Smart Sensor Calibration with Auto-Rotating Perceptrons

Daniel Saromo [1 2] **Leonardo Bravo** [2] **Elizabeth Villota** [2]

Abstract

Sensor calibration is vital to have valid measurements of physical activities. Herein we deal with adjusting the signal from a wearable force sensor against a reference scale. By using a few samples and data augmentation, we trained a neural-based regression model to correct the wearable output. For this task, we tested the novel Auto-Rotating Perceptrons (ARP). We found that a neural ARP model with sigmoid activations can outperform an identical neural network based on classic perceptrons with sigmoid and even ReLU activation.

1. Introduction

In recent years, wearable technology has seen an increase in sports applications. Unlike conventional motion capture systems, wearables can gather data outside the laboratory environments (Adesida et al., 2019). Sensing of ground reaction forces (GRF) in particular is of great interest for performance evaluation in sports (Evans et al., 2018). Wearable technology for GRF sensing mainly consists of insole-type wearables that measure plantar pressure distribution (Nagamune & Yamada, 2018). Calibration of these devices has been performed with classical statistical approaches (Eguchi & Takahashi, 2018) and also using neural networks (Eng et al., 2018). For instance, an LMBP-ANN model served to predict validated load signals from a nonlinear pressure sensor (Almassri et al., 2018), a WNN model predicted GRFs using plantar information from insole pressure sensors (Sim et al., 2015), and an LSTM model predicted the strain state from a stretch sensor (Oldfrey et al., 2019).

This paper presents the calibration of an insole-type GRF sensor using the recently developed Auto-Rotating Perceptrons (ARP) for artificial neural networks. We aim to implement and train a feedforward network model in order to have a valid posterior measurement without requiring reference signals.

[1]PUCP Artificial Intelligence Research Group [2]PUCP Biomechanics and Applied Robotics Research Laboratory. Correspondence to: Daniel Saromo <daniel.saromo@pucp.pe>, Leonardo Bravo <leonardo.bravot@pucp.pe>, Elizabeth Villota <evillota@pucp.edu.pe>.

LXAI Workshop at 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

2. Problem definition

A WEarable VErtical GRF Sensor system (WEVES) composed of an insole with piezoelectric sensors and an electronic housing -to be attached to the leg- was developed, see Figure 1(a). WEVES was designed to measure forces up to 2600 N. A 65-kg male amateur athlete wore WEVES and performed a standardized movement on an AMTI force plate (OPT400600 model), see Figure 1(b). AMTI and WEVES signals were recorded for 30 s, which we show in Figure 1(c). Even though AMTI and WEVES signals are quite similar in shape, there exists a difference in scale, and this is recurrent in all sample pairs recorded. Let K be the scaling factor between AMTI and WEVES measurements. Then, WEVES calibration consists in finding that factor, which varies across the signal pairs.

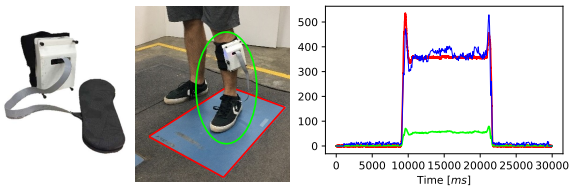

Figure 1. (a) WEVES mechatronic system: electronic box and insole. (b) Stand-up straight position test with AMTI (red) and WEVES (green) sensors. (c) Force measurements (N): AMTI (red), uncalibrated WEVES (green), and scaled WEVES (blue).

3. Methodology

WEVES calibration was performed employing a neural regression model that predicts the K factor. We compared four model types with the same architecture, half of them had classic perceptrons, and the others used the novel ARP units (Saromo et al., 2019). ARP and classical perceptrons were tested for both sigmoid and ReLU activation functions. Previous ARP implementations have reported that this neural unit can lead to better performance in image classification tasks, provided that the sigmoid is the network activation function (Saromo et al., 2019).

Dataset preparation: In order to find the target K, we need to quantify the similarity between AMTI platform signal and filtered WEVES outputs. These signals are represented by the vectors $\mathbf{p}$ and $\mathbf{w}$, respectively. We define the root mean square error (RMSE) as the loss function, so that finding the K factor can be posed as an optimality problem:

$$K = \arg\min_{K} \{\text{RMSE}(\mathbf{p},\, K\mathbf{w})\},\ K > 0.$$

Several algorithms find the best solution for such an optimization problem (Karaboga, 2005; Wang et al., 2018; Kumari & Kaur, 2019). We have implemented the Artificial Bee Colony (ABC) and the Particle Swarm Optimization (PSO) algorithms to obtain K values for each signal pair $(\mathbf{p}, \mathbf{w})$. The K factor chosen corresponds to the lowest cost function value found. Descriptive statistics and information related to these experiments are shown in Table A1.

WEVES signals were obtained at 100 Hz and then interpolated to reach 1 MHz, which is the frequency of the AMTI output data. We generated the K values and then downsampled both signals by a factor of 15. Hence, we reduced the input data dimension to 2000. We then filtered WEVES signals by using a 4th-order low-pass Butterworth filter with a normalized cutoff frequency of 0.15, as implemented in literature for these signal types (Pandit et al., 2018). Data was then circularly shifted, retaining important information -the signal's middle part, see Figure 1(c). By this artifice, the number of pair samples increased from 12 to 8787. Resulting signals were further scaled to the range $[0, 1]$. The final dataset was divided into three groups: 7029 for training, 879 for validation, and the remaining 829 for testing.

Architecture definition: Aiming for a real-time implementation, we chose a feedforward neural network with only one hidden layer and the following architecture: 2000-256-1. The scalar output is the scaling factor K that the regression neural network needs to predict. We assigned the same initial weights and biases to the four network types before each execution. For the experiments, the ARP hyperparameters are defined as: $x_Q = 2.6$, since the maximum input value is $1 < x_Q$; and $L = 3$, because the unipolar sigmoid derivative $\sigma'(z)$ is not very small for $|z| \leq 3$.

4. Experiments: Classic perceptrons vs ARP

Figure 2 shows the loss evolution throughout the epochs. We observe that the networks with vanilla units activated by sigmoids rapidly stagnate in a constant loss value. This undesired premature convergence happens for both the train and validation losses. By far, this is the worst-performing model type. The other model family, which uses classic perceptrons with ReLU, presents a loss decay consistent over time and performs better than the first model type. The following two model groups had ARP in their hidden layer neurons. The ReLU-activated ARP model family exhibit an initial better performance than the other ReLU network type with classical perceptrons, but then is overtaken. The remaining two curves show information of the last model set, an ARP network with sigmoid activations. This model type has the best performance, reaching the lowest loss values, even when compared to the ReLU networks. Notably, with ARP, the test loss of the sigmoid networks was reduced by a factor of 15 at the cost of increasing the execution time by $\sim$12% (for details see Figure A1).

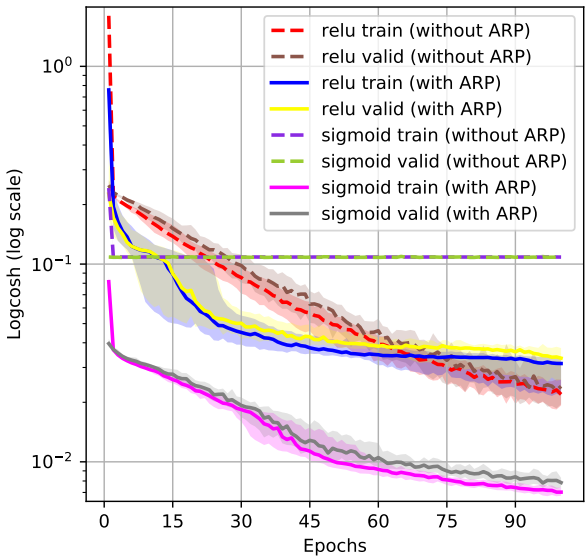

Figure 2. Loss function of the four model types. Central bold lines are median curves, and shades represent interquartile ranges. Number of executions per model type: 50. Epochs per iteration: 100. Batch size: 64. Optimizer: Nadam (lr = 0.003).

5. Conclusions

In this work, we employed the ARP neural unit aiming to calibrate a wearable GRF sensor. We compared the ARP performance against models with classic perceptrons using two different activation functions. The results show that in a neural network with sigmoid activations, changing the classic perceptrons to ARP can lead to a better performance, even beating the same neural model with vanilla perceptrons and ReLU activations. Also, even though the ARP units were designed to be employed with saturated functions, we can see that ARP can improve the model performance for the early epochs, when using them with a non-saturated function like ReLU. A more in-depth analysis needs to be performed to explain this behavior appropriately.

The calibration pipeline herein presented could be extrapolated to other cases where it is required to calibrate a sensor output to make it close to a reference scaled signal.

Acknowledgments

This work is funded by CONCYTEC-FONDECYT Peru (contract number E041-01/058-2018-FONDECYT-BM-IADT-AV).

References

Adesida, Y., Papi, E., and McGregor, A. Exploring the role of wearable technology in sport kinematics and kinetics: A systematic review. *Sensors*, 2019. doi: 10.3390/s19071597.

Almassri, A., Wan Hasan, W., Ahmad, S., Shafie, S., Wada, C., and Horio, K. Self-calibration algorithm for a pressure sensor with a real-time approach based on an artificial neural network. *Sensors*, 18(8):2561, 2018. doi: 10.3390/s18082561.

Eguchi, R. and Takahashi, M. Accessible calibration of insole force sensors using the wii balance board for kinetic gait analysis. In *2018 IEEE SENSORS*, pp. 1–4, 2018.

Eng, S., Al-Mai, O., and Ahmadi, M. A 6 dof, wearable, compliant shoe sensor for total ground reaction measurement. *IEEE Transactions on Instrumentation and Measurement*, 67(11):2714–2722, 2018.

Evans, M., Colyer, S., Cosker, D., and Salo, A. Foot contact timings and step length for sprint training. In *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV 2018)*, 03 2018. doi: 10.1109/WACV.2018.00184.

Karaboga, D. An idea based on honey bee swarm for numerical optimization. Technical report, Erciyes University, 2005.

Kumari, K. and Kaur, P. Use of nature-inspired computing techniques in real world applications (2010-2019) a brief review. *International Journal of Computer Applications & Information Technology*, 11(2):258–264, 2019.

Nagamune, K. and Yamada, M. A wearable measurement system for sole pressure to calculate center of pressure in sports activity. In *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2018)*, pp. 1333–1336, 10 2018.

Oldfrey, B., Jackson, R., Smitham, P., and Miodownik, M. A deep learning approach to non-linearity in wearable stretch sensors. *Front. Robot. AI 6: 27. doi: 10.3389/frobt*, 2019.

Pandit, S., Godiyal, A., Vimal, A., Singh, U., Joshi, D., and Kalyanasundaram, D. An affordable insole-sensor-based trans-femoral prosthesis for normal gait, 2018.

Saromo, D., Villota, E., and Villanueva, E. Auto-Rotating Perceptrons. *LatinX in AI Workshop at NeurIPS 2019 (arXiv:1910.02483)*, 2019.

Sim, T., Kwon, H., Oh, S., Joo, S.-B., Choi, A., Heo, H., Kim, K., and Mun, J. Predicting complete ground reaction forces and moments during gait with insole plantar pressure information using a wavelet neural network. *Journal of biomechanical engineering*, 137, 06 2015.

Wang, D., Tan, D., and Liu, L. Particle swarm optimization algorithm: an overview. *Soft Computing*, 22(2):387–408, 2018.

A. Supplementary Material

Table A1. Comparison of the algorithms used to calculate K. Both algorithms ran 100 times with 50 elements (food sources and particles for the ABC and PSO algorithms, respectively).

Algorithm		ABC	PSO
Final RMSE	Mean	32.712	29.440
	Std. Dev.	7.458	7.159
Execution time [s]	Mean	31.908	3.121
	Std. Dev.	0.206	0.059

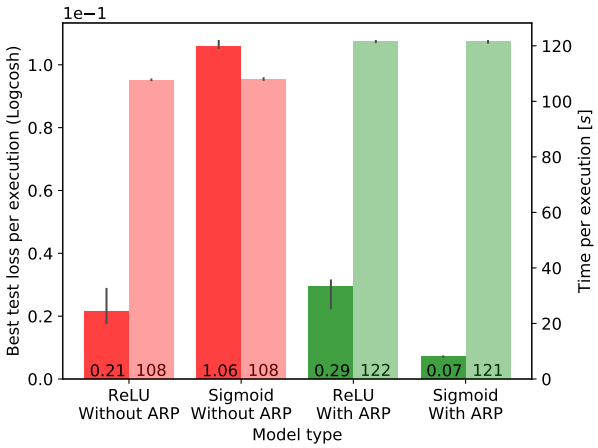

Figure A1. Best test loss value (left vertical axis) and corresponding processing time (right vertical axis) per execution. Each one of the executions has 100 epochs. The loss function is the logarithm of the hyperbolic cosine of the network prediction error. The bars represent the median value obtained after 50 executions. The vertical lines at the top of the bars represent the interquartile range.