Coincidence Detection Is All You Need

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper demonstrates that the performance of coincidence detection - a classic neuromorphic signal processing method found in Rosenblatt's perceptrons with distributed transmission times, can be competitive to a state-of-the-art deep learning method for pattern recognition. Hence, we cannot remain comfortably numb to the prevailing dogma that efficient matrix-vector operations is all we need; but should enquire with greater vigour if more advanced continual learning methods (running on spiking neural network hardware with neuromodulatory mechanisms at multiple timescales) can beat the accuracy of task-specific deep learning methods.

1 Introduction

Frank Rosenblatt and his team (1957-1971) built and analyzed several kinds of perceptrons [1, 2, 3, 4] - networks of sensory, association and receptor neurons; which in contemporary deep learning terminology relates to the input, hidden and output layers. The propagating signals were binary (compatible with a spike-based view), the synaptic delays (transmission times) and weights (memory states) could be analog, the network could be recurrent and was often randomly interconnected, and learning often meant tuning the weights of the association-receptor subnetwork by some error-corrective reinforcement. The synaptic delays were not learnt but instead randomly distributed in Rosenblatt's Tobermory perceptrons [5], and this was rich enough to realize concentration-invariant and uniform time-warp invariant spatiotemporal classification by logarithmic encoding and coincidence detection. However, the processing speed of commercial Von Neumann computers advanced exponentially and outperformed neuromorphic hardware on yesterdecade's benchmarks [6]. The Tobermory perceptron was forgotten, nevertheless, the utility of logarithmic encoding and coincidence detection was formalized by John Hopfield [7] as an efficient solution to the *analog match* problem in pattern recognition.

Now, half a century after the accidental demise of Rosenblatt, neuromorphic signal processors are making a comeback. For example, (1) Intel's Loihi with spike-time dependent plasticity mechanisms for learning olfactory pattern recognizers [8]; (2) Physical reservoir computing networks [9] where the interconnectivity of the hidden layer is unchanged, closer to the spirit of Rosenblatt's randomly interconnected sensory-association subnetwork.

Here, to strengthen the case for revisiting classic methods on novel and modern hardware, we evaluate the performance of coincidence detection in comparison to a deep learning method. Nothing more, nothing less, although this work was triggered by a rabid interest in employing artificial intelligence to sniff out infections and prevent future pandemics.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Table 1: Test accuracy (%)

ResNet-26	Coincidence detection
82.2± 0.3 (from [10])	82.7 (this work)

2 Methods

Here, we consider the work [10] of an interdisciplinary team, where a 26 layer convolutional neural network with residual connections (ResNet-26) was successfully trained for classifying pathogenic bacteria by Raman spectroscopy. In their work, there are $N = 30$ classes of bacterial isolates and they begin with a ResNet-26 pre-trained on $N \times 2000$ spectra, then for each class $n = 1 : N$ there are $M = 100$ training spectra, and similarly $N \times M = 3000$ test spectra. Each spectrum $\boldsymbol{x}$ contains 1000 floating-point numbers ranging between 0 and 1. Although compute intensive, their deep learning method proved to be a tool of great convenience for pattern recognition in a challenging dataset, where intra-isolate spectra were often more dissimilar than inter-isolate spectra.

Our method to tackle the above dataset, is inspired by the theory of how coincidence detection [7] in animal brains is fundamental for odour classification in complex and turbulent mixtures. Each class n has a vector representation $\boldsymbol{w}_n$ that is learnt, and an input vector $\boldsymbol{x}$ results in an output class $y(\boldsymbol{x}) = \arg_n \max(\boldsymbol{x} \bigwedge \boldsymbol{w}_n)$ where we introduce the operator $\bigwedge$ to represent the coincidence between two signals. The analytical nature of coincidence detection depends on the specificities of the ion-channels and the membranes involved [11], and may even incorporate nonlinear leaky-integrate [12] multiple timescale mechanisms. We do not yet have a complete theory of neuromorphic signal processing, so here we introduce an approximation for the translation and scale-invariant property of coincidence detection as

$$\arg_n \max(\boldsymbol{x} \bigwedge \boldsymbol{w}_n) \approx \arg_n \max(\boldsymbol{w}_n \cdot \hat{\boldsymbol{x}}), \tag{1}$$

where $\hat{\boldsymbol{x}}$ is the zero-mean unit-variance normalization of $\boldsymbol{x}$.

Thus, the approximation in Eq. (1) allows $y(\boldsymbol{x})$ to be learnt by a logistic regression on the normalized dataset. We discard the pre-training data, pre-process the training and test spectra by a range-1 mean filter, and use the default method for logistic regression in Wolfram Mathematica (L2-regularization = 0.0001, optimization method = limited-memory BFGS). Code is provided in the supplemental material for reproducibility.

3 Result and outlook

The coincidence detection (via normalized logistic regression) method introduced here achieves a test accuracy greater than ResNet-26 (see Table 1), and it took less than 3 seconds to train the classifier on a modern desktop (without any special-purpose GPUs). Check the Appendix for a confusion matrix plot of the training and test data. Note that the training data was fit all at once to a 100% accuracy. With a more neuromorphic coincidence detection method and a learning method that adapts the synaptic delays $\boldsymbol{w}$ continually, to keep track under changing environmental conditions, we may achieve even greater accuracies.

References

[1] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, Inc. Report no. 85-460-1, 1957.

[2] Frank Rosenblatt. *The perceptron: A theory of statistical separability in cognitive systems*. Cornell Aeronautical Laboratory, Inc. Report no. VG-1196-G-1, 1958.

[3] Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Cornell Aeronautical Laboratory, Inc. Report no. 1196-G-8, 1961.

[4] Frank Rosenblatt. Cognitive systems research program. Technical report, Cornell University, Ithaca, New York, 1971.

[5] Frank Rosenblatt. A description of the tobermory perceptron. In *Collected Technical Papers*, volume 2. Cornell University, Ithaca, New York, 1963.

[6] George Nagy. Neural networks-then and now. *IEEE Transactions on Neural Networks*, 2(2):316–318, 1991.

[7] John J Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.

[8] Nabil Imam and Thomas A Cleland. Rapid online learning and robust recall in a neuromorphic olfactory circuit. *Nature Machine Intelligence*, 2(3):181–191, 2020.

[9] G. Tanaka, T. Yamane, J.B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.

[10] Chi-Sing Ho, Neal Jean, Catherine A Hogan, Lena Blackmon, Stefanie S Jeffrey, Mark Holodniy, Niaz Banaei, Amr AE Saleh, Stefano Ermon, and Jennifer Dionne. Rapid identification of pathogenic bacteria using raman spectroscopy and deep learning. *Nature communications*, 10(1):1–8, 2019.

[11] Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206–221, 2008.

[12] Wondimu Teka, Toma M Marinov, and Fidel Santamaria. Neuronal spike timing adaptation described with a fractional leaky integrate-and-fire model. *PLoS computational biology*, 10(3):e1003526, 2014.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Table 1.
   (b) Did you describe the limitations of your work? [Yes] Equation (1) makes it clear that we employ an approximation for coincidence detection.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Check supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] qualitatively, in the results section
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes] In the supplemental information

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

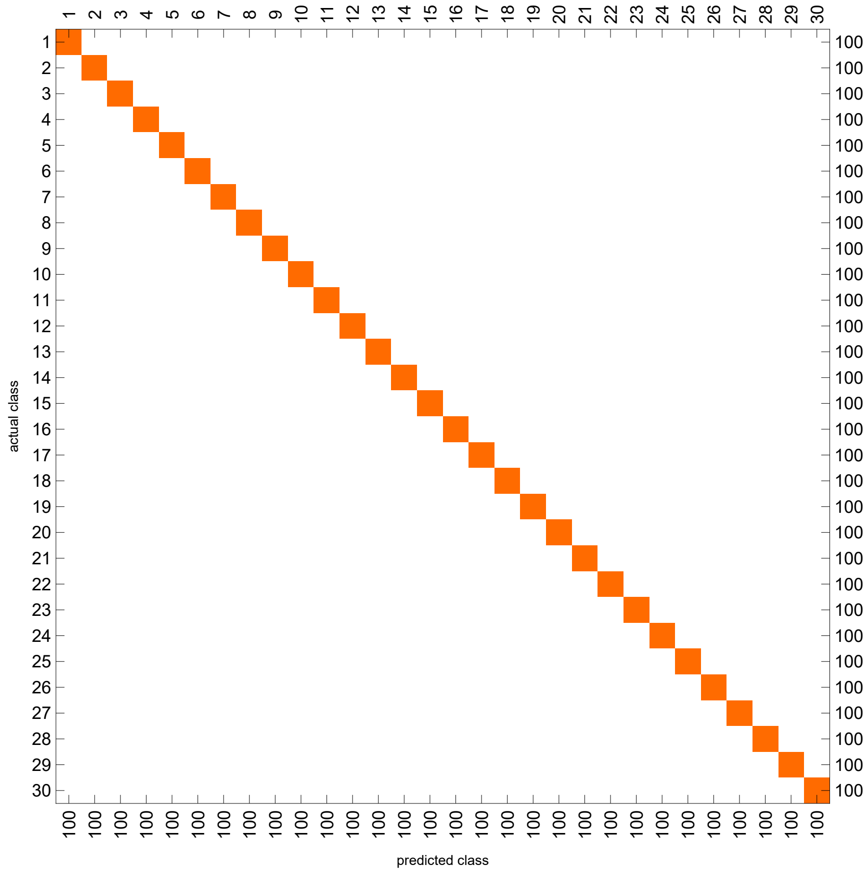
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

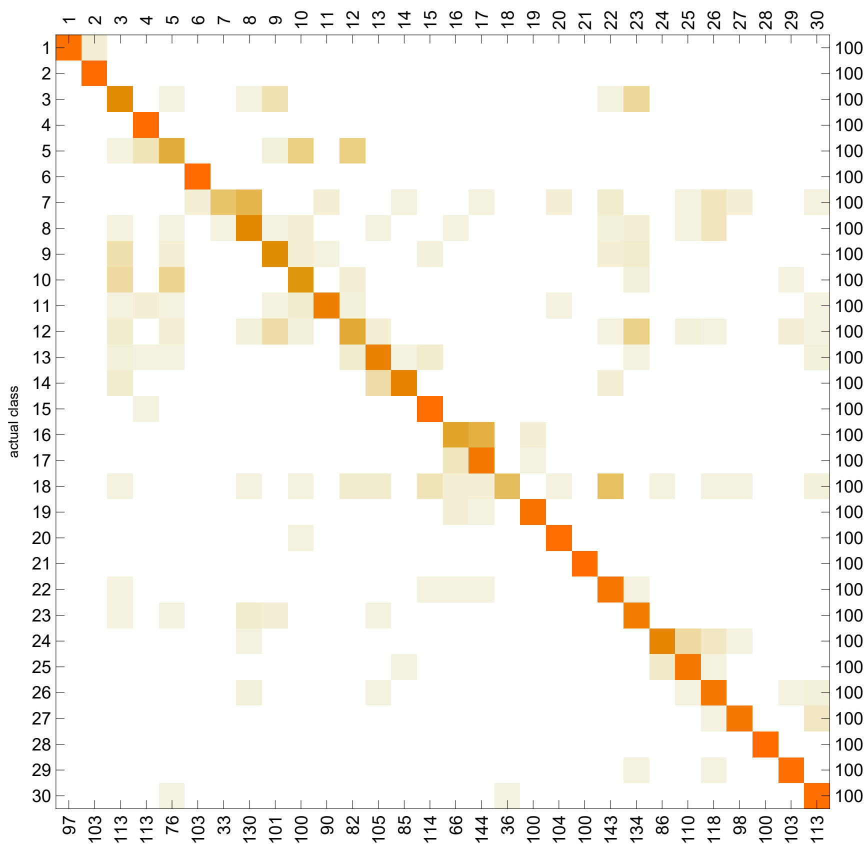
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Confusion matrix of the training and test data. Wolfram Mathematica code to reproduce these results is provided as supplemental material.


actual class
predicted class


actual class
predicted class

134