University of Stuttgart
Institute for Parallel and Distributed Systems
Analytic Computing

December 23, 2022

ML in the Science
Deep Equilibrium Models

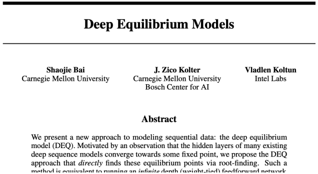
Daniel Frank

Motivation

1

Equilibrium Networks in Deep Learning

Where do they come from?



Deep Equilibrium Models

Shaojie Bai
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI

Vladlen Koltun
Intel Labs

Abstract

We present a new approach to modeling sequential data: the deep equilibrium model (DEQ). Motivated by an observation that the hidden layers of many existing deep sequence models converge towards some fixed point, we propose the DEQ approach that *directly* finds these equilibrium points via root-finding. Such a method is equivalent to running an *infinite* depth (weight-tied) feedforward network

Figure 1: From NeurIPS 2019, *about 400 citations* [Bai et al., 2019]

Equilibrium Networks in Deep Learning

Where do they come from?

Deep Equilibrium Models

Shaojie Bai
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI

Vladlen Koltun
Intel Labs

Abstract

We present a new approach to modeling sequential data: the deep equilibrium model (DEQ). Motivated by an observation that the hidden layers of many existing deep sequence models converge towards some fixed point, we propose the DEQ approach that *directly* finds these equilibrium points via root-finding. Such a method is equivalent to running an *infinite* depth (weight-tied) feedforward network

Figure 1: From NeurIPS 2019, *about 400 citations* [Bai et al., 2019]

Monotone operator equilibrium networks

Ezra Winston
School of Computer Science
Carnegie Mellon University
Pittsburgh, United States
`ewinston@cs.cmu.edu`

J. Zico Kolter
School of Computer Science
Carnegie Mellon University
& Bosch Center for AI
Pittsburgh, United States
`zkolter@cs.cmu.edu`

Abstract

Implicit-depth models such as Deep Equilibrium Networks have recently been shown to match or exceed the performance of traditional deep networks while being much more memory efficient. However, these models suffer from unstable convergence to a solution and lack guarantees that a solution exists. On the other hand, Neural ODEs, another class of implicit-depth models, do guarantee existence of a unique solution but perform poorly compared with traditional net-

Figure 2: From NeurIPS 2020, *about 70 citations* [Winston and Kolter, 2020]

Equilibrium Networks in Deep Learning

Where do they come from?

Deep Equilibrium Models

Shaojie Bai
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI

Vladlen Koltun
Intel Labs

Abstract

We present a new approach to modeling sequential data: the deep equilibrium model (DEQ). Motivated by an observation that the hidden layers of many existing deep sequence models converge towards some fixed point, we propose the DEQ approach that *directly* finds these equilibrium points via root-finding. Such a

Figure 1: From NeurIPS 2019, *about 400 citations* [Bai et al., 2019]

Monotone operator equilibrium networks

Ezra Winston
School of Computer Science
Carnegie Mellon University
Pittsburgh, United States
ewinston@cs.cmu.edu

J. Zico Kolter
School of Computer Science
Carnegie Mellon University
& Bosch Center for AI
Pittsburgh, United States
zkolter@cs.cmu.edu

Abstract

Implicit-depth models such as Deep Equilibrium Networks have recently been shown to match or exceed the performance of traditional deep networks while being much more memory efficient. However, these models suffer from unstable convergence to a solution and lack guarantees that a solution exists. On the other hand, Neural ODEs, another class of implicit-depth models, do guarantee existence of a unique solution but perform poorly compared with traditional net-

Figure 2: From NeurIPS 2020, *about 70 citations* [Winston and Kolter, 2020]

2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Recurrent Equilibrium Networks: Unconstrained Learning of Stable and Robust Dynamical Models

Max Revay, Ruigang Wang, Ian R. Manchester

***Abstract*— This paper introduces *recurrent equilibrium networks* (RENs), a new class of nonlinear dynamical models for applications in machine learning and system identification. The new model class has "built in" guarantees of stability and robustness: all models in the class are contracting – a strong form of nonlinear stability – and models can have prescribed Lipschitz bounds. RENs are otherwise very flexible: they can**

the works [10], [11], [13], [14], [15] are guaranteed to find contracting models.

Beyond stability, model *robustness* can be characterised in terms of sensitivity to small perturbations in the input. It has recently been shown that recurrent neural network models can be extremely fragile [17], i.e. small changes to

Figure 3: From CDC 2021, *8 citations* [Revay et al., 2021]

Equilibrium Networks in Deep Learning

Where do they come from?

Deep Equilibrium Models

Shaojie Bai
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI

Vladlen Koltun
Intel Labs

Abstract

We present a new approach to modeling sequential data: the deep equilibrium model (DEQ). Motivated by an observation that the hidden layers of many existing deep sequence models converge towards some fixed point, we propose the DEQ approach that *directly* finds these equilibrium points via root-finding. Such a

Figure 1: From NeurIPS 2019, *about 400 citations* [Bai et al., 2019]

Monotone operator equilibrium networks

Ezra Winston
School of Computer Science
Carnegie Mellon University
Pittsburgh, United States
ewinston@cs.cmu.edu

J. Zico Kolter
School of Computer Science
Carnegie Mellon University
& Bosch Center for AI
Pittsburgh, United States
zkolter@cs.cmu.edu

Abstract

Implicit-depth models such as Deep Equilibrium Networks have recently been shown to match or exceed the performance of traditional deep networks while being much more memory efficient. However, these models suffer from unstable convergence to a solution and lack guarantees that a solution exists. On the other hand, Neural ODEs, another class of implicit-depth models, do guarantee existence of a unique solution but perform poorly compared with traditional net-

Figure 2: From NeurIPS 2020, *about 70 citations* [Winston and Kolter, 2020]

2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Recurrent Equilibrium Networks: Unconstrained Learning of Stable and Robust Dynamical Models

Max Revay, Ruigang Wang, Ian R. Manchester

***Abstract*—This paper introduces *recurrent equilibrium networks* (RENs), a new class of nonlinear dynamical models for applications in machine learning and system identification. The new model class has "built in" guarantees of stability and robustness: all models in the class are contracting – a strong form of nonlinear stability – and models can have prescribed Lipschitz bounds. RENs are otherwise very flexible: they can**

the works [10], [11], [13], [14], [15] are guaranteed to find contracting models.

Beyond stability, model *robustness* can be characterised in terms of sensitivity to small perturbations in the input. It has recently been shown that recurrent neural network models can be extremely fragile [17], i.e. small changes to

Figure 3: From CDC 2021, *8 citations* [Revay et al., 2021]

Connection between recurrent neural networks and linear time-invariant (LTI) systems

- Recurrent neural networks are a special case of LTI systems with nonlinear disturbance
- Equilibrium networks further generalize the network structure
- Robustness analysis known from robust control can be applied to such systems

Inverted pendulum with torque input

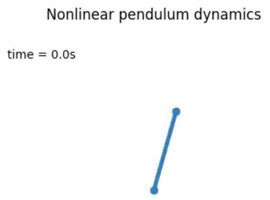
Classical toy example in control

Difference equation of pendulum dynamics

$$\mathcal{G}\begin{cases} x^{k+1} = \begin{pmatrix} 1 & \delta \\ \frac{g\delta}{l} & 1-\frac{\mu\delta}{ml^2} \end{pmatrix} x^k + \begin{pmatrix} 0 \\ -\frac{g\delta}{l} \end{pmatrix} u^k + \begin{pmatrix} 0 \\ \frac{\delta}{ml^2} \end{pmatrix} w^k \\ y^k = \begin{pmatrix} 1 & 0 \end{pmatrix} x^k \\ z^k = \begin{pmatrix} 1 & 0 \end{pmatrix} x^k \end{cases} \tag{1}$$

$$w^k = \Delta(z^k) = z^k - \sin(z^k) \tag{2}$$

- The states $x^k = [\phi,\ \dot{\phi}]^{\mathrm{T}}$ represents the *angle* and *angular velocity*, friction is denoted by μ and sample time by $\delta = 0.01$
- Input u^k represents a torque
- Output y^k is the *angle* ϕ
- One stable and one unstable equilibrium point



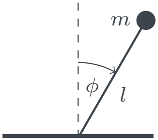


Figure 4: Parameters of single pendulum.

Inverted pendulum with torque input

Classical toy example in control

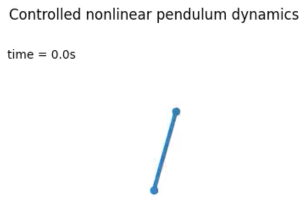
Difference equation of pendulum dynamics

$$\mathcal{G}\begin{cases} x^{k+1} = \begin{pmatrix} 1 & \delta \\ \frac{g\delta}{l} & 1-\frac{\mu\delta}{ml^2} \end{pmatrix} x^k + \begin{pmatrix} 0 \\ -\frac{g\delta}{l} \end{pmatrix} u^k + \begin{pmatrix} 0 \\ \frac{\delta}{ml^2} \end{pmatrix} w^k \\ y^k = \begin{pmatrix} 1 & 0 \end{pmatrix} x^k \\ z^k = \begin{pmatrix} 1 & 0 \end{pmatrix} x^k \end{cases} \tag{1}$$

$$w^k = \Delta(z^k) = z^k - \sin(z^k) \tag{2}$$

- The states $x^k = [\phi,\ \dot{\phi}]^{\mathrm{T}}$ represents the *angle* and *angular velocity*, friction is denoted by μ and sample time by $\delta = 0.01$
- Input u^k represents a torque
- Output y^k is the *angle* ϕ
- One stable and one unstable equilibrium point

Linearized system can be used to design a controller that stabilizes the unstable equilibrium point



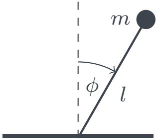


Figure 4: Parameters of single pendulum.

Background

2

Generalization of Recurrent Neural Networks

LTI system with nonlinear disturbance

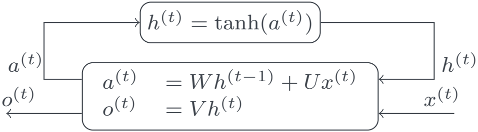


Figure 5: Recurrent neural network from [Goodfellow et al., 2016]

Generalization of Recurrent Neural Networks

LTI system with nonlinear disturbance

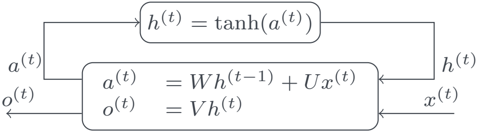


Figure 5: Recurrent neural network from [Goodfellow et al., 2016]

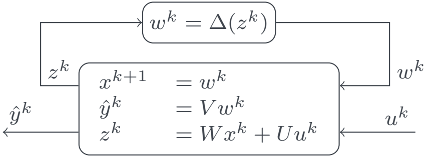


Figure 6: Using common notation from system theory.

Generalization of Recurrent Neural Networks

LTI system with nonlinear disturbance

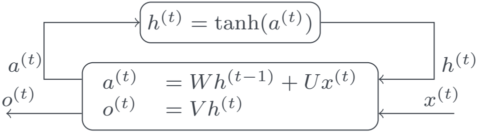


Figure 5: Recurrent neural network from [Goodfellow et al., 2016]

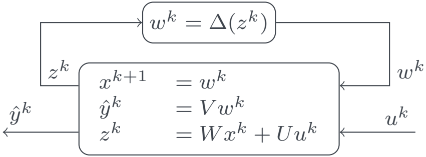


Figure 6: Using common notation from system theory.

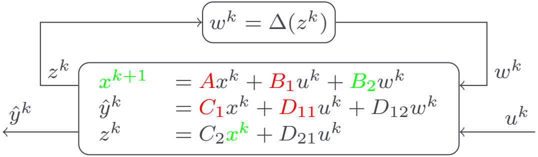


Figure 7: LTI with non-linear disturbance.

Generalization of Recurrent Neural Networks

LTI system with nonlinear disturbance

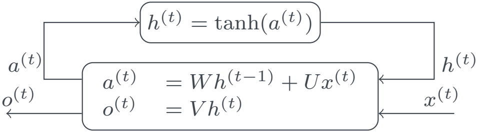


Figure 5: Recurrent neural network from [Goodfellow et al., 2016]

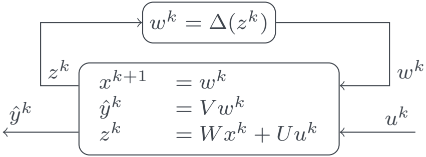


Figure 6: Using common notation from system theory.

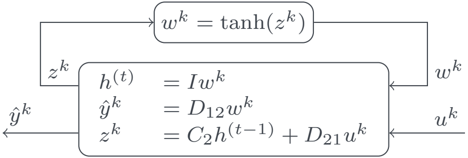


Figure 7: LTI with non-linear disturbance.

- For $A = B_1 = C_1 = D_{11} = 0$, $B_2 = I$, $\Delta(\cdot) = \tanh(\cdot)$ and $h^{t-1} = x^k$ the networks in Figure 5 and Figure **??** are equivalent

Pendulum is also in LTI structure

Same representation as RNNs

Difference equation of pendulum dynamics

$$
\mathcal{G}\begin{cases}
x^{k+1} = \underbrace{\begin{pmatrix} 1 & \delta \\ \frac{g\delta}{l} & 1-\frac{\mu\delta}{ml^2} \end{pmatrix}}_{A} x^k + \underbrace{\begin{pmatrix} 0 \\ -\frac{g\delta}{l} \end{pmatrix}}_{B_1} u^k + \underbrace{\begin{pmatrix} 0 \\ \frac{\delta}{ml^2} \end{pmatrix}}_{B_2} w^k \\
y^k = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C_1} x^k \\
z^k = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_{C_2} x^k
\end{cases}
$$

$$
w^k = \Delta(z^k) = z^k - \sin(z^k)
$$

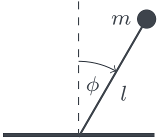


Figure 8: Parameters of single pendulum.

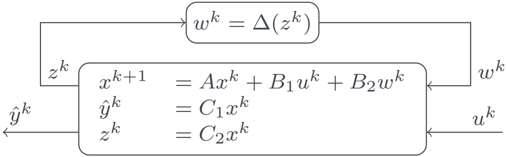


Figure 9: Pendulum dynamics in LTI structure with nonlinearity $\Delta(\cdot)$

Why is this connection interesting?

Linear systems are well understood

- Stability depends on eigenvalues of A matrix
- Well established controller design techniques

Figure 10: [Lunze and Lunze, 1996]

Why is this connection interesting?

Linear systems are well understood

- Stability depends on eigenvalues of A matrix
- Well established controller design techniques

Figure 10: [Lunze and Lunze, 1996]

Linear system with disturbances

- For known disturbances robust control methods can be applied
- Activation function of neural networks can be seen as disturbance

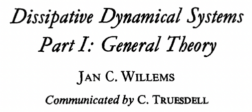


Figure 11: [Willems, 1972]

System Identification Problem

If the equations of the pendulum are not available but we are given data $\mathcal{D} = \{(u, y)_i\}_i^N$, which contains input output measurements. Finding the parameters A, B_1, B_2, C_1, C_2 can be seen as a deep learning problem.

Why is this connection interesting?

Linear systems are well understood

- Stability depends on eigenvalues of A matrix
- Well established controller design techniques

Figure 10: [Lunze and Lunze, 1996]

Linear system with disturbances

- For known disturbances robust control methods can be applied
- Activation function of neural networks can be seen as disturbance

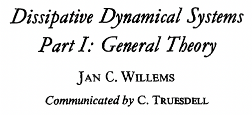


Figure 11: [Willems, 1972]

System Identification Problem

If the equations of the pendulum are not available but we are given data $\mathcal{D} = \{(u, y)_i\}_i^N$, which contains input output measurements. Finding the parameters A, B_1, B_2, C_1, C_2 can be seen as a deep learning problem.

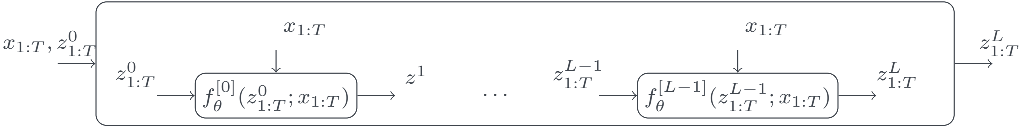
Neural networks that can be represented in LTI structure with nonlinear disturbances can be analyzed with well established tools from robust control.

Deep Equilibrium Models

[Bai et al., 2019]

- Deep sequence model $\mathcal{DEQ}$ that maps an input sequence $x_{1:T}$ to an output sequence $z^L_{1:T}$

Deep Sequence Model $\mathcal{DEQ}$

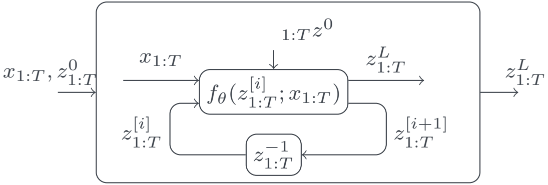


Deep Equilibrium Models

[Bai et al., 2019]

- Deep sequence model $\mathcal{DEQ}$ that maps an input sequence $x_{1:T}$ to an output sequence $z^L_{1:T}$
- Layers are weight tied $f_\theta(z^0_{1:T}; x) = f^{[i]}_\theta(z^0_{1:T}; x_{1:T})$ for all $i = 0, \ldots, L-1$

Deep Sequence Model $\mathcal{DEQ}$

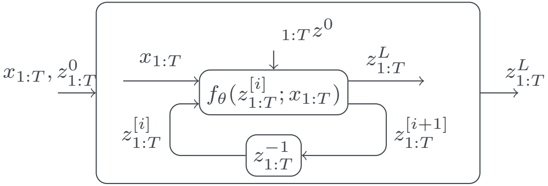


Deep Equilibrium Models

[Bai et al., 2019]

- Deep sequence model $\mathcal{DEQ}$ that maps an input sequence $x_{1:T}$ to an output sequence $z^L_{1:T}$
- Layers are weight tied $f_\theta(z^0_{1:T}; x) = f^{[i]}_\theta(z^0_{1:T}; x_{1:T})$ for all $i = 0, \ldots, L-1$

Deep Sequence Model $\mathcal{DEQ}$



Core idea of deep equilibrium models

- Equilibrium point $z^*_{1:T} = f_\theta(z^*_{1:T}; x_{1:T})$
- Find equilibrium point via root finding method e.g. *Newton's method*

$$z^*_{1:T} = \mathrm{RootFind}(g_\theta; x_{1:T}), \tag{3}$$

were $g_\theta(z^*_{1:T}; x_{1:T}) = f_\theta(z^*_{1:T}; x_{1:T}) - z^*_{1:T}$
- No gradients on each layer, classical backpropagation not possible.

Equilibrium Models for System Identification

3

Conclusion

4

University of Stuttgart

Daniel Frank
Institute for Parallel and Distributed Systems
Analytic Computing

eMail	daniel.frank@ipvs.uni-stuttgart.de
phone	+49 711 685-88107
address	Universitaetstraße 32 70569 Stuttgart 2.204

Literature I

[Bai et al., 2019] Bai, S., Kolter, J. Z., and Koltun, V. (2019).
Deep equilibrium models.
In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 688–699.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

[Lunze and Lunze, 1996] Lunze, J. and Lunze, J. (1996).
Regelungstechnik 1, volume 10.
Springer.

[Revay et al., 2021] Revay, M., Wang, R., and Manchester, I. R. (2021).
Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models.
In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2282–2287. IEEE.

[Willems, 1972] Willems, J. C. (1972).
Dissipative dynamical systems part i: General theory.
Archive for rational mechanics and analysis, 45(5):321–351.

Literature II

[Winston and Kolter, 2020] Winston, E. and Kolter, J. Z. (2020).
Monotone operator equilibrium networks.
Advances in neural information processing systems, 33:10718–10728.