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

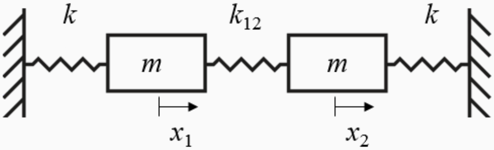
UNIVERSITÀ DI PISA

Introduction to the subject

The Coupled Oscillatory Recurrent Neural Network (coRNN) is a novel RNN architecture inspired by the ability of **biological neural circuits** to express a rich set of outputs while keeping the gradients of state variables bounded
$\longrightarrow$ mitigation of the **exploding and vanishing gradient problem**!

CoRNN characteristics

- time-discretization of a system of second-order ordinary differential equations $\longrightarrow$ modeling networks of controlled nonlinear oscillators
- performs comparably to the state-of-the-art on a variety of benchmarks



Model description based on 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 $0 < \Delta < 1$, when $t_n = n\Delta t \in [0, 1]$:

RNN hidden states

IMEX discretization of the first-order system:

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

Model description based on ODEs

Considering the following initial conditions:

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

RNN hidden states

IMEX discretization of the first-order system:

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

Exploding Vanishing Gradient Problem (EVGP)

BPTT and EVGP

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

CoRNN approach

Taking inspiration from **biological neurons**, the proposed solution is **coupling** networks of controlled non-linear forced and damped **oscillators**.

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

Bounds on hidden states

Energy bounds

Let y_n, z_n be the hidden states of an RNN for $1 \leq n \leq N$.

y_n and z_n satisfy the following energy bounds:

$$\mathrm{y}_n^\top y_n + \mathrm{z}_n^\top z_n \leq nm\Delta t = mt_n \leq m.$$

This bound rules out chaotic behavior of hidden states.

UNIVERSITÀ DI PISA

Bounds on hidden state gradients

Proposition 1 - setting

Let y_n, z_n be the hidden states generated by an RNN.

We assume that the time step $\Delta t \ll 1$ can be chosen such that:

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

Denoting $\delta = \frac{1}{1+\Delta t}$ as the gradient of the loss function $\mathcal{E}$:

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

Exploding gradient management

Proposition 1

With respect to any parameter $\theta \in \Theta$, δ is bounded as:

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

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

As the entire gradient of the loss function $\mathcal{E}$ is bounded with respect to the weights and biases of the network, **the exploding gradient problem is mitigated for the considered RNN.**

Vanishing gradient management

Proposition 2

Let y_n be the hidden states generated by the RNN. Assuming that $y_i^n = \mathcal{O}(\sqrt{t_n})$ for every $1 \leq i \leq m$ and considering Proposition 1, the gradient for long-term dependencies satisfies:

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

This bound shows that though $\mathcal{O}(\Delta t^{\frac{3}{2}})$ can be small, it is **independent of** k, $\rightarrow$ **long-term dependencies contribute to gradients at much later steps** and **mitigating the EVG.**

Experiments

CoRNN has been tested for the following tasks:

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.

The tests did **non** exploit additional performance enhancing tools!

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.

Neural Oscillators

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 $z \in \mathbb{R}^q$.

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

Università di Pisa

Proof of universality

Fundamental Lemma

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

For every $\epsilon > 0$ there exist:

- $N \in \mathbb{N}$
- frequencies $\omega_1, ...\omega_N$
- continuous mapping $\Psi : \mathbb{R}^{p\times N} \times [0, T^2/4] \to \mathbb{R}^q$

such that: $|\Phi(u)(t) - \Psi(\mathcal{L}_t u(\omega_1), \dots, \mathcal{L}_t u(\omega_N); t^2/4)| \leq \epsilon$

for every $u \in K$ and $t \in [0,T]$.

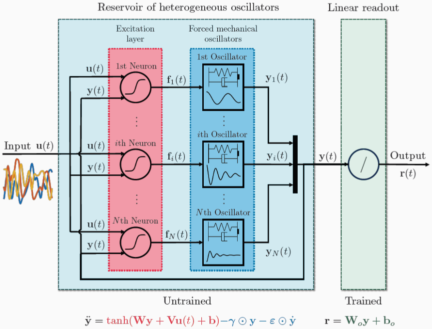
Università di Pisa

An application: Random Oscillators Network (RON)

Architecture description

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

$$
\begin{aligned}
y_{k+1} &= y_k + \tau z_{k+1}, \\
z_{k+1} &= z_k + \tau(\tanh(W y_k + V u_{k+1} + b) - \gamma \odot y_k - \epsilon \odot z_k).
\end{aligned}
$$



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:

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

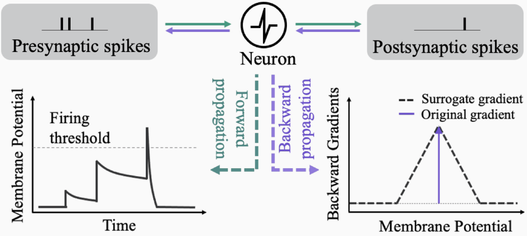
Computational efficiency of oscillatory-based archetypes

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. $\hookrightarrow$ 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.