Supervised Learning with Small Training Set for Gesture Recognition by Spiking Neural Networks

Natabara Máté Gyöngyössy, Márk Domonkos, János Botzheim, Péter Korondi
Department of Mechatronics, Optics and Mechanical Engineering Informatics
Faculty of Mechanical Engineering
Budapest University of Technology and Economics
4-6 Bertalan Lajos Street, 1111 Budapest, Hungary
e-mail: natabara@gyongyossy.hu, {domonkos,botzheim,korondi}@mogi.bme.hu

***Abstract*—This paper proposes a novel supervised learning algorithm for spiking neural networks. The algorithm combines Hebbian learning and least mean squares method and it works well for small training datasets and short training cycles. The proposed method is applied in human-robot interaction for recognizing musical hand gestures based on the work of Zoltán Kodály. The MNIST dataset is also used as a benchmark test to verify the proposed algorithm's capability to outperform shallow ANN architectures. Experiments with the robot also provided promising results by recognizing the human hand signs correctly.**

***Index Terms*—Spiking Neural Networks, Supervised Learning, Gesture Recognition, Human-Robot Interaction**

I. Introduction

In the history of robotics we saw many revolutions. From the first industrial robots, the UNIMATION, developed by George C. Devol in 1950's [1] to nowadays when robots are equipped with high performance computers, sensors, and actuators so they can manipulate faster and safer, we made a huge step forward. Recent trends show that the industry is starting to orientate in a new direction, where the requirement of performance increased in robotcells that are sharing their working area with operators. For this request robot manufacturers started their research on cooperative robots, in short cobots. Instead of the classical matrix transformation and model-based robotics, a model-free control solutions is suggested [2]. Model-free description is an important step to the intelligent robotics.

A new direction in robotics research is cognitive robotics, where the robot has some kind of cognitive abilities like facial expression recognition, path finding, gesture recognition, etc. This last ability, and our application for this kind of purposes, will be also detailed in this paper.

This paper is focusing mainly on spiking neural networks (SNN) and gesture recognition. We made a test environment with a UR3e cobot playing on a Yamaha keyboard, based on camera images and the output of the SNN.

Artificial neural networks (ANN) are well known computation elements of machine learning. They pass values between each other represented by float numbers, which is unlikely to happen in human brain. Spiking neural networks solve this problem and give us a more detailed, dynamic model of our brain cells, leading us to the third-generation of neural networks [3]. In these networks neurons code information to spike trains which are simple non-linear functions of time. With proper information coding spatio-temporal learning can take place between these neurons. SNNs have been successfully used for image processing and gesture recognition tasks in the last few years [4] [5]. Although they appear to be a novel topic of computational intelligence, these networks have already been used to create interactive human-robot interfaces and develop social skills [6]. Spiking neural networks proved their performance in several convolutional, real-time applications, too [7] [8] [9]. These works point out that SNNs need less data than regular deep networks do. In this article we intend to focus on these low-end solutions both in network architecture and available data. Furthermore, we propose a supervised learning method based on spike timing and the adaptation of Widrow's least mean squares (LMS) algorithm [10] keeping the Hebbian perspective of spike-timing-dependent plasticity (STDP) learning rule. This algorithm is first evaluated on the MNIST dataset, then it is applied in an adaptive gesture recognition scenario.

Effective and ergonomical communication between the human consumer and the robot is necessary in a large variety of situations no matter if it is in social or industrial robotics. Fasola and Matarić designed an antropomorph robot for elderly people as a training coach for seated aerobic exercises. In their work their socially assistive robot system sucsessfully motivated and engaged the users [11].

Knox et al in a case study describe a robot system which can be trained for five different navigational tasks via human feedback [12].

Similarly to us Hoffman and Weinberg presented a musician robot, Shimon the marimba player, for human-robot joint jazz improvisation, where the sequences of notes, as an improvisation, were used as a gesture [13] [14].

Our gestures are based on Zoltán Kodály's solmizational gestures [15] on the first hand while the secondary hand shows the rhythm. Zoltán Kodály was a Hungarian pioneer of ethnomusicology and a groundbreaking educator, who based on the work of John Curwen's old tonic solfa established a new type of solmization mainly for the Hungarian education of music [16].

Fig. 1. Solmization hand signs

The structure of this paper is as follows. In Section II the investigated problem is introduced. Our proposed algorithm is presented in Section III after a short introduction to spiking neural networks. Section IV shows experimental results on the MNIST dataset and in recognition of gestures while playing music. At last Section V draws conclusions.

II. Problem Statement

In the recent years the interest in studying human-computer (HCI) and human-robot interaction (HRI) has increased significantly. Researchers believed that HCI and HRI can work much more effectively if artificial intelligence systems are capable of detecting correctly certain sociocognitive or socioemotional behaviors. To develop this line of research a multidisciplinary collaboration is starting. The main goal of this collaboration is to enable human to interact with artificial systems in his natural "human" way.

Social interaction is also one of the main contributions of this paper. The data exchange between the robot and a higher level intelligent controller is the info-communication level. It is also called info-communication when the operator gives direct commands to the robot. We move to the cognitive info-communication level when the operator can instruct the robot using cognitive communication channels [17], including gesture-driven robots. The basic concept is that the gesture system is the result of a mutual learning process.

A. The Implemented Scenario

The robot is waiting for an unexperienced volunteer, who receives a table of solmization hand signs shown in Fig. 1. He/she tries to show solmization hand signs to the robot by his/her right hand and shows the rhythm by the left hand. The robot presses the corresponding key of a piano following the actual rhythm. The main challange is that the robot instead of gathering a big database for gesture recognition starts a mutual learning process and tries to learn the specific hand motion of the actual volunteer.

Gesture recognition is performed by biologically inspired neural networks and a new learning method STDP-LMS. Using a low complexity neural model we aimed to create a method capable of quick learning without defining a gradient or leaving possible human-related methods out completely. The idea of combining the well known methods of computation technology and neurology was inspired by Widrow's Hebbian-LMS solution [18]. The proposed algorithm is able to learn from a limited number of samples, which enables real-time social interaction in general. In our case we use this algorithm for playing music with the robot. Figure 2

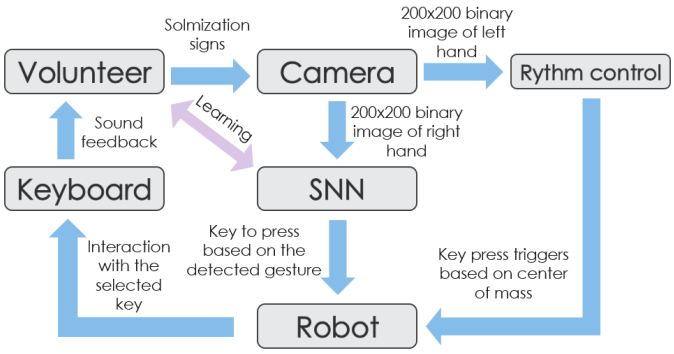

Fig. 2. The information flow of the scenario

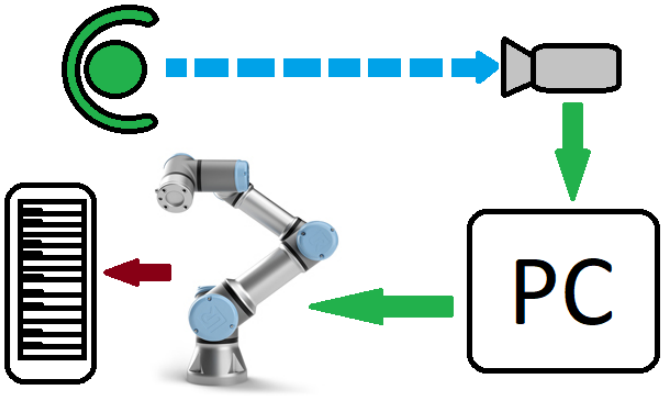

Fig. 3. The layout of the experimental set up

summarizes the main concepts of communication between the elements of this scenario.

The aim of the research presented in this paper is to use a supervised learning algorithm with some novel aspects using SNN architecture. This architecture is interpreted to be able to recognize patterns in pictures so it can categorize them. Since every picture shows a hand sign representing a musical note, i.e. gesture, this implementation of the SNN architecture is also able to recognize gestures.

B. Experimental Set Up

In this scenario, as shown in Fig. 3, a UR3e collaborative robot is used as a manipulator with a 3D printed one finger opened hand, shown in Fig. 4. A Basler ACA800-510UC camera with an objective with focal length of 6 mm and a PC are installed. The camera's exposure time is 2 ms to avoid the motion blur of hands during gesture recognition. Pictures are sent to the computer with a fix 3 ms cycletime, what the computer preprocesses, so only the regions of interests will be processed. The algorithm decides which kind of gesture is shown and sends the robot to the corresponding positions of the note on the Yamaha keyboard.

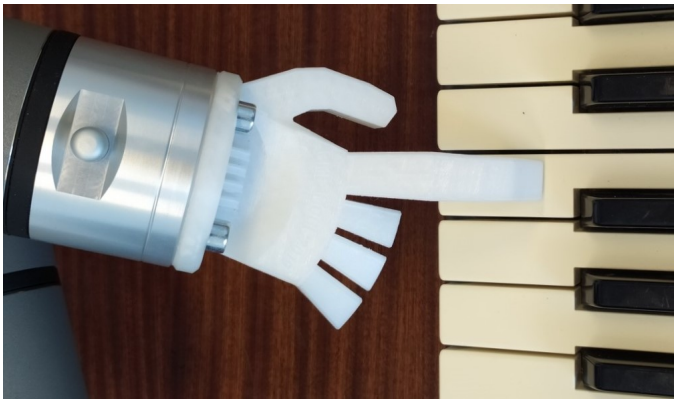

Fig. 4. The 3D printed hand

III. Proposed Algorithm

A. Spiking Neural Networks

1) Spiking Neuron Models: Spiking neurons are simplified models of nerve cells. In reality these cells are controlled by numerous ion channels, which often gate each other [3]. Thus some simplifications are necessary. Many models including the Hodgkin-Huxley model [19] reduce their model of a neuron to two dimensions and several non-linear functions of these two variables time and position. Some simplified models even reduce the dimensions to time only. In this subsection we will shortly introduce the concept of one dimensional neuron models focusing on their computational importance, rather than diving deep into neurobiology. At the end of the subsection we describe the model used in proposed algorithm.

Spiking neurons are represented by a value which is related to the voltage between the cell and its surroundings, called membrane potential. Membrane potential changes by time based on incoming stimulus which we call spike trains. When a neuron's membrane potential reaches a specific threshold it emits a spike, which is a short but huge positive jump in the potential of neurons. Neurons pass information between each other by these spikes. Their timing, phase or even frequency could encode some information. After a neuron fires its potential returns to a refractory state in which it cannot fire so easily. Some networks define inhibitory or excitatory synapses between neurons based on the effect of the incoming spike train to the post-synaptic neuron's potential.

This effect could be described as the discrete convolution of discrete Kronecker-delta spike trains and a kernel function. These kernel functions almost always depend on the weight between the two neurons which represents the strength of the given synapse.

2) Non-leaky Integrate and Fire Model: Keeping the above principles and the intention of developing a low-end solution in mind we use a non-leaky integrate and fire model also used in [9]. When taking time in account we used discrete dimensionless timesteps to keep track of events and values.

The membrane potential of non-leaky neurons won't decay over time, it remains the same, until an input spike train changes it. The change of membrane potential over time could be described by Eq. (1).

$$h_m(t) = h_m(t-1) + \sum_{n=1}^{N} s_n(t-1) \cdot w_{nm}(t-1) + h_m^{ext}(t) \quad (1)$$

where n and m are indices of pre- and postsynaptic neurons, respectively, N is the number of presynaptic neurons, $h_m(t)$ is the membrane potential of neuron m, $s_n(t-1)$ is the presynaptic spike train, w_{nm} is the weight between neuron n and m of the two layers, $h_m^{ext}(t)$ refers to an external input which comes from outside of the network. From Eq. (1), it could be clearly read that the weights themselves are used as a constant kernel function.

Spike generation is defined in Eq. (2).

$$s_m(t) = \begin{cases} 1 & \text{if } h_m(t) \geq \theta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $s_m(t)$ is the spike train emitted by neuron m, and θ meaning the threshold, which is constant in the proposed model. These spike trains one-by-one are equal to a Kronecker-delta function $\delta_m(t, \hat{t}_m)$, where $\hat{t}_m$ is the spike time of m, which is equal to the timestep where neuron m reaches its threshold.

$$\hat{t}_i = \begin{cases} t & \text{where } s_i(t) = 1 \\ t_{max} + 1 & \text{if } s_i(t) \equiv 0 \end{cases} \quad (3)$$

Equation (3) gives the proper definition of spike time for any neuron with index i, introducing a pseudo-spike time for every neuron which has not fired defining them as the incrementation of t_{max}, where t_{max} is the number of timesteps an input is presented to the network for. Introducing this pseudo-spike time is necessary to be able to use the spike time based coding technique described later.

After a neuron fired, its potential is reset to zero and kept at this value, until a new input is presented to our network. In some cases we might apply a rule called winner-take-all, which means putting every neuron of a layer into refractory state after a single neuron from that layer fired.

For the presented gesture recognition and image processing applications a two-layer architecture is used, which, in case of SNNs is possible, even without losing the ability to interpolate higher order functions. The input layer 0 has a neuron for each pixel of the input images, while output layer 1 represents the categories with a neuron for each. The connection between these two layers is represented by an $n \times m$ dimensional weight matrix. An example of this architecture is shown in Fig. 5.

B. STDP-LMS Method

1) Spike-Timing-Dependent Plasticity: Spike-timing-dependent plasticity is a native learning method for SNNs. STDP is based on Hebb's learning rule [20] "fire together, wire together", with the ability to develop anti-Hebbian and non-Hebbian qualities as well. In general STDP means weight modification based on the delay between the pre- and

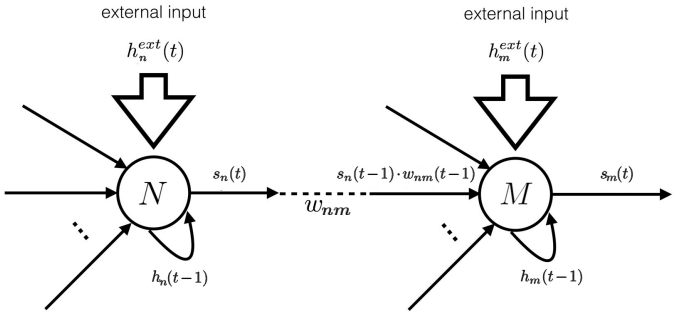

Fig. 5. Structure of the non-leaky integrate and fire model

postsynaptic spike times. These rules vary from high to low complexity, here we provide a simple approach which is used in the proposed novel algorithm.

$$\Delta w_{nm}(\hat{t}_m - \hat{t}_n) = \begin{cases} a^{pre,post}(\hat{t}_m - \hat{t}_n) & \text{if } \hat{t}_m - \hat{t}_n > 0 \\ a^{post,pre}(\hat{t}_m - \hat{t}_n) & \text{if } \hat{t}_m - \hat{t}_n < 0 \end{cases} \tag{4}$$

where $\hat{t}_m - \hat{t}_n$ is the delay between the two spike trains, $a^{pre,post}$ describes the learning rule when the presynaptic neuron fires first, $a^{post,pre}$ is the other case. In a "pre, post" case the presynaptic neuron contributes to the postsynaptic membrane potential, thus neuron n might be a cause of neuron m firing. According to Hebb's postulate this results in a positive weight change, while the other case changes the weight in the opposite direction.

STDP could be much more complex, when Eq. (4) is just a component of the learning rule, called the learning window, but in this article we will stick only to this simplified rule.

2) Least Mean Squares for SNNs: The basic idea of the least mean squares algorithm is that the weight modification is based on the least mean square of the error function i.e. the difference between the desired output and the actual output. The main equation of this algorithm is

$$\Delta w_{nm} = \gamma \cdot e_m \cdot x_{nm} \tag{5}$$

where γ is a constant learning rate, e_m is the error of postsynaptic neuron m and x_{nm} is the input of postsynaptic neuron m coming from presynaptic neuron n.

To measure a neuron's error we propose to use the deviation of the neuron's spike time from the desired spike time.

$$e_m = \hat{t}_m - \hat{t}_{target} \tag{6}$$

where $\hat{t}_m$ is the neuron's spike time and $\hat{t}_{target}$ is our target spike time for neuron m.

The input related part x_{nm} is derived from the spike trains, which are the only connections between two neurons.

$$x_{nm} = \begin{cases} \sum_{t=0}^{t_{max}} s_n(t) \cdot w_{nm}(t) = \sum_{t=0}^{t_{max}} \delta(t, \hat{t}_n) \cdot w_{nm}(t) = \\ = w_{nm}(\hat{t}_n) & \text{if } \hat{t}_n < \hat{t}_m \\ a^- & \text{if } \hat{t}_m < \hat{t}_n < t_{max} + 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where s_n is the input spike train, $\delta(t, \hat{t}_n)$ is a Kronecker-delta function, w_{nm} is the weight between the two neurons, $a^- < 0$ is the anti-Hebbian constant.

Equation (7) points out, that if a presynaptic fire occurs, the input is clearly the value of the weight between the two neurons at presynaptic spike time. Otherwise it would clearly be zero. We overwrite this value in one case, specifically when the presynaptic fire occurs before the postsynaptic. In this case the presynaptic neuron did not contribute to the potential of the postsynaptic neuron before firing. The usage of a negative constant in analogy of the negative $a^{post,pre}$ function is proposed.

With this modification the proposed STDP-LMS algorithm is driven by an error minimization rule, while keeping the spike-timing-dependent behavior.

IV. Experimental Results

A. MNIST Benchmark Tests

MNIST is a dataset of handwritten numbers (0-9) it has a training set of 60000 images and a test set of 10000 normalized images. It is a commonly used database for classification and image recognition problems presented in [21]. We evaluated our learning method on this dataset using only 2500 randomly selected training pictures but using the whole test set to determine the network's error.

Each selected image was presented for 20 ticks. We applied winner-take-all rule on the output layer to find the neuron representing the most relevant category. Fitting the input dimension the input layer consists of 784 neurons, the output layer has 10 neurons. The parameters of the algorithm were set based on preliminary tests. We set the threshold to 0.5 for every neuron, the weight decay is set to -0.2, accompanied with a 0.02 learning rate. All the weights are initialized randomly by a normal distribution around 0.75 with a deviance of 0.05. The input data was scaled linearly between 0 and 0.05, as a low-end solution we used the input neurons' ability of non-leaky integration to get spikes related to the value of inputs. We interpreted the output labels by defining target spike time vectors of the value 8 for the neuron related to the category and 21 for all the other neurons we don't want to fire.

After 3 epochs of training our results are promising. With an average of 81% accuracy we were able to outperform a shallow ANN architecture with 100 hidden neurons, sigmoid activation and stochastic gradient descent optimizer, which architecture reached an average of 77% accuracy on the test set.

Although a 3-layer ANN with more test data and epochs or hidden neurons to train can outperform the proposed SNN, in the early phase of convergence the proposed 2-layer network

Fig. 6. The robot gets a task to reach the position of “so” (left figure), then presses the not (middle figure), and repositions himself to “re” (right figure).

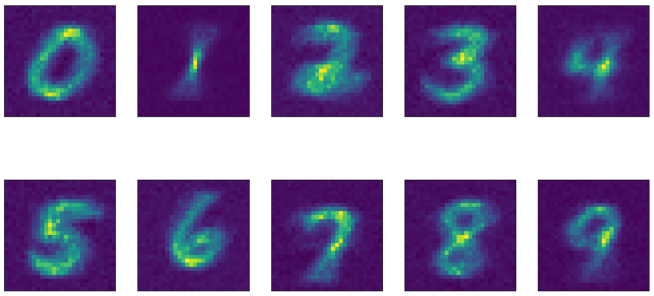

Fig. 7. Visualization of the weights of the output layer after training (MNIST).

could top a regular ANN architecture. Note that the SNN here runs without any runtime optimizers. The phenomenon described above shows the advantages of SNNs even in this low-end case, which is the ability to adapt to incomplete training sets and converge quickly. Results are shown in Fig. 7 where the weights are visualized by color intensity starting from the dark blue.

B. Solmization based Application

The robotics application, which we introduced in Section II, aims to create a platform for human-robot communication via a sign language. We chose a simple sign language, namely the solmization sign system introduced by Zoltán Kodály and used a keyboard to get feedback from the robot.

Using a region of interest (ROI) method to extract data from the image, we preprocessed our ROIs for both hands by substracting the background and using an adaptive color filtering method to get a binary image which we directly fed to our SNN. The rhythm is detected by measuring the position of the center of mass on axis y using constant thresholds.

With 20-20 pictures taken of all the solmization signs and after 3 epochs of training we were able to reach a 100% accuracy even with using only half of the pictures for training using the same hyperparameters as we used in subsection A of this section. Without any real test data available, we measured the recognition accuracy based on a large number of real world inputs. The network reached an accuracy of 93% when feeding

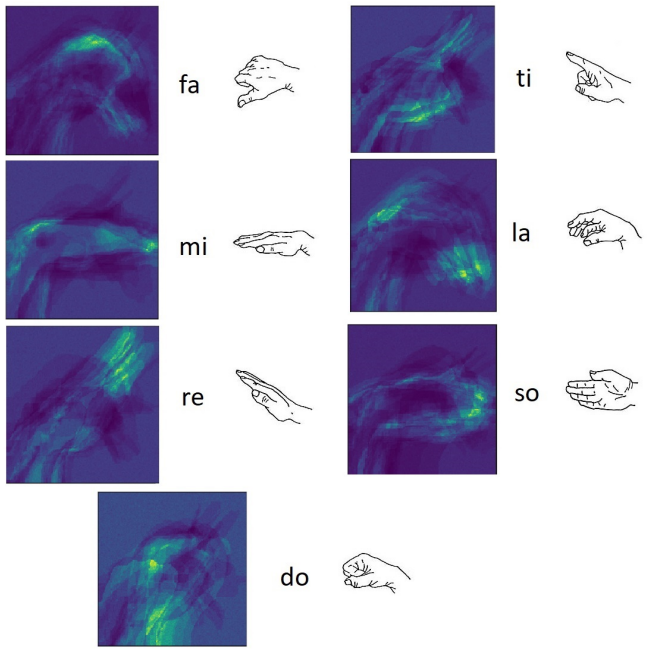

Fig. 8. Visualization of the weights of the output layer after training the signs.

it with input from the same individual it got its training data from. The whole process takes about two minutes and the robot is ready to communicate with human partners in the way shown in Figure 6.

We could observe the same fast convergence and relatively high capability to fit the training data (which is not a disadvantage in our usecase) as we have seen with the MNIST dataset. Figure 8 lists the weight maps representing the learned categories, the labels are provided as images of the corresponding hand sign. In these figures one can observe how the network, just like our human perception, focuses on similarities and differences between the training images.

With the system above we were able to play simple folk

songs and aid singers in their preparation by providing the tones they wanted to sing.

V. Conclusions

The supervised learning algorithm based on SNN architecture containing Hebbian learning and least mean sqares method presented in this paper provided promising results outperforming a shallow ANN on small datasets, including a restricted MNIST dataset. The test application with gesture recognition and UR3e robot playing the piano provided also highly promising results in social robotics and cobot application technology.

This kind of SNN architecture can be more efficient in the development of a human-robot sign language since its nature mimics the learning of human brain more accurately compared to other ANN models. The results presented in this paper have the potencial for further works also in ethorobotics for an ergonomical communication between the human and the robot mainly in learning the social skills fast in social robotics and in the fields of joint learning.

Acknowledgment

Authors would like to thank Bertalan Pizág to provide help in setting up the robotic environment.

János Botzheim was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

This work was supported by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/FM) and the National Research, Development and Innovation Office grant (K120501).

References

[1] L.Ballard, "Robotics' founding father George C. Devol-serial entrepreneur and inventor," in *Robot-Congers Issue 31*, 2011, p. 58.

[2] R.-C. Roman, R.-E. Precup, and R.-C. David, "Second order intelligent proportional-integral fuzzy control of twin rotor aerodynamic systems," *Procedia Computer Science*, vol. 139, pp. 372–380, Oct 2017.

[3] W. Gerstner and W. M. Kistler, *Spiking neuron models*. Shaftesbury Road, Cambridge: Cambridge University Press, 2002.

[4] D. Niu, D. Li, R. Yan, and H. Tang, "A gesture recognition method based on spiking neural networks for cognition development," in *Neural Information Processing*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham: Springer International Publishing, 2018, pp. 582–593.

[5] J. Botzheim, T. Obo, and N. Kubota, "Human gesture recognition for robot partners by spiking neural network and classification learning," in *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, Nov 2012, pp. 1954–1958.

[6] J. Woo, J. Botzheim, and N. Kubota, "Emotional empathy model for robot partners using recurrent spiking neural network model with Hebbian-LMS learning," *Malaysian Journal of Computer Science*, vol. 30, no. 4, pp. 258–285, 2017. [Online]. Available: https://ejournal.um.edu.my/index.php/MJCS/article/view/9889

[7] R. Vaila, J. Chiasson, and V. Saxena, "Deep convolutional spiking neural networks for image classification," *CoRR*, vol. abs/1903.12272, 2019. [Online]. Available: http://arxiv.org/abs/1903.12272

[8] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, pp. 54–66, 05 2015.

[9] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56 – 67, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608017302903

[10] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *IRE WESCON Conv. Record*, vol. 4, pp. 96–104, 1960.

[11] J. Fasola and M. J. Matarić, "A socially assistive robot exercise coach for the elderly," *J. Hum.-Robot Interact.*, vol. 2, no. 2, pp. 3–32, Jun. 2013. [Online]. Available: https://doi.org/10.5898/JHRI.2.2.Fasola

[12] W. B. Knox, P. Stone, and C. Breazeal, "Training a robot via human feedback: A case study," in *Proceedings of the 5th International Conference on Social Robotics*, Bristol, UK, 2013, pp. 460–470.

[13] G. Hoffman and G. Weinberg, "Gesture-based human-robot jazz improvisation," in *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, 2010, pp. 582–587.

[14] ——, "Synchronization in human-robot musicianship," in *19th IEEE International Symposium on Robot and Human Interactive Communication*, Principe di Piemonte - Viareggio, Italy, 2010, pp. 718–724.

[15] E. Hegyi and Z. Kodály, *Solfege According to the Kodály-concept: Chapters I to V*. Zoltán Kodály pedagogical Institute of music, 1975, vol. 1.

[16] International Kodály Society. Kodály's Life & Work. [Online]. Available: https://www.iks.hu/index.php/zoltan-kodalys-life-and-work

[17] C. Horváth and S. Kovács, "New cognitive info-communication channels for human-machine interaction," *RECENT INNOVATIONS IN MECHATRONICS*, vol. 4, no. 1, pp. 1–9, 2017.

[18] B. Widrow, Y. Kim, and D. Park, "The Hebbian-LMS learning algorithm," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 37–53, Nov 2015.

[19] A. L. Hodgkin and A. F. Huxley, "A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes," *J. Physiol.*, vol. 117, pp. 500–544, 1952.

[20] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

[21] Y. LeCun and C. Cortes, "MNIST handwritten digit database," http://yann.lecun.com/exdb/mnist/, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/