ANTISYMMETRICRNN:

A dynamical system view on recurrent neural networks

Jacopo Raffi - 598092
j.raffi@studenti.unipi.it

University of Pisa
MsC Computer Science - Curriculum AI
Intelligent Systems for Pattern Recognition
A.Y. 2023/2024

Introduction

- Recurrent Neural Networks (RNN) have gained widespread use in modeling sequential data such as time series;
- RNN have two main problems:
  - Hard to parallelize;
  - Difficult to learn long-term dependencies due to exploding/vanishing gradients.
- Different approaches have been proposed to solve the second issue but they often require more computational effort, e.g:
  - LSTM;
  - GRU.
- The authors propose an ODE (Ordinary Differential Equations) principled approach to learning long-term dependencies that avoids significant computational overhead.

Model Description

To solve the problem of learning long-term dependencies, the authors introduce the AntisymmetricRNN (A-RNN), a recurrent neural network architecture inspired by the theory of ODE.
In particular, the authors start from the ODE **(1)**, then applying the Euler discretization method they obtain equation **(2)**, which can be seen as an RNN without input data and bias term, with the addition of a residual connection.
Thus, in order to study the stability of the discretized version, the authors start by studying the stability of the continuous version (considering also the input data and the bias term). Equation **(3)** is the final discretized version to be stabilized.

$$h'(t) = \tanh(\boldsymbol{W} h(t)) \qquad (1)$$

$$h_t = h_{t-1} + \epsilon \tanh(\boldsymbol{W} h_{t-1}) \qquad (2)$$

$$h(t) = h(t\text{-}1) + \epsilon \tanh(W_1 h(t\text{-}1) + W_2 x(t) + b) \qquad (3)$$

To achieve stability and avoid gradient problems the authors pose $\boldsymbol{W_1 = W - W^T - \gamma I}$, where $\boldsymbol{W - W^T}$ is an antisymmetric matrix and $\boldsymbol{\gamma I}$ is a *diffusion* term.

Mathematical Background

Given the ODE $h'(t) = f(h(t))$:

(1) The solution of an ODE is stable if:

$max_i\ Re(\lambda_i(J)) \leq 0, \quad \forall\ t \geq 0$

J is the Jacobian of f, $Re(\lambda_i(J))$ denotes the real part of the *i-th* eigenvalue of the Jacobian;

(2) Stability alone is not enough to capture long-term dependencies (avoid gradients problems).
If $\forall\ i\ Re(\lambda_i(J)) \approx 0$, the system maintains the long-term dependencies of the inputs while exhibiting stability;

(3) A matrix is antisymmetric if $M = -M^T$.

Model Equation

An ODE solution is stable if the long-term behavior of the system does not depend significantly on the initial conditions. Differentiating equation **(1)** with respect to the initial state h(0) on both sides, we have the following sensitivity analysis, shown in equation **(2)**; where ***A(t) = dh(t) / dh0***.

$$\boldsymbol{h}'(t) = f(\boldsymbol{h}(t)) \quad (1) \qquad\qquad \frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t} = \boldsymbol{J}(t)\boldsymbol{A}(t), \quad \boldsymbol{A}(0) = \boldsymbol{I} \quad (2)$$

The solution of the ODE is $\boldsymbol{A(t) = P\, e^{\Delta(J)*t}\, P^{-1}}$ ($\boldsymbol{\Delta(J)}$ are the eigenvalues of the Jacobian and the columns of ***P*** are the corresponding eigenvectors). When $\boldsymbol{Re(\Delta(J)) \approx 0}$, $\boldsymbol{Re(\Delta(J))}$ denotes the real part of the eigenvalues of the Jacobian, the magnitude of ***A(t)*** is approximately constant in time, so there are no vanishing/exploding gradient. To satisfy this condition the authors exploit an interesting property of antisymmetric matrices; the eigenvalues of antisymmetric matrices are all imaginary (the real part of the eigenvalues is 0).

In the case of RNN, $\mathbf{h'(t) = }\boldsymbol{tanh(W_1h(t-1) + W_2x(t) + b)}$ and the Jacobian of ***f*** is $\mathbf{J(t) = diag[}\boldsymbol{tanh'(W_1h(t-1)+W_2x(t)+b)]W_1}$.

The authors proved that if the eigenvalues of $\boldsymbol{W_1}$ are all imaginary, then the eigenvalues of **J** are all imaginary ($\boldsymbol{Re(\Delta(J)) = 0}$); to achieve this they posed $\boldsymbol{W_1 = W - W^T}$. The use of an antisymmetric matrix ($\boldsymbol{W - W^T}$) allows to avoid gradients problems in the continuous ODE, but not also in the Euler discretized version. Therefore, the authors add *diffusion* (γI) to the equation, as shown in **(3)**, the A-RNN final model equation.

$$h_t = h_{t-1} + \epsilon \tanh\left((W_h - W_h^T - \gamma \boldsymbol{I})h_{t-1} + V_h x_t + b_h\right) \quad (3)$$

Empirical Results

A-RNN (and its gated version) was compared to other approaches on four image classification tasks with long-range dependencies.

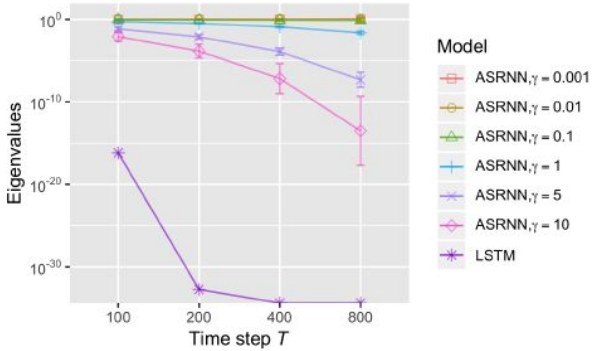
The evaluation results show how the proposed approach outperform LSTM (except in one case where the results are almost comparable) with a significantly lower number of parameters.

Accuracy on MNIST and permuted MNIST (a harder version)

method	MNIST	pMNIST	# units	# params
LSTM (Arjovsky et al., 2016)[1]	97.3%	92.6%	128	68k
FC uRNN (Wisdom et al., 2016)	92.8%	92.1%	116	16k
FC uRNN (Wisdom et al., 2016)	96.9%	94.1%	512	270k
Soft orthogonal (Vorontsov et al., 2017)	94.1%	91.4%	128	18k
KRU (Jose et al., 2017)	96.4%	94.5%	512	11k
AntisymmetricRNN	98.0%	**95.8%**	128	10k
AntisymmetricRNN w/ gating	**98.8%**	93.1%	128	10k

Accuracy on CIFAR-10 and noise padded CIFAR-10 (a harder version)

method	pixel-by-pixel	noise padded	# units	# params
LSTM	59.7%	11.6%	128	69k
Ablation model	54.6%	46.2%	196	42k
AntisymmetricRNN	58.7%	48.3%	256	36k
AntisymmetricRNN w/ gating	**62.2%**	**54.7%**	256	37k



Another interesting result is the eigenvalues of the Jacobian matrix. The plot shows the mean and standard deviation of the eigenvalues of LSTM and A-RNN (with different γ values). LSTM quickly approaches zero, while A-RNN (with lower γ) has a mean close to 1 and standard deviation close to 0, indicating non-exploding and non-vanishing gradients.

Conclusion

- Novelties:
  - Trainability of RNN is presented from the perspective of dynamical systems (ODE);
- Strength:
  - A-RNN demonstrates competitive performance in learning long-term dependencies;
  - It requires significantly fewer parameters compared to popular recurrent models such as LSTM.
- Weaknesses:
  - Still hard to parallelize;
  - Additional hyperparameters:
    - Diffusion coefficient (γ);
    - Euler step size (ϵ).

Thank you for your attention!