UNIVERSITÀ DI PISA

Coupled oscillatory recurrent neural network (coRNN)

An accurate and (gradient) stable architecture for learning long time dependencies.

Rusch, T. Konstantin, and Siddhartha Mishra

Seminar by Francesca Poli [f.poli12@studenti.unipi.it]

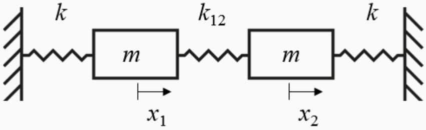
M.Sc. Computer Science - AI curriculum

Computational Neuroscience, a.y. 2023/24

Index

The Coupled Oscillatory Recurrent Neural Network (coRNN) is a novel RNN architecture inspired by the adaptability and efficiency of **biological neural circuits**.

CoRNNs are networks of coupled oscillators, interesting for their ability to express a rich set of outputs while keeping the gradients of state variables bounded $\longrightarrow$ mitigation of the **exploding and vanishing gradient problem**!



Model description

CoRNNs are described through time-discretization of a system of second-order ODEs.

$$y'' = \sigma(Wy + \mathcal{W}y' + Vu + b) - \gamma y - \epsilon y'$$

- $t \in [0, 1]$ continuous time variable
- $u = u(t) \in \mathbb{R}^d$ time-dependent input signal
- $y = y(t) \in \mathbb{R}^m$ hidden state of the RNN
- $W, \mathcal{W} \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{m \times d}$ are weight matrices
- $b \in \mathbb{R}^m$ bias vector
- $0 < \gamma, \epsilon$ oscillation frequency and damping parameters respectively
- $\sigma : \mathbb{R} \mapsto \mathbb{R}$, here set as $\sigma(u) = \tanh(u)$, is the activation function

Model description based on ODEs

Introducing **velocity**: $z = y'(t) \in \mathbb{R}^m$ to rewrite y'' as a first-order system:

CoRNN as first-order ODE

$$y' = z$$

$$z' = \sigma(Wy + \mathcal{W}z + Vu + b) - \gamma y - \epsilon z$$

Considering a fixed timestep $0 < \Delta t < 1$, when $t_n = n\Delta t \in [0, 1]$:

RNN hidden states - IMEX discretization

$$y_n = y_{n-1} + \Delta t z_n$$

$$z_n = z_{n-1} + \Delta t \sigma(W y_{n-1} + \mathcal{W} z_{n-1} + V u_n + b) - \Delta t \gamma y_{n-1} - \Delta t \epsilon z_n$$

Exploding Vanishing Gradient Problem (EVGP)

BPTT and EVGP

BPTT algorithm for training RNNs requires computing products of the Jacobians of the underlying hidden states over **very long time scales** $\longrightarrow$ gradient can grow to infinity or decay to zero exponentially fast with respect to the number of recurrent interactions.

CoRNN approach

The proposed solution is to use **coupling** networks of controlled, non-linear, forced and damped **oscillators**.

- preserves expressivity;
- constrains the dynamics of state variables and their gradients

Exploding gradient management

Mean Squared Error

The loss function $\mathcal{E}$ to minimize is defined by:

$$\mathcal{E} := \frac{1}{N}\sum_{n=1}^{N}\mathcal{E}_n, \qquad \mathcal{E}_n = \frac{1}{2}\|\mathrm{y_n} - \overline{\mathrm{y}}_\mathrm{n}\|_2^2$$

Exploding gradient management

Proposition 1

Assuming that the time step $\Delta t \ll 1$ be such that

$$max \left\{ \frac{\Delta t(1 + \|W\|_\infty)}{1 + \Delta t}), \frac{\Delta t(\|\mathcal{W}\|_\infty)}{1 + \Delta t}) \right\} = \eta \leq \Delta t^r, \quad \frac{1}{2} \leq r \leq 1,$$

the gradient of $\mathcal{E}$ with respect to any parameter $\theta \in \Theta$, is bounded as:

$$|\frac{\partial \mathcal{E}}{\partial \theta}| \leq \frac{3}{2}(m + \overline{Y}\sqrt{m})$$

where $\overline{Y} = \max_{1 \leq n \leq N} \|\overline{y}_n\|_\infty$ is a bound on the underlying training data.

Università di Pisa

Vanishing gradient management

Gradient for long-term dependencies

Considering $X_n = [y_n, z_n]$, the gradient is obtained using the **chain rule**:

$$\frac{\partial \mathcal{E}_n}{\partial \theta} := \sum_{1 \leq k \leq n} \frac{\partial \mathcal{E}_n^{(k)}}{\partial \theta}$$

where

$$\frac{\partial \mathcal{E}_n^{(k)}}{\partial \theta} := \frac{\partial \mathcal{E}_n}{\partial X_n} \frac{\partial X_n}{\partial X_k} \frac{\partial^+ X_k}{\partial \theta}$$

and $\frac{\partial^+ X_k}{\partial \theta}$ is the partial derivative of X_k with respect to θ with the other arguments kept constant.

Vanishing gradient management

Proposition 2

Assuming that $y_i^n = \mathcal{O}(\sqrt{t_n})$ for every $1 \leq i \leq m$ and considering Proposition 1, we have:

$$\frac{\partial \mathcal{E}_n^{(k)}}{\partial \theta} = \mathcal{O}(\widehat{c}\delta\Delta t^{\frac{3}{2}}) + \mathcal{O}(\widehat{c}\delta(1+\delta)\Delta t^{\frac{5}{2}}) + \mathcal{O}(\Delta t^3),$$

where $\widehat{c} = sech^2(\sqrt{k\Delta t}(1 + \Delta t))$, $k \ll n$.

This bound shows that even though the gradient can be small, it is in fact **independent of** k, ensuring that **long-term dependencies contribute to gradients at much later steps**.

UNIVERSITÀ DI PISA

Experiments

Learning tasks

- Adding problem, to test the ability of an RNN to learn (very) long-term dependencies;
- Sequential (permuted) MNIST and Noise padded CIFAR-10 classification benchmarks, with astonishing results;
- Human activity recognition, also with great results;
- IMDB sentiment analysis.

Experiments

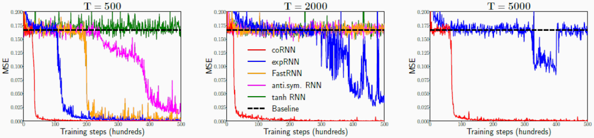


Figure 1: Results of the adding problem for coRNN, expRNN, FastRNN, anti.sym. RNN and tanh RNN based on three different sequence lengths T, i.e. $T = 500$, $T = 2000$ and $T = 5000$.

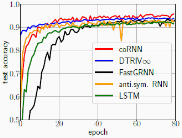


Figure 2: Performance on psMNIST for different models, all with 128 hidden units and the same fixed random permutation.

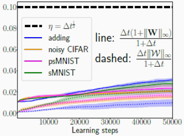


Figure 3: Weight assumptions (8), with $r = \frac{1}{2}$, evaluated during training for all LTD experiments (mean and standard deviation of 10 different runs for each task).

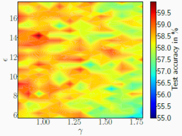


Figure 4: Ablation study on the hyperparameters ϵ, γ in (3) using the noise padded CIFAR-10 experiment.

Università di Pisa

Universality of Neural Oscillators

Universality of Neural Oscillators is a gateway

The Universality theorem for neural oscillators sets their potential to approximate continuous operators between appropriate function spaces. Proving this, we can:

- Establish firm mathematical foundation for the deployment of oscillator-based NNs, such as CoRNN and UnicoRNN, pushing a more widespread use;
- Show how networks of oscillators can approximate a large class of mappings, a non-trivial feature competing with traditional NNs.

General Form

Given $u : [0, T] \to \mathbb{R}^p$ as an input signal, for every final time $T \in \mathbb{R}_+$ consider the following system of **neural ODEs** for the evolution of dynamic hidden variables $y \in \mathbb{R}^m$, coupled to a linear readout to obtain the output $z \in \mathbb{R}^q$.

$$\begin{cases} \ddot{y}(t) = \sigma(Wy(t) + Vu(t) + b), \\ y(0) = \dot{y}(0) = 0, \\ z(t) = Ay(t) + c. \end{cases} \tag{1}$$

Universal approximation Theorem

Theorem

Let $\Phi : K \subset C_0([0,T];\mathbb{R}^p) \rightarrow C_0([0,T];\mathbb{R}^q)$ be a causal and continuous operator.
Let $K \subset C_0([0,T];R^p)$ be compact.

Then for any $\epsilon > 0$, there exist hyperparameters $L, m_1, \dots, m_L$, weights $w^l \in \mathbb{R}^{m_l}$, $V_l \in \mathbb{R}^{m_l \times m_{l-1}}$, $A \in \mathbb{R}^{q \times m_L}$ and bias vectors $b_l \in \mathbb{R}^{m_l}$, $c \in \mathbb{R}^q$, for $l = 1, \dots, L$, such that the output $z : [0,T] \rightarrow \mathbb{R}^q$ of a multi-layer neural oscillator satisfies

$$\sup_{t \in [0,T]} |\Phi(u)(t) - z(t)| \leq \epsilon, \quad \forall u \in K$$

Università di Pisa

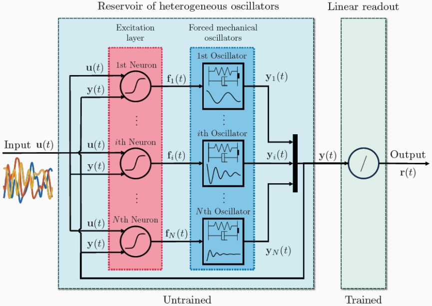
An application: Random Oscillators Network (RON)

Architecture description

RON is a discrete-time RNN model whose update reads:

$$y_{k+1} = y_k + \tau z_{k+1},$$

$$z_{k+1} = z_k + \tau(\tanh(Wy_k + Vu_{k+1} + b) - \gamma \odot y_k - \epsilon \odot z_k).$$



Architecture description

Layers

The Random Oscillators Network consists of:

- N harmonic oscillators forced by coupled neurons with tanh activations.
- A linear output layer that maps the states of the mechanical oscillators in the desired output.
- for **time series tasks**, a stacked linear layer transforming the hidden state y to an output state r

$$r_{k+1} = W_o y_{k+1} + b_o$$

Computational efficiency of oscillatory-based archetypes

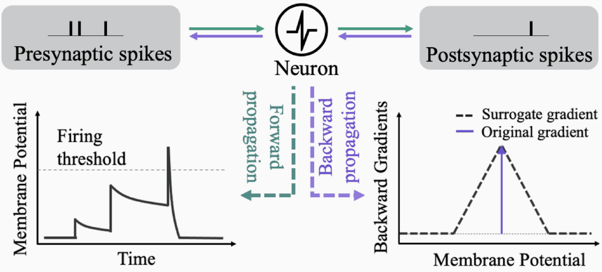
RON vs CoRNN

- coRNN model is fully trained, while RON does not need BPTT thanks to the reservoir layer $\rightarrow$ efficiency in time and energy consumption;
- oscillatory-based recurrent models require more hyperparameters tuning, and coRNN is very sensitive to parameter selection;
- CoRNN does not use heterogeneous oscillators;
- CoRNN requires an additional hidden-to-hidden adaptive matrix W.

Oscillatory-based NNs and Spiking Neural Networks as archetypes

Spiking Neural Networks: NN archetypes based on neurons that can either fire or not based on binary inputs.

$\longrightarrow$ Spiking behavior could be modulated by the oscillatory dynamics!



Oscillatory-based NNs and Spiking Neural Networks as archetypes

Some ideas

- **Spike-Timing Dependent Plasticity (STDP)** adjusts synaptic weights based on the timing of pre-synaptic and post-synaptic spikes.
  - Modulating the input current to the spiking neurons based on the oscillatory phase.
  - Adjusting the synaptic weights based on the timing of spikes to reinforce the desired oscillatory patterns.
- **Hebbian learning** principles can be applied to promote synchronization and oscillatory coupling.

Conclusions

Possible future developments

- Exploit oscillatory models' potential in stability, robustness and expressivity to enhance their performances and introduce a bias towards oscillatory-based NNs;
- Study and develop a greater variety of possible biological neurons-inspired RNNs to explore a bigger area of application, such as Neuromorphic Computing;
- Including Deep Learning theories into oscillatory RNNs, as RON does with Reservoir Computing principles;
- Integrating oscillatory-based models with other archetypes such as Spiking Neural Networks.

Possible future developments

- Exploit oscillatory models' potential in stability, robustness and expressivity to enhance their performances and introduce a bias towards oscillatory-based NNs;
- Study and develop a greater variety of possible biological neurons-inspired RNNs to explore a bigger area of application, such as Neuromorphic Computing;
- Including Deep Learning theories into oscillatory RNNs, as RON does with Reservoir Computing principles;
- Integrating oscillatory-based models with other archetypes such as Spiking Neural Networks.

Possible future developments

- Exploit oscillatory models' potential in stability, robustness and expressivity to enhance their performances and introduce a bias towards oscillatory-based NNs;
- Study and develop a greater variety of possible biological neurons-inspired RNNs to explore a bigger area of application, such as Neuromorphic Computing;
- Including Deep Learning theories into oscillatory RNNs, as RON does with Reservoir Computing principles;
- Integrating oscillatory-based models with other archetypes such as Spiking Neural Networks.

Possible future developments

- Exploit oscillatory models' potential in stability, robustness and expressivity to enhance their performances and introduce a bias towards oscillatory-based NNs;
- Study and develop a greater variety of possible biological neurons-inspired RNNs to explore a bigger area of application, such as Neuromorphic Computing;
- Including Deep Learning theories into oscillatory RNNs, as RON does with Reservoir Computing principles;
- Integrating oscillatory-based models with other archetypes such as Spiking Neural Networks.

Bibliography

- **Rusch, T. Konstantin, and Siddhartha Mishra. (2020) *Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies.* arXiv preprint arXiv:2010.00951**
- Ceni, Andrea, et al. (2024) *Random Oscillators Network for Time Series Processing.*, International Conference on Artificial Intelligence and Statistics. PMLR, 2024.
- Keller, T. Anderson, and Max Welling. (2023) *Neural wave machines: learning spatiotemporally structured representations with locally coupled oscillatory recurrent neural networks.* International Conference on Machine Learning. PMLR, 2023.
- Lanthaler, Samuel, T. Konstantin Rusch, and Siddhartha Mishra. (2024) *Neural oscillators are universal.*, Advances in Neural Information Processing Systems 36.

Bibliography

- Rusch, T. Konstantin, and Siddhartha Mishra. (2021) *Unicornn: A recurrent model for learning very long time dependencies.*, International Conference on Machine Learning. PMLR, 2021.
- De Geeter, Florent and Ernst, Damien and Drion, Guillaume. (2024) *Spike-based computation using classical recurrent neural networks.*, arXiv preprint arXiv:2306.03623.