Implicit Bias of Linear RNNs





Melika Emami [1,] , Mojtaba Sahraee-Ardakan [1,2],
Parthe Pandit[1,2], Sundeep Rangan[3], Alyson K. Fletcher [1,2]

ECE, UCLA , [2]STAT, UCLA, [3]ECE, NYU

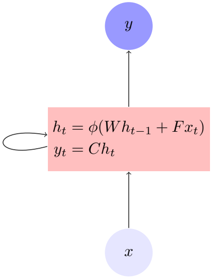
ICML 2021

Recurrent Neural Networks (RNNs)

- **RNN model**:

$$h_t = \phi(Wh_{t-1} + Fx_t), \quad y_t = Ch_t \tag{1}$$

  - Sequence-to-sequence mapping: $(x_0, \ldots, x_{T-1}) \to (y_0, \ldots, y_{T-1})$
  - Parameters: $\theta_{\mathrm{RNN}} = (W, F, C, h_{-1})$
  - h_t is n dimensional
  - Modeling sequential data

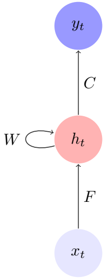


- **Empirically known**: RNNs learned from data cannot capture long-term dependencies
  - Short-term memory **bias**
- **Questions**:
  - How "short" is short-term memory?
  - Why is there short-term memory?
  - Can we control it?
- **Our contribution**:
  - Precisely characterize this short-term bias
  - Show connections to initialization

Linear RNNs

- **Linear RNN model**:

$$h_t = \frac{1}{\sqrt{n}} W h_{t-1} + F x_t, \quad y_t = \frac{1}{\sqrt{n}} C h_t \tag{2}$$



  - Parameters: $\theta_{\mathrm{RNN}} = (W, F, C)$ and $h_{-1} = 0$
  - State dimension: n
  - Mapping: $y = f_{\mathrm{RNN}}(x, \theta_{\mathrm{RNN}})$
  - **Non-linear** parameterization → hard to analyze
  - State-space representation of a linear system

Convolutional Equivalent System

- **Key observation**:

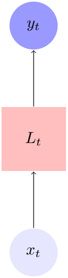
  Functional equivalence of linear RNNs and 1D convolutional model

- **1D convolutional model**:

$$y_t = \sum_{j=0}^{t} L_j x_{t-j}, \qquad L_j = \frac{1}{n^{(j+1)/2}} C W^j F. \qquad (3)$$

  - Linear systems theory:

    Identical input-output mapping with RNNs
  - Different parameterizations
  - **Linear** in parameters L_j
  - L_j: dimensions independent of n



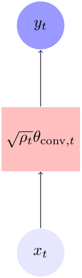
$$y_t = L_t * x_t$$

Scaled Convolutional Model

- **Scaled** 1D convolutional model:

$$y_t = \sum_{j=0}^{t} \sqrt{\rho_j}\,\theta_{j,\text{conv}}\, x_{t-j} \tag{4}$$

  - $\rho_j > 0$: positive fixed scaling factors
  - Learnable parameters: θ_{conv}
  - Mapping: $y = f_{\text{conv}}(x, \theta_{\text{conv}})$
  - **Linear** in parameters θ_{conv}
  - **Short-term memory** if ρ_j decreases with j
  - Dimensions of θ_{conv} independent of n



$$y_t = \sqrt{\rho_t}\theta_{\text{conv},t} * x_t$$

Main Result: Equivalence in Training

Linear RNN

- Initialize with:

$$W_{ij} \sim N(0, \nu_W), F_{ij} \sim N(0, \nu_F), C_{ki} \sim N(0, \nu_C)$$

- Non-linear parameterization
- Hard to analyze

Scaled 1D convolutional

- Fixed ρ_j
- Linear parameterization
- Easy to analyze

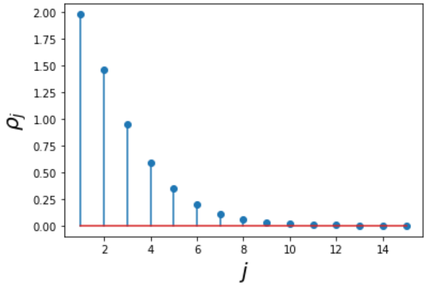
Theorem (Equivalence in Training)

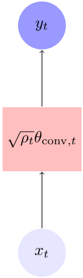
(a) *Under gradient descent, as $n \to \infty$, learning the linear RNN is equivalent to learning the scaled convolutional model when*

$$\rho_j = \nu_C(j\nu_F\nu_W^{j-1} + \nu_W^j) + \nu_F\nu_W^j. \tag{5}$$

(b) *The equivalence holds throughout training.*

Implications





- Consequence of equivalence:
  - Stability: $\frac{1}{\sqrt{n}}\lambda_{\max}(W) < 1 \Rightarrow \nu_W < 1$
  - ρ_j decays geometrically with ν_W^j
  - Coefficients with high lag are given low weight

Main Result: Implicit Bias Toward Short-Term Memory

- Consider step ℓ of gradient descent training of the linear RNN model:

$$\theta^{\ell}_{\mathrm{RNN}} = (W^{\ell}, F^{\ell}, C^{\ell}), \qquad L^{\ell}_{\mathrm{RNN},j} = n^{-(j+1)/2} C^{\ell} (W^{\ell})^j F^{\ell} \tag{6}$$

Theorem (Implicit Bias of Linear RNNs)

(a) At initialization, we have

$$\lim_{n\to\infty} \mathbb{E}\|L^{0}_{\mathrm{RNN},j}\|_F^2 = n_x n_y \nu_C \nu_F \nu_W^j. \tag{7}$$

(b) For all steps ℓ, there exists constants B_1 and B_2 such that if $\eta < B_1$,

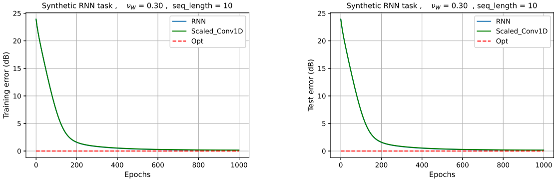
$$\limsup_{n\to\infty} \|L^{\ell}_{\mathrm{RNN},j} - L^{0}_{\mathrm{RNN},j}\|_F \le B_2 \rho_j \eta \ell = O(\ell \, \nu_W^j) \tag{8}$$

where the convergence is in probability.

- Learning the effect at delay j needs exponential number of steps with j
- To learn dependencies at delay j, $\|L^{\ell}_{\mathrm{RNN},j} - L^{0}_{\mathrm{RNN},j}\|$ need to be large

Numerical Experiments

- Synthetic data: true system generated via an RNN with 4 hidden states



- Experiment: Synthetic data with delay: $y_t = x_{t-\text{delay}} + \text{noise}$
  - Low delay: Implicit bias helps $\rightarrow$ lower variance error
  - High delay: Implicity bias hurts $\rightarrow$ cannot capture delay