Probability in the Engineering and Informational Sciences, **35**, 2021, 51–61.
doi:10.1017/S0269964819000147

ACCURATE, ENERGY-EFFICIENT CLASSIFICATION WITH SPIKING RANDOM NEURAL NETWORK

Khaled F. Hussain and Mohamed Yousef Bassyouni
Computer Science Department, Faculty of Computers and Information, Assiut University, Asyut 71515, Egypt
E-mail: khussain@aun.edu.eg; mohamed.mahdi@fci.au.edu.eg

Erol Gelenbe
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK
E-mail: e.gelenbe@imperial.ac.uk

Artificial Neural Networks (ANNs)-based techniques have dominated state-of-the-art results in most problems related to computer vision, audio recognition, and natural language processing in the past few years, resulting in strong industrial adoption from all leading technology companies worldwide. One of the major obstacles that have historically delayed large-scale adoption of ANNs is the huge computational and power costs associated with training and testing (deploying) them. In the mean-time, Neuromorphic Computing platforms have recently achieved remarkable performance running the bio-realistic Spiking Neural Networks at high throughput and very low power consumption making them a natural alternative to ANNs. Here, we propose using the Random Neural Network, a spiking neural network with both theoretical and practical appealing properties, as a general purpose classifier that can match the classification power of ANNs on a number of tasks while enjoying all the features of being a spiking neural network. This is demonstrated on a number of real-world classification datasets.

Keywords: artificial neural network, neuromorphic computing, random neural network, spiking neural networks

1. INTRODUCTION

Despite being first proposed about 60 years ago (The Perceptron model [69]), only in the past few years, artificial neural network(ANNs) had become the de facto standard machine learning model [55] achieving state-of-the-art results for wide range of problems ranging from vision problems such as image classification [44,54,75], object detection [39,68], semantic segmentation [45,58], face recognition [67,71], and text recognition [40,73], to speech recognition [2,41,46], to natural language processing problems like machine translation [51,74], language modeling [3], and question answering [4]. This has resulted in huge industry-wide adoption from leading technology companies like Google, Facebook, Microsoft, IBM, Yahoo!, Twitter, Adobe, and a quickly growing number of start-ups. One of The prominent reasons

 0269-9648/19 $25.00

for this recent revival is that in order for ANNs to achieve such performance they need big labeled datasets and huge computational power at a scale that only recently came to the hands of individual researchers in the form of GPUs [66], which kick-started the deep learning revolution in 2012 [54]. Since then, the trend for demanding more computation and more power has largely increased.

Despite being bio-inspired architectures, ANNs have subtle differences from actual biological neurons in their work (how computations is performed by neurons), structure (connection patterns and topologies of neurons), learning (how neurons adopt themselves to new observations), and communication (how inter-neuron data is encoded and passed). One of the main reasons of the inefficiency of ANNs compared to biological neurons is how communication is done. While biological neurons use asynchronus trains of spikes in an event-based, data-driven manner that adopts locally to its external stimulating pattern to communicate and encode data (though the specific encoding mechanism used by neurons is not totally understood), ANNs communicate in dense, continuous valued activations, whcich means all the neurons are working in the same time thus using lots of computation and power to operate. The idea behind spiking neural networks is to leverage this benefit from biological neurons and communicate asynchronously in trains of spikes. Thus, spiking neural networks incorporates the concept of time, and instead of all neurons firing in the same time as the case with ANNs, in spiking neural networks neurons fire only when thier intrinsic potential (i.e., membrane voltage) reaches a specific threshold [38,47]. Neuroscientists have historically suggested a large number of models for simulating how biological neurons communicate, one of the simplest models that is widely used in many spiking neural networks models is the integrate-and-fire (IF) model [1], in which the change in the membrane voltage v_{mem} is given by

$$\frac{dv_{mem}(t)}{dt} = \sum_i \sum_{s \in S_i} w_i \delta(t - s) \tag{1}$$

where w_i is the weight of the ith incoming synapse, $\delta(.)$ is the delta function, and $S_i = t_i^0, t_i^1, \ldots$ contains the spike times of the ith presynaptic neuron. If the membrane voltage crosses the spiking threshold v_{thr}, a spike is generated and the membrane voltage is reset to a reset potential v_{res} [10]. Some other models exist, such as spike response model (SRM) [52], and the Izhikevich neuron model [50].

One of the prominent differences between spiking neural networks and ANNs is how they learn and adopt to new signals. While ANNs have been predominantly trained in literature using Backpropagation [80] and some variant of stochastic gradient descent (SGD), which can be summarized as moving the vector of network parameters or weights θ in the direction of the negative gradient of some loss function L that characterizes the deviation network's current output from the ground truth labels of input data. Training spiking neural networks, on the other hand, is still an open research issue with many proposed solution and no consensus [76]. One of the most popular and biologically plausible learning methods in spiking neural networks is unsupervised learning using the Spike Timing Dependent Plasticity (STDP) [7,59], in which the synaptic weight is adjusted in accordance with the relative spike times of the presynaptic and postsynaptic neurons. An important problem that has always faced using the popular gradient-based optimization algorithms in spiking neural networks is that both spike trains and the underlying membrane voltage are not differentiable at the time of spikes, researchers tried different approaches to alleviate this problem, one of the most successful has been the workaround of first training an ANN and then converting it to a corresponding spiking neural networks [10,62,63].

Though von Neumann architectures [78] worked very well running ANNs efficiently, it was suggested as early as 1980s that they were not adequate for running the more realistic spiking neural networks models efficiently, and a new architecture was needed to realize their power and computational efficiencies [83]. The recent success of ANNs have pushed this trend much faster. The main idea behind Neuromorphic Computing was to design Integrated Circuits (ICs) that are arranged and behave like living neurons (i.e., to mimic how the brain performs computation) [60], the spiking neural networks model of a biological neuron has historically been used as a guiding design in this process. After years of trails, the past few years saw the demonstration of Neuromorphic Computing platforms with millions of neurons while requiring only milliWatts of power for their operation such as TrueNorth [61], SpiNNaker [16], and Loihi [9]. A number of SSN-based pattern classification applications were demonstrated to run efficiently and accurately on these chips while being orders of magnitude more efficient in terms of power consumption than an ANN on a von Neumann CPU or GPU running a similar task. The main source of this power saving is the asynchronous working and firing of spiking neural networks described earlier, so neurons fire and the chip consumes power only when needed this is completely different than what happens in an ANN when all neurons are obliged to fire synchronously together which costs a lot of an unnecessary energy and computation. This efficiency can be even increased by consuming input from neuromorphic sensors such as silicon retinas [72] or cochleas [57], which create sparse, framefree, and precisely timed train of signals, with substantially reduced latencies compared to traditional frame-based approaches which produce large volumes of redundant data and therefore consumes lots of power. This line of neuromorphic sensor design has been applied to vision sensors, auditory sensors, and olfactory sensors [77].

The rest of the paper is structured as follows: Random Neural Network are described and reviewed in Section 2; our experimental setup is described and results presented in Section 3; the conclusions and future work are drawn in Section 4.

2. RANDOM NEURAL NETWORK (RNN)

G-Networks [20] are a family of queueing networks with a convenient and computationally efficient product form mathematical solution. The computation of the state of a G-Network is obtained via a simple fixed-point iteration, and the existence and uniqueness of the solution to the key G-Network state equation is easily verified [31]. G-Networks incorporate useful primitives, such as the transfer of jobs between servers or the removal of batches of jobs from excessively busy servers, that were developed in a series of successive papers [12,21,22,33]. They have a wealth of diverse applications as a tool to analyze and optimize the effects of dynamic load balancing in large-scale networks and distributed computer systems [30]. They are also used to model Gene Regulatory Networks [24,53]. A recent application of G-Networks is to the modeling of systems which operate with intermittent sources of energy, known as Energy Packet Networks [13,17,26,27,29].

The simplest version of G-Networks, known as the RNN [19] has a powerful property of approximating continuous and bounded real-valued functions [36]. This property serves as the foundation for RNN-based learning algorithms [23] and Deep Learning [32,82].

The RNN has been used for modeling natural neuronal networks [18], and for protein alignment [65]. It has been used in several image processing applications including the accurate evaluation of tumors from brain MRI scans [34] and the compression of still and moving images [8,35,43]. It was recently introduced as a tool for predicting the toxicity of chemical compounds [42].

https://doi.org/10.1017/S0269964819000147 Published online by Cambridge University Press

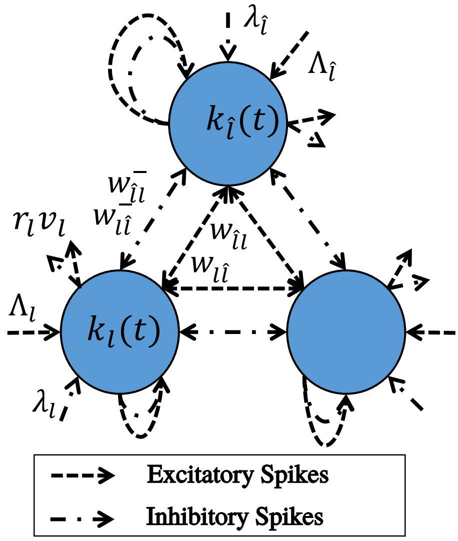

FIGURE 1. Schematic representation of a RNN [81].

In the field of computer network performance, the RNN has been used to build distributed controllers for quality of service routing in packet networks [5,25,28] and in the design of Software Defined Network controllers for the Internet [14,15]. Real-time optimized task allocation algorithms in Cloud systems [79] have also been built and tested. Recent applications has addressed the use of the RNN to detect network attacks [70] and attacks on Internet of Things (IoT) gateways [6].

Figure 1 gives a schematic diagram of a RNN. Assume a system with L neurons. The state of neuron l at time t is represented by a non-negative integer called its potential, denoted by $k_l(t) \geq 0$. Network state at time t is a vector $k(t) = (k_1(t), \ldots, k_l(t), \ldots, k_L(t))$. When an excitation signal arrives to neuron l, the state of neuron l is changed from state $k_l(t)$ to $k_l(t)+1$. When an inhibition signal arrives to neuron l, the state of neuron l is changed from state $k_l(t)$ to $k_l(t)-1$. Neuron l emits a spike if $k_l(t)$ is positive (i.e., it is excited); the state of neuron l is changed from state $k_l(t)$ to $k_l(t)-1$. The spikes are sent out from neuron l at a rate r_l which is exponentially distributed.

Spikes are sent from the outside world to neuron l as a positive signal according to Poisson processes of rate Λ_l or also as a negative signal according to Poisson processes of rate λ_l

Spikes are sent out from neuron l to neuron $\hat{l}$ as a positive signal with probability $p_{l,\hat{l}}$ or as a negative signal with probability $p^-_{l,\hat{l}}$, or they depart from the network with probability v_l. The sum of these probabilities must be one.

$$d_i + \sum_{j=1}^{L} [p_{l,\hat{l}} + p^-_{l,\hat{l}}] = 1, \quad \forall l \tag{2}$$

The spikes are sent out from neuron l to neuron $\hat{l}$ at rates:

$$w_{l,\hat{l}} = r_l \, p_{l,\hat{l}} \geq 0, \tag{3}$$

$$w^-_{l,\hat{l}} = r_l \, p^-_{l,\hat{l}} \geq 0, \tag{4}$$

$w_{l,\hat{l}}$ and $w^-_{l,\hat{l}}$ are also called the excitatory and inhibitory weights.

Combining Eqs. (2)–(4) we can get:

$$r_l = \frac{\sum_{j=1}^{L}[w_{l,\hat{l}} + w^-_{l,\hat{l}}]}{1 - v_l} \tag{5}$$

Let $q_l = \lim_{t\to\infty} \text{Prob}(k_l(t) > 0)$ denote stationary excitation probability of the neuron l. The total arrival rates of positive signals Ω_l and negative signals Ω_l^-, for $l = 1, \ldots, n$, can be calculated from the following nonlinear system of equations:

$$\Omega_l = \Lambda_l + \sum_{j=1}^{L} q_l w_{l,\hat{l}} \tag{6}$$

$$\Omega_l^- = \lambda_l + \sum_{j=1}^{L} q_l w^-_{l,\hat{l}} \tag{7}$$

It has been proven that q_l can be directly calculated by the following system of equations:

$$q_l = \min\left\{1, \frac{\Omega_l}{r_l + \Omega_l^-}\right\} \tag{8}$$

The existence of a solution to the system of N non-linear Eq. (8) and its uniqueniess has been proven [23]. Therefore, the states of the RNN can be efficiently obtained by solving it.

3. EXPERIMENTAL RESULTS

Here we present experimental results on a number of benchmark real world classification datasets that are widely used in literature. We show that spiking neural network-based RNNs are empirically at least as powerful as ANNs in this category of classification problems (It was theoretically shown that RNNs are universal function approximaters [36,37], Thus, as computationally capable as ANNs [48]).

3.1. Evaluation Setup

Table 1 shows statistics about the used datasets, we use Iris, Breast Cancer, and Glass datasets from the UCI machine learning repository [56] and the Ovarian cancer dataset [64]. We train the RNN using the same procedure described in Hussain and Moussa [49], which can be summarized as:

Assume the given dataset has K pairs of input training patterns x_k and associated output class label y_k.

TABLE 1. Names and statistics of the used datasets

Dataset	# Attributes	# Features	# Output classes
Iris	4	150	3
Breast Cancer Wisconsin	9	699	2
Glass	9	214	7
Ovarian cancer	100	216	2

https://doi.org/10.1017/S0269964819000147 Published online by Cambridge University Press

1. Initialize the weights $w_{l,\hat{l}}$ and $w^-_{l,\hat{l}}$, $\forall l, \hat{l}$, to random values between zero and one.
2. Set the inhibitory rates to zero.
3. Set the excitatory input rates for the input neurons, $\Lambda_k = x_k$, where x_k is the kth input training pattern.
4. Solve the nonlinear system of Eq. (8) to obtain the neuron stationary excitation probability $q_l, \forall l$
5. Iterate through the RNN learning algorithm till convergence, updating in each step the weights $w_{l,\hat{l}}$ and $w^-_{l,\hat{l}}$, $\forall l, \hat{l}$, which minimize the following error function:

$$E = \sum_{k=1}^{K} \sum_{l=1}^{L} [q_{lk} - y_{lk}]^2 \tag{9}$$

3.2. Dataset Description

Here we give a brief description of the four datasets used for evaluation.

1. Iris dataset [56]: Each instance is described by four plants attributes (sepal length and width, and petal length and width) all are real numbers and the task is to recognize which class of Iris plants (Iris Setosa, Iris Versicolour, or Iris Virginica) a given test instance belongs to.
2. Breast Cancer Wisconsin dataset [56]: Each instance is described by 9 numerical attributes, that range from 1 to 10. The attributes include the clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. This breast cancer databases were obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The task is to recognize the class of the breast cancer (benign or malignant).
3. Glass dataset [56]: Each instance has 9 continuous attributes, including the refractive index and the unit measurements of sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron. There are in total 7 types of glass, while there are instances of only 6 types of glass in the dataset. The task is to using the 9 attributes to recognize which type of glass this instance belongs to (whether it is windows glass or non-window glass).
4. Ovarian Cancer dataset [64]: From the FDA-NCI Clinical Proteomics Program Databank, the dataset comprises 216 patients, out of which 121 are ovarian cancer patients and 95 are normal patients. Each instance has 100 attributes, each of which represents the ion intensity level at a specific mass-charge value . The task is to recognize the class of the ovarian cancer (benign or malignant).

3.3. Results

Tables 2–5 display confusion matrices of RNN on the four datasets. Table 6 compares the accuracy of RNNs against ANNs some UCI datasets. For the ANN results, we use results obtained on UCI datasets from the comprehensive study in Fernández-Delgado et al. [11]. We can clearly see in Table 1 that RNNs are at least as powerful as ANNs in these datasets and can deliver excellent classification accuracy.

https://doi.org/10.1017/S0269964819000147 Published online by Cambridge University Press

TABLE 2. Confusion matrix for Iris dataset

Class	Setosa	Versicolour	Virginica
Setosa	1.0	0.0	0.0
Versicolour	0.0	1.0	0.0
Virginica	0.0	0.0	1.0

TABLE 3. Confusion matrix for Breast Cancer dataset

Class	Positive	Negative
Positive	0.984	0.016
Negative	0.067	0.933

TABLE 4. Confusion matrix for Glass dataset

Class	Positive	Negative
Positive	1.0	0.0
Negative	0.127	0.8139

TABLE 5. Confusion matrix for Ovarian Cancer dataset

Class	Positive	Negative
Positive	1.0	0.0
Negative	0.1064	0.8936

TABLE 6. Accuracy comparison between RNN, and ANN. Best result in each dataset is in bold

Dataset	RNN	ANN
Iris	**1.0**	0.959
Breast Cancer Wisconsin	**0.964**	0.963

4. CONCLUSIONS

In this paper, we have motivated the need for power and computation-wise efficient models for classification and how ANNs, despite being very powerful classifiers, are not efficient neither in their power nor computational demands. We presented and reviewed spiking neural networks and neuromorphic computing as a possible alternative that have received a lot of interest in past years due to its power usage and computation efficiency, but also suffered from its own problems, namely, being harder to train and less robust in its generalization performance. We also presented a special kind of spiking neural networks, the Random Neural Network, that was first introduced in Gelenbe [19]. RNN's special analytical properties makes it much easier to train, we have also empirically shown that it provides generalization performance that is at least as powerful as conventional ANNs in a number of real world

https://doi.org/10.1017/S0269964819000147 Published online by Cambridge University Press

classification datasets, while entertaining the efficiencies associated with being a spiking neural network.

FUNDING

This research received no external funding.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

References

1. Abbott, L.F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin* 50(5-6): 303–304.
2. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L.V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., & Zhu, Z. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pp. 173–182.
3. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research* 3: 1137–1155.
4. Bordes, A., Chopra, S., & Weston, J. (2014). Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676.
5. Brun, O., Wang, L., & Gelenbe, E. (2016). Big data for autonomic intercontinental overlays. *IEEE Journal on Selected Areas in Communications* 34(3): 575–583.
6. Brun, O., Yin, Y., Gelenbe, E., Kadioglu, Y.M., Augusto-Gonzalez, J., & Ramos, M. (2018). Deep learning with dense random neural networks for detecting attacks against iot-connected home environments. In *Security in Computer and Information Sciences: First International ISCIS Security Workshop 2018 (Euro-CYBERSEC 2018)*, London, UK, 26–27 February 2018. Lecture Notes CCIS No. 821, Springer-Verlag.
7. Caporale, N. & Dan, Y. (2008). Spike timing–dependent plasticity: a hebbian learning rule. *Annual Review of Neuroscience* 31: 25–46.
8. Cramer, C. & Gelenbe, E. (2000). Video quality and traffic qos in learning-based subsampled and receiver-interpolated video sequences. *IEEE Journal on Selected Areas in Communications* 18(2): 150–167.
9. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C.-K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y.-H., Wild, A., Yang, Y., & Wang, H. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38(1): 82–99.
10. Diehl, P.U., Neil, D., Binas, J., Cook, M., Liu, S.C., & Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *IEEE 2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.
11. Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?. *The Journal of Machine Learning Research* 15(1): 3133–3181.
12. Fourneau, J.M. & Gelenbe, E. (2017). G-networks with adders. *Future Internet* 9(3): 34.
13. Fourneau, J., Marin, A., & Balsamo, S. (2016). Modeling energy packets networks in the presence of failures. In*24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2016)*, London, United Kingdom, 19–21 September 2016, pp. 144–153.
14. François, F. & Gelenbe, E. (2016). *Optimizing secure sdn-enabled inter-data centre overlay networks through cognitive routing*. MASCOTS: IEEE Computer Society, pp. 283–288.

15. François, F. & Gelenbe, E. (2016). Towards a cognitive routing engine for software defined networks. In *IEEE The International Conference on Communications (ICC)*, pp. 1–6.
16. Furber, S.B., Galluppi, F., Temple, S., & Plana, L.A. (2014). The spinnaker project. *Proceedings of the IEEE* 102(5): 652–665.
17. Gelenbe, E. & Ceran, E.T. (2016). Energy packet networks with energy harvesting. *IEEE Access* 4: 1321–1331.
18. Gelenbe, E. & Cramer, C. (1998). Oscillatory corticothalamic response to somatosensory input. *Biosystems* 48(1-3): 67–75.
19. Gelenbe, E. (1989). Random neural networks with negative and positive signals and product form solution. *Neural Computation* 1(4): 502–510.
20. Gelenbe, E. (1991). Product-form queueing networks with negative and positive customers. *Journal of Applied Probability* 28(3): 656–663.
21. Gelenbe, E. (1993). G-networks by triggered customer movement. *Journal of Applied Probability* 30(3): 742–748.
22. Gelenbe, E. (1993). G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences* 7(3): 335–342.
23. Gelenbe, E. (1993). Learning in the recurrent random neural network. *Neural Computation* 5(1): 154–164.
24. Gelenbe, E. (2007). Steady-state solution of probabilistic gene regulatory networks. *Physical Review E* 76, 031903.
25. Gelenbe, E. (2009). Steps toward self-aware networks. *Communications of the ACM* 52(7): 66–75.
26. Gelenbe, E. (2012). Energy packet networks: Adaptive energy management for the cloud. In *CloudCP '12 Proceedings of the 2nd International Workshop on Cloud Computing Platforms* (ACM 2012), p. 1.
27. Gelenbe, E. & Abdelrahman, O.H. (2018). An energy packet network model for mobile networks with energy harvesting. *Nonlinear Theory and Its Applications, IEICE* 9(3): 1–15.
28. Gelenbe, E. & Kazhmaganbetova, Z. (2014). Cognitive packet network for bilateral asymmetric connections. *IEEE Transactions on Industrial Informatics* 10(3): 1717–1725.
29. Gelenbe, E. & Marin, A. (2015). Interconnected wireless sensors with energy harvesting. In *Proceedings 22nd International Conference Analytical and Stochastic Modelling Techniques and Applications (ASMTA 2015)*, Albena, Bulgaria, 26–29 May 2015, pp. 87–99 .
30. Gelenbe, E. & Morfopoulou, C. (2010). A framework for energy-aware routing in packet networks. *The Computer Journal* 54(6): 850–859.
31. Gelenbe, E. & Schassberger, R. (1992). Stability of product form g-networks. *Probability in the Engineering and Informational Sciences* 6(3): 271–276.
32. Gelenbe, E. & Yin, Y. (2016). Deep learning with random neural networks. In *IEEE 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1633–1638.
33. Gelenbe, E., Glynn, P., & Sigman, K. (1991). Queues with negative arrivals. *Journal of Applied Probability* 28(1): 245–250.
34. Gelenbe, E., Feng, Y., & Krishnan, K.R.R. (1996). Neural network methods for volumetric magnetic resonance imaging of the human brain. *Proceedings of the IEEE* 84(10): 1488–1496.
35. Gelenbe, E., Sungur, M., Cramer, C., & Gelenbe, P. (1996). Traffic and video quality with adaptive neural compression. *Multimedia Systems* 4(6): 357–369.
36. Gelenbe, E., Mao, Z.H., & Li, Y.D. (1999). Function approximation with spiked random networks. *IEEE Transactions on Neural Networks* 10(1): 3–9.
37. Gelenbe, E., Mao, Z.H., & Li, Y.D. (2006). Function approximation by random neural networks with a bounded number of layers. In *Computer System Performance Modeling In Perspective: A Tribute to the Work of Prof Kenneth C Sevcik*, pp. 35–58. World Scientific.
38. Gerstner, W., Kistler, W.M., Naud, R., & Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge, UK: Cambridge University Press.
39. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587.
40. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5): 855–868.
41. Graves, A., Mohamed, A.r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649.

42. Grenet, I., Yin, Y., Comet, J.P., & Gelenbe, E. (2018). Machine learning to predict toxicity of compounds. In *27th Annual International Conference on Artificial Neural Networks (ICANN18)*. Springer-Verlag, accepted for publication.
43. Hai, F., Hussain, K.F., Gelenbe, E., & Guha, R.K. (2001). Video compression with wavelets and random neural network approximations. In *Applications of Artificial Neural Networks in Image Processing VI* Vol. 4305, pp. 57–65. International Society for Optics and Photonics.
44. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
45. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.
46. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6): 82–97.
47. Hodgkin, A.L. & Huxley, A.F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117(4): 500–544.
48. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5): 359–366.
49. Hussain, K.F. & Moussa, G.S. (2016). On-road vehicle classification based on random neural network and bag-of-visual words. *Probability in the Engineering and Informational Sciences* 30(3): 403–412.
50. Izhikevich, E.M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks* 14(6): 1569–1572.
51. Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2014). On using very large target vocabulary for neural machine translation. arXiv preprint arXiv:1412.2007.
52. Jolivet, R., Timothy, J., & Gerstner, W. (2003). The spike response model: a framework to predict neuronal spike trains. In *Artificial Neural Networks and Neural Information Processing–ICANN/ICONIP 2003*, Springer, pp. 846–853.
53. Kim, H. & Gelenbe, E. (2012). Stochastic gene expression modeling with hill function for switch-like gene responses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9(4): 973–979.
54. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105.
55. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* 521(7553): 436.
56. Lichman, M. & Bache, K. (2013). UCI machine learning repository.
57. Liu, S.C. & Delbruck, T. (2010). Neuromorphic sensory systems. *Current Opinion in Neurobiology* 20(3): 288–295.
58. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.
59. Markram, H., Gerstner, W., & Sjöström, P.J. (2011). A history of spike-timing-dependent plasticity. *Frontiers in Synaptic Neuroscience* 3: 4.
60. Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE* 78(10): 1629–1636.
61. Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S.K., Appuswamy, R., Taba, B., Amir, A., Flickner, M.D., Risk, W.P., Manohar, R., & Modha, D.S. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345(6197): 668–673.
62. Neil, D., Pfeiffer, M., & Liu, S.C. (2016). Learning to be efficient: algorithms for training low-latency, low-compute deep spiking neural networks. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ACM, pp. 293–298.
63. O'Connor, P., Neil, D., Liu, S.C., Delbruck, T., & Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience* 7: 178.
64. Petricoin, E.F., III, Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C., & Liotta, L.A. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet* 359(9306): 572–577.
65. Phan, H.T.T., Sternberg, M.J.E., & Gelenbe, E. (2012). Aligning protein-protein interaction networks using random neural networks. In *2012 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2012)*, Philadelphia, PA, USA, 4–7 October 2012, pp. 1–6.
66. Raina, R., Madhavan, A., & Ng, A.Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, pp. 873–880.

67. Ranjan, R., Patel, V.M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41: 121–135.
68. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788.
69. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6): 386.
70. Sakellari, G. & Gelenbe, E. (2010). Demonstrating cognitive packet network resilience to worm attacks. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, (CCS 2010)*, Chicago, Illinois, USA, 4–8 October 2010, pp. 636–638.
71. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823.
72. Serrano-Gotarredona, T. & Linares-Barranco, B. (2013). A 128×128 1.5% contrast sensitivity 0.9% fpn 3 μs latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *Journal of Solid-State Circuits* 48(3): 827–838.
73. Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(11): 2298–2304.
74. Sutskever, I., Vinyals, O., & Le, Q.V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112.
75. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.
76. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., & Maida, A.S. (2018). Deep learning in spiking neural networks. arXiv preprint arXiv:1804.08150.
77. Vanarse, A., Osseiran, A., & Rassau, A. (2016). A review of current neuromorphic approaches for vision, auditory, and olfactory sensors. *Frontiers in Neuroscience* 10: 115.
78. Von Neumann, J. (1993). First draft of a report on the edvac. *IEEE Annals of the History of Computing* 4: 27–75.
79. Wang, L. & Gelenbe, E. (2018). Adaptive dispatching of tasks in the cloud. *IEEE Transactions on Cloud Computing* 6(1): 33–45.
80. Werbos, P. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. dissertation, Harvard University.
81. Yin, Y. (2018). *Random neural networks for deep learning*. Ph.D. thesis, Imperial College, London.
82. Yin, Y. & Gelenbe, E. (2017). Single-cell based random neural network for deep learning. In *2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 86–93.
83. Zhao, W., Agnus, G., Derycke, V., Filoramo, A., Bourgoin, J., & Gamrat, C. (2010). Nanotube devices based crossbar architecture: toward neuromorphic computing. *Nanotechnology* 21(17): 175202.