A. Brömme, C. Busch, N. Damer, A. Dantcheva, M. Gomez-Barrero, K. Raja,
C. Rathgeb, A. Sequeira, and A. Uhl (Eds.): BIOSIG 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021

Gait Authentication based on Spiking Neural Networks

Enrique Argones Rúa[1], Tim Van hamme[2], Davy Preuveneers[2], Wouter Joosen[2]

Abstract: In this paper we address gait authentication using a novel approach based on spiking neural networks (SNNs). This technology has proven advantages regarding energy consumption and it is a perfect match with some proposed neuromorphic hardware chips, which can lead to a broader adoption of user device applications of artificial intelligence technologies. One of the challenges when using this technology is the training of the network itself, since it is not straightforward to apply well-known error backpropagation, massively used in traditional artificial neural networks (ANNs). In this paper we propose a new derivation of error backpropagation for the spiking neural networks that integrates lateral inhibition and provides competitive results when compared to state of the art ANNs in the context of IMU-based gait authentication.

Keywords: Spiking neural network, membrane potential, gait authentication, average spiking rate, normalized membrane potential, synaptic time dependent plasticity, gait recognition system, error backpropagation, surrogate function, angular velocity, variable length gait sequence, continuous authentication, open set biometric authentication, IMU gait authentication.

1 Introduction

Artificial Neural Networks (ANNs) have become the most prevalent pattern recognition tool, being used in a multiplicity of applications. Regarding biometrics, it is already used in most biometric modalities, such as speaker authentication, face recognition, fingerprint recognition, hand-based biometrics, electrocardiogram-based recognition, handwritten online signature recognition, or inertial gait recognition. However, deep learning neural networks often come at the cost of larger complexity and computational requirements in terms of memory and processing power, which may thwart its deployment on constrained user devices, especially for continuous authentication. In recent years there has been an increasing interest in a different type of neural network, the Spiking Neural Networks (SNN) [Ma97a]. This interest has been driven mainly by the possibility to use these networks within ultra-low power consumption specific hardware modules called neuromorphic hardware [Ba20], ideally suited for instance for continuous authentication. However, the drawback of this technology is the lack of mature learning approaches, as opposed to standard ANNs. The most widely used method for SNNs is Synaptic Time-Dependent Plasticity (STDP), which is a biologically inspired unsupervised method based on Hebbian

[1] imec – COSIC, Dept. of Electrical Engineering, KU Leuven. Kasteelpark Arenberg 10, bus 2452, B-3001 Heverlee, Belgium, enrique.argonesrua@esat.kuleuven.be
[2] imec – DistriNet, Dept. of Computer Science, KU Leuven. Celestijnenlaan 200A B-3001 Heverlee, Belgium, {tim.vanhamme, davy.preuveneers, wouter.joosen}@cs.kuleuven.be

rules [Ma97b], reaching limited discrimination performance when compared to gradient descent for ANNs. However, some approximations to gradient descent have been recently proposed, as we will discuss later. In this paper we propose to use SNNs topologically organized in columns, and a novel backpropagation method that allows to perform competitive supervised learning. We demonstrate the potential of this approach in the challenging biometric problem of inertial gait recognition. Inertial measurement unit (IMU) based gait recognition uses the inputs captured by IMU sensors placed somewhere in the subjects' body. Thus, gait is modeled as a six dimensional time series: 3D linear acceleration and 3D angular velocity. We augment this signal and represent it as a 26D time sequence as described in prior work [Va18].

To the best of our knowledge, this constitutes the first work on stream-based biometrics using this technology. We compare the obtained results in terms of authentication performance to ANNs.

2 Previous work on Spiking Neural Networks

Although SNNs are a relatively new neural network paradigm, a lot of research efforts have focused on it lately. Regarding the learning technique, initially most of the works focused on diverse versions of Synaptic Time Dependent Plasticity, a non-supervised technique that produces generative models, which produce spikes related to the most relevant or frequent inputs observed by the network. There are plenty of examples, such as [TM19, DC15, Kh18]. However, all of them suffer from one intrinsic limitation, its generative nature makes them not as accurate when dealing with classification tasks as other state of the art discriminative approaches. The main problem is the non-differentiability of the spikes, the output of the networks, which makes traditional error backpropagation (EBP) not directly applicable. There exist two main approaches to tackle this problem when using spiking neural networks, which consist of converting conventional neural networks trained with traditional supervised EBP-based techniques, or developing a framework where it is possible to perform EBP on the spiking neural network. A comparison of the mentioned approaches can be found in [Ta19]. Although both approaches can provide reasonable performances, specific training can be theoretically more accurate, especially if SNNs have special topological features which do not easily map from ANNs, as the columns used in this paper. This motivated us to use a specific training approach. Developments shown in [Mo18, Pa20, Wu18] are some representative examples of EBP on SNNs. All the backpropagation frameworks made for SNNs so far share that the objective function is either specialized for classification tasks (trying to get a specific label as the average spiking rate at the output layer), or it tries to provide spike trains very similar in different inputs, implying homogeneous dimensionality and duration (if applicable) of input signals. In both cases, it is hard to use these ideas to get a universal (open set, where the same network will be used to process samples from individuals not included in the training set) network for sequence-based biometric authentication, where the length of the sequence is variable (non-homogeneous, making the use of sequence approximation cumbersome).

3 Gait recognition system using SNNs

In this paper we use Leaky Integrate and Fire (LIF) neurons as fundamental computation units. Our networks will be constituted of columns of these neurons, organized in layers. Each layer feeds with the outputs of the columns from previous layers, after an optional max pool layer. Regarding the input layer, it is fed with the min-max normalized 26-D available gait signal and their negative counterparts, since synaptic weights are constrained to be positive in our SNNs.

We train the SNNs in two phases. In the first one, we use the unsupervised method Synaptic Time-Dependent Plasticity to get a good baseline for the second phase, which is a novel supervised Gradient Descent. In the following sections we describe in detail the topological elements of the networks and the learning algorithms.

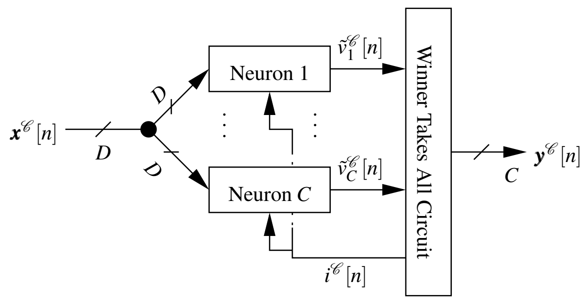

Fig. 1: Column of neurons with lateral inhibition

Columns of LIF neurons As mentioned above, neurons are organized in columns, as shown in Fig. 1. Neurons in a column share the synaptic field (input space), and only one can emit a spike at a given instant in time. This behaviour is modelled using a Winner Takes All (WTA) circuit, which chooses the neuron with the maximum normalized membrane potential above unity as the spiking one. When a neuron spikes, a lateral inhibition signal $i^{\mathscr{C}}[n]$ is fed back to the neurons in the column, which will put them in a refractory state, as explained below.

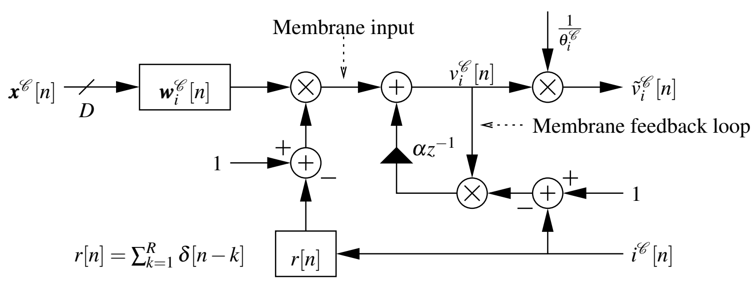

Fig. 2: Diagram of neuron i in column $\mathscr{C}$.

LIF neurons The LIF neurons used in this paper, shown in Fig. 2, work in discrete time. They can be in two different states: *active* and *refractory*. During the active state, the membrane potential leaky integrates the inputs filtered by their corresponding synaptic response filters, as shown in the membrane feedback loop. Before the membrane potential is fed into the WTA circuit, it is divided by θ, which can be understood as a neuron-specific activation threshold: if the membrane potential exceeds this value, the normalized membrane potential $\tilde{v}[n]$ will exceed unity. The WTA circuit then performs:

$$y_i^{\mathscr{C}}[n] = \begin{cases} 1 & \iff \quad i = \mathrm{argmax}_j \left\{ \hat{v}_j^{\mathscr{C}}[n] \mid \hat{v}_j^{\mathscr{C}}[n] \geq 1 \right\} \\ 0 & \iff \quad \nexists j \mid \hat{v}_j^{\mathscr{C}}[n] \geq 1 \end{cases} \quad , \; i^{\mathscr{C}}[n] = \max_i \left\{ y_i^{\mathscr{C}}[n] \right\} \tag{1}$$

When any neuron in the column emits a spike, each neuron in the column receives an inhibition impulse from the column WTA circuit (i.e., $i^{\mathscr{C}}[n] = 1$), and goes into refractory state, where the membrane potential is reset and synaptic inputs are ignored, by using $1 - i^{\mathscr{C}}[n]$ and $1 - i^{\mathscr{C}}[n] * r[n]$ as gate signals in the membrane potential feedback loop and input respectively, with $r[n] = \sum_{k=1}^{R} \delta[n-k]$, and R the refractory period duration. It must be noticed that if $R = 0$ the inhibition only affects the feedback loop. Both R and the membrane persistence α (or equivalently are the same in all the neurons in our model, since we understand that these are parameters related to the neurons' physiology in this biologically-inspired system.

Unsupervised initialization using STDP Synaptic weights are randomly initialized from Gaussian distributions, and clamped into the interval $[0,1]$. Then, STDP is used to adapt these weights to the statistics of the gait input sequences. Different update rules for the weights have been proposed for STDP, but the main idea is to implement (i) Long Term Potentiation (LTP) to weights recently contributing to the potential of a neuron that spikes, i.e., increasing this weight; and (ii) Long Term Depression (LTD) to weights that did not contribute to the potential of a neuron that spikes. In other words, neurons that spike reinforce the weights that contribute to the spiking, and weaken those weights that do not contribute. In our case, we implement LTP and LTD by increasing the weights proportionally to the observed input contribution. Given the synaptic filter $\boldsymbol{w}[n] = \sum_{i=0}^{L} w^k \delta[n-k]$, if a spike is emitted at time s, then $\Delta w^k = p \sum_{j=0}^{k} x[s-(k+j)]\alpha^j$ if $\sum_{j=0}^{k} x[s-(k+j)]\alpha^j > \varepsilon$, and $\Delta w = -d$ otherwise. A homeostatic rule is also used to avoid too dominant neurons in a column, decreasing the activation thresholds of neurons that spike below the columns' average spiking rate, and increasing them for neurons with spiking rate higher than the columns' average. These averages are computed using first order average estimators $\bar{r}[n] = \tau^{-1} r[n] + (1-\tau^{-1})\bar{r}[n-1]$, with τ equal to 5 times the average gait sequence length. Thresholds are updated at the end of each sequence STDP by $\Delta\theta_i^{\mathscr{C}} = \beta(\bar{r}^{\mathscr{C}} - C\bar{r}_i^{\mathscr{C}})$, where C is the number of neurons in column $\mathscr{C}$, $\bar{r}^{\mathscr{C}}$ and $\bar{r}_i^{\mathscr{C}}$ are the average spiking rate of the column and its i^{th} neuron respectively. After the update, the thresholds are clamped to the interval $[1,10]$.

Supervised training using EBP The main obstacle to perform EBP on SNNs is the non-differentiability of the neuron's output. Its discontinuous (it is either 0 or 1) and homogeneous (all the spikes look the same) nature require adopting a different approach to the one used in traditional ANNs. One of the most common is to define a surrogate function $\boldsymbol{f}$ that is differentiable, and substitute the output of the neuron by that surrogate during gradient computations, i.e., $\delta \boldsymbol{y} \sim \delta \boldsymbol{f}$. One sensible approach is to make this surrogate function model the probability of that neuron firing. A sensible surrogate function for the proposed neuron columns should be based on the normalized membrane potential signals $\tilde{v}_i^{\mathscr{C}}[n]$, which are differentiable with respect to the neurons' input and synaptic filter weights. In this regard, it should also:

1. Be monotonically increasing with its normalized membrane potential, and monotonically decreasing with the normalized membrane potential of the other neurons in the column.
2. Saturate when the membrane potential of the neuron dominates the other neurons' membrane potential in the column.

Taking these into account, we propose to use the softmax function of the normalized membrane potential as surrogate function:

$$\boldsymbol{f}^{\mathscr{C}}[n] = \operatorname{softmax}\left(\tilde{\boldsymbol{v}}^{\mathscr{C}}[n]\right) = \left(\frac{e^{\hat{v}_1^{\mathscr{C}}[n]}}{\sum_{j=1}^{C} e^{\hat{v}_j^{\mathscr{C}}[n]}}, \ldots, \frac{e^{\hat{v}_C^{\mathscr{C}}[n]}}{\sum_{j=1}^{C} e^{\hat{v}_j^{\mathscr{C}}[n]}}\right)^t \tag{2}$$

We evaluate this differentiable surrogate function during the backward phase of EBP *only at the spiking times*, i.e., when one neuron in the column emits a spike. By doing this, we only take into account the spiking events, which are the ones forwarding real information, thus removing influence of instants where the neurons do not get enough evidence to spike, and saving a lot of computation power and training time. Thus, spikes serve as noise removing and energy efficiency mechanisms.

Objective function, optimizer and boosting Although there could be information on time dependencies among spiking events (local information), we only use the average spiking rate at the output layer neurons. We use a siamese architecture, where a reference sequence $\mathscr{S}_R$, belonging to class C_R, and a probe sequence $\mathscr{S}_P$ belonging to class C_P make a forward pass through the network, obtaining two average output average spiking rates $\boldsymbol{r}_R^o$ and $\boldsymbol{r}_P^o$. We then maximize the following cosine similarity based objective function:

$$O(\mathscr{S}_R, \mathscr{S}_P) = (2\delta[C_R - C_P] - 1)\, \frac{\boldsymbol{r}_R^{o\,t} \boldsymbol{r}_P^o}{|\boldsymbol{r}_R^o|\,|\boldsymbol{r}_P^o|} \tag{3}$$

We use the Nadam optimizer as described in [Do16] to improve convergence, abiding the parameters suggested in this paper. We also incorporated boosting to train the neural network, grouping the pairs of sequences matching a given enrollment sequence in the same minibatch, as in tuplet loss [YT19], and performing the updates based only on the hard negative sequences and the positive sequence (if any of the negative examples is hard, or if the positive sequence is hard itself).

4 Gait recognition system using ANNs

We built a gait recognition system based on ANNs which serves as a baseline to adequately demonstrate the potential of SNNs to model gait. We choose an architecture that led to competitive results for other gait sequence modelling tasks [Va19]. Specifically, we chose temporal convolutional networks (TCN), which are proven to be an effective strategy to model time series data in general [BKK18]. Furthermore, to accommodate the needs of authentication, i.e. an open world assumption with limited amounts of enrollment data, we leverage metric learning to train the system, i.e. we use the triplet loss [SKP15].

As can be seen in Fig. 3, our TCN has a fairly simple architecture with only three layers. The final embedding is a vector of size 128. We trained the network for 500 epochs with the adam optimzer with a learning rate of 0.001. During training we feed fixed length sequences of size 180 to the network, due to the sample rate of $100Hz$ this corresponds to $1.8s$ of data. We chose this value such that at least one gait cycle is captured. Moreover, which part of the full variable length gait sequence is retained is chosen at random. During testing the whole variable length gait sequence is used.

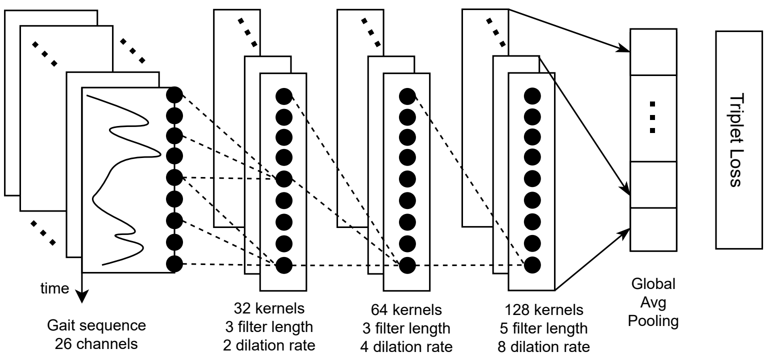

Fig. 3: The architecture of the ANN system which is based on 1 dimensional convolutions with dilation and trained with the triplet loss.

5 Database and experimental setup

To train and evaluate our systems we use the IMU sequences contained in the OU-ISIR dataset labeled as *level walk* and captured by the center sensor. Usually for each user there are two sequences, of which we use one for enrollment and one for testing. We only consider the 483 users with two valid walking sequences for all three sensors, i.e. left, right and center located sensors. Only the center sensor is used for the experiments.

The walk sequences of the OU-ISIR dataset is represented as a six dimensional (6D) time series: 3D linear acceleration and 3D angular velocity. We augment this signal and represent it as a 26D time sequence as described in prior work [Va18]. Eight dimensions are derived from the *gyroscopic dynamics (8D)*: the angular velocity, their first order differences, and the magnitude of both vectors. Six dimensions are related to the *vertical and horizontal components (6D)*, which are an approximation of the vertical and horizontal acceleration in the world pane. They are complemented with their first order differences and jerks. Twelve dimensions are related to *roll and pitch (12D)*, which are approximated from the linear accelerations. We also improve this approximation by fusing it with an estimation of roll and pitch from the angular velocity. We keep both approximations and complement them with their first-order and second-order differences.

We adopt a 5-fold cross-validation protocol, where we randomly split the 483 users in 5 disjoint sets which contain either 96 or 97 users: $S_s = \{U_0^s, U_1^s, \ldots, U_x^s\}$ with $x \in [96, 97]$ and $s \in [0, 4]$. Thus, we do 5 rounds, where during each round k we select S_k as the test set, and the remaining 4 sets as training set. We report the Equal Error Rate (EER) in the test set, i.e. the threshold at which False Rejection Rate (FRR) is equal to False Acceptance Rate (FAR). Besides, we plot the Detection Error Trade-off curves for a visual comparison.

6 Experimental results

In our experiments we tried different SNN topologies, with one and two layers, using max pooling between them. We present here the performance obtained by the TCN architecture presented in Section 4, together with the fllowing SNNs:

N1 1 layer, 4 columns, 32 neurons/column, 1 coeff. filter, 32-D column input.

N2 1 layer, 4 columns, 32 neurons/column, 4 coeff. filter, 32-D column input.

N3 2 layers, each with 4 columns, 32 neurons/column, and 4 coeff. filters. First layer has 32-D and second layer has 96-D column inputs. Max-pooling with window and stride 2 is used between layers.

N4 1 layer, 32 columns, 32 neurons/column, 2 coeff. filter, 32-D column input.

7 Conclusions

In this work we presented a novel column-based SNN architecture and EBP approach for supervised learning. We tested this approach on the challenging task of authenticating persons using IMU gait signals in an open set protocol. Although results obtained by EBP on SNNs are yet behind from the ones shown by state of the art ANN architectures, a clear improvement over STDP is demonstrated. The low power consumption shown by SNN hardware implementations, together with the development of EBP techniques and reduced performance gap with respect to ANNs encourage further research on the application of SNN for biometrics, but also for a wider range of applications.

Label	Algorithm	Test fold 1	2	3	4	5	Average ± Std. dev.
N1	STDP	7.97	4.88	6.19	10.66	6.25	7.19 ± 2.23
	EBP	3.98	2.51	3.09	6.25	5.21	4.21 ± 1.53
N2	STDP	9.28	5.96	8.25	10.71	11.46	9.13 ± 2.17
	EBP	4.16	2.13	4.12	7.27	4.17	4.37 ± 1.84
N3	STDP	10.53	7.22	11.16	15.62	11.46	11.20 ± 3.00
	EBP	7.62	2.80	4.99	6.33	8.33	6.01 ± 2.20
N4	STDP	5.62	4.12	5.15	9.38	6.25	6.10 ± 1.99
	EBP	2.49	1.03	2.06	3.12	2.08	2.16 ± 0.76
TCN	EBP	1.03	1.92	1.92	2.08	1.06	1.60 ± 0.46

Tab. 1: Authentication performance in terms of test EER(%) of the different networks.

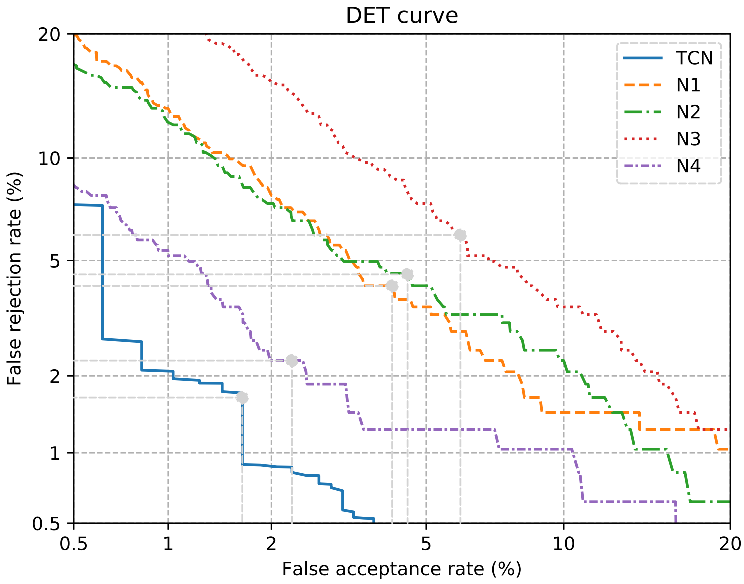

Fig. 4: DET curve for the EBP-trained SNNs and TCN.

References

[Ba20] Balaji, Adarsha; Das, Anup; Wu, Yuefeng; Huynh, Khanh; Dell'Anna, Francesco G.; Indiveri, Giacomo; Krichmar, Jeffrey L.; Dutt, Nikil D.; Schaafsma, Siebren; Catthoor, Francky: Mapping Spiking Neural Networks to Neuromorphic Hardware. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(1):76–86, 2020.

[BKK18] Bai, Shaojie; Kolter, J. Zico; Koltun, Vladlen: , An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018.

[DC15] Diehl, Peter; Cook, Matthew: Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience, 9:99, 2015.

[Do16] Dozat, Timothy: Incorporating Nesterov Momentum into Adam. In (Bengio, Yoshua; LeCun, Yann, eds): 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016.

[Kh18] Kheradpisheh, Saeed Reza; Ganjtabesh, Mohammad; Thorpe, Simon J.; Masquelier, Timothée: STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks, 99:56–67, 2018.

[Ma97a] Maass, Wolfgang: Networks of spiking neurons: The third generation of neural network models. Neural Networks, 10(9):1659–1671, 1997.

[Ma97b] Markram, Henry; Lübke, Joachim; Frotscher, Michael; Sakmann, Bert: Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. Science, 275(5297):213–215, 1997.

[Mo18] Mostafa, H.: Supervised Learning Based on Temporal Coding in Spiking Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 29(7):3227–3235, 2018.

[Pa20] Patiño-Saucedo, Alberto; Rostro-Gonzalez, Horacio; Serrano-Gotarredona, Teresa; Linares-Barranco, Bernabé: Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the SpiNNaker neuromorphic platform. Neural Networks, 121:319–328, 2020.

[SKP15] Schroff, Florian; Kalenichenko, Dmitry; Philbin, James: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823, 2015.

[Ta19] Tavanaei, Amirhossein; Ghodrati, Masoud; Kheradpisheh, Saeed Reza; Masquelier, Timothée; Maida, Anthony: Deep learning in spiking neural networks. Neural Networks, 111:47–63, 2019.

[TM19] Tavanaei, Amirhossein; Maida, Anthony: BP-STDP: Approximating backpropagation using spike timing dependent plasticity. Neurocomputing, 330:39–47, 2019.

[Va18] Van hamme, Tim; Argones Rúa, Enrique; Preuveneers, Davy; Joosen, Wouter: Gait template protection using HMM-UBM. In (Brömme, Arslan; Busch, Christoph; Dantcheva, Antitza; Rathgeb, Christian; Uhl, Andreas, eds): 2018 International Conference of the Biometrics Special Interest Group, BIOSIG 2018, Darmstadt, Germany, September 26-28, 2018. volume P-282 of LNI. GI / IEEE, pp. 1–8, 2018.

[Va19] Van hamme, Tim; Garofalo, Giuseppe; Argones RÃºa, Enrique; Preuveneers, Davy; Joosen, Wouter: A Systematic Comparison of Age and Gender Prediction on IMU Sensor-Based Gait Traces. Sensors, 19(13), 2019.

[Wu18] Wu, Yujie; Deng, Lei; Li, Guoqi; Zhu, Jun; Shi, Luping: Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. Frontiers in Neuroscience, 12:331, 2018.

[YT19] Yu, B.; Tao, D.: Deep Metric Learning With Tuplet Margin Loss. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6489–6498, 2019.