Dual Propagation

Accelerating Contrastive Hebbian Learning with Dyadic Neurons

Rasmus Høier, D. Staudt, Christopher Zach

Chalmers University of Technology
Email: hier@chalmers.se
Code repo: github.com/rasmuskh/dual-propagation
Slideshow: github.com/rasmuskh/dualprop-slideshow

Motivation

Contrastive Hebbian learning (CHL) [1] and equilibrium propagation (EP) [2]

- Biological plausibility
- Neuromorphic computing

Motivation

Contrastive Hebbian learning (CHL) [1] and equilibrium propagation (EP) [2]

- Biological plausibility
- Neuromorphic computing

However, certain issues limit their usability.

1. First order gradient estimate
2. Extremely slow inference
3. Two globally synchronized inference phases

Motivation

Contrastive Hebbian learning (CHL) [1] and equilibrium propagation (EP) [2]

- Biological plausibility
- Neuromorphic computing

However, certain issues limit their usability.

1. First order gradient estimate
2. Extremely slow inference
3. Two globally synchronized inference phases

A dyadic objective

$$\mathcal{L}_\alpha(\theta) := \min_{z^+} \max_{z^-} \alpha \ell(z_L^+) + (1-\alpha)\ell(z_L^-) \\ + \sum_{k=1}^{L} \tfrac{1}{\beta_k} \Big(G_k(z_k^+) - G_k(z_k^-) + (z_k^- - z_k^+)^\top W_{k-1}(\alpha z_{k-1}^+ + (1-\alpha) z_{k-1}^-) \Big)$$

- Linear units: $G_k = \|\cdot\|^2/2$
- ReLU units: $G_k = \|\cdot\|^2/2 + \imath_{\geq 0}(\cdot)$
- $\alpha \in [0, 1]$

Choice of α

- $\alpha = 1$ and $\alpha = 0$:
  - LPOM-like [3, 4] pure minimization objectives
  - Slow iterative inference

Choice of α

- $\alpha = 1$ and $\alpha = 0$:
  - LPOM-like [3, 4] pure minimization objectives
  - Slow iterative inference
- $\alpha \in (0, 1)$ excluding $\alpha = 1/2$
  - Fast inference
  - Lacks convergence guarantee

Choice of α

- $\alpha = 1$ and $\alpha = 0$:
  - LPOM-like [3, 4] pure minimization objectives
  - Slow iterative inference
- $\alpha \in (0, 1)$ excluding $\alpha = 1/2$
  - Fast inference
  - Lacks convergence guarantee
- $\alpha = 1/2$:
  - Fast inference
  - Convergence guarantee (details in paper)

Assume $\alpha = 1/2$ from now on.

Connection to CHL and EP

Defining energy functions E_k as in CHL allows reformulating $\mathcal{L}_{\frac{1}{2}}$.

- $E_k(z_k, z_{k-1}) := G_k(z_k) - z_k^T W_{k-1} z_{k-1}$
- $\bar{z}_k := \frac{1}{2}(z_k^+ + z_k^-)$

$$\mathcal{L}_{\frac{1}{2}}(\theta) = \min_{z^+} \max_{z^-} \ell(z_L^+) + \ell(z_L^-) + \sum_{k=1}^{L} \frac{1}{\beta_k}\left(E(z_k^+, \bar{z}_{k-1}) - E(z_k^-, \bar{z}_{k-1})\right)$$

Similar to CHL and EP objectives but z_k^+ and z_k^- are inferred simultaneously and ”tethered” via $\bar{z}_{k-1}$.

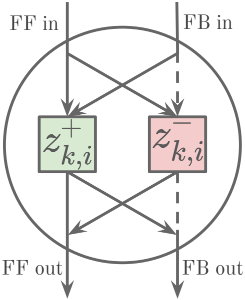
Inference

- Neurons within layers are not coupled.
- Fully optimize $\mathcal{L}_{\frac{1}{2}}$ with respect to z_k^+ and z_k^- in a single block-coordinate descent step!

$$z_k^\pm \leftarrow f_k\left(\tfrac{1}{2}W_{k-1}(z_{k-1}^+ + z_{k-1}^-) \pm \tfrac{\beta_k}{2\beta_{k+1}}W_k^\top(z_{k+1}^+ - z_{k+1}^-)\right)$$

- Same runtime as BP and >100X faster than EP and CHL.

Dyadic neurons



$$z_k^{\pm} \leftarrow f_k\left(\underbrace{\tfrac{1}{2}W_{k-1}(z_{k-1}^{+}+z_{k-1}^{-})}_{\text{FF in}} \pm \underbrace{\tfrac{\beta_k}{2\beta_{k+1}}W_k^{\top}(z_{k+1}^{+}-z_{k+1}^{-})}_{\text{FB in}}\right)$$

$$\text{FF out} = \frac{1}{2}(z_{k,i}^{+}+z_{k,i}^{-})$$

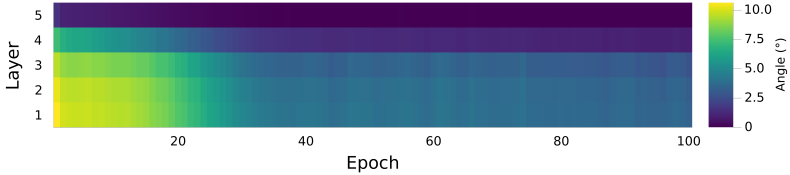
$$\text{FB out} = \frac{1}{2}(z_{k,i}^{+}-z_{k,i}^{-})$$

Learning

- Fully local contrastive Hebbian gradient
- Second order gradient estimate
- Discounting not needed ($\beta_k = 1 \; \forall \; k$ employed in experiments)

$$
\begin{aligned}
\frac{\partial}{\partial W_{k-1}} \mathcal{L}_{\frac{1}{2}} &= \frac{1}{\beta_k} \left(\frac{E(z_k^{+}, \bar{z}_{k-1})}{\partial w_{k-1}} - \frac{E(z_k^{-}, \bar{z}_{k-1})}{\partial w_{k-1}} \right) \\
&= \frac{1}{2\beta_k} \left(z_k^{-} - z_k^{+} \right) \left(z_{k-1} + z_{k-1} \right)^{\top}.
\end{aligned}
$$

- Excellent gradient alignment between BP and DP



Experiments: MLP (MNIST)

- DP: cost-efficient (forwards + backwards)
- MS-DP: Multiple steps of inference and weight updates per datapoint
- L-DP: ”Lazily” let old states persist providing potentially harmful feedback.
- P-DP: parallel neuron updates
- R-DP-100: Random update sequence

Method	BP	DP	MS-DP	L-DP	P-DP	R-DP-100
Test acc (%)	98.45 $\pm$0.04	98.43 $\pm$0.03	98.40 $\pm$0.02	98.42 $\pm$0.07	98.47 $\pm$0.04	98.48 $\pm$0.11

Experiments: VGG16

DP matches back-propagation both in terms of runtime and accuracy.

Method		BP	DP	KP-DP†	EP* [5]	DTP* [6]
CIFAR10	Top-1	92.26 ± 0.23	92.30 ± 0.11	91.84 ± 0.11	88.6 ± 0.2	89.38 ± 0.20
CIFAR100	Top-1	69.63 ± 0.24	69.57 ± 0.51	70.40 ± 0.25	61.6 ± 0.1	—
	Top-5	88.13 ± 0.22	88.36 ± 0.13	88.57 ± 0.15	86.0 ± 0.1	—
ImageNet32x32	Top-1	41.28 ± 0.19	41.48 ± 0.19	—	36.5 ± 0.3	36.81
	Top-5	64.89 ± 0.11	64.90 ± 0.13	—	60.8 ± 0.4	60.54

(†) Dual propagation with Kolen-Pollack learning of feedback weights.

(*) High computational costs limit EP (Laborieux 2022) and DTP (Ernoult 2022) to 5-7 layer VGG-like networks.

Experiments: VGG16

DP matches back-propagation both in terms of runtime and accuracy.

Method		BP	DP	KP-DP†	EP* [5]	DTP* [6]
CIFAR10	Top-1	92.26 ± 0.23	92.30 ± 0.11	91.84 ± 0.11	88.6 ± 0.2	89.38 ± 0.20
CIFAR100	Top-1	69.63 ± 0.24	69.57 ± 0.51	70.40 ± 0.25	61.6 ± 0.1	—
	Top-5	88.13 ± 0.22	88.36 ± 0.13	88.57 ± 0.15	86.0 ± 0.1	—
ImageNet32x32	Top-1	41.28 ± 0.19	41.48 ± 0.19	—	36.5 ± 0.3	36.81
	Top-5	64.89 ± 0.11	64.90 ± 0.13	—	60.8 ± 0.4	60.54

(†) Dual propagation with Kolen-Pollack learning of feedback weights.

(*) High computational costs limit EP (Laborieux 2022) and DTP (Ernoult 2022) to 5-7 layer VGG-like networks.

Conclusion

- Unlike previous CHL methods DP computes errors across compartments rather than across time.
- The dyadic neuron model enables closed-form inference rules and a second order gradient estimate.
- Many viable update schemes including random and parallel updates.
- The efficient DP implementation matches back-propagation both in terms of accuracy and runtime.

Future work: Continuous inference and learning in a streaming setting.

Code repo: github.com/rasmuskh/dual-propagation
Slideshow: github.com/rasmuskh/dualprop-slideshow

Extra slides

Back of the envelope runtime comparisson

- CIFAR10 runtime per epoch in seconds for different implementations.
- Estimates based on numbers from supplemental material of [5] and [6].

Method	DP	H-EP [5]	DTP [6]
Seconds/epoch	3.5	1700	240
Layers	16	7	6

- Differences in hardware, software, model size and batch size makes this comparisson indicative only.

References

[1] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, p. 24, 2017.

[2] X. Xie and H. S. Seung, "Equivalence of backpropagation and contrastive hebbian learning in a layered network," *Neural computation*, vol. 15, no. 2, pp. 441–454, 2003.

[3] J. Li, C. Fang, and Z. Lin, "Lifted proximal operator machines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4181–4188, 2019.

[4] C. Zach, "Bilevel programs meet deep learning: A unifying view on inference learning methods," *arXiv preprint arXiv:2105.07231*, 2021.

[5] A. Laborieux and F. Zenke, "Holomorphic equilibrium propagation computes exact gradients through finite size oscillations," *arXiv preprint arXiv:2209.00530*, 2022.

[6] M. M. Ernoult, F. Normandin, A. Moudgil, S. Spinney, E. Belilovsky, I. Rish, B. Richards, and Y. Bengio, "Towards scaling difference target propagation by learning backprop targets," in *International Conference on Machine Learning*, pp. 5968–5987, PMLR, 2022.