nature REVIEWS NEUROSCIENCE

SUPPLEMENTARY INFORMATION

In format as provided by the authors

Backpropagation and the brain

Timothy P. Lillicrap , Adam Santoro, Luke Marris, Colin J. Akerman and Geoffrey Hinton

https://doi.org/10.1038/s41583-020-0277-3

Appendix: Backpropagation and the brain

Timothy P. Lillicrap[1,2*], Adam Santoro[1*], Luke Marris[1], Colin Akerman[3], Geoffrey Hinton[4,5]

[1]DeepMind, London, UK. [2]Centre for Computation, Mathematics and Physics, University College London, London, UK. [3]Department of Pharmacology, University of Oxford, Oxford, UK. [4]Department of Computer Science, University of Toronto, Toronto, Canada. [5]Google Brain, Toronto, Canada.

* Denotes equal contribution.
Correspondence: Timothy Lillicrap countzero@google.com, Geoffrey Hinton geoffhinton@google.com

Spiking neurons

While neurons in the brain communicate using spikes, most artificial neural networks are trained using neurons that communicate real values. This discrepancy has sometimes been viewed as a stumbling block for linking backprop with learning in the brain[1–3]. However, recent work in machine learning suggests this apparent issue may not present a significant impediment to understanding how cortex approximates backprop[4–6]. In a spiking neuron, the spike train is often a noisy realization of the underlying firing rate. In this case, if errors are represented as activity differences, the post-synaptic term in the learning rule needs to measure changes in this underlying rate. It can do this by applying a derivative filter to the post-synaptic spike train. The derivative filter compares the firing rate just before a pre-synaptic spike with the firing rate just after a pre-synaptic spike. This looks exactly like spike-time dependent plasticity[7,8]. Naturally, the output of the derivative filter will be a very noisy estimate of the change in the underlying firing rate, but stochastic gradient descent is extremely robust to noise, provided the noise is unbiased. Indeed, adding random noise to neural activities during training has been shown to greatly improve the ability of neural networks to generalize well to novel data[6], so rather than viewing spikes as a clumsy way to convey an underlying firing rate, we can view them as a very effective regularizer that allows us to fit large neural nets to relatively small amounts of data.

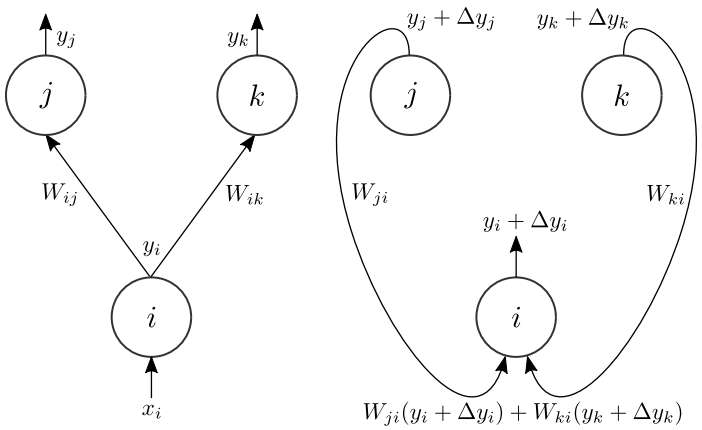
Connections with unsupervised learning

A distinction is often made between backprop and unsupervised algorithms[9]. However, this is a false dichotomy that likely arose from the fact that backprop was first developed in a supervised setting[10]. Unsupervised algorithms are characterized by the lack of output labels for targets, but there is no issue with employing backprop in these algorithms. Without output labels, learning may occur *within* a modality by trying to predict one part of an input from the remainder of the

input[11–13]. Unsupervised learning can also occur across modalities[14] or across time[15–17]: i.e., where the future activity of a network is the target for the prediction. Not only is backprop compatible with unsupervised learning, it underlies the most powerful unsupervised algorithms developed to date[13,18,19]. In short, the issue of effective learning across multiple layers exists in the case of unsupervised learning as well and backpropagation is well suited to the task.

Backpropagation-through-time

In this review we address the question of how the brain might learn across networks with multiple layers. We have not addressed the difficult issue of how the brain might optimize recurrent networks that process time-varying inputs. To learn from temporal data, artificial neural networks make extensive use of backpropagation-through-time[20–22] (BPTT). It is much harder to see how BPTT could be implemented in cortex because each neuron must remember its activity value at many different time steps during the forward pass and then use these remembered activities to compute weight updates during the subsequent backward pass. Rather than using BPTT, we suspect the cortex may rely on approximations such as eligibility traces[23–25], or approaches wherein fast temporary changes in synaptic weights are used to store recent hidden activity vectors[26]. In the latter case, the cortex could do associative retrieval of relevant recent activity vectors in order to learn long-term dependencies without having to explicitly go back through the intermediate time-steps.



Supplementary Figure 1: The two last layers of a simple feedforward network. The total input to unit i is x_i and its output is $y_i = f(x_i)$ where f is a smooth non-linear function. After an initial forward pass, the network's initial output vector (y_j, y_k) is compared with the target and is moved towards the target by a small amount, $(\Delta y_j, \Delta y_k)$, that is proportional to the difference. For linear output units with a quadratic loss or for logistic output units with a cross-entropy loss, this ensures that $\Delta y_j, \Delta y_k$ represent the derivatives of the loss with respect to the *total inputs*, x_j and x_k to the output units. To compute the derivatives in earlier layers, we can now make use of the following curious fact: If the perturbation in the *input* to a unit represents the derivative of the loss w.r.t. the *output* of the unit, the resulting perturbation of its *output* represents the derivative w.r.t. its *input*. The modified output vector is used to reconstruct the activity in the previous layer via backwards connections that have the same weights as the forward connections (i.e. $W_{ij} = W_{ji}$ and $W_{ik} = W_{ki}$)). We assume that these weights have already been trained to be a perfect autoencoder, so if the output vector had not been perturbed, the top-down input of $W_{ij}y_j + W_{ik}y_k$ to unit i would produce the same output, y_i, as was computed on the forward pass. The small additional input $W_{ij}\Delta y_j + W_{ik}\Delta y_k$ will be converted into a small additional output which will be the additional input times the gradient of the non-linear function f. So, to first order, $\Delta y_i = dy_i/dx_i(W_{ij}\Delta y_j + W_{ik}\Delta y_k)$. This is exactly the derivative prescribed by backpropagation, so a perturbation in the output layer that represents the derivatives of the loss with respect to the inputs to that layer, causes a perturbation in the previous layer that represents the same quantity for the previous layer. This can be repeated for as many earlier layers as required. The learning rule is then to modify each incoming weight in proportion to the product of the pre-synaptic activity and the change in the post-synaptic activity.

References

1. Crick, F. The recent excitement about neural networks. *Nature* **337,** 129 (1989).
2. Guergiuev, J., Lillicrap, T. P. & Richards, B. A. Deep learning with segregated dendrites. *arXiv preprint arXiv:1610.00161* (2016).

3. Samadi, A., Lillicrap, T. P. & Tweed, D. B. Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural computation* (2017).

4. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247* (2014).

5. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. *Binarized neural networks* in *Advances in neural information processing systems* (2016), 4107–4115.

6. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15,** 1929–1958 (2014).

7. Hinton, G. How to do backpropagation in a brain. *NIPS 2007 Deep Learning Workshop* (2007).

8. Bengio, Y., Mesnard, T., Fischer, A., Zhang, S. & Wu, Y. Stdp-compatible approximation of backpropagation in an energy-based model. *Neural computation* (2017).

9. Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science* **11,** 23–63 (1987).

10. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagation errors. *Nature* **323,** 533–536 (1986).

11. Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. *Greedy layer-wise training of deep networks* in *Advances in neural information processing systems* (2007), 153–160.

12. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. *Extracting and composing robust features with denoising autoencoders* in *Proceedings of the 25th international conference on Machine learning* (2008), 1096–1103.

13. Oord, A. v. d., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).

14. Ngiam, J. *et al. Multimodal deep learning* in *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), 689–696.

15. Srivastava, N., Mansimov, E. & Salakhudinov, R. *Unsupervised learning of video representations using lstms* in *International conference on machine learning* (2015), 843–852.

16. Mikolov, T., Karafiát, M., Burget, L., Černocky, J. & Khudanpur, S. *Recurrent neural network based language model* in *Eleventh Annual Conference of the International Speech Communication Association* (2010).

17. Gemici, M. *et al.* Generative Temporal Models with Memory. *arXiv preprint arXiv:1702.04649* (2017).

18. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

19. Gregor, K., Danihelka, I., Graves, A., Rezende, D. J. & Wierstra, D. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* (2015).

20. Rumelhart, D. E., McClelland, J. L., Group, P. R., *et al. Parallel distributed processing* (IEEE, 1988).

21. Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **78,** 1550–1560 (1990).

22. Lillicrap, T. P. & Santoro, A. Backpropagation through time and the brain. *Current opinion in neurobiology* **55,** 82–89 (2019).

23. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63,** 544–557 (2009).

24. Miconi, T. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife* **6** (2017).

25. Bellec, G. *et al.* Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv preprint arXiv:1901.09049* (2019).

26. Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z. & Ionescu, C. *Using fast weights to attend to the recent past* in *Advances in Neural Information Processing Systems* (2016), 4331–4339.