Machine Learning: The Past, Present and the Future

Narayan Srinivasa
Intel Corporation

Abstract

Machine learning algorithms of the past were designed to capture the learning capabilities of the brain but with a high level of abstraction of its learning mechanisms. These abstractions resulted in shallow learning models that relied on hand crafted feature extraction on a problem specific basis with limited practical applications. The deep learning models of today represent the next generation of machine learning algorithms that can be trained from raw data using multiple processing layers due to novel modifications to the learning architecture compared to shallow learners. This development combined with the availability of raw data to train these models and the availability of fast affordable computers have enabled a great surge in its utility for many applications including video and audio pattern recognition. By exploiting the fundamentally different computing architecture and mechanisms prevalent in the brain, we believe that the next generation of machine learning called neuromorphic computing will advance the state-of-the-art in this field. In particular, it has the potential to realize energy efficient learning machines that could support a wide range of applications including internet of things, sensor processing, cybersecurity, robotics, mobile devices, diagnostics and prognostics and exoscale computing systems.

INTRODUCTION

Can we build machines that can exhibit intelligent behavior? The answer to this question has drawn our attention to understand and mimic how the human brain functions. So far, most attempts to understand brain function has focused on how the brain can compute intelligent behavioral responses from internal representation of stimuli and stored representations of information of past experience. The roots of this approach can be traced back to two key ideas. Alan Turing's pioneering work [1] in machine theory defined computation as formally equivalent to the manipulation of symbols in a temporary buffer. Similarly, the pioneering work on telephone communication by Shannon and Weaver [2] resulted in a formal definition of information where informational content of a signal is inversely related to the probability of that signal arising from randomness.

As these developments launched computer science into prominence, and as computers grew in functional complexity, so did the analogy between computers and brain. The basic premise for this analogy was that computers and the brain received information from the external environment and both acted upon this information in complex ways. This analogy (also known as the computer metaphor) provided a candidate mechanism to explain intelligent behavior as akin to a digital computer program that can manipulate internal representation according to a set of rules. Furthermore, mental entities in the brain were akin to software, whereas physical mechanisms were akin to hardware.

The extensive use of the computer metaphor resulted in the brain models using network of neural cell like computational elements [3], or artificial neural networks (ANN), as a proxy for the computation in the brain. This was the origin of the field of machine learning. In this paper, we will briefly outline the key ideas that shaped this field in the past and its status today. We will then contrast it with the emergence of neuromorphic computing as the next generation of machine learning, highlight its salient features and related challenges, discuss potential applications of this technology and how it could influence our future.

ARTIFICIAL NEURAL NETWORKS

In an artificial neural network, simple artificial "processing elements" or "neurons", are connected together in layers to form a network which mimics a biological neural network. The connections between the artificial neurons represent tunable adaptive weights (Fig. 1(a)) that serve as a proxy for the synapse in biological neural networks. These neural networks (Fig. 1(b)) are abstracted versions of biological neural networks but similar in performing of functions collectively and in parallel, rather than there being a clear delineation of subtasks to which individual neurons are assigned.

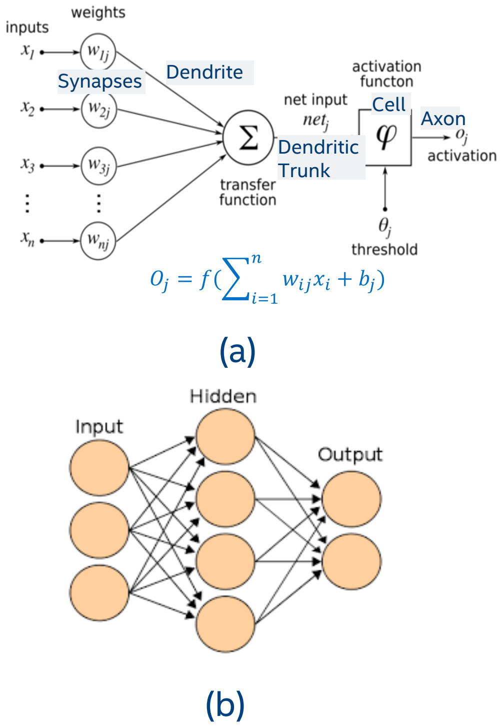


Fig. 1 (a) the artificial neuron model with the biological analogs appropriately marked. The output was a nonlinear function *f* of the weighted inputs. (b) A three layered network with each circle representing one of the artificial neurons connected.

The first interesting network called the *perceptron* was a two-layer network designed for pattern recognition [4]. After going through road bumps with the basic perceptron in its inability to solve the exclusive-or problem [5], the creation of the back-propagation (BP) algorithm [6] not only solved the exclusive-or problem but also enabled the training of these networks resulting in the first generation of practical machine learning models.

Shallow Machine Learning Models

The first generation of machine learning models was commonly designed to be supervised where the goal of the system is to classify input patterns into desired output class labels. During training, the machine is shown a sample from the input data and the corresponding output score for all the categories is provided as ground truth. The ideal outcome after the training process is for the machine to able to infer the correct class for inputs both within and outside the training data. To accomplish this, the training process leverages the BP algorithm. The first step in the process is to compute an objective function that measures the distance (L_2 norm) between the desired and actual scores for the class labels. The BP algorithm then modifies its internal adjustable parameters such as the adaptive weights to minimize this distance across all training data. The BP algorithm computes a gradient vector for each weight that evaluates how the error changes (either up or down) as a function of weight changes (by a small amount up or down). The learning step then adjusts this weight in the opposite direction of the gradient vector. This process is repeated for all the weights in all the layers in the network [7].

After training, the performance of the system is measured on a new data set called a test set that is used to compute the accuracy of the classification on new inputs that the machine has never seen before. While this algorithm worked very well for hand crafted input features in the training data, it did not work well for raw data. This is because with small initial weights, error gradients using BP in early layers are very small and this gets worse with depth (number of layers) of network. Thus only *shallow networks* (depth <= 3) could be trained. Furthermore, BP algorithm was also found to be susceptible to becoming stuck in local minima with large initial weights. This made shallow learners sensitive to irrelevant details such as illumination changes or variations in the pitch or accent of speech [8]. Thus, there was a need for considerable skills to create hand crafted features that could ensure high performance. This was a key limitation in the adoption of shallow learners for real-world applications.

Deep Machine Learning Models

The most common machine learning models used today are deep learning models that are composed of multiple stacks of layers or modules with millions of parameters for tuning (Fig. 2). Each layer computes a nonlinear input-output mapping that increases both the selectivity and invariance of the representation. This feature of deep learning models enables the network to compute intricate functions of inputs from raw data that are sensitive to object details while being insensitive to irrelevant variations –illumination etc. There were two flagship models that achieve these capabilities. The first deep learning model are the deep belief nets based on the Restricted Boltzmann machine (RBM) and the second is the convolutional neural network.

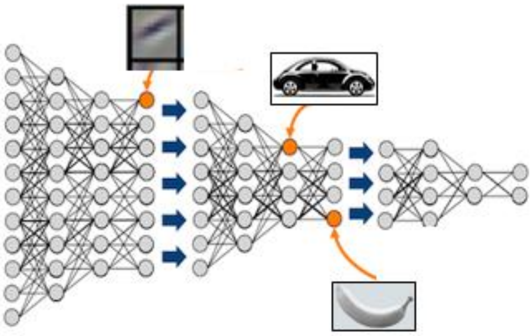


Fig. 2, A deep learning model with multiple layers where the neurons in the earlier layers represent low level features such as oriented edges while the neurons in the higher level represent high level concepts such as cars and banana.

The RBMs were invented by Geoffrey Hinton and his group [9] for the purpose of unsupervised of features from unlabeled data [10]. The objective of each layer was to reconstruct or model activities of features (or raw inputs) in the layer below. As a first step, the raw data in the first layer was used to construct the feature detectors in the second layer and the second layer in turn was used to reconstruct the raw data (Fig. 3). This process was repeated pairwise for all pairs of layers to create a deep auto encoder also known as the RBM. This unsupervised process forms a crucial first step in training RBM based deep learning networks and is commonly referred to as *pretraining.* Using pre-training enabled the deep learning system to be seeded with sensible initial weights in an *unsupervised* fashion and the process was scalable to arbitrarily large depths of network layers.

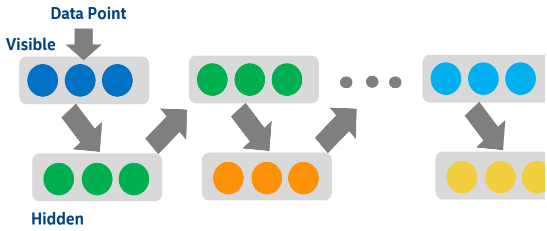


Fig.3. the pre-training process for the RBM is illustrated here. The input data point forms the visible layer that the second layer of neurons are trained to represent during learning. This hidden layer then forms the visible layer for the second pair of layers and the process is repeated until all sequential pairs of layers in the forward direction of the deep learning network are trained.

As a second step, all the layers are then combined and the full network is trained using BP with these initialized weights. This step is referred to as *fine tuning.* The interesting aspect of fine tuning is that the network converges readily compared to shallow learners while producing very low errors. Another huge advantage is that both pre-training and fine tuning scale linearly in space and time with the number of training cases. This enables RBM based deep learning models to be trained with very large data sets but without needing much training time.

Inspired by the hierarchical visual cortex model by Fukushima [11], another deep learning model was developed by Lecun [12] called convolutional neural networks. This model was designed to process data that come in the form of multiple arrays – 2D/3D images, signal sequences including language etc. This deep learning model was composed of a series of stages (Fig. 4). Each stage composed of two layers: convolutional layers and pooling layers [12]. The neurons of the convolutional layer is organized into feature maps where each neuron in this map is connected to local patches in map of previous layer via a set of weights called a filter bank. Since this filtering operation is a discrete convolution, the deep learning network was called convolutional networks (ConvNets).

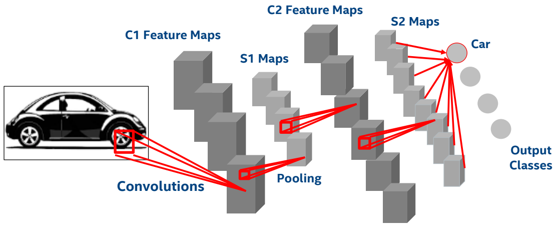
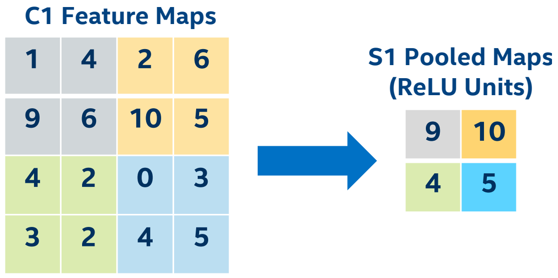


Fig. 4. A two layered convolutional network showing the convolution (C#) and pooling (S#) operations where # represents the layer number. The last layer represents the classification layer or the outputs of the network.

The pooling layer merges semantically similar features into one. A typical pooling neuron (also known as a rectified linear unit (ReLU)) computes the *maximum* of a local patch of neurons in the feature map (Fig. 5). Neighboring pooling neurons take input from patches that are shifted by more than one row/column thus providing shift and distortion invariance in its processing. Multiple stages of convolution and pooling with a final coding layer forms the ConvNet.

The training for the ConvNets was based on the BP algorithm where the weights in all the layers of the deep learning model was adapted based on the errors in classification. Unlike shallow learners, these networks also converged readily since the ReLU units offers more structure to the network and the weights are well-conditioned as a result for learning.

There are three factors that enabled the rapid adoption of deep learning models as a flagship approach in machine learning today. The first factor is the ability of these networks to be trained even with many layers of depth thanks to innovations such as the RBM and ConvNets. The second factor is the availability of fast enough computers with affordable large memories that enabled these deep networks with many millions of parameters to be trained. The third factor was the availability of large data sets for training and benchmarking due to emergence of the Internet and fast computers that provided the raw data (such as images and audio data) with ground truth. These approaches are now being explored from a software perspective by major companies like Google, Facebook, Microsoft, IBM, Yahoo!, Twitter, and Adobe. They are rapidly developing image understanding products and services using deep learning models. In parallel, these models have been implemented in hardware by companies such as Intel, NVIDIA, Mobileye, Qualcomm and Samsung to enable real-time vision applications for smartphones, cameras, robots and self-driving cars [8].



C1 Feature Maps

1	4	2	6
9	6	10	5
4	2	0	3
3	2	4	5

S1 Pooled Maps (ReLU Units)

9	10
4	5

Fig. 5. A ReLU based computational process where the max of each local patch of features is extracted by the pooling process.

NEUROMORPHIC COMPUTING

The original term in this field was the term “neuromorphic electronics” coined by Carver Mead [13] at CalTech in the late 80’s to describe electronic *analog* circuits that *mimic* neurobiological circuits and architectures in the nervous system. Lately the term “neuromorphic engineering/computing” was introduced to expand the scope to include analog, digital, mixed-mode analog/digital VLSI and software systems and algorithms. This is a multidisciplinary field with skills ranging from computer science and engineering, physics, mathematics, neurobiology, psychology and computational modeling. It is also a very dynamic field where our understanding of the nervous system is changing all the time due in part to better measurement tools such as optogenetics, viral tracing, better understanding of cellular features, better models that capture this understanding and new insights and theories that makes this field both exciting but also challenging.

The foundation of neuromorphic computing unlike the machine learning algorithms lies in understanding and exploiting how biological computation is very different from digital computers of today. Brain is composed of very noisy analog computing elements including neurons and synapses. Neurons operate as relaxation oscillators. Synapses are implicated in memory formation in the brain and can only resolve between 3-4 bits of information at each synapse [14]. The dynamics of these elements are asynchronous and thus *clock-free* [15, 16]. Since these synapses are fully distributed and they are implicated in memory, this implies that, in general, there are no single synapse or single neural firing activity that corresponds to a particular item or concept and so are *symbol free* [16]. The effective integration of heterogeneous and non-local sources of information for multiple goals and is a hallmark of human-like cognition [16, 17]. The brain thus operates in a *grid-free* fashion. Finally, the scale of neuronal interactions can span from a few neurons all the way to the entire network of neurons in the brain during various time instances that depends upon on the context of its interaction with its environment. This implies that the brain is a complex physical system whose dynamics is *scale-free* [18].

While these features of neuromorphic computing are foundational, there are four clear focus areas that are being actively explored today to design and architect the next generation of machine learning algorithms and hardware systems. We will now highlight these areas and highlight the progress and challenges in all these four focus areas.

Spike Based Representations

Brains operate using spike-based codes that often appear sparse in time and across populations of neurons. A neuron generates these spikes in response to spikes from other neurons by integrating the current that leaks

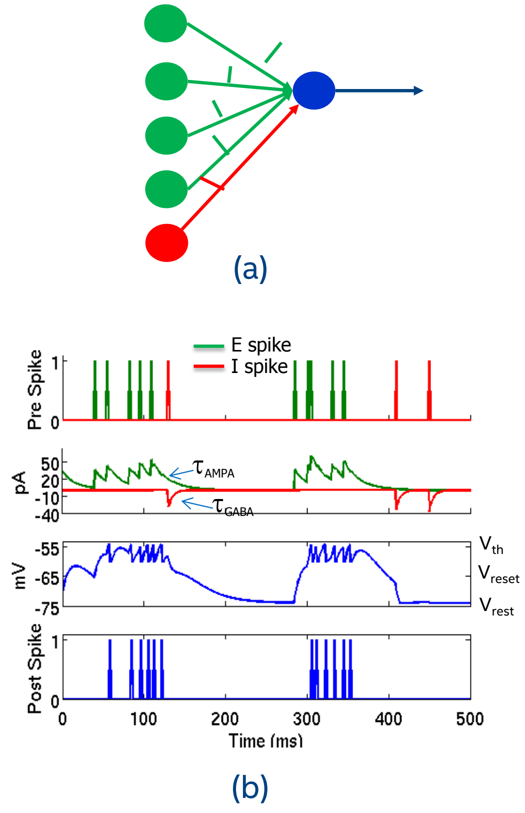


Fig. 6. (a) A simple 5 input neuron network with four excitatory neurons and one inhibitory neuron to a single post-synaptic neuron. (b) The input spikes shown in the top row gets converted to input currents (the green and red traces – the second row). The membrane voltage and the corresponding spikes of the post-synaptic neuron are shown in the last two rows.

into the soma and generating a spike of its when the net current exceeds a threshold (Fig. 6). Spike-based representations offer the advantage that sensory information is encoded by relatively small populations of spikes and their precise relative timing.

This capability can be very useful because it offers an efficient solution by generating sparse codes [19]. In particular, it provides an approach to extract and represent useful information from high-dimensional data such as video images by extracting the most relevant information from just a few spikes [20]. There seems to be some understanding on how such spike-based representations can be realized for small scaled feedforward and recurrent spiking neural models [21, 22].

Spiking neural networks (SNN) based architectures have recently been implemented [23, 24] to mimic deep ConvNets. The communication between the various stages of ConvNets are through spikes but the synaptic weights are trained off-line and programmed into the model.

Benchmarking of these models using CIFAR-10 datasets composed of real world images show that the performance of these models are comparable with state-of-the-art ConvNets algorithms while realizing energy efficiencies of ~10-20x. A recent spiking model [25] inspired by auditory processing in mammals was found to exhibit feature sensitivity by being robust to a range of variations in the stimulus such as found in naturalistic stimuli and human speech samples. In robotics, applications of spiking neural models including the control of a simulated robot arm [26] and application in Robocup competition [27] have recently appeared.

There are several recent hardware systems that have utilized spike based representations. In the case of sensor processing, a silicon retina was developed [36]. This sensor is a spike-event generating image sensor consisting of 128 x 128 pixels which asynchronously outputs streams of address events in response to relative light-intensity changes (Fig. 7). The events are tagged with the address of the creating pixel, a time stamp, and an ON or OFF polarity tag, which indicates whether the event was created in response to an increase or decrease of light intensity over that pixel. The sensor is able to respond to these changes with extremely low latencies of 15 µs. An address event based silicon cochlea [28] was developed to mimic the biological cochlea. This device transforms input sounds into streams of spikes in 64 channels responsive to different frequency ranges (350 – 1200 Hz) where the inter-spike-intervals was a more precise indicator of the frequency of pure input tones than the distributions of channels from which the spikes originated. These sensors were also recently fused to perform real-time classification that is comparable to conventional CovNets based algorithm performance on the MNIST dataset [29]. There have also been several recent applications that have exploited these types of sensors for real world applications including motion sensing [30] and visual shape tracking [31] and navigation [32] and also for visual information processing [33].

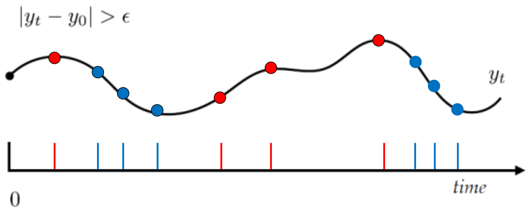


Fig. 7. An event or asynchronous coding of intensity y_t via spikes where the red spikes represent a nominal upward change in intensity compared to a baseline y_0 and blue spikes represent a nominal downward change. The parameter € provides the threshold for change.

IBM developed a TrueNorth chip [34] that integrates 1 million programmable spiking neurons and 256 million configurable synapses. Chips can be tiled in two dimensions via an interchip communication interface. The architecture applied for example, multiobject detection and classification but the synaptic weights were obtained in an offline fashion. For a 400-pixel-by-240-pixel video input at 30 frames per second, the chip consumes 63 mW. Inspired by the brain’s inherent dynamics and plasticity, HRL has developed an efficient, scalable, and flexible non–von Neumann architecture that leverages contemporary silicon technology to learn on-chip [35]. They recently demonstrated the application of this chip on a nano air vehicle [36] to learn on-chip to recognize three rooms in under 8 minutes with ~50x improvement in energy and ~10x improvement in throughput compared to a conventional processor.

The Neurogrid project [37] at Stanford provides an option for brain based simulations by using analog computation to simulate ion-channel activity and uses digital communication to simulate synaptic connections. The Neurogrid can simulate a million neurons. There also other efforts in Europe including the work on FACETS and BrainScaleS projects [38-40] and in the UK SpiNNaker Project [41]. The current version of the Neuromorphic Physical Model (NM-PM) from the BrainScales project [42] incorporates $50*10^6$ plastic synapses and 200,000 biologically realistic neuron models on a single 8-inch silicon wafer in 180 nm process technology.

Challenges with Spike Based Representations: There are gaps in our understanding of how large scale spiking networks could help compute and stably represent information using recurrent architectures that seem to be ubiquitous in the brain. There are interesting hypothesis [44, 45] emerging on how this may be performed but still lack strong experimental support due to lack of experimental tools that can monitor large populations of neurons at the spike level. Another gap in our understanding comes from the fact that spike-based representations are inherently noisy (e.g., latency, jitter etc.). The fact that the brain appears to embrace this as a feature rather than avoid it remains to be fully understood. There are some interesting recent ideas (e.g., Bayesian computation [46-48]) on how this could happen in the brain based on spikes as its representation. Finally, the spike based representations in the brain are invoked both by sensory stimuli and spontaneously when there are no stimuli. Another key gap to address is in our understanding of the role of these two modes of operation for computing (unlike Von Neumann machines that do nothing if there is no input) and interacting with a changing environment.

Asynchronous Computing

It is well known that the brain operates using a plethora of brain rhythms (see [16]) but without any global clock. This asynchronous mode of operation seems to rely on surprisingly precise timing information that cannot be recovered by counting spikes over long temporal windows [53-57]. A recent study [58] suggests that acquiring temporally precise spikes from a visual sensor [28] provides up to 70% more information than conventional spikes generated from a frame-based acquisition as used in standard artificial vision thus enabling more accurate classification. Furthermore, the response times of these sensors are much faster thus enabling the detection of features that cannot even be sensed with regular cameras. This advantage combined with lower energy consumption in producing these precisely timed spikes will result in dramatically lower energy and high throughput products making next generation computing systems endowed with sensors that are more sensitive, efficient and fast in its processing capability.

Recent neuromorphic research chips have demonstrated the same mixed-mode model of computation as found in biological neural systems, namely local analog computation with global digital asynchronous communication [34-43]. Unlike traditional computing applications, these systems typically forego the determinism and reliability of synchronized digital computation. The systems loosely synchronize through the dynamics of time-based (spike) interactions (see [55]).

These neuromorphic architectures optimize for energy efficiency above all else, relaxing the performance, reliability, and deterministic computational constraints of conventional ASIC (Application Specific Integrated Circuit) and CPU architectures. Through its 2011 acquisition of Fulcrum Microsystems [56], Intel now has the most advanced and commercially proven asynchronous design technology and tools in the industry [57] demonstrating best-in-class latency, bandwidth, and power metrics. Today this technology is being directed to more basic research including neuromorphic computing.

Challenges with Asynchronous Commputing: The gap in our understanding stems from how the brain resolves this temporal precision in the cortex and deeper brain areas (i.e., far away from the sensory inputs and motor outputs) given the amount of noise due to the stochastic nature of neural firing and also the interaction between millions of neural cells that are all affected by this muddling of precisely timed spikes with imprecise noise. One school of thought suggests that the cortical areas of the brain operate in a Bayesian fashion [48, 58] to help resolve this ambiguity. Another view point is that while the brain is not deterministic, neuronal variability may often be overestimated due to use of inappropriate reference times and also due to uncontrolled internal variables [59].

The argument is that the problem of reference time can be largely avoided by recording multiple neurons at the same time and looking at the statistical structures in relative latencies with the added advantage that they are insensitive to the variability that is shared across neurons.

While much of the work on spike latency based models emphasize synchrony of firing to enable robust learning in a sea of asynchronous spikes, an alternate viewpoint called polychronization [60] was recently developed. This alternate viewpoint relies on the observation that axonal conduction delays in the mammalian neocortex has a large range depending upon the type and location of neurons [61]. Using these delays, a model developed in [60] showed that when spikes originating from these set of neurons with an appropriate pattern of delays become coincident at a recipient neuron that neuron can generate a strong postsynaptic potential. This in turn can cause synapses to adapt thus forming a polychronous group between these neurons. The number of such polychronous groups far exceeds the number of neurons in the network resulting in an unprecedented memory capacity. New experimental work with the ability to measure large populations of neurons in the future will be needed to help resolve some of these differences in viewpoints in how the brain operates under asynchronous conditions.

Plasticity and Learning

The shallow and deep learning machine learning algorithms relies on the BP algorithm which is not a biologically plausible algorithm. In biology, the synaptic weights are adapted based more local influences. In particular, each synaptic weight between a pair of neurons modifies its strength based on the temporal correlations in spiking activity between the

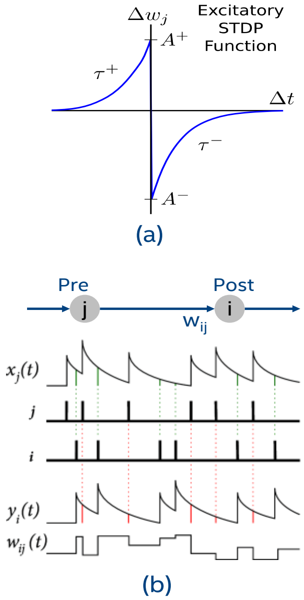


Fig. 8. (a) The temporally asymmetric spike timing dependent plasticity (STDP) rule. (b) The additive STDP changes (red for negative and green for positive) using this rule for synaptic weight w_{ij} as a function of the pre spikes j and post spikes i is illustrated here.

pair. This idea was originally postulated by Donald Hebb [62] and subsequently many forms of Hebbian plasticity have been discovered in the brain. The key observation is that brain exhibits plasticity of various kinds that operates at many spatial and temporal time scales and is continuously on. Thus, the brain is learning using these plasticity rules online.

Online learning enables a computer to directly and continuously interact with its environment thereby enabling a self-organized approach to understanding its world. If the interaction is such that there is feedback provided on the computers performance, then it allows the system to also continuously improve its ability to perform as per the expectations of its user/environment.

The learning mechanisms at short time scales are becoming clear and seems to primarily consist of Hebbian homosynaptic spike-timing dependent plasticity [63-65] (Fig. 8) and other heterosynaptic plasticity [66, 67]. Mechanisms for long-term plasticity appears to involve synaptic consolidation and maintenance [68]. There are other non-standard forms of plasticity including homeostatic plasticity [69], structural plasticity [70, 71] and short-term plasticity [72]. In addition there seems to be neuromodulatory influences [73, 74] on learning as well.

There have been many applications of online learning but using models that are not biologically plausible such as the Adaptive Resonance Theory [75] based models. These models exhibit online learning but address the question of stability by preventing any overwriting of already formed memories using a reset mechanism. As a result, the network does not scale gracefully with increasing number of classes to be stored. The other common online learning approaches are based on the deep learning and its variants [8-10, 76].

An example of an electronic chip exhibiting plasticity was the HRL spiking neuromorphic system. This system has short-term plasticity, spike-timing dependent plasticity and homeostatic plasticity all integrated into a single chip. The demonstration of online learning was performed using this hardware for the application described in [37]. However, in that application, only the short time scale based spike-timing dependent plasticity was used. As mentioned above, this chip experiment resulted in a ~50x improvement in energy and ~10x improvement in throughput compared to a conventional processor and learning happened in self-organized fashion without any human intervention.

Challenges with Plasticity and Learning:
One key gap in our understanding is how these various mechanisms interact during learning to form memories (but see [77] for a recent attempt). The other gap is in our understanding of how memories formed during online learning can be stabilized despite a constant barrage of new information being encountered by the brain and in the presence of noise. A related problem of an extreme kind is that of one-shot learning [78] where the system is only exposed to stimuli once and yet is capable of reliably responding to future instances of that one time event. Finally, there is no clear consensus on how the brain can learn complex episodes (spatiotemporal patterns) and then recall information from its learned memory in a robust and yet rapid fashion.

Co-located memory and computation

The average operating speed of neural circuits is in the order of a few Hz compared to current generation computers that operate in the tens to hundreds of GHz range. Evolution appears to have designed these low speed brain circuits to optimize on energy by avoiding having to transmit information long distance thus facing communication delays and interference. But unlike “weak co-location” in current generation computers where memory units are physically co-located with the processing units but serve the same decoupled functions, neural circuits in the brain are “strongly co-located” (Fig. 9) where in addition to physical proximity of memory and computation, there are learning and plasticity mechanisms co-located with memory and computing. In other words, in this

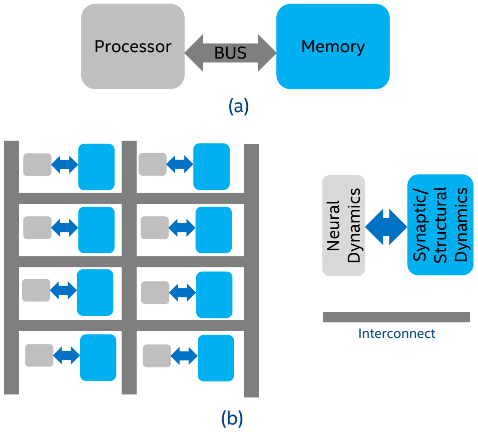


Fig. 9. (a) Memory and processing are fully decoupled in both physical location and functional role in a Von Neumann architecture (b) In a neuromorphic architecture, they are strongly co-located both physically and functionally where neuronal dynamics influences synaptic dynamics and vice versa. Furthermore the connectivity (1 neuron connects to 10,000 synapses) is very large for each neuronal processing unit.

system all memory resources, both for routing digital events and for storing synaptic and neural circuit parameters are tightly integrated with the synapse and neuron computing circuits. The changes in the timing of spike streams passing through a synapse effects changes in the synaptic strength which in turn affects the timing of the spikes and communication speeds to other neurons in the future. Furthermore, since this learning happens in a fully distributed fashion among many synapses, memory of events being processed is stored in the dynamics of the circuits rather than at a single location making it more robust to faults and tolerant to damage unlike current computational systems.

There have been several recent neuromorphic chips that have incorporated co-located memory and computation elements. NeuroGrid [38] designed at Stanford and the TrueNorth [35] designed at IBM are two examples of chips with a large number of processing synapses and neurons where different memory structures are distributed across the network (e.g., in the form of routing tables, parameters, and state variables). The ability of the shared synapses to integrate incoming spikes reproducing biologically plausible dynamics provide the system with computational primitives that can hold and represent the system state for tens to hundreds of milliseconds. However, the design choice to use linear synapses in the system excluded the possibility to implement synaptic plasticity mechanisms at each synapse, and therefore the ability to model on-line learning or adaptive algorithms without the aid of additional external computing resources.

The HRL neuromorphic chip [36] overcomes this bottleneck with a variety of plasticity mechanisms that support complex synaptic dynamics. However, that chip is limited in the number of neurons and synapses supported to address large real world tasks. The BrainScales [39] effort in Europe implements a wafer scale neural simulation platform with co-located neuron and synapse elements. However, in order to maximize the number of processing elements in the wafer, they chose to implement relatively small capacitors for modeling the synapse and neuron capacitances. As a consequence, given the large currents produced cannot achieve the long time-constants required for interacting with the environment in real-time. Rather, their dynamics are “accelerated” with respect to typical biological times by a factor of 10^3 or 10^4. This has the advantage of allowing very fast simulation times which can be useful e.g., to investigate the evolution of network dynamics over long periods of time, once all the simulation and network configuration parameters have been uploaded to the system. But it has the disadvantage of requiring very large bandwidths and fast digital, high-power, circuits for transmitting and routing the spikes across the network.

Applications of Neuromorphic Computing

By enabling autonomous computing for a low energy cost, neuromorphic computing hardware could play a big role in the future of large scale data centers and exoscale computing systems. The main contribution here will come from a mixed mode of computing where computing sub-tasks that can tolerate approximate solutions will be enabled by energy efficient neuromorphic hardware that is integrated within a Von Neumann machine to provide large gains in energy efficiency.

The energy efficient neuromorphic hardware will also create a whole new set of applications from intelligent mobile device to intelligent and efficient IoT devices and systems. This is because neuromorphic chips offer size, weight and area efficiencies as well. Furthermore, since these systems can learn online, they can enable these devices to operate in a self-organized fashion without any manual intervention.

The complex spatiotemporal learning capability will also enable autonomous self-driving cars and drones as well as provide novel cyber security solutions. This online learning capability will also usher in a whole new class of sensors such as cameras that can learn about the pictures it may capture and label the pictures when they similar objects in the future. They will also enable novel home appliances and health monitoring devices that could learn to customize the user experience to their tastes and state of health respectively.

Since these system are capable of learning in a supervised fashion as well, they will usher in novel computing frameworks that learn the physics of complex systems by observation and then replace these systems with matching performance while being faster and much more energy efficient in comparison.

CONCLUSIONS

In this paper, we discussed the field of machine learning by highlighting the salient aspects of the past and present. We believe that the future will be based on neuromorphic computing. With architectural features including asynchronous processing, distributed memory and computation and spike based representations combined with online learning capability, this next generation of machine learing will usher in a whole new computing paradigm and potentially revolutionize our way of living.

REFEENCES

1. A. M. Turing, “On computable numbers with an application to the Entscheidungs problem,” *Proc. London Math. Soc.*, vol. 2, no. 42, pp. 230–265, 1936.
2. C. Shannon and W. Weaver, *The Mathematical Theory of Information.* Urbana, IL: Univ. Illinois Press, 1949.
3. W. McCulloch and W. Pitts, “A logical calculus of ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, 5(4), pp. 115-133, 1943.
4. F. Rosenblatt, "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". *Psychological Review* 65 (6): 386–408, 1958.
5. M. Minsky, S. Papert. *An Introduction to Computational Geometry*. MIT Press, 1969. ISBN 0-262-63022-2.
6. P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph. D. Thesis, Harvard University, 1975.
7. D. E. Rumelhart, G. E. Hinton and R J. Willliams, “Learning Representations by back-propagating errors”, *Nature* 323, pp. 533-536, 1986.
8. Y. Lecun, Y. Bengio and G. Hinton, “Deep Learning”, *Nature*, vol. 521, pp. 436-444, 2015.
9. G. E. Hinton, S. Osindero, Y. W. A. Teh, “A fast learning algorithm for deep belief nets”, *Neural Computation*, 18, 1527-1554, 2006.
10. G. E. Hinton and R. Salakhutdinov, “reducing the dimensionality of data with neural networks,” *Science*, 213, 504-507, 2006.
11. K. Fukushima and S. Miyake, “Neocognitron: a new algorithm for pattern recognition tolerant to deformations and shifts in position,” *Pattern Recognition*, 15, 455-469, 1982.

12. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient based learning applied to document recognition. *Proc. of IEEE* 86, 2278-2324, 1998.
13. C. A. Mead, Neurmorphic Electronics Systems, *Proc. of IEEE*, vol. 78, no. 10, pp. 1629-1636, 1990.
14. A. B. Barrett, M. C. W. van Rossum, “Optimal Learning Rules for Discrete Synapses,” *PLoS Computational Biology* 4(11): e1000230. doi:10.1371/journal.pcbi.1000230, 2008.
15. A. Renart, J. De la Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, K. D. Harris, “The asynchronous state in cortical circuits,” *Science*, 327, pp. 587–590, 2010.
16. G. Buzsaki, Rhythms of the Brain, Oxford University Press, USA, 2009.
17. Edelman, G. M., *The Remembered Present: A Biological Theory of Consciousness*, Basic Books, NY, 1989.
18. W. J. Freeman, “A field-theoretic approach to understanding scale-free neocortical dynamics,” *Biological Cybernetics*, 92(6):350-359, 2005.
19. D. J. Graham, D. J. Field, “Sparse coding in the neocortex”. In: Kass JH, editor. *Evolution of the nervous system*, Vol III. Oxford: Academic Press. pp. 181–187, 2007.
20. R. Van Rullen, and R. Guyonneau, and S. J. Thorpe, “Spike times make sense”, *Trends in Neuroscience*, 28:1-4, 2005.
21. N. Srinivasa, and Q. Jiang, “Stable learning of functional maps in self-organizing spiking neural networks with continuous synaptic plasticity, *Front. In Comp. Neuroscience*, doi: 10.3389/fncom.2013.0001, 2011.
22. N. Srinivasa, and Y. K. Cho, “Unsupervised discrimination of patterns in spiking neural networks with excitatory and inhibitory synaptic plasticity,” *Front. In Comp. Neuroscience*, doi:10.3389/fncom.2014.00159, 2014.
23. Y. Cao, Y. Chen, and D. Khosla, “Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition,” *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2014.
24. E. Hunsberger, and C. Eliasmith. “Spiking Deep Networks with LIF Neurons.” arXiv:1510.08829v1, 2015.
25. M. Coath, S. Sheik, E. G. Chicca, G. Indiveri, and S. L. T. Wennekers, “A robust sound perception model suitable for neuromorphic implementation,” doi:10.3389/fncom.2013.00278, 2014.
26. P. Joshi, and W. Maass, “Movement generation and control with generic neural microcircuits, in Proc. of BIO-AUDIT, pp. 16-31, 2004.
27. M. Oubatti, P. Levi, M. Schanz, T. Buchheim, “Velocity control of an omnidirectional robocup player with recurrent neural networks,” In Proc. of Robocup Symposium, pp. 691-701, 2005.
28. P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor,” *IEEE J. Solid-State Circ*. 43, 566–576, 2008.
29. S. Liu, A. Van Schaik, B. Minch, and T. Delbruck, “Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms,” in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2027–2030, 2010.
30. P. O’Connor, D. Neil, S. C. Liu, T. Delbruck, and M. Pfeiffer, “Real-time classification and sensor fusion with a spiking deep belief network,” *Front. in Neuromorphic Engineering*, http://dx.doi.orgns/10.3389/fnins.2013.00178, 2013.
31. G. Orchard, and R. E. Cummings, “Bioinspired Visual Motion Estimation,” *Proc of the IEEE*, vol. 102, 10, 1520-1536, 2014.
32. Z. Ni, B. Bolopion, J. Agnus, R. Benosman and S. Regnier, “Asynchronous Event-Based Visual Shape Tracking for Stable Haptic feedback in Microrobotics. IEEE Trans. On Robotics, Vol. 28, no. 5, pp. 1081-1089, 2012

33. T. K. Horiuchi, "A spike-latency model for sonar-based navigation in obstacle fields," *IEEE Trans on Circuits and Systems – I Regular Papers*, vol. 56, no. 11, pp. 2293-2401, 2009.
34. R. J. Vogelstein, U. Malik, E. Culurceillo, G. Cauwenberghs, R. Etienne-Cummings, "A multichip neuromorphic system for spike-based visual information processing," *Neural Computation* 19:2281-2300, 2007.
35. P. A. Merolla, J. A. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, et al, "A million spiking neuron integrated circuit with a scalable communication network and interface," *Science*, Vol. 345, Issue 6197, pp. 668-673, 2014. DOI: 10.1126/science.1254642.
36. J. Cruz-Albrecht, T. Derosier, and N. Srinivasa, "Scalable neural chip with synaptic electronics using CMOS integrated memristors, *Nanotechnology, Special Issue on Synaptic Electronics*, vol. 24, 384011 (11pp), doi:10.1088/0957-4484/24/38/384011, 2013.
37. http://www.technologyreview.com/news/532176/a-brain-inspired-chip-takes-to-the-sky/
38. B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran , J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE*, vol 102, no 5, pp 699-716, 2014.
39. *Brain-inspired multiscale computation in neuromorphic hybrid systems (BrainScaleS)* FP7 269921 EU Grant, 2011–2015.
40. G. Indiveri, E. Chicca and R. Douglas, A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity, *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211-221, 2006.
41. R. Vogelstein, U. Mallik, J. Vogelstein and G. Cauwenberghs, Dynamically Reconfigurable Silicon Array of Spiking Neurons With Conductance-Based Synapses, *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 253-265, 2007.
42. N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska and G. Indiveri, A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses, *Frontiers in Neuroscience*, vol. 9, no. 141, 2015.
43. S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, "The SpiNNaker Project. *Proceedings of the IEEE*: 1. doi:10.1109/JPROC.2014.2304638, 2014.
44. A. M. Bastos, W. M Usrey, R. A. Adams, G. R. Mangun, P. Fries, K. J. Friston, "Canonical microcircuits for predictive coding," *Neuron* 76(4):695-711, 2012.
45. R. P. Rao, and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, 2(1):79-87, 1999.
46. W. Maass, "Noise as a resource for computation and learning in networks of spiking neurons," *Special Issue of the Proc. of the IEEE on Engineering Intelligent Electronic Systems based on Computational Neuroscience*, 102(5):860-880, 2015.
47. D. Kappel, S. Habenschuss, R. Legenstein, R., and W. Maass, "Network plasticity as Bayesian inference," *PLOS Computational Biology*, 11(11):e1004485, 2015.
48. B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass, "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity," *PLoS Computational Biology*, vol. 9, no. 4, p. e1003037, 2013.
49. D. S. Reich, F. Mechler, and J. D. Victor, "Independent and redundant information in nearby cortical neurons," *Science*, 294(5551), 2566-2568, 2001.
50. P. Kara, P. Reinagel, and R. C. Reid, "Low response variability in simultaneously

recorded retinal, thalamic and cortical neurons," *Neuron*, 27, 635-646, 2000.

51. B. Haider, M. R. Krause, A. Duque, Y. J. Yu, J. Touryan, J. A. Mazer, D. A. McCormick, "Efficient discrimination of temporal patterns by motion sensitive neurons in primate visual cortex," *Neuro*n, 65, 107-121, 2010.
52. G. T. Buracas, A. M. Zador, M. R. DeWeese, and T. D. Albright, "Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex," *Neuron*, 20, 959-969, 1998.
53. S. Panzeri, N. Brunel, N. Logothetis, and C. Kayser, "Sensory neural codes sing multiplexed temporal scales," *Trends in Neuroscience*, 33, 111-120, 2010.
54. H. Akolkar, C. Meyer, Z. Clady, O. Marre, C. Bartolozzi, and S. Panzeri. "What can Neuromorphic Event-Driven Precise Timing Add to Spike-Based Pattern Recognition?," *Neural Computation*, 27, 561-593, 2015
55. J. Binas, G. Indiveri and M. Pfeiffer, Spiking Analog VLSI Neuron Assemblies as Constraint Satisfaction Problems. arXiv, 2015.
56. The Register, "Intel snaps up network chipper Fulcrum," [Online]. Available: http://www.theregister.co.uk/2011/07/19/intel_acquires_fulcrum_microsystems/.
57. R. Wilson, "Fulcrum," EE Times, 6 9 2011. [Online].Available: http://www.eetimes.com/document.asp?doc_id=1279057.
58. L. Büsing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons," PLoS Comput. Biol. 7:e1002211. doi: 10.1371/journal.pcbi.1002211, 2011.
59. P. Tiesinga, J. M. Fellous, and T. J. Sejnowski, "Regulation of spike timing in visual cortical circuits," *Nature Review Neuroscience*, 9, 97–107, 2008.
60. E. M. Izhikevich, "Polychronization: Computation with Spikes," *Neural Computation* 18, 245-282, 2006.
61. H. A. Swadlow, and S. G. Waxman, Observations on impulse conduction along central axons. *PNAS* 72, 5156-5159, 1975.
62. D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.
63. H. Markram, J. Lubke, M. Frotsche, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science* 275, 213-215, 1997.
64. G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength and postsynaptic cell type," *Journal of Neuroscience*, 18, 10464-10472, 1998.
65. P. J. Sjostrom, G. G. Turrigiano, and S. B. Nelson, "Rate, timing and cooperativity jointly determine cortical synaptic plasticity," *Neuron* 32, 1149-1164, 2001.
66. M. Chistiakova, N. M. Bannon, M. Bazhenov and M. Volgushev, "Heterosynaptic plasticity: multiple mechanisms and multiple roles," *Neuroscientist* 20, 483-498, 2014.
67. G. S. Lynch, T. Dunwiddie, and V. Gribkoff, "Heterosynaptic depression: a postsynaptic correlate of long-term potentiation," *Nature* 266, 737-739, 1977.
68. U. Frey and R. G. M. Morris, "Synaptic tagging and long term potentiation," *Nature* 385, 533-536, 1997.
69. G. G. Turrigiano, "Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor Perspective in Biology*, 4, a005736, 2012.
70. A. Stepanyants, P. R. Hof, D. B. Chklovskii, "Geometry and structural plasticity of synaptic connectivity," *Neuron* 34, 275-288, 2002.
71. J. T. Trachtenberg JT et al, "Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex," *Nature* 420, 788-794, 2002.
72. H. Markram, Y. Wang, and M. Tsodyks, "Differential signaling via the same axon

of neocortical pyramidal neurons," *Proc. Natl Acad. Sci USA*, 95 5323-5328, 1998.

73. V. Pawlak, J. R. Wickens, A. Kirkwood, and J. N. D. Kerr, "Timing is not everything: neuromodulation opens the STDP gate," *Front. Synaptic Neuroscience* 2, 146, 2010.
74. W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science* 275, 1593-1599, 1997.
75. G. A. Carpenter, and S. Grossberg, "Adaptive resonance theory," *Encyclopedia of Machine Learning and Data Mining*. C. Sammut and G. Webb, Eds. Berlin: Springer-Verlag, 2015.
76. J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks* 61: 85–117, 2015.
77. F. Zenke, E. J. Agnes, and W. Gerstner, "Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks," *Nature Comm*. Doi: 10.1038/ncomms7922, 2015.
78. P. D. Balsam, M. R. Drew, et al, "Time and associative learning," *Comparative Cognition and Behavior Review*, 5, 1-22, 2010.