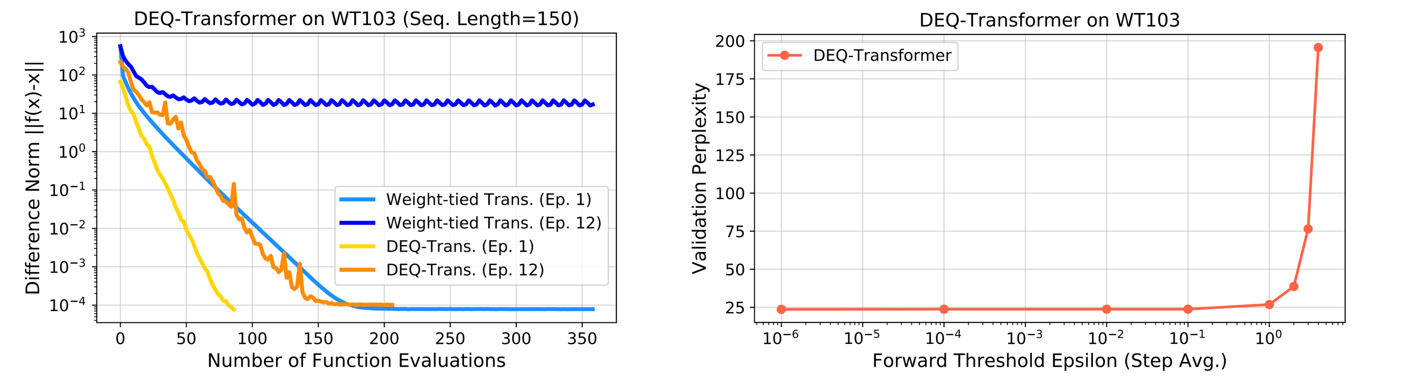


Figure 5: Left: DEQ-Transformer finds the equilibrium in a stable and efficient manner. Right: DEQ can be accelerated by leveraging higher tolerance ε.

- DEQ represent the largest-scale practical application of implicit layers in deep learning of which we are aware.

[1] Luis B Almeida. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In *Artificial Neural Networks*. 1990.
[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Trellis networks for sequence modeling. In *International Conference on Learning Representations (ICLR)*, 2019.
[3] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Neural Information Processing Systems*, 2018.
[4] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv:1604.06174*, 2016.
[5] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *International Conference on Learning Representations (ICLR)*, 2019.
[6] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Neural Information Processing Systems*, 2017.

NeurIPS 2019 - Vancouver, Canada