# Deep Equilibrium Models

Трошин Сергей

Высшая Школа Экономики

https://arxiv.org/pdf/1909.01377.pdf

# Deep Equilibrium Models

**Shaojie Bai**
Carnegie Mellon University

**J. Zico Kolter**
Carnegie Mellon University
Bosch Center for AI

**Vladlen Koltun**
Intel Labs

- Advances in Neural Information Processing Systems (NeurIPS), 2019 (Selected for spotlight oral presentation)

# Outline

- Deep Learning for Sequence Modelling
- Deep Equilibrium Models
- Experiments
- Convergence, Universality

# Deep Learning for Sequence Modelling

# Sequence modeling task

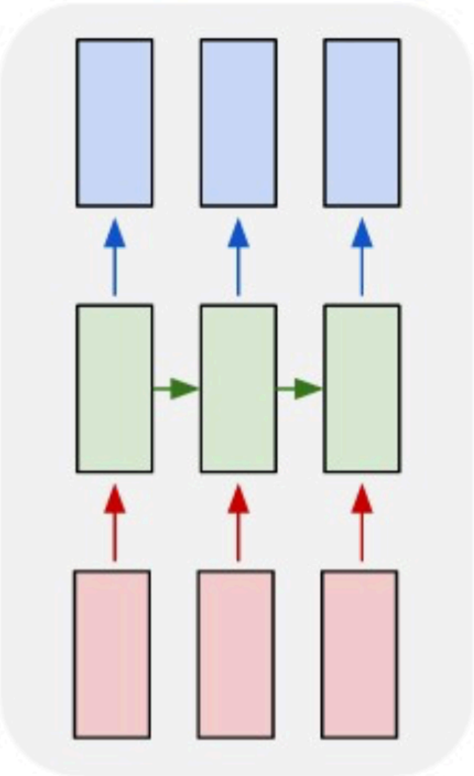
$$x_{[1:T]} = [x_1, \ldots, x_T]$$

$$y_{[1:T]} = [y_1, \ldots, y_T]$$

Constraint: causality

Applications:
- Language modeling
- Time series tasks

$y_1, \ldots, y_T$



$x_1, \ldots, x_T$

# Limitations of using very deep neural networks.

- Need O(L) memory for training, L – the number of layers.

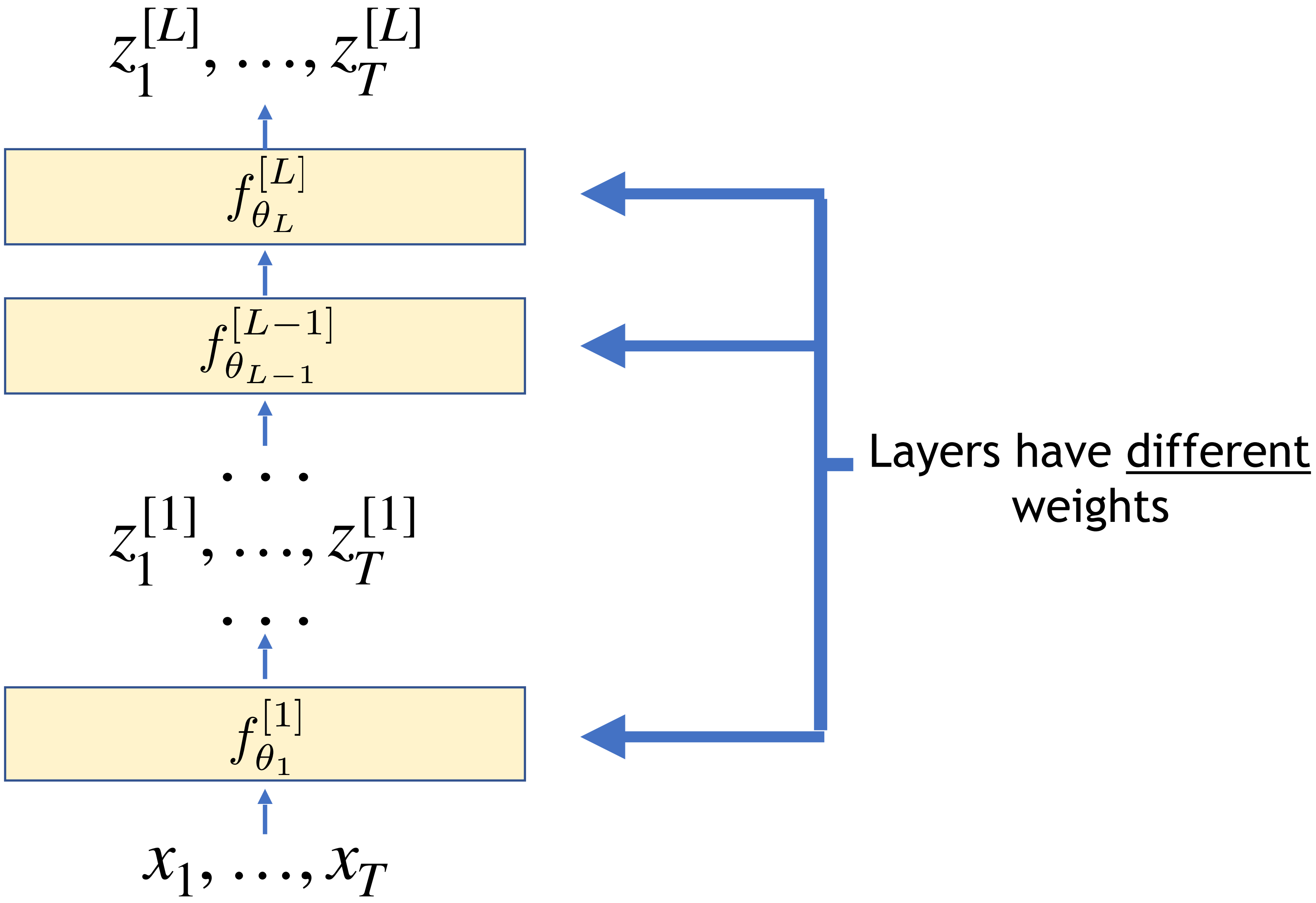
Solutions:

- Gradient Checkpointing (2016): O($\sqrt{L}$ )
- Neural ODEs(2018): Constant (using black-box solver for backward pass)

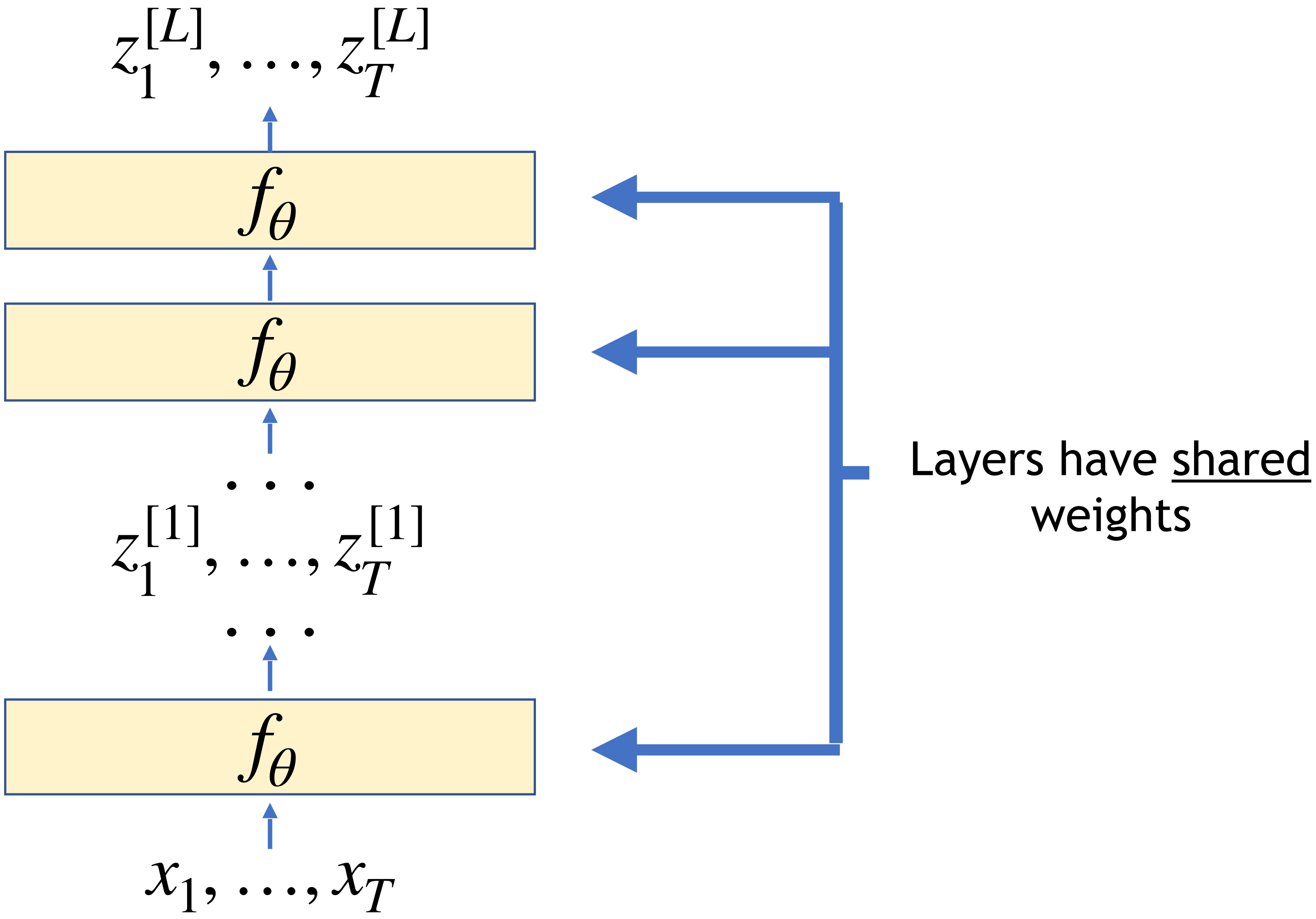
https://github.com/cybertronai/gradient-checkpointing

https://arxiv.org/abs/1806.07366

# Common deep sequence model



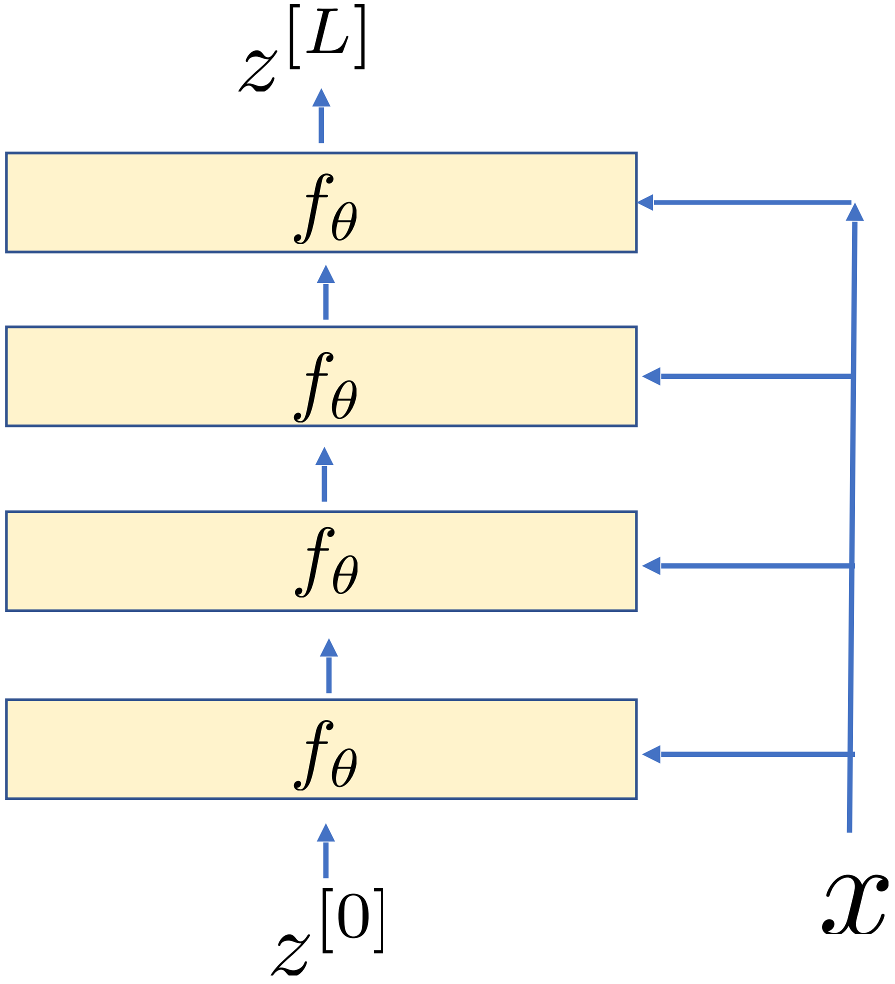
# Weight-tied deep sequence model



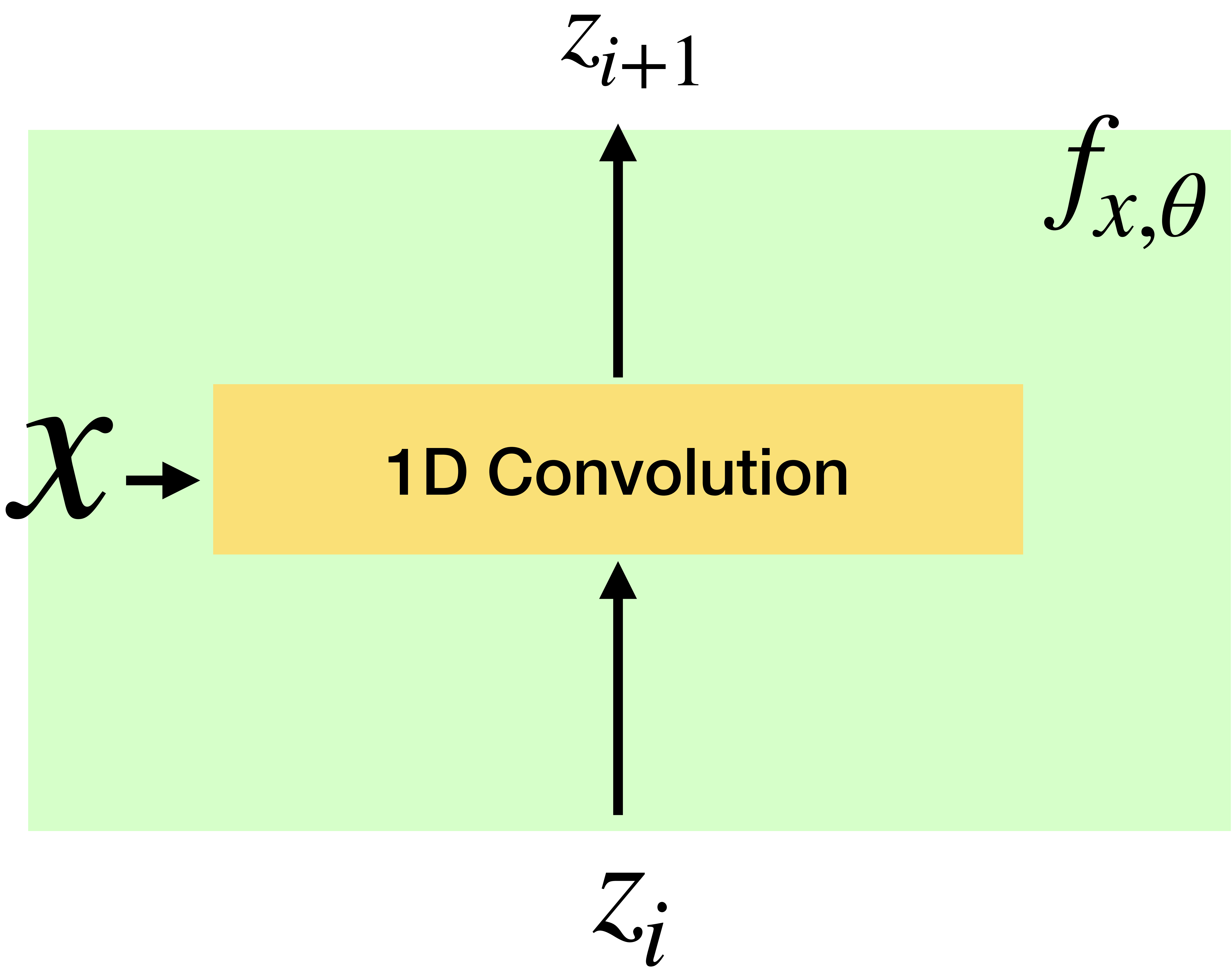
# Weight-tied Input-Injected DNN

$$z^{[i+1]} = f_{\theta,x}(z^{[i]})$$

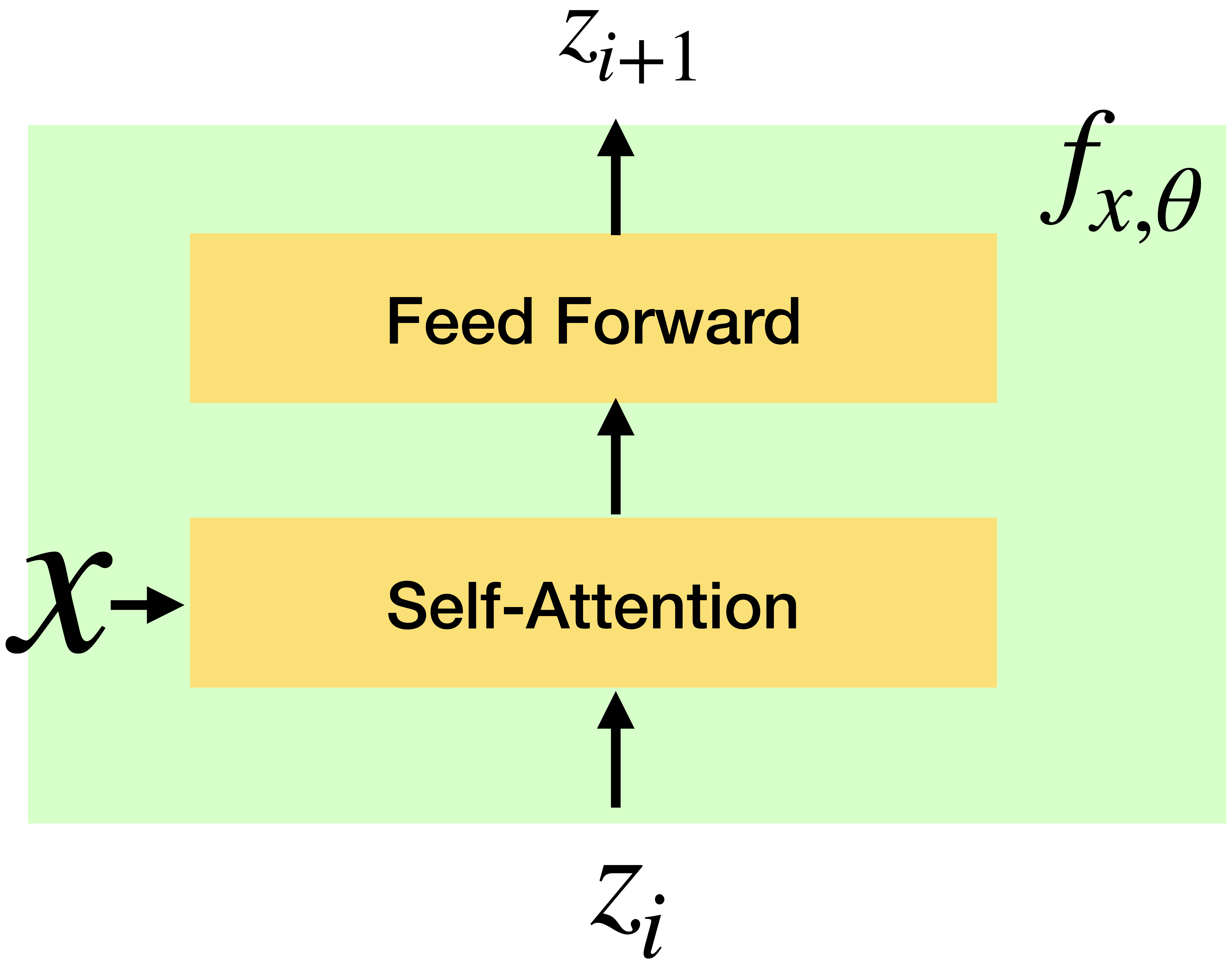
$$z^{[0]} = 0$$



Trellis networks (2019)



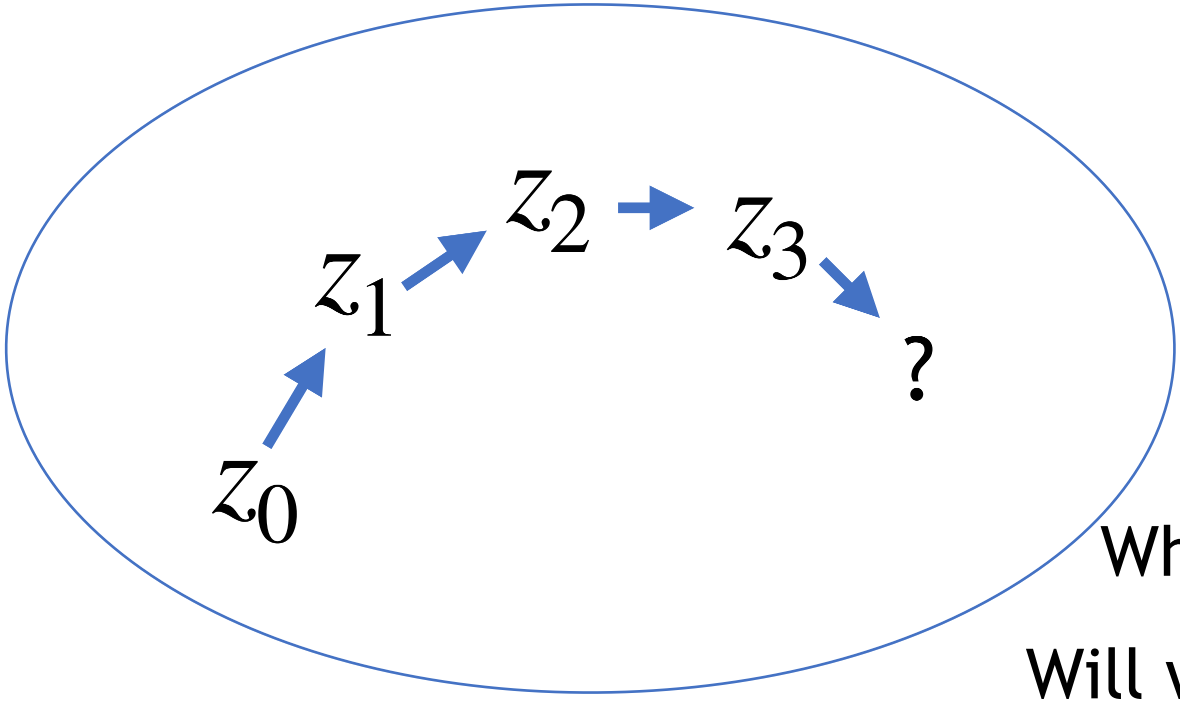
# Universal Transformer (2019)



# DEQ

# What if we increase the number of layers

What is the dynamics of our outputs?

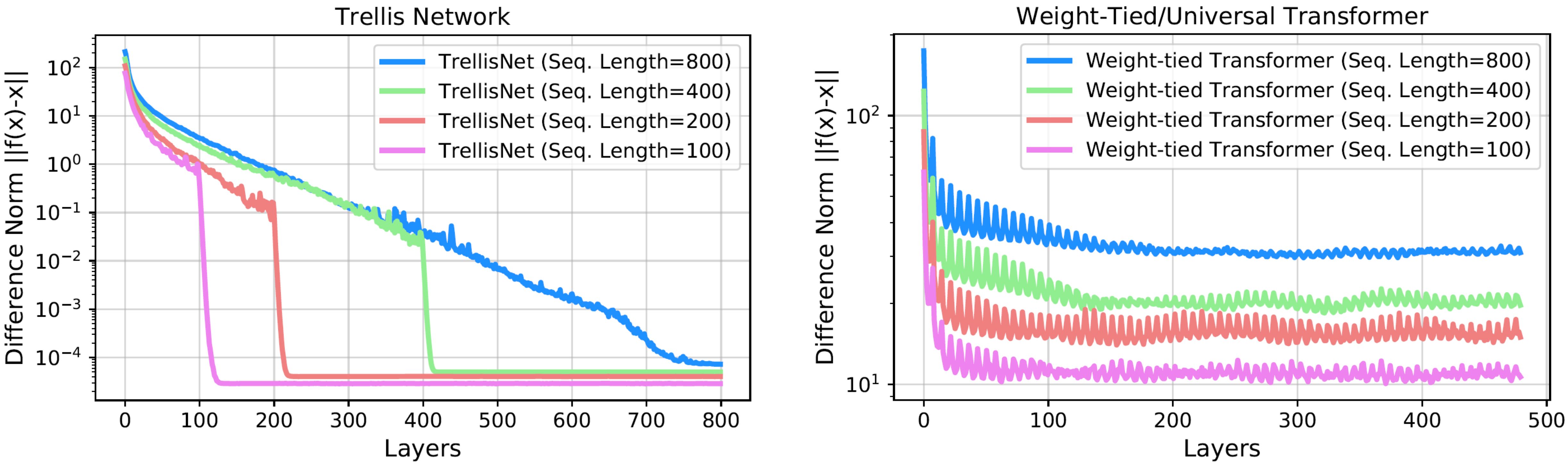




What if we apply our layer many times?

Will we converge to some attractor?

# Empirical Evidence




A tendency of layers to converge

# Convergence to the same point

$$\lim_{i \to \infty} \mathbf{z}_{1:T}^{[i]} = \lim_{i \to \infty} f_\theta\left(\mathbf{z}_{1:T}^{[i]}; \mathbf{x}_{1:T}\right) \equiv f_\theta\left(\mathbf{z}_{1:T}^{\star}; \mathbf{x}_{1:T}\right) = \mathbf{z}_{1:T}^{\star}$$

Equilibrium Point

# Implicit Function Theorem

**The Implicit Function Theorem for** $\mathbb{R}^2$ :

**Consider a continuously differentiable function** $G : \mathbb{R}^2 \to \mathbb{R}^2$

**and a point** $(x_0, z_0) \in \mathbb{R}^2$ **so that** $G(x_0, z_0) = 0$.

**If** $\frac{\partial G}{\partial z}(x_0, z_0) \neq 0$, **there is a neighbourhood of** $(x_0, z_0)$

**so that whenever** $x$ **is sufficiently close to** $x_0$ **there is a unique** $z$ **,**

**so that** $G(x, z) = 0$.

**Implicit differentiation:**

$$G(x, z(x)) = 0 \Rightarrow \frac{dG}{dx} = \frac{\partial G}{\partial x}\frac{dx}{dx} + \frac{\partial G}{\partial z}\frac{dz}{dx} = \frac{\partial G}{\partial x} + \frac{\partial G}{\partial z}\frac{dz}{dx} = 0$$

$$\Rightarrow \frac{dz}{dx} = -\left(\frac{\partial G}{\partial z}\right)^{-1}\frac{\partial G}{\partial x}$$

# Commentary: $\partial$ vs $d$

**For a function** $G(x, z(x)) : \mathbb{R}^2 \to \mathbb{R}$,

$$\frac{\partial G}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, z(x))}{\Delta x}$$

$$\frac{dG}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, z(x + \Delta x))}{\Delta x}$$

$$z^* = f_{\theta,x} \circ f_{\theta,x} \circ \ldots \circ f_{\theta,x}(z_0), \quad z_0 = 0$$

$$f_{\theta,x}(z^*) = z^*$$ **Equilibrium equation**

$$y := \{x, \theta\}$$

$$G(y, z) := f_{x,\theta}(z) - z$$

$$G(y, z^*) = 0 \xrightarrow[\text{Implicit Function Th.}]{} G(y, z^*(y)) = 0$$

$$\frac{dG}{dy} = \frac{\partial G}{\partial y} + \frac{\partial G}{\partial z}\frac{dz}{dy}$$

$$\frac{dG}{dy} = \frac{\partial G}{\partial y} + \frac{\partial G}{\partial z}\frac{dz}{dy} = 0,\ if\ \ z = z^*$$

$$\frac{dz}{dy} = -\left(\frac{\partial G}{\partial z}\right)^{-1}\frac{dG}{dy} = -\left(\frac{\partial f}{\partial z} - I\right)^{-1}\frac{df}{dy}$$

**Finally!**

**A derivative of loss function w.r.t parameters:**

$$\frac{\partial \ell}{\partial y} = \frac{\partial \ell}{\partial z^*}\frac{dz^*}{dy} = -\frac{\partial \ell}{\partial z^*}\left(\frac{\partial f}{\partial z} - I\right)^{-1}\frac{df}{dy}$$

# Reliable estimation of equilibrium

- Unfortunately $\lim_{i\to\infty} \underbrace{f_{\theta,x} \circ \ldots \circ f_{\theta,x}}_{i} (z_0)$ may not exist or a convergence may be very slow in practice.

- Fortunately, the dimensionality of $z$ is quite low e.g. 500 (comparing with a number of parameters in neural networks). Hence, we can use one of Quasi-Newton methods, e.g. Broyden method (will be shown to reliably find an equilibrium point)

# Reliable estimation of equilibrium: forward pass

**Broyden method:**

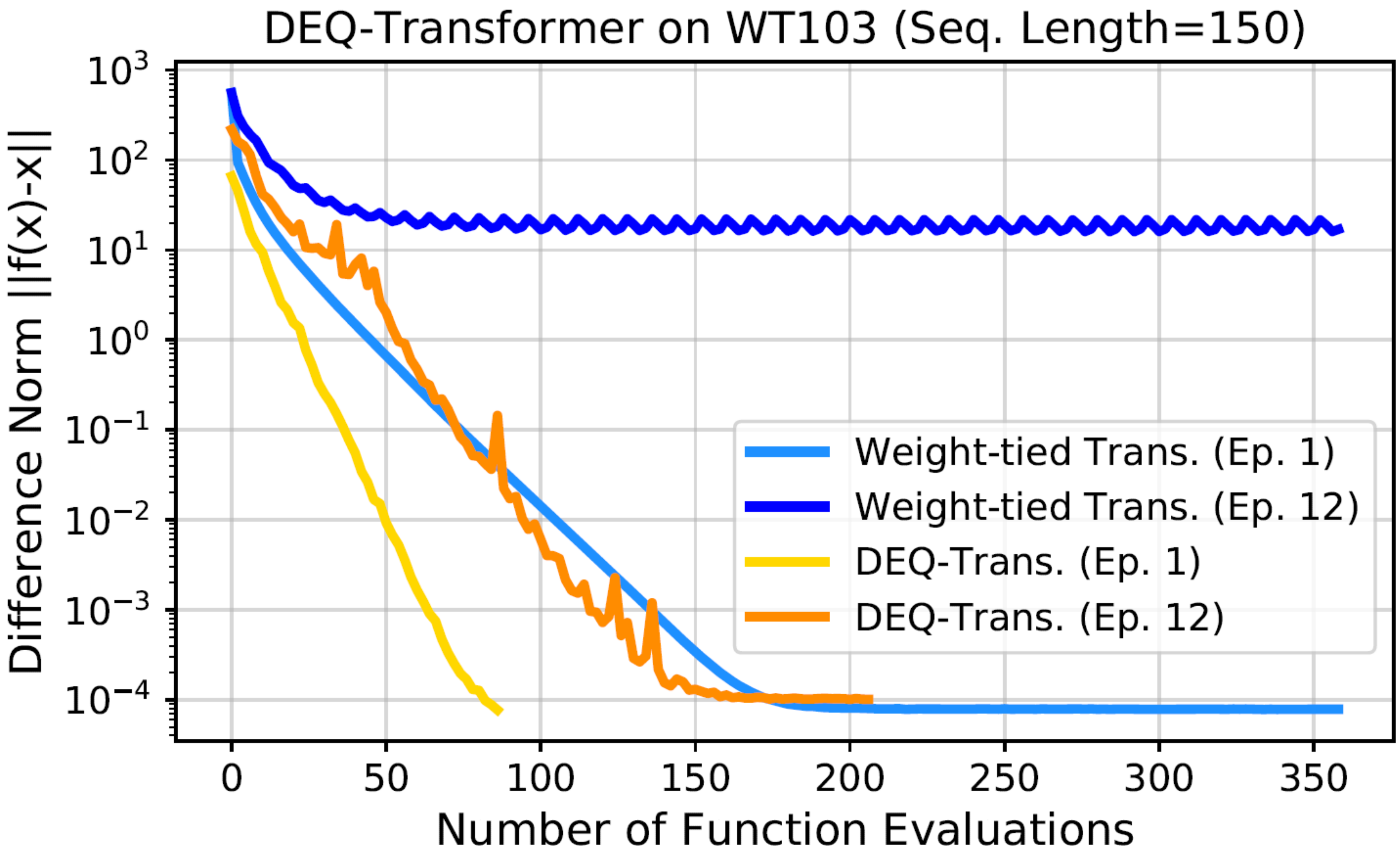
**finds** $z$**, such that** $G(y, z) = 0$

**or more formally,** $z = \arg\min_{z} \| G(y, z) \|_2$

$z_{i+1} = z_i - \alpha\, B\, G(y, z_i)$, **for** $i = 0,1,2,\ldots$

$B \approx J_G^{-1}|_{z_i}$ **– low rank approximation**

$\alpha$ **– step size**




DEQ-transformer finds equilibrium more reliably

## Forward Pass:

$$\mathbf{z}^\star_{1:T} = \mathsf{RootFind}(g_\theta; \mathbf{x}_{1:T})$$

## Backward Pass:

$$\frac{\partial \ell}{\partial (\cdot)} = -\frac{\partial \ell}{\partial \mathbf{z}^\star_{1:T}} \left( J^{-1}_{g_\theta} \Big|_{\mathbf{z}^\star_{1:T}} \right) \frac{\mathrm{d} f_\theta(\mathbf{z}^\star_{1:T}; \mathbf{x}_{1:T})}{\mathrm{d}(\cdot)} = -\frac{\partial \ell}{\partial h} \frac{\partial h}{\partial \mathbf{z}^\star_{1:T}} \left( J^{-1}_{g_\theta} \Big|_{\mathbf{z}^\star_{1:T}} \right) \frac{\mathrm{d} f_\theta(\mathbf{z}^\star_{1:T}; \mathbf{x}_{1:T})}{\mathrm{d}(\cdot)}$$

# Forward & Backward

**What we already have:**

- Forward: $z^* = \text{Broyden}(f, x, z_0)$
- Backward: $\dfrac{\partial \ell}{\partial y} = -\dfrac{\partial \ell}{\partial z^*}\left(J_G^{-1}|_{z^*}\right)\dfrac{df}{dy}$

# Accelerating Backward

- Backward: $\frac{\partial \ell}{\partial y} = -\frac{\partial \ell}{\partial z^*}\left(J_G^{-1}|_{z^*}\right)\frac{df}{dy}$

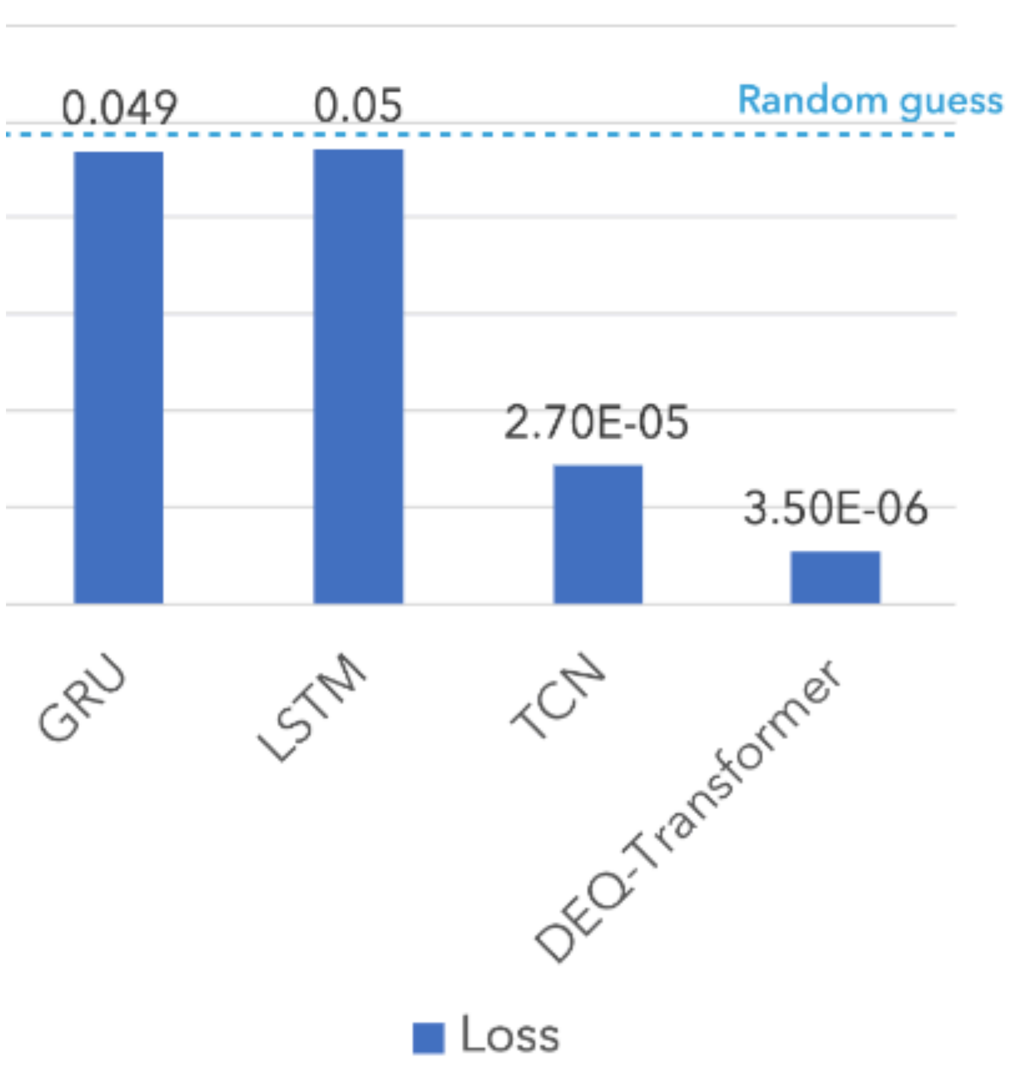
- Instead of calculating $-\frac{\partial \ell}{\partial z^*}\left(J_G^{-1}|_{z^*}\right)$ directly, we can hack and solve a linear system:

- let $\mathbf{b} = -\left(\frac{\partial \ell}{\partial z^*}\right)^T$, $\quad \mathbf{A} = J_G^T|_{z^*}$, $\quad \mathbf{x} = -\frac{\partial \ell}{\partial z^*}\left(J_G^{-1}|_{z^*}\right)$

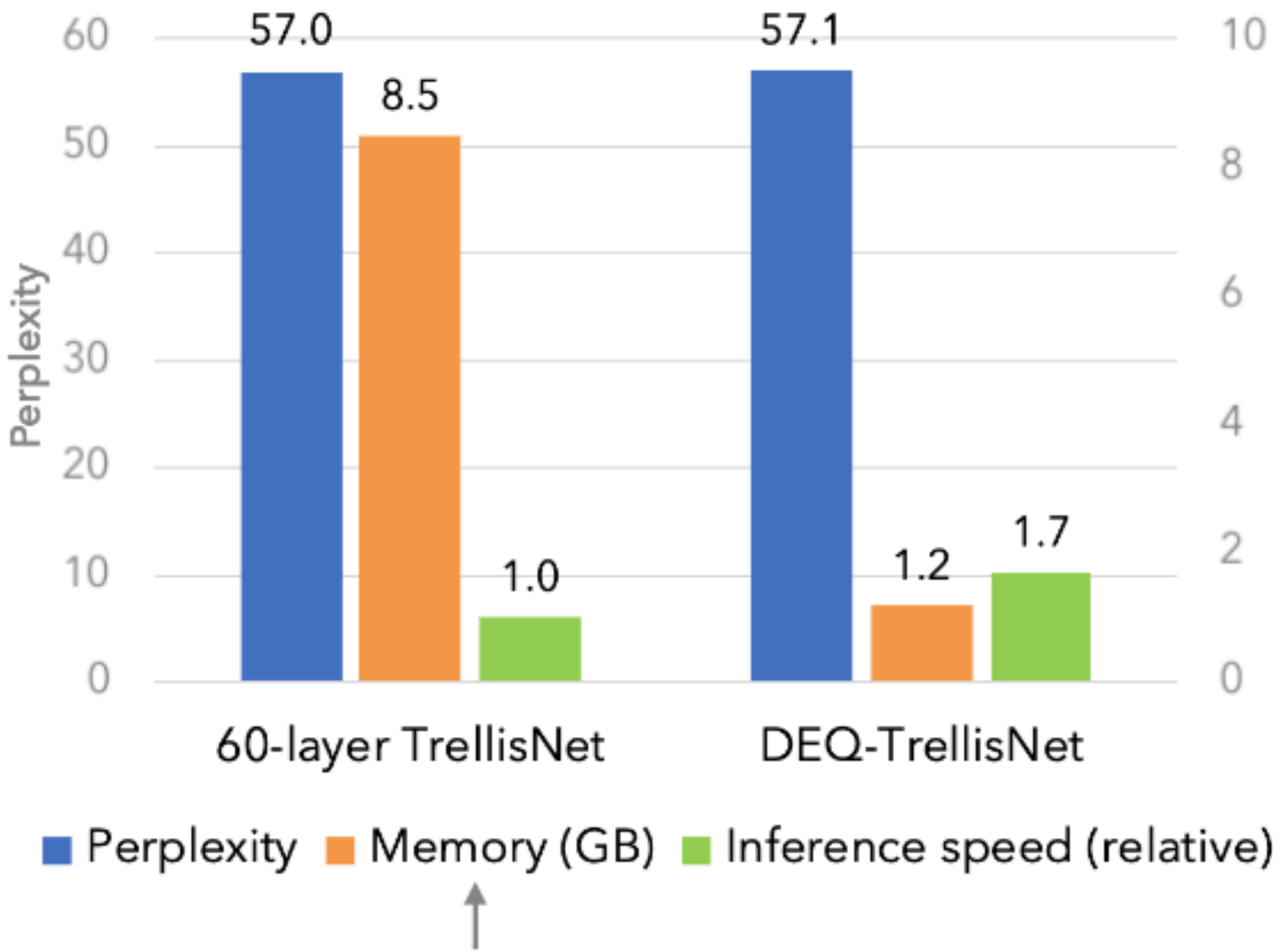
- solve $\mathbf{Ax} + \mathbf{b} = 0$ for unknown $\mathbf{x}$ again with Broyden

# Experiments

(Long-Range) Copy Memory Task

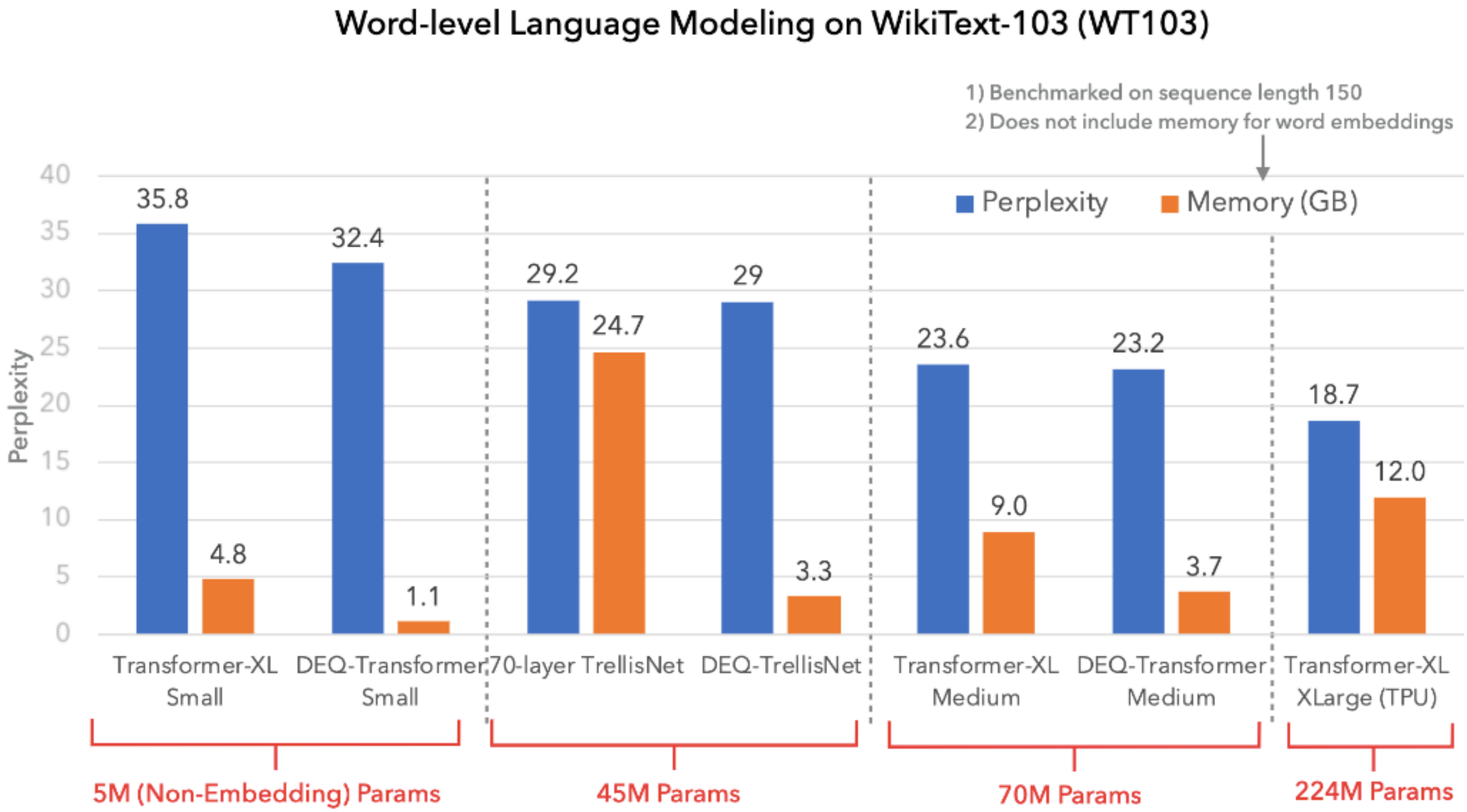



Word-level Language Modeling on Penn Treebank (PTB)




1) Benchmarked on sequence length 150
2) Does not include memory for word embeddings

https://github.com/locuslab/deq/blob/master/presentations/DEQ_slides.pdf




https://github.com/locuslab/deq/blob/master/presentations/DEQ_slides.pdf

# Convergence, Universality

# Is fixed point unique?

1) Upper diagonal matrix condition

Input

$$\begin{aligned} \mathbf{z}^{[1]} &= \sigma(W_1\mathbf{x} + b_1) \\ \mathbf{z}^{[2]} &= \sigma(W_2\mathbf{z}^{[1]} + b_2) \\ \mathbf{z}^{[3]} &= \sigma(W_3\mathbf{z}^{[2]} + b_3) \end{aligned} \iff \begin{bmatrix} \mathbf{z}^{[1]} \\ \mathbf{z}^{[2]} \\ \mathbf{z}^{[3]} \end{bmatrix} = \sigma\left( \begin{bmatrix} 0 & 0 & 0 \\ W_2 & 0 & 0 \\ 0 & W_3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z}^{[1]} \\ \mathbf{z}^{[2]} \\ \mathbf{z}^{[3]} \end{bmatrix} + \begin{bmatrix} W_1 \\ 0 \\ 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

(Apply this **three** times to $\begin{bmatrix} \mathbf{z}^{[1]} & \mathbf{z}^{[2]} & \mathbf{z}^{[3]} \end{bmatrix}^\top = \mathbf{0}$ )

https://github.com/locuslab/deq/blob/master/presentations/DEQ_slides.pdf

# Is fixed point unique?

2) Contractive mapping condition

Equation $z = \sigma(Az + Ux)$ has unique solution if

$$\forall z_1, z_2 \;\; |\sigma(z_1 - z_2)| \leq |z_1 - z_2|$$

$$\forall z, |z| \leq 1 \;\; |Az| \leq 1$$

https://arxiv.org/pdf/1908.06315.pdf

# Is fixed point for Transformer of Trellis Network unique?

None of these two conditions can be applied :(

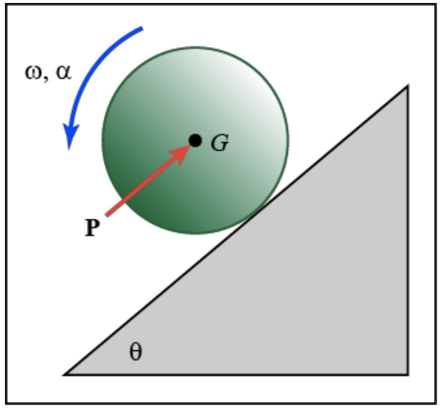
But there are guys who try to enforce one of them to make problem well-posed (Implicit Deep Learning, https://arxiv.org/pdf/1908.06315.pdf)

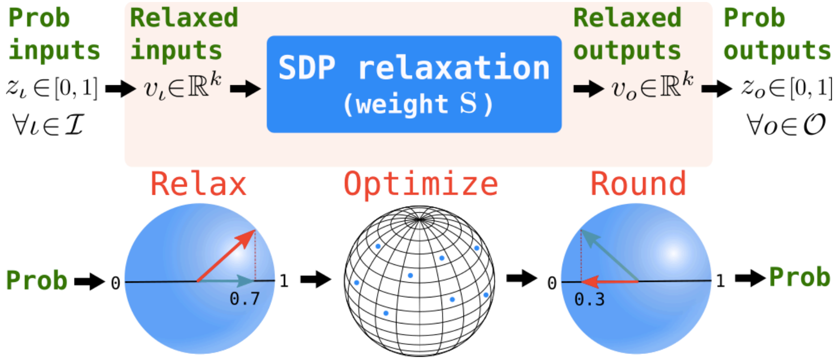
# More Implicit Layers

Far not a complete list:

- OptNet [Amos and Kolter, 2017]
- Differentiable Physics [Belbute-Peres et al., 2018]
- Combinatorial optimisation [Wang et al., 2019]

$$z_{i+1} = \underset{z}{\operatorname{argmin}} \ \frac{1}{2} z^T Q(z_i) z + q(z_i)^T z$$
$$\text{subject to } \ A(z_i) z = b(z_i)$$
$$G(z_i) z \leq h(z_i)$$




http://www.conceptualdynamics.com/files/rbnewt/rbnewt_g_ep2.htm https://github.com/locuslab/SATNet/blob/master/images/poster_forward.png

# Conclusions

- DEQ – a memory efficient model for sequential data, but can be slow to train.
- When some optimal conditions holds we can forget about the path and directly solve for a Jacobian through them.
- Every feed-forward deep model can be made implicit
- Further theoretical research is required

# References


- https://arxiv.org/pdf/1909.01377.pdf - Deep Equilibrium Models
- https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf - Transformer
- https://arxiv.org/pdf/1807.03819.pdf - Universal Transformer
- https://arxiv.org/pdf/1810.06682.pdf - Trellis Networks
- https://github.com/cybertronai/gradient-checkpointing - Gradient Checkpointing
- https://arxiv.org/abs/1806.07366 - Neural ODEs
- https://arxiv.org/pdf/1908.06315.pdf - Implicit Deep Learning
- https://arxiv.org/pdf/1703.00443.pdf - OptNet
- https://arxiv.org/pdf/1905.12149.pdf - SATNet