Spiking neural networks for vision tasks

ADVANCED SEMINAR

submitted by
Henry Martin

NEUROSCIENTIFIC SYSTEM THEORY
Technische Universität München

Prof. Dr Jörg Conradt

Supervisor: M.Sc. Lukas Everding
Final Submission: 15.12.2015

TECHNISCHE UNIVERSITÄT MÜNCHEN
NEUROWISSENSCHAFTLICHE SYSTEMTHEORIE
PROF. DR. JÖRG CONRADT

2015-10-01

A D V A N C E D S E M I N A R

Spiking neural networks for vision tasks

Problem description:

Neural networks have achieved striking results in object recognition tasks lately. However, most networks, like standard convolutional networks, work on full images/frames and are expensive with respect to computing resources. This heavily restricts their use in real-time applications. To overcome this, research has been going in the direction of fast networks and more efficient visual coding. One example of this are frame-free spiking convolutional nets: they use event-based vision streams generated by novel vision sensors (DVS [1]) instead of full frames - as generated by conventional cameras - as input and process data asynchronously. For this project, we want you to have a look into the capabilities and limits of spiking neural nets for machine vision tasks and compare them to traditional approaches.

- Get familiar with the fundamentals of convolutional neural nets
- Explain spiking neural networks in vision and their advantages/ disadvantages
- Compare them with regular CNNs in terms of performance, areas of use etc.
- Give examples for applications

Supervisor: Lukas Everding

(Jörg Conradt)
Professor

Bibliography:

[1] Lichtsteiner, P., Posch, C. and Delbruck, T. *A 128 times; 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor* IEEE Journal of Solid-State Circuits Feb. 2008 p. 566-576

Abstract

In the past few years, convolutional neural networks had tremendous success in computer vision tasks such as object detection or face recognition. Despite their success, high computational complexity and energy consumption are limiting their usage for mobile applications and robotics. Therefore, scientists are working on the next generation of neural networks which use event-based spikes to encode information. Spiking neural networks seem to be more efficient in terms of power consumption and algorithmical complexity, but what are the capabilities and limits of this type of networks and how do they perform in comparison with regular convolutional neural networks?

To answer this question, this work gives a rough overview on regular neural networks, the basic neuron models and the benefits of a convolutional architecture. It then proceeds with an introduction to spiking neural networks. Both network types are compared in terms of availability of training data, technology readiness level, speed and efficiency.

At last, the most relevant examples of spiking neural network applications in computer vision are presented and literature for further reading is proposed.

Contents

1 Regular Neural Networks

1.1 Introduction

Even if neural networks are known for over 50 years, their widespread use began only in the last few years. Although they achieved impressive results in simpler computer vision applications like handwritten digits recognition [1] they where believed to be unsuitable for more complex problems like object detection. It was not before 2012 when A. Krizhevsky et. al [2] proposed a deep convolutional neural network at the ImageNet Large Scale Visual Recognition Challenge(ILSVRC) which outperformed its competitors by far. Since then, CNNs are successfully used to solve various computer vision problems like object detection or face recognition.

1.2 Generation and neuron models

In figure 1 we can see, a neural network with three layers. In the conventional architecture, all neurons are connected to all the neurons of an adjacent layer. When zooming in on a single neuron, like it is done in figure 2. We see the edges of the neurons of the proceeding layer which are weighted and summed to form the input to the neurons activation function. It is important to note that every neuron has only one output value, which then may be weighted independently by every proceeding neuron. The choice of the activation function has a critical impact on the behavior of the neural networks and defines its generation.[8]

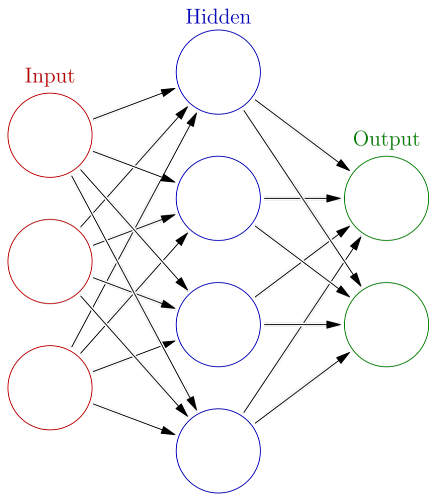


Figure 1: Example of a fully connected neural network with 3 layers[1]

Neurons of the first generation have only two possible outputs (0 and 1), this made the networks unstable because a very small change near the threshold could have a huge influence on the whole network. The neurons of the second generation have continuous output which solves this problem.[5]. Figure 3 shows the activation function of first generation neurons and two examples for continuous activation functions. While sigmoid activation functions where very popular, due to the advantage of the learning behavior,[5] the trend shifted back

[1] https://en.wikipedia.org/wiki/File:Colored_neural_network.svg

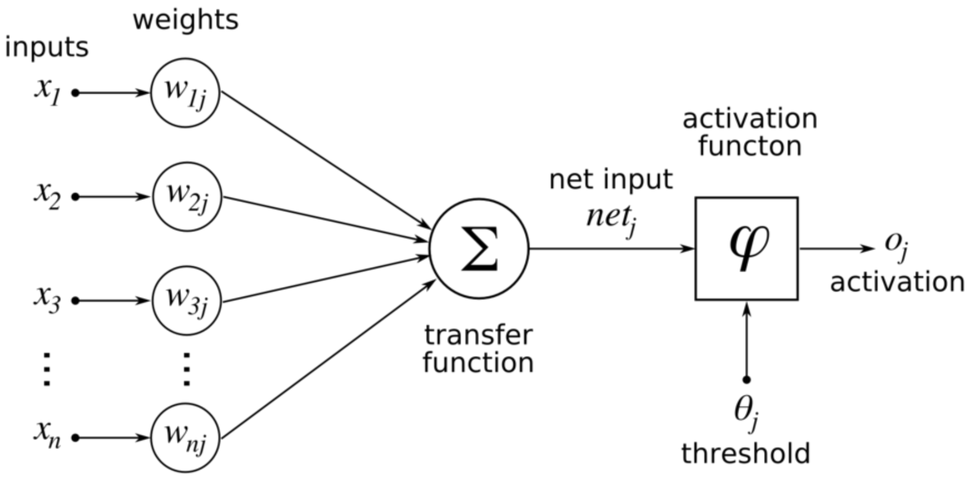


Figure 2: Overview of the elements in a classical neuron model.[2]

to rectified linear activation functions when networks grew bigger. The reason is, that gradient based learning methods, such as the error backpropagation algorithm, are multiplying gradients of the activation functions of many connected neurons and as sigmoid activation functions have a gradient in between 0 and 1, their product becomes infinitely small. This effect is known as the vanishing gradient problem[3] and can be avoided by using linear activation functions with a gradient of 1.

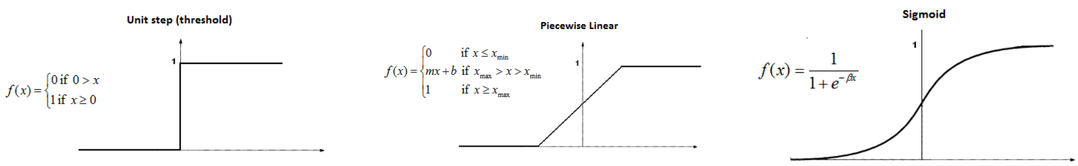


Figure 3: The three most common activation functions for neurons. From the left, to the right Step function for neurons of the first generation, rectified linear function and sigmoid function for neurons of the second generation.[3]

1.3 Convolutional Architecture

At the end of the 1990 one of the most important limitations for neural networks was still the computational complexity of learning with lots of variables.[6] Even if CPUs and GPUs got faster, the convolutional architecture of neural networks was the key to their success. The key contribution of the convolutional architecture

[2] https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions

[3] http://chem-eng.utoronto.ca/~datamining/dmc/artificial_neural_network.htm

is the dimensional reduction. In a convolutional layer, neurons are connected section-wise to the next neuron. These sections overlap partially like in figure 4 and the edges of every section share the same weight. Another layer which is part of the convolutional architecture is the so called pooling layer. The pooling layer, example shown in figure 5, summarizes information of a number of input neurons into one single output neuron. Commonly used pooling functions are the maximum or the mean function.

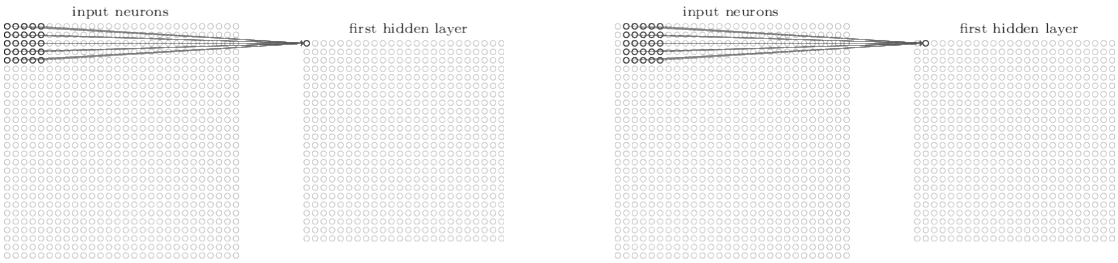


Figure 4: Shows the connection of neurons in a convolutional layer. The inputlayer is connected section wise to the neurons of the next layer. The overlapping of these sections is similar to a mathematical convolution which named this layer. Edges of the same section share the same weight which also reduces the number of variables in a network.[4]

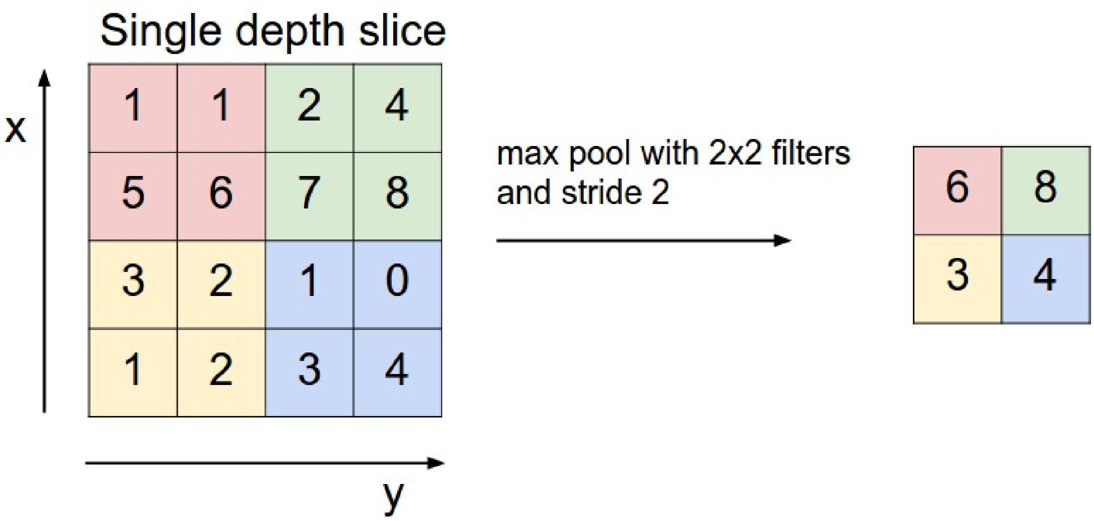


Figure 5: shows an example of a pooling layer. The main goal of a pooling layer is to reduce the complexity of a network. Therefore the inputs are summarized using a pooling function such as choosing the maximum of the inputs or calculating their mean. In this figure, the input is a 2x2 matrix and the output is the maximum value of the matrix. The grid is then moved by the stride. [5]

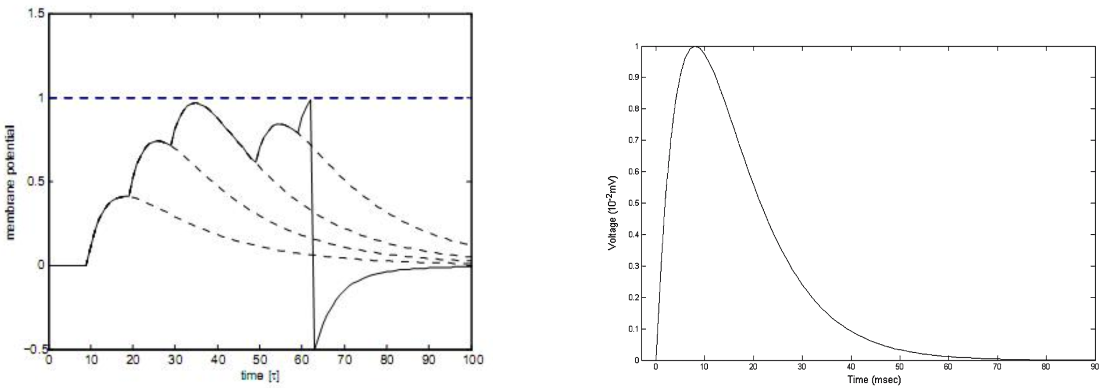
[4] http://neuralnetworksanddeeplearning.com/chap6.html
[5] http://cs231n.github.io/convolutional-networks/#pool

2 Spiking Neural Networks (SNN)

2.1 Introduction

The name spiking neural network as well as the term *neural network of the third generation* applies to the deployed neuron model which uses spike formed impulses instead of a constant time invariant value as output. Unlike conventional neurons, spiking neurons do not operate on a discrete time basis but will fire a spike whenever their membrane potential crosses the firing threshold. This can be seen in figure 6a, where the membrane potential is increasing due to incoming spikes (also called events) until the the firing threshold is crossed. The membrane potential then drops and a spike, as seen in figure 6b, is fired to all connected neurons. The incoming spike will then increase their membrane potential depending on the weight of the connection.

While the information in conventional neuron models is encoded in the amplitude of the output, the amplitude of a spike is constant. There are different ways to encode information using spikes, examples are spike-rate dependent coding or spike-timing dependent coding. The way information is encoded in the brain is still an open research topic and not in scope of this work. If interested, the reader can get a quick overview in [8] or find detailed information in [9].



a: Membrane potential of a leaky integrate and fire neuron

b: Simple example of an output spike

Figure 6: a) shows the curve of the membrane potential of a Leaky-Integrate-and-Fire neuron. Incoming spikes increase the membrane potential,depending on how they are weighted. Membrane potential decreases over time over time (dashed line shows the curve if there would have been no new input). If the potential crosses the firing threshold, a spike, as shown in b), is released. Spikes may be released at moment whenever the threshold is crossed.[6]

[6] `http://neuralnetworksanddeeplearning.com/chap6.html`

2.2 Motivation

Even if the convolutional architecture manages to reduce the complexity of neural networks significantly, the trend towards bigger and deeper networks is steadily increasing the needed amount of computational power. As this demand increases faster then the advance in hardware development, regular CNNs are unsuitable for mobile applications. Therefore research is looking for new, more efficient methods to implement neural networks. Inspired by the brain, which is able to solve complex tasks with very low power consumption, research goes towards biological more plausible neuron models which operates event-based and are therefore more efficient in terms of energy consumption and computational complexity [7]

2.3 Spiking neuron models

Similar to the neurons of the second generation, there are different artificial neuron models which model different parts of the biological neuron. Choosing a neuron model is usually a trade-off between biological plausibility and complexity which can be seen in figure 7. Following is a short overview over the most common neuron models.

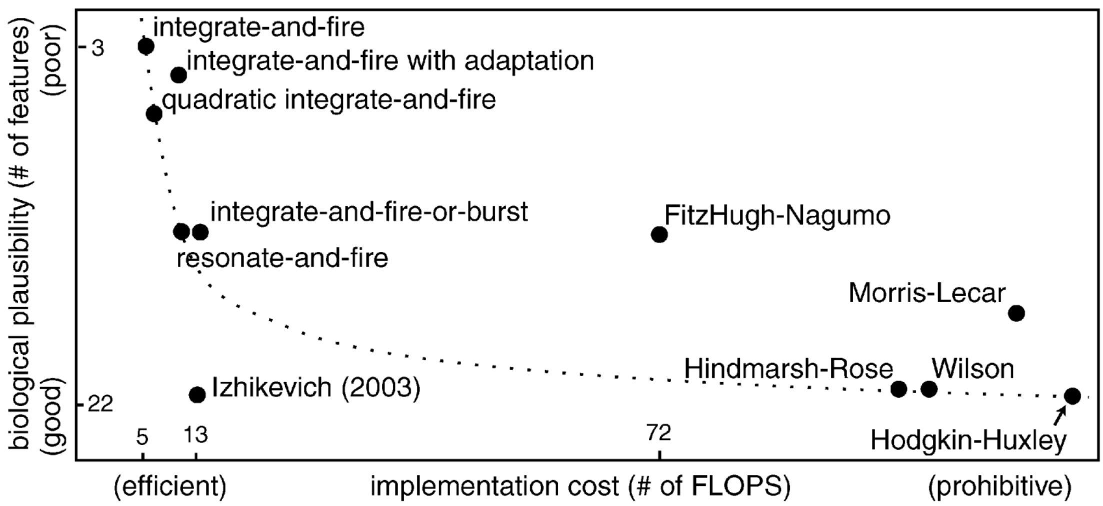


Figure 7: Figure from [10], ranks different neuron models by biological plausibility, which is defined by the number of different spiking behaviors or features a model can reproduce and by the computational complexity which is needed to simulate a model

Hodgkin-Huxley

Presented by A. L. Hodgkin and A. F. Huxley in 1954 and awarded with the Nobel price in medicine in 1963, this model is one of the most known neuron models. Its key feature is the accuracy with which it models the biological behavior of real neurons. The price for this accuracy is a high complexity which disqualifies the model for usage in bigger networks. It is nevertheless important, as it can be used to derive simpler model.

Leaky Integrate-and-Fire (LIF) neuron

The Leaky Integrate-and-Fire is one of the most common spiking neuron models used. It is Its main advantage is its simplicity. The LIF neuron integrates all input spikes which increases the membrane potential until it reaches the firing threshold, then it drops and the neuron sends a spike as output. If no input event occurs, the membrane potential slowly decreases (leaking) to zero[7]. An example of a LIF neuron is shown in figure 6.

Izhikevich neuron

In 2003 E. Izhikevich presented a simple neuron model [10] which is able to reproduce most of the biological spiking behaviors without being particularly complex. Due to the fact that this model seems to be both, efficient and plausible, it has attracted lots of attention. Today it is not yet widely used in neural networks, as more complex spiking behaviors are not yet controllable for learning and information coding. [13]

3 Comparison of regular convolutional neural networks and (convolutional) spiking neural networks

3.1 Availability of suitable training data

While frame based neural networks are widely used, spiking neural Networks are still in their infancy. One point that slows down the research on SNNs is a lack of available datasets.[11] While there are millions of ground truth annotated Images available, the number of labeled, event-based, frame free datasets is small. Especially big, event-based benchmarking datasets, which would encourage competition, are rare. One of the reasons for this deficit is the difficulty of annotating event-based video data.[11] Until real event-based benchmarking datasets are available, a compromise is to transform frame-based datasets into frame-free ones, like it is done in [12]. These transformations allow to develop first applications for SNNs but it can only be an interim solution as it is unlikely that SNN will be able to show their full potential in terms of speed and mobility without datasets which are tailored to their needs[11][12]. Another problem is, that those datasets are often flawed like it is shown in figure 8, where the monitor refresh rate is seen in the event-based dataset.

3.2 Technology readiness level

The understanding of spiking neural networks is not yet as broad as of regular neural networks. Reasons are, that the focused research on spiking neural networks began recently after regular neural networks have become successful and that biological inspired neurons are more complex and more difficult to understand than neurons of the first two generations. This low technology readiness level makes working with spiking neural networks more difficult, efficient learning algorithms are for example still an open research problem. [13]

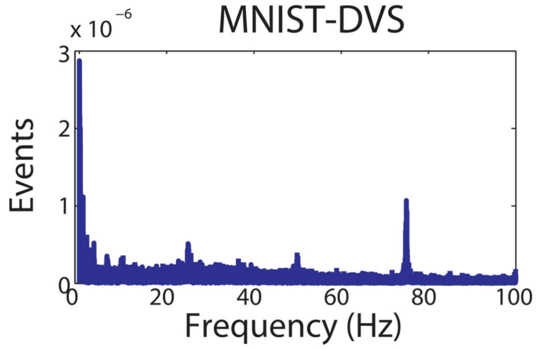


Figure 8: Shows a problem that occurs when transforming frame-based datasets into frame-free datasets by filming them with a DVS camera is that the temporal resolution of the DVS camera is higher than the discrete refresh rate of the screen. Here we can see a Fourier analysis of the MNIST-DVS dataset. We can observe a significant peak at 75Hz which is refresh rate of the monitor.[12]

3.3 Speed, energy efficiency and computational complexity

A big difference in between regular neural networks and spiking neural networks is the speed with which information can be processed. While regular NNs process information synchronized, that means that every neuron in a layer is evaluated before the information can progress to the next layer, SNNs process information in an asynchronous, event-based way. Event-based processing has several advantages, first of all, a neuron is only activated, when it is addressed by an event. This allows the spiking neural network to be much more energy efficient. An example for a very energy efficient implementation is presented in [18], where a processor was build based on spiking neurons. It consumes up to 1000 times less energy than common processors. That a neuron is only activated when a event occurs, means also, that the neuron needs only to be evaluated in this case, while in a conventional neural network all neurons have to be evaluated in every time step. This makes SNNs less computational demanding.

Secondly, a neuron can respond directly to a event and does not have to wait until all the neurons in a layer are evaluated, nor to the next discrete time step to fire its response. The ability of an SNN to process information without delay is also called pseudo simultaneity[13]. This is especially important in combination with

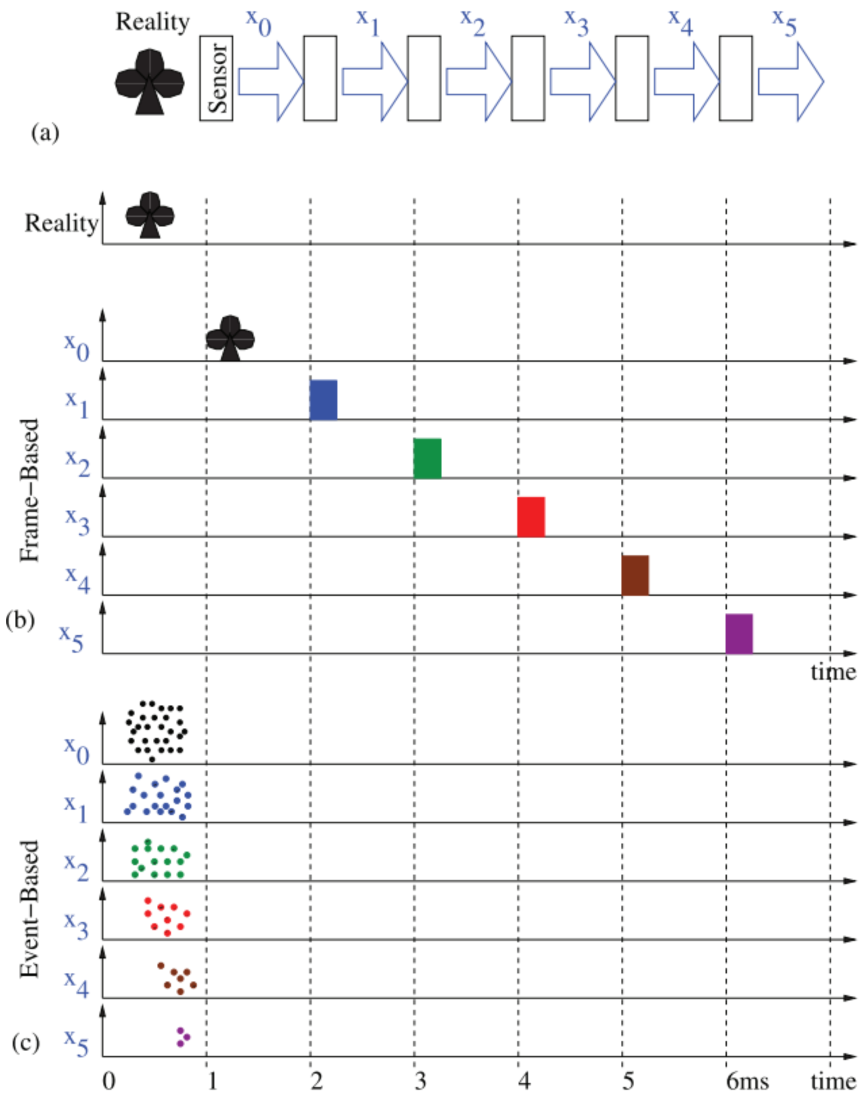


Figure 9: This figure shows the differences in speed of regular frame-based vision system connected to a CNN and an event-based vision systems connected to a SNN. a) shows an abstract view of the architecture and the input. Both systems get a clubs symbol as input and have 5 processing stages. While the regular system in b) is dependent on the frame time of 1ms, the event-based system in c)can process information as it comes. It can be seen that a deeper layer starts fire spikes when the first stage is not yet done with processing.[13]

event-based neuromorphic hardware such as DVS cameras. Figure 9 shows the difference in speed when processing event-based information. The regular CNN works with discrete time steps and information can only progress one layer in every time step. On the contrary, information in the SNN is processed as it arrives and can progress through the network without having to wait for the next discrete time step.

4 Examples for Spiking Neural Networks applications in computer vision

4.1 Mapping conventional learned CNNs to SNNs with applications to poker symbol recognition

A method that can simplify the work with SNN was proposed in [13]. While learning methods are very well developed for frame based CNNs, they are still an open research problem for frame free spiking neural networks. the presented method avoids this problem by transforming a regular CNN, trained with conventional learning methods, into a SNN which is then able to solve the same problem.

The result wished in [13], is a convolutional SNN that recognizes the card symbols off an event-based dataset. It is build from a DVS recording of hands browsing a poker deck. An example can be seen in figure 10. To train the regular CNN, the data has to be frame-based, therefore images are generated by collecting events during a frame times of 30ms. These images where then used to train the frame based CNN using error back propagation.

After the learning procedure, a SNN with the same architecture and the same neuron connections is created. In [13] a set of equations to parametrize the LIF neurons calculate their weights is presented. After the mathematical transformation, simulated annealing optimization routines are used to fine tune the parameters.

The resulting SNN was fed with the testset and was able to recognize in between 97.3% and 99.6% of the symbols. The approach presented by [13] looks very promising as it evades the difficulties that directly learning a SNN poses and allows to use the knowledge from CNNs.

There are similar approaches where fuully learned regular CNNs are transformed into SNNs but where conventional frame-based datasets are preprocessed into frame-free event-based datasets. This approach was taken by [16] to recognize handwritten digits which can be seen in section 4.2 and by [15] and [19] to detect objects as it is presented in section 4.3.

4.2 Handwritten digits recognition

Regular CNNs have achieved striking results in computer vision applications, therefore and because of the attractive combination with event-based DVS cameras, research on SNN is heading towards computer vision applications. Due to the poor availability of suitable datasets, research is concentrating on the few existing benchmarking datasets. One of the most commen ones is the MNIST-DVS dataset. An event-based version of the MNIST, handwritten digits, dataset which was used in [1]. It is notable, that the MNIST-DVS dataset consist only of 10'000 samples of handwritten digits while the original MNIST dataset consists of 60'000 samples.

In [16] an approach similar to [13] presented in section 4.1 was taken, only here, preprocessing is used to convert the frame-based dataset into a event-based dataset by generating Poisson distributed spike trains based on the intensity value of a pixel. This allows the usage of the whole 60'000 samples for training and testing. In [16] the prediction error using a deep convolutional SNN was reduced to 0.9% compared to 0.21% with regular CNNs[7]. Notable is, that with a comparable prediction performance, the SNN evaluates the result five times as fast. Another approach is presented in [17] which uses the tempotron learning rule to directly learn a SNN from the MNIST-DVS dataset. They achieved a recognition rate of 88.14% but had only the reduced number of samples available.

4.3 Object classification on the CIFAR-10 dataset

In [19] Y. Cao et al. present a method to convert a regular CNN into a convolutional SNN which then can be implemented on more efficient neuromorphic hardware. Unlike the method presented in section 4.1, Y. Cao et al. do not train their regular network with DVS data which was converted into frames, but they train their regular network on the original dataset, convert the trained network into a spiking neural network and use a preprocessing step to convert the regular input images into frame-free event based input for the SNN.

This approach allows them to test their spiking neural network on a wide-range of available frame-based datasets. They are benchmarking their network on the CIFAR-10 dataset, which consists of 60'000, 32 x 32 pixel, labeled images from ten categories (for examples: Bird, dog, airplane or truck). CIFAR-10 is a well known classification benchmarking dataset which allows to compare the results of regular CNNs with SNNs. As regular neural networks achieve error rates below 1% the successor CIFAR-100 was introduced.

[7] Recent estimation results can be seen under: `http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#4d4e495354`

Their transformed SNN achieves an error-rate of 22.57% which is worse than the original network from A. Krizhevsky et al. presented in [2] which achieved 14.63% and which they used as model. In [15] Hunsberger et al. use a similar approach. But they present a new way which allows to transform a regular CNN into a SNN made from LIF neurons. They do this by smoothing the LIF response function so that rectified linear activation functions from the regular neuron model can be fitted to the slightly modified LIF neurons. In their paper they present the first deep convolutional spiking neural network which uses LIF neurons and achieves an error of only 17.05% which is close to the original result from A. Krishevsky in [2] which was also the model for them.

4.4 Hand posture estimation

In [14] Q. Liu and S. Furber trained a spiking neural network to recognize simple hand postures. The network runs on a spiNNaker chip, a computer architecture for SNNs, and gets input from a DVS camera. They present a big and a small version of the same network to fulfill this task. Both are tested under real live condition, where they can recognize hand postures in real time with an accuracy of 93% for the big one and 86.4% for the smaller network. Notable is, that the smaller network uses only 10% of the resources while it still achieves 92,9% of the performance.

4.5 Human action recognition

In [17] a network of leaky integrate-and-fire neurons was successfully used to recognize human action. The network was trained with the tempotron learning rule using the *AER Posture Dataset*, an event-based dataset of small video sequences which show humans performing simple actions like walking, sitting or bending. Their network reaches a recognition rate of 99.48% which probably means, that the dataset is to simple. Nevertheless their work is an interesting example on how SNNs can be used for event-based video sequences.

4.6 Examples of non-vision applications

There are some other applications of SNNs which might be interesting but are not in the scope of this work and are therefore only mentioned. A recent example from robotics is [20] where a SNN is used for the indoor navigation of a robot. Another area for SNN applications is the analysis of spatio-temporal data such as it was done in [21] for speech recognition applications or in [22] where a SNN is used to analyze and understand brain data. Next to the famous research topics there

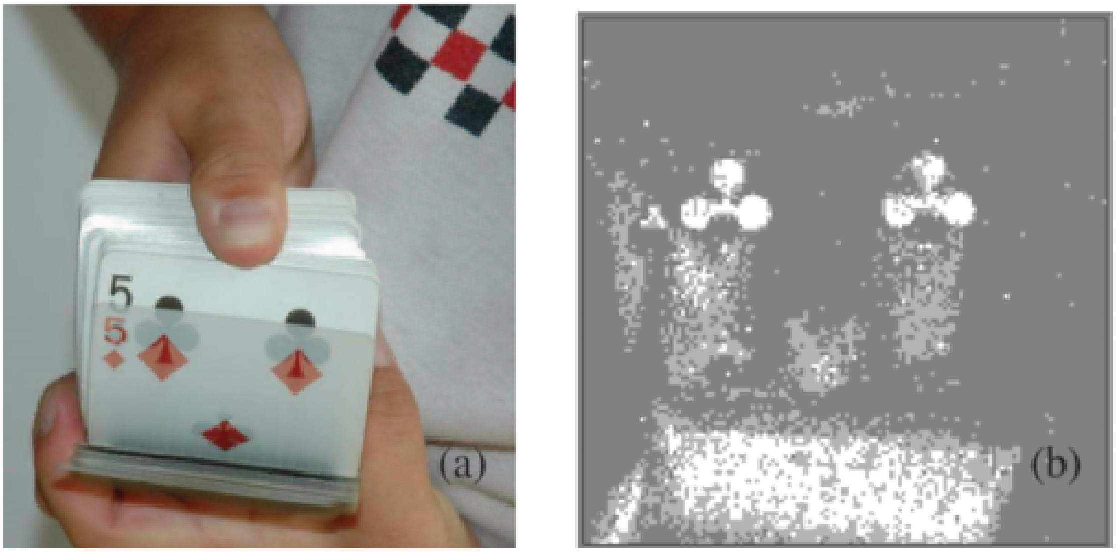


Figure 10: The left picture shows the creation of the dataset with a normal frame-driven camera, the picture on the right shows the same picture from a frame-free camera obtained by collecting events for 5ms.

Figure 11: Examples of the different hand postures used in [14]. The postures are from left to the right: *Fist*,*Index Finger*,*Victory Sign*, *Full Hand*,*Thumb up*.

are also some niches like in [23] where a SNN is used to build a biological more plausible nose which is then used for tea odour classification.

5 Conclusion

Frame-free spiking networks have important advantages over regular, frame-based neural networks. They are more energy efficient, less computational complex and faster. These are important requirements for mobile and robotic applications. But spiking neural networks cannot compete yet with regular neural networks in terms of performance. Reasons therefore are a lack of suitable, event-based datasets and not yet fully developed learning algorithms.

It is possible today to avoid those problems by recording regular datasets with a DVS camera or by transforming fully learned regular neural networks into spiking neural networks. This simplifies the application of spiking neural networks for vision tasks such as handwritten digit recognition or object recognition.

Even though these are good preliminary solutions, it is unlikely that spiking neural networks can develop their full potential while working with datasets or algorithms which are tailored for the needs of frame-based neural networks. To explore the full potential of spiking neural networks, the development of efficient learning algorithms and the creation of real event-based benchmarking datasets is needed.

List of Figures

References

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner Gradient-Based Learning Apllied to Document Recognition Proceedings of the IEEE, 1998

[2] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton ImageNet Classification with Deep Convolutional Neural Networks Advances in Neural Information Processing Systems 25 (NIPS), 2012

[3] Sepp Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6.02 (1998): PP. 107-116.

[4] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. International Conference on Artificial Intelligence and Statistics. 2011.

[5] Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka, Elements of Artificial Neural Networks Elements of artificial neural networks. MIT press, 1997: Pp 9-16

[6] Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka, Elements of Artificial Neural Networks Elements of artificial neural networks. MIT press, 1997: P. 85

[7] O'Connor, P., Neil, D., Liu, S. C., Delbruck, T., & Pfeiffer, M. Real-time classification and sensor fusion with a spiking deep belief network Frontiers in neuroscience, 7. ,2013

[8] Daniel Kunkle, Chadd Merrigan Pulsed neural networks and their application Computer Science Dept., College of Computing and Information Sciences, Rochester Institute of Technology, 2002

[9] Gerstner, W., & Kistler, W. M. Spiking neuron models: Single neurons, populations, plasticity. Cambridge university press. (2002).

[10] Eugene M Izhikevich, Which model to use for cortical spiking neurons? IEEE transactions on neural networks 15.5 (2004): 1063-1070

[11] Tan, Cheston, Stephane Lallee, and Garrick Orchard. Benchmarking neuromorphic vision: lessons learnt from computer vision. Frontiers in Neuroscience 9 (2015).

[12] Orchard, G., Jayawant, A., Cohen, G., Thakor, N. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades Frontiers in Neuroscience 2015

[13] Pérez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S. and Linares-Barranco, B. Mapping from Frame-Driven to Frame-Free Event-Driven Vision Systems by Low-Rate Rate Coding and Coincidence Processing–Application to Feedforward ConvNets. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(11), 2706-2719. (2013)

[14] Liu, Q., and Furber, S. Real-Time Recognition of Dynamic Hand Postures on a Neuromorphic System. World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, 9(5), 432-439 (2015)

[15] Hunsberger, Eric, and Chris Eliasmith. Spiking Deep Networks with LIF Neurons. arXiv preprint arXiv:1510.08829 (2015).

[16] Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C. and Pfeiffer, M. Fast-Classifying, High-Accuracy Spiking Deep Networks Through Weight and Threshold Balancing. International Joint Conference on Neural Networks (IJCNN). 2015;

[17] Zhao, B., Ding, R., Chen, S., Linares-Barranco, B., and Tang, H. Feedforward categorization on AER motion events using cortex-like features in a spiking neural network. IEEE Transactions on Neural Networks and Learning Systems, (2014).

[18] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... & Brezzo, B. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 345(6197), 668-673, 2014

[19] Cao, Y., Chen, Y., & Khosla, D. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition. International Journal of Computer Vision, 113(1), 54-66, 2015

[20] Beyeler, M., Oros, N., Dutt, N., & Krichmar, J. L. A GPU-accelerated cortical neural network model for visually guided robot navigation. Neural Networks, 72, 75-87, 2015

[21] Zhang, Y., Li, P., Jin, Y., & Choe, Y. A Digital Liquid State Machine With Biologically Inspired Learning and Its Application to Speech Recognition. Preprint, 2015

[22] Kasabov, N. K. NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. Neural Networks, 52, 62-76, 2014

[23] Sarkar, S. T., Bhondekar, A. P., Macaš, M., Kumar, R., Kaur, R., Sharma, A., ... & Kumar, A. Towards biological plausibility of electronic noses: A spiking

neural network based approach for tea odour classification. Neural Networks, 71, 142-149, 2015

I hereby certify that this advanced seminar has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged.