Auto-Rotating Neural Networks

Daniel Saromo-Mori[1] and Matias Valdenegro-Toro[2]

[1] Czech Technical University in Prague (CTU) and [2] University of Groningen (RUG)

dnyAI

CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE

university of groningen

Objective: Addressing the VGP

Neural networks with saturating activations are often not used due to vanishing gradients. This problem is frequently tackled by modifying the activation function or applying Batch Normalization (BN) techniques, but we propose to use a different approach: the Auto-Rotation (AR). An existing AR-based method is the Auto-Rotating Perceptron (ARP), which generalizes Rosenblatt's Perceptron and alleviates vanishing gradients by limiting the pre-activation to a region where the neurons do not saturate. However, this method is only defined for dense layers and requires additional hyperparameter tuning. In this poster, we present an extension of the ARP concept: the Auto-Rotating Neural Networks (ARNN). With them, we have convolutional layers and learnable pre-activation saturation regions. In all our experiments, we got that the AR outperforms the BN approach in terms of preventing vanishing gradients. The results support our hypothesis that by changing from classical to AR layers, the nets can achieve better learning performance.

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.

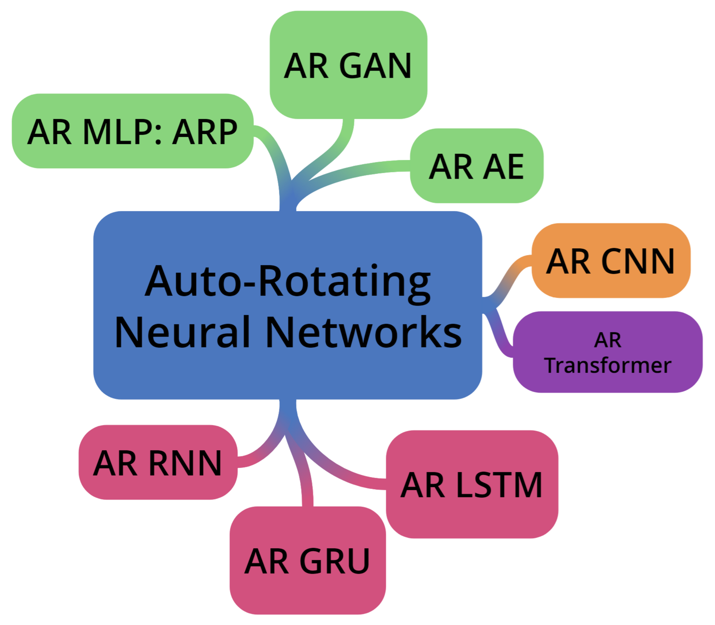


Figure 1: Some aplications of the ARNNs.

- ARNN = ANN + AR.
- The AR operation is a generalization of the dot product.

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 keeping the z inputs of the perceptron activation $\sigma(z)$ near zero, but with no learning alteration.
- The modification is achieved by multiplying the linear transformation $f(\mathbf{x})$ with a scalar coefficient ρ, before going through 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 $L \in \mathbb{R}$.

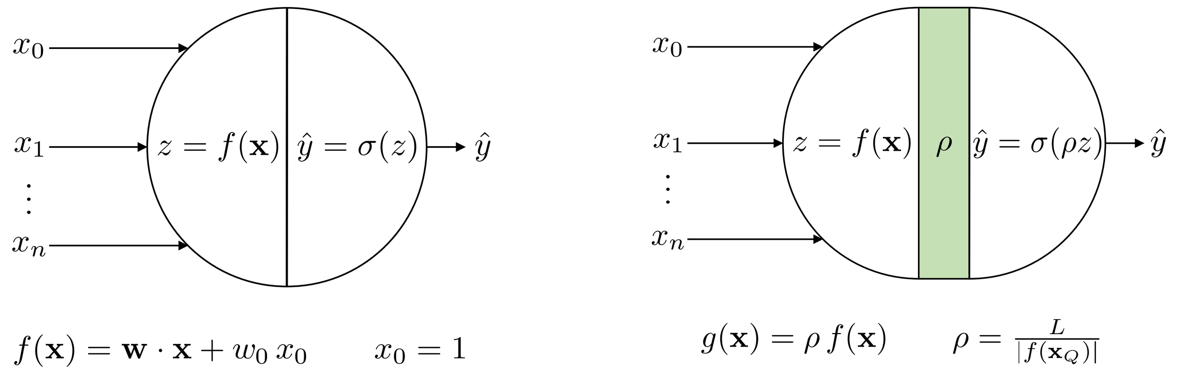
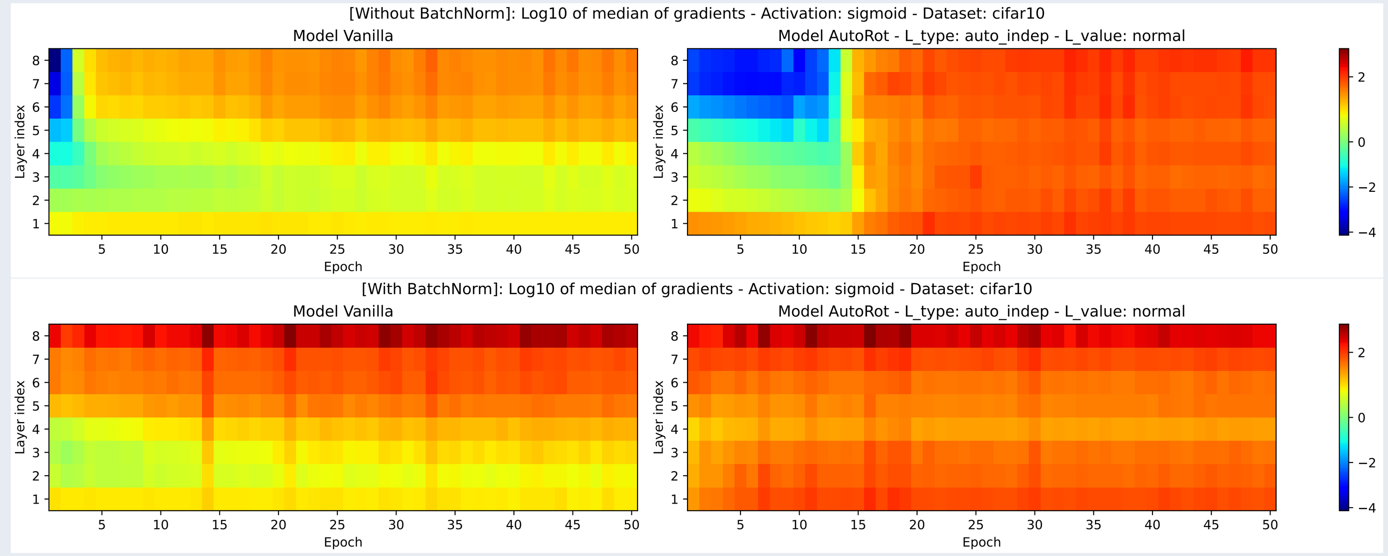
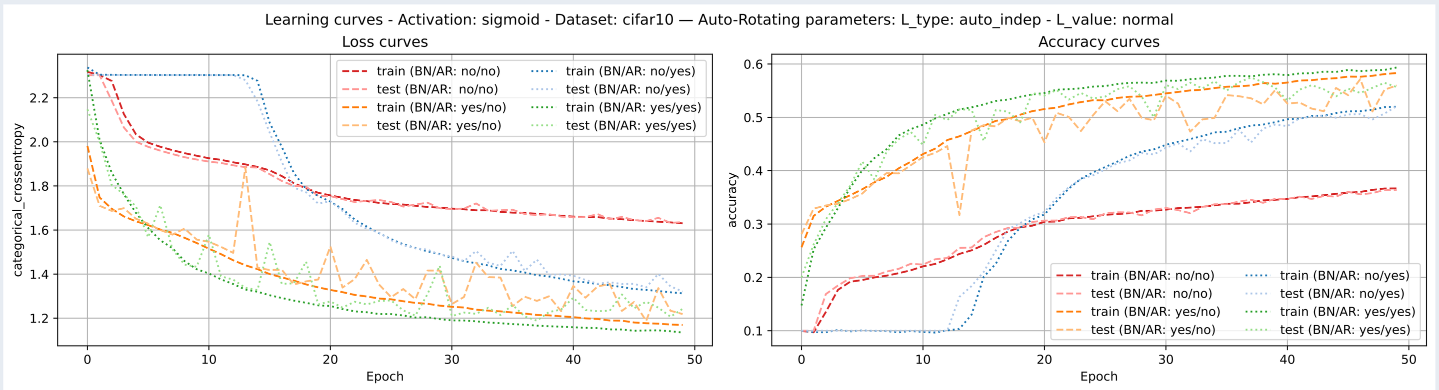


Figure 2: Classic perceptron (left) and ARP (right).

Experimental results

- We trained four neural nets with the same initial weights, but with four different settings (with/without BN, and traditionnal/ARNN aproach). *Architecture:* (Conv2D, BN, MaxPool2D) × 3 + Flatten + (Dense layer of size 10) × 5. At conv. layers: input size preserved and zero padding used.
- *On heatmaps:* Lower layer index → Layer is closer to the input. We see that with BN and AR, the VGP diminishes. Also, with AR we got much more uniform and stable gradients (across the layers and epochs, respectively). Remark: the best results were obtained when using the AR along with BN.





- For ReLU, see Figure 2 of paper [2]: ARNN models are initially outperformed by the classical nets. On later epochs, their performances are similar.

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.

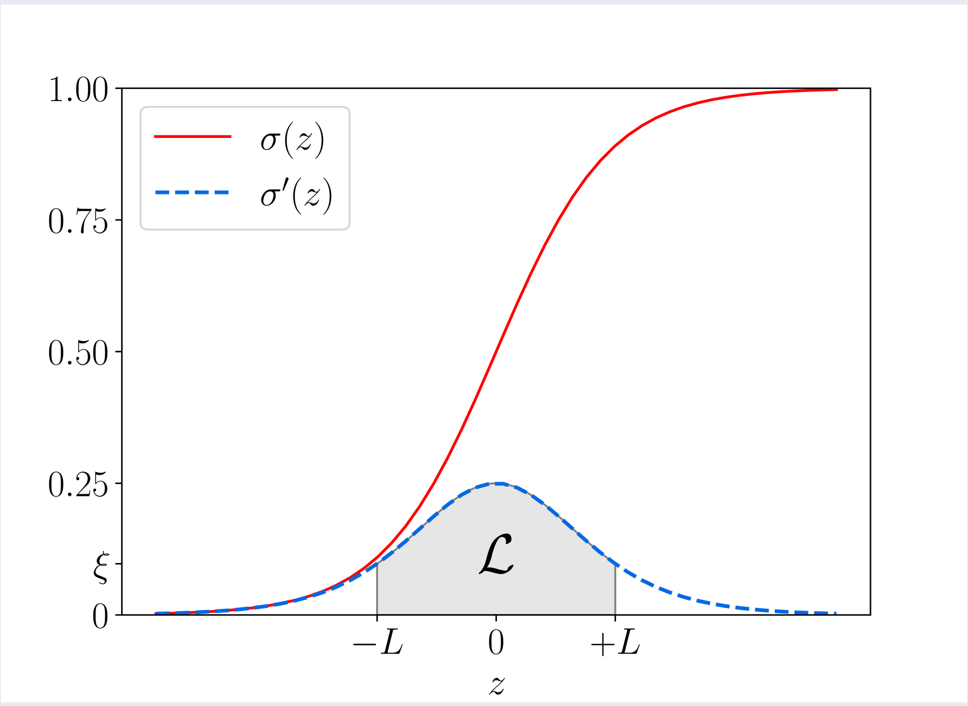


Figure 3: Sigmoid activation function $\sigma(z)$ and its derivative (solid red line and dashed blue line, respectively). The shaded gray area is the projection of the dynamic region $\mathcal{L}(L)$ on the curve $\sigma'(z)$.

- Perceptrons need to operate in their dynamic region $\mathcal{L}$.
- How can we do it? Only by changing σ we avoid VGP?
- We propose to not touch the activation but to change the nature of the pre-activation process.

2) Pre-activation phase

- New *feature axis* Z augments the input space $\mathbb{R}^n$.
- Boundary Γ holds the neuron's inference structure.
- $\varphi := \langle \mathbf{x}, f(\mathbf{x}) \rangle \subset \mathbb{R}^{n+1}$ and $\Gamma \subset \mathbb{R}^n \,|\, \Gamma := \langle \mathbf{x}, 0 \rangle \cap \varphi$.
- 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.

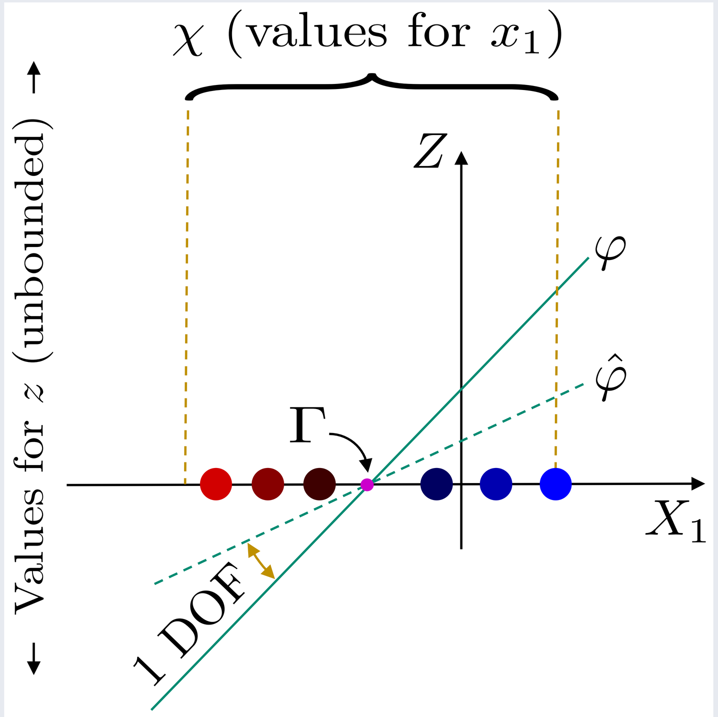


Figure 4: DOF at classic perceptrons with 1D inputs.

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.

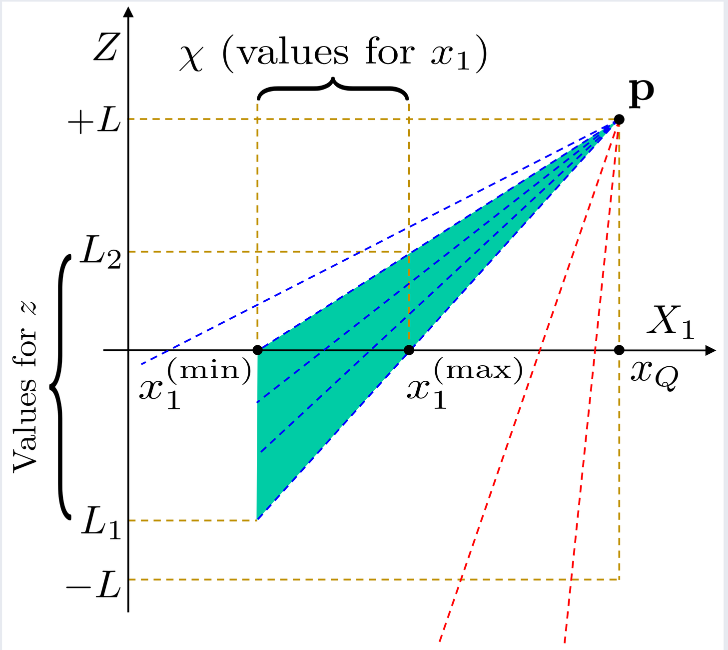


Figure 5: Rotation bounds z to the desired dynamic region.

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. It is useful to use AR with BN.
- Lastest ARNN version does not have hyperparameters.
- Keras Library: www.github.com/DanielSaromo/ARP.

References

[1] Saromo, D., Villota, E., and Villanueva, E. "Auto-Rotating Perceptrons," LXAI Workshop at NeurIPS 2019. Vancouver, Canada. Paper: `https://shorturl.at/2HQ9X`. Video: `https://shorturl.at/sET2A`.

[2] Saromo, D., Bravo, L., and Villota, E. "Smart Sensor Calibration with Auto-Rotating Perceptrons," LXAI Workshop at ICML 2020. Vienna, Austria. Paper: `https://shorturl.at/fU8id`. Video: `https://shorturl.at/jfOj7`.