Auto-Rotating Neural Networks

Daniel Saromo Mori[1,2], Edwin Villanueva Talavera[2], and Matias Valdenegro-Toro[1]

[1] German Research Center for Artificial Intelligence (DFKI), [2] Pontifical Catholic University of Peru (PUCP)

Asian Conference on Machine Learning

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

PUCP

Objective: Addressing the VGP

This work proposes an improved type of neural network designed to mitigate the **vanishing gradient problem (VGP)**. This nuisance appears when training deep artificial neural networks with bounded activation functions. This new design, named **Auto-Rotating Neural Networks (ARNN)**, has a mechanism to ensure that the node always operates in the **dynamic region of the activation function** and thus avoids perceptron (and layer) saturation. The proposed method, derived from the Auto-Rotating Perceptrons (ARP), **does not change the inference structure learned** at each layer. We tested the effect of using ARNN units in network architectures that operate with the most popular activation functions: sigmoid, ReLU, tanh, arctan, and leaky ReLU. The results support our hypothesis that neural networks with ARNN layers **can achieve better learning performance** than equivalent models with classic layers.

A new type of neural networks

- We have implemented well-known Artificial Neural Networks (ANN) types using the Auto-Rotation.
- We extrapolated the Auto-Rotating operation [1] from dense to convolutional and recurrent layers (see Figure 1). Thus, we created the ARNNs.

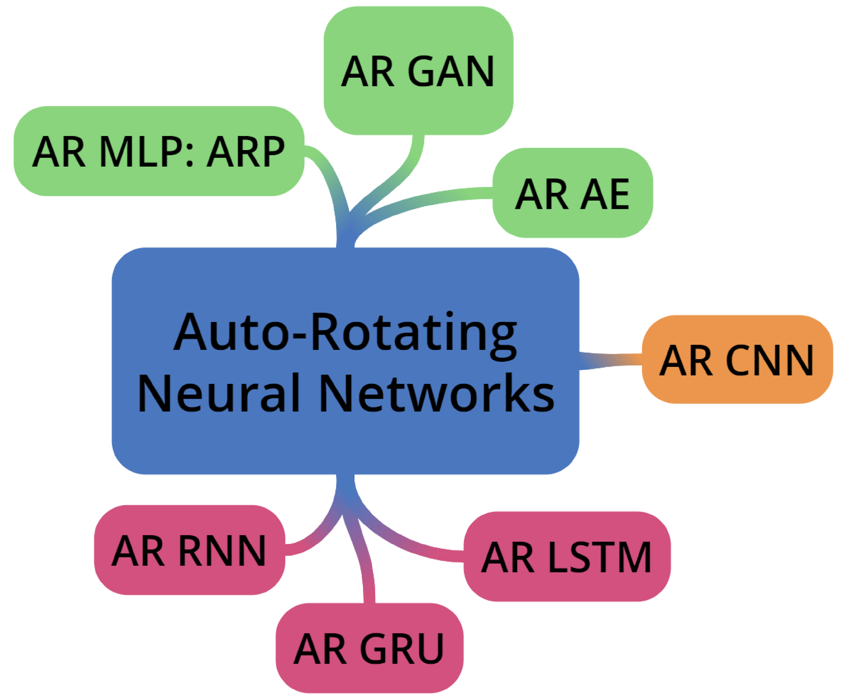
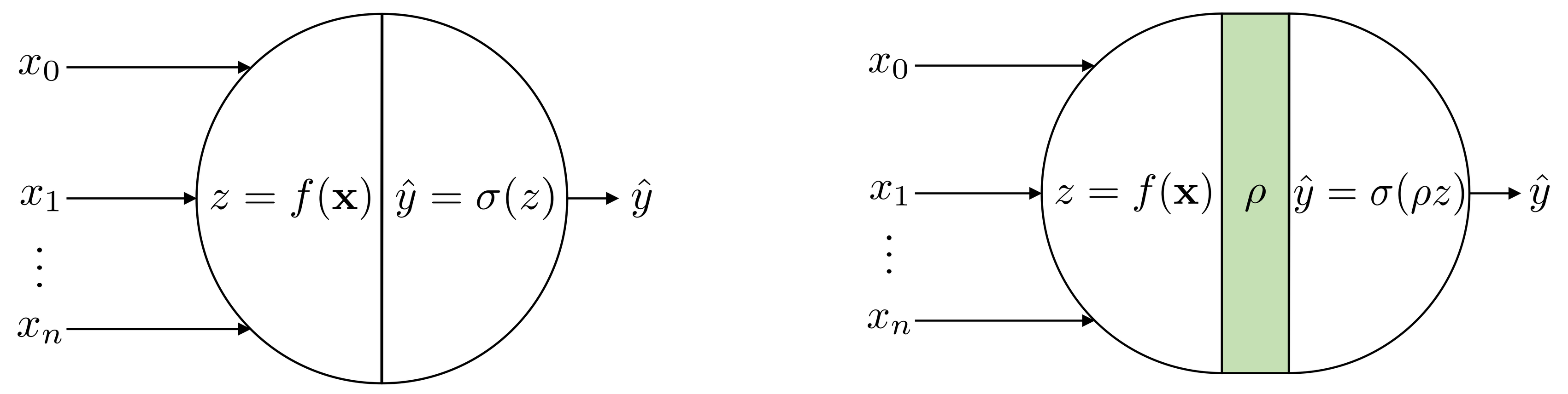


Figure 1: Some aplications of the ARNNs.

- ARNN = ANN + AR.

Background: What is an Auto-Rotating Perceptron (ARP)?

- The ARP, proposed by Saromo et al. [1], is an innovative neural unit that aims to avoid the vanishing gradient problem (VGP) by making the z inputs of the perceptron activation $\sigma(z)$ near zero with no learning alteration.
- The modification is achieved by multiplying the linear transformation $f(\mathbf{x})$ with an scalar coefficient ρ, before the activation function $\sigma(z)$. The ARP has two hyperparameters: $\mathbf{x}_Q = \langle x_Q, \cdots, x_Q \rangle \in \mathbb{R}^n$ and $L \in \mathbb{R}$.



$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0\, x_0 \qquad x_0 = 1 \qquad\qquad g(\mathbf{x}) = \rho\, f(\mathbf{x}) \qquad \rho = \frac{L}{|f(\mathbf{x}_Q)|}$

Figure 2: Classic perceptron (left) and ARP (right).

Experimental results

- Two identical neural networks. Only one difference: One has classic perceptrons and the other has ARP.
- Architecture: 2, 50, **50**, 50, 1. || $L = 6$. The last ARNN version automatically calculates the hyperparameter x_Q.

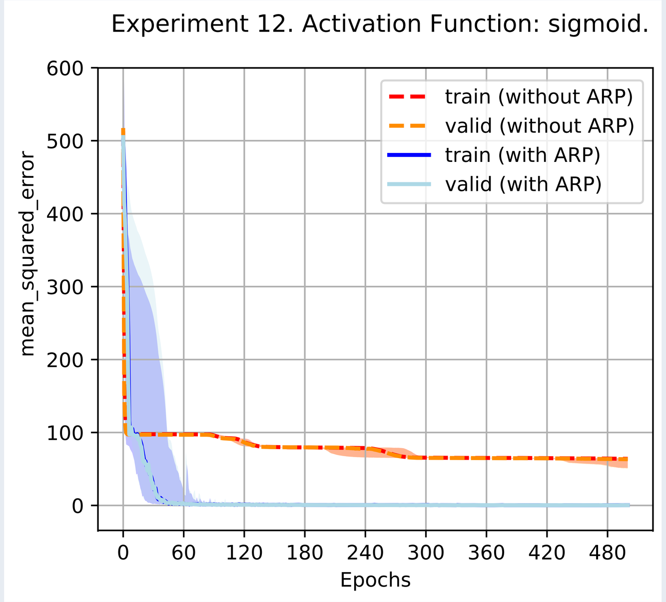


Figure 3: Loss curves of 30 network pairs. Dataset: Rastrigin.

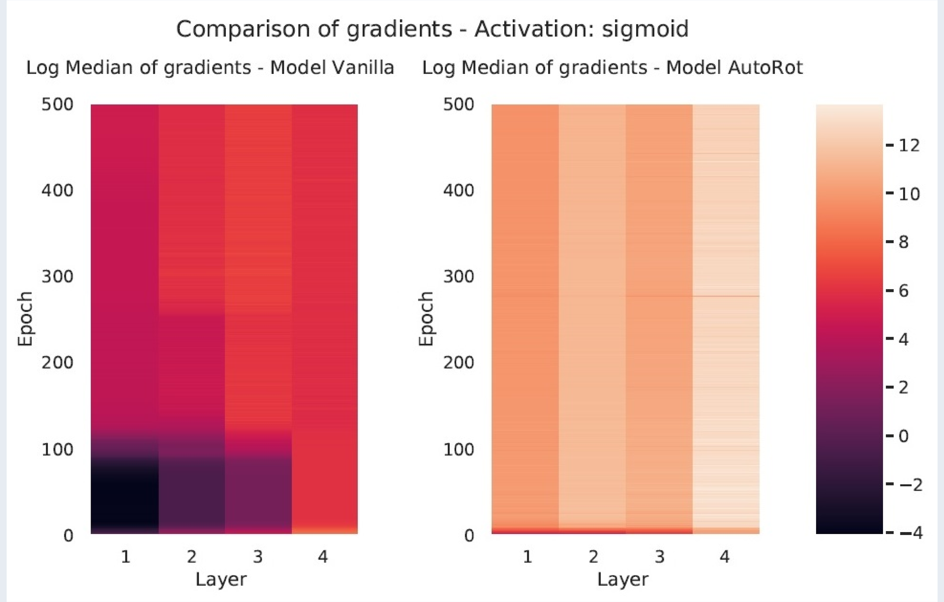


Figure 4: Gradients of the models' weights at **this** layer.

Working principle behind the ARNNs: The Auto-Rotation (AR)

1) Dynamic region of the neurons

- If $\sigma'(z) \approx 0 \rightarrow$ Unwanted node saturation: VGP.

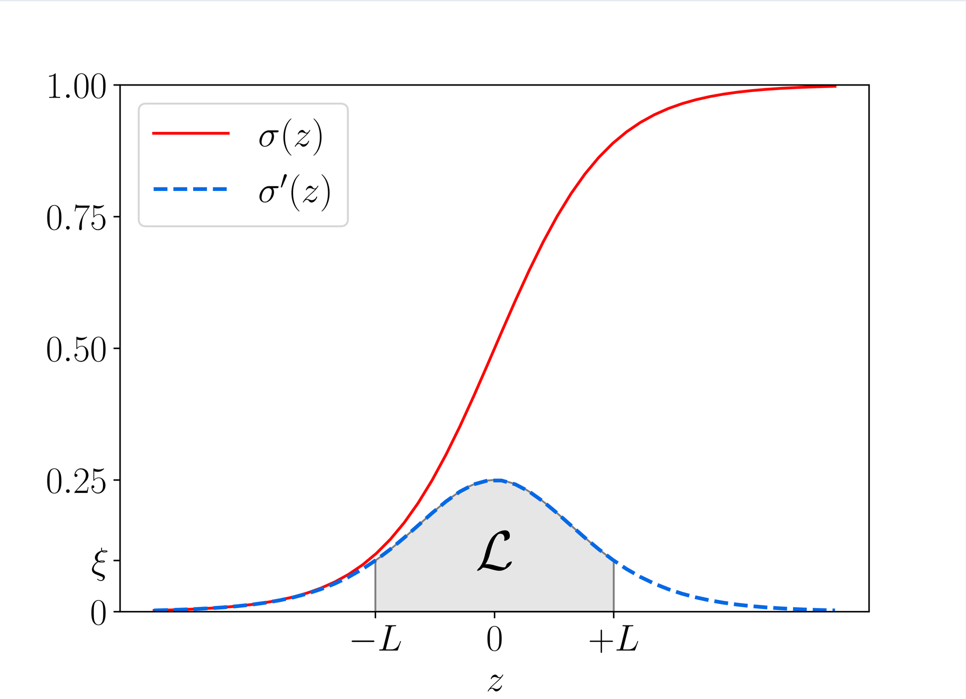


Figure 5: Sigmoid activation function $\sigma(z)$ and its derivative (solid red line and dashed blue line, respectively). The shaded gray area is the projection of the dynamic region $\mathcal{L}(L)$ on the curve $\sigma'(z)$.

- Perceptrons need to operate in their dynamic region $\mathcal{L}$.
- How can we control it? Is it possible to achieve that?
- Do we need to change $\sigma(z)$ to avoid VGP?

2) Pre-activation phase

- New *feature axis* Z augments the input space.
- Boundary Γ holds the neuron's inference structure.
- $\varphi := \langle \mathbf{x}, f(\mathbf{x}) \rangle \subset \mathbb{R}^{n+1}$ and $\Gamma \subset \mathbb{R}^n \mid \Gamma := \langle \mathbf{x}, 0 \rangle \cap \varphi$.
- φ not unique: that rotational DOF can be exploited.
- Z is unbounded. We need to avoid node saturation.
- ARP wisely chooses $\hat{\varphi}$ and preserves Γ.

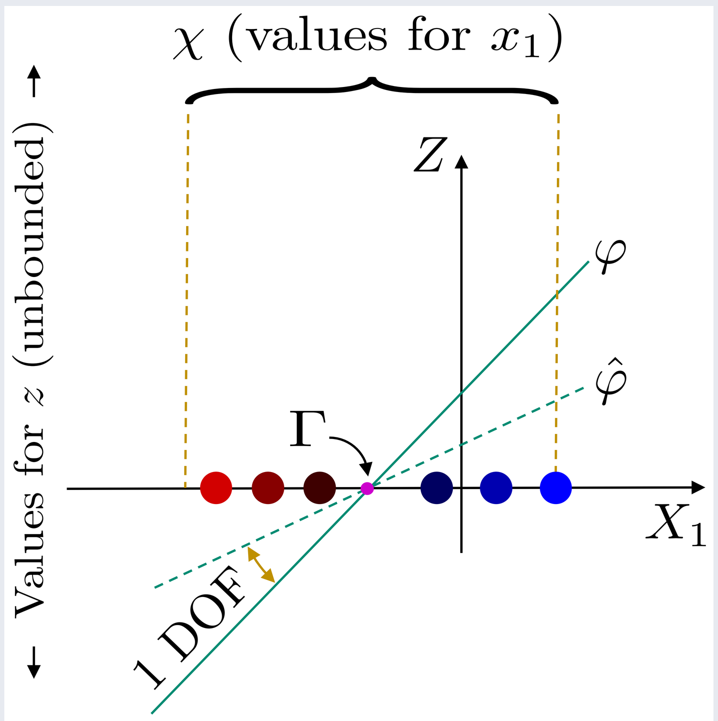


Figure 6: DOF at classic perceptrons with 1D inputs.

3) Controlling the rotation

- **Hyperparameters:** $L \in \mathbb{R}$ and $\mathbf{x}_Q \in \mathbb{R}^n$.
- **Conditions:** $\langle \mathbf{x}_Q, L \rangle \in \hat{\varphi}$ and $\Gamma \subset \hat{\varphi}$.
- **Green region:** all possible positions for $\hat{\varphi} \supset \Gamma$.
- **In reality:** $z \in [L_1, L_2]$. **Consider:** $|z| \leq L$.
- The rotation depends on $\rho := \rho(L, \mathbf{x}_Q)$.
- **Result:** Limit the z values that will enter $\sigma(z)$.
- Formulation extrapolated to the n-dimensional case.

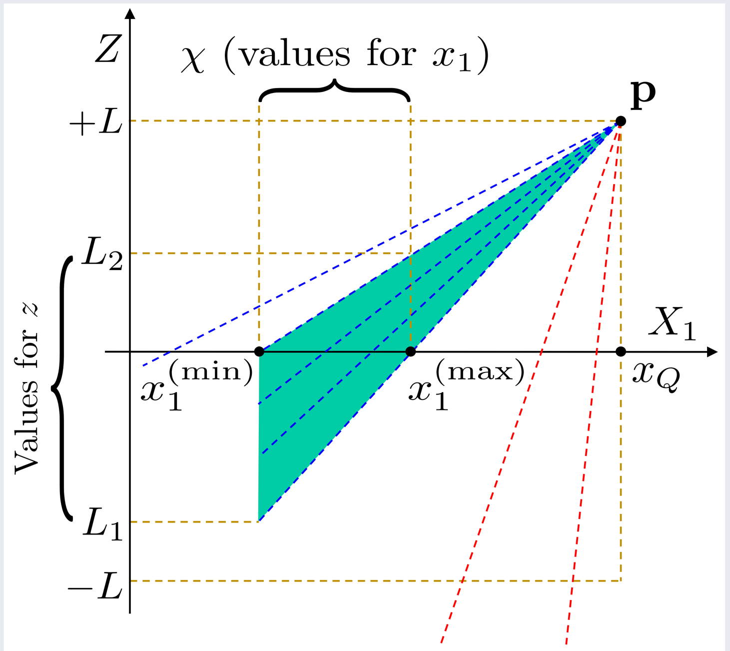


Figure 7: Rotation bounds z to the desired dynamic region.

Key takeaways

- The Auto-Rotation [1] is a mathematical non-linear operation that changes the perceptron's internal core and can boost its inference capabilities.
- There is evidence that if we change the perceptrons of sigmoid-based regression networks to ARP, the **test loss is reduced by a factor of 15** at the cost of increasing the execution time by ~12% [2].
- **Main contribution:** The proposed principle allowed us to create a new neural network type that can be used wherever ANNs are currently applied, and potentially improve their performance.
- Last ARNN version just needs L as hyperparameter.
- ARP Library: www.github.com/DanielSaromo/ARP.

References

[1] Saromo, D., Villota, E., and Villanueva, E. "Auto-Rotating Perceptrons," *LatinX in AI Research Workshop at NeurIPS 2019*. Vancouver, Canada. 2019. arXiv: https://arxiv.org/abs/1910.02483.

[2] Saromo, D., Bravo, L., and Villota, E. "Smart Sensor Calibration with Auto-Rotating Perceptrons," *LatinX in AI Research Workshop at ICML 2020*. Vienna, Austria. 2020. Hover link: available on ResearchGate.