The Loss Landscape of Dense Associative Memory

Robin Thériault[1]*
Supervised by Daniele Tantari[2]

[1]Scuola Normale Superiore di Pisa
[2]Department of Mathematics, University of Bologna

October 31, 2024

*robin.theriault@sns.it

Review

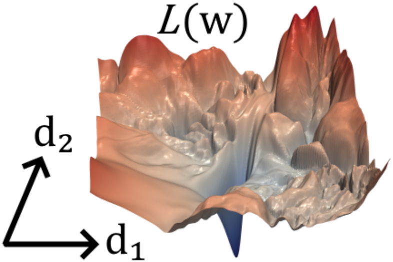
Purpose of (generative) ML: fit data $\mathbf{x}$ and labels y with $\mathrm{P}\left(\mathbf{x}, y \middle| \mathbf{w}\right)$.

[Lecun et al., 1998]: MNIST digits.

One typically minimizes a loss $L\left(\mathbf{w}; \mathbf{x}, y\right)$ to find the parameters $\mathbf{w}$.

Neural Network Loss Landscape

Previous works investigated the loss $L(\mathbf{w})$ of neural networks (NN) as a function of parameters $\mathbf{w}$ to study how they learn.

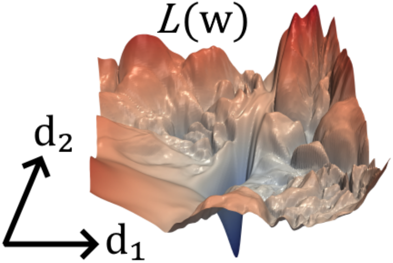


[Li et al., 2018]: visualizing the loss landscape.

- [Choromanska et al., 2015]: local mins almost as good as global.
- [Dauphin et al., 2014]: large NNs have much more saddles than local mins.

Neural Network Loss Landscape

Previous works investigated the loss $L(\mathbf{w})$ of neural networks (NN) as a function of parameters $\mathbf{w}$ to study how they learn.

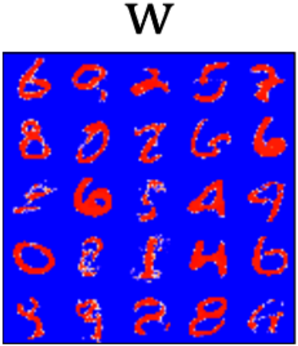


[Li et al., 2018]: visualizing the loss landscape.

Characterizing and classifying critical points is still an open problem. See [Zhang et al., 2021] for recent progress.

Dense Associative Memory (DAM)

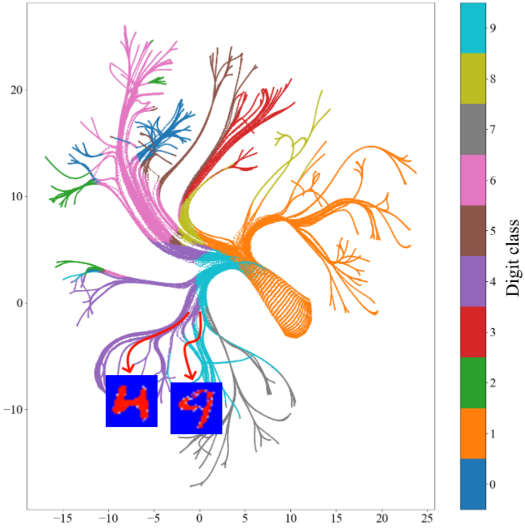
A way forward: dense associative memory (DAM), related to transformers [Ramsauer et al., 2020].
DAM learns data archetypes or *memories* as min of $L(\mathbf{w})$.



[Krotov and Hopfield, 2016]

Can the other critical points be characterized as well?

Dense Associative Memory (DAM)

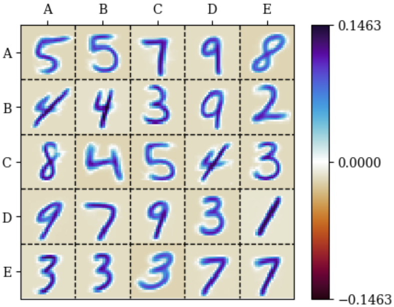


[Boukacem et al., 2024]: Parameters during training

Our DAM

Our work: characterize DAM saddle points.

- We design a new DAM, different form [Krotov and Hopfield, 2016, Boukacem et al., 2024].
- Easier to study using statistical mechanics.
- Verify it learns similar memories $\mathbf{w}$ as the previous version:



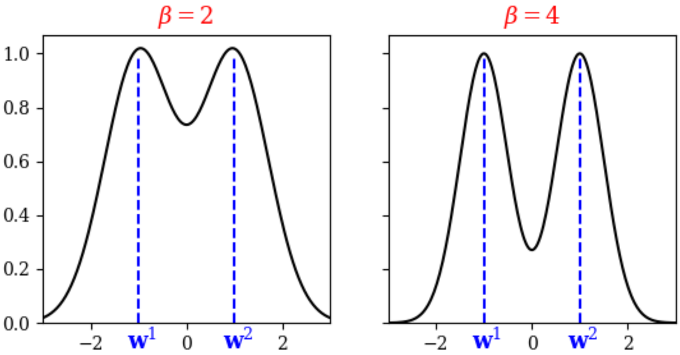
Our DAM

Recall (generative) ML fits data $\mathbf{x}$ and labels y with $\mathrm{P}(\mathbf{x}, y|\mathbf{w})$.

Inverse temperature

$$\text{Our DAM: } \mathrm{P}(\mathbf{x}, y|\mathbf{w}, \mathbf{g}) = \sum_{\mu=1}^{P} \mathbf{g}_y^\mu \exp\left(\beta \sum_{i=1}^{N} \mathbf{w}_i^\mu \mathbf{x}_i\right)$$

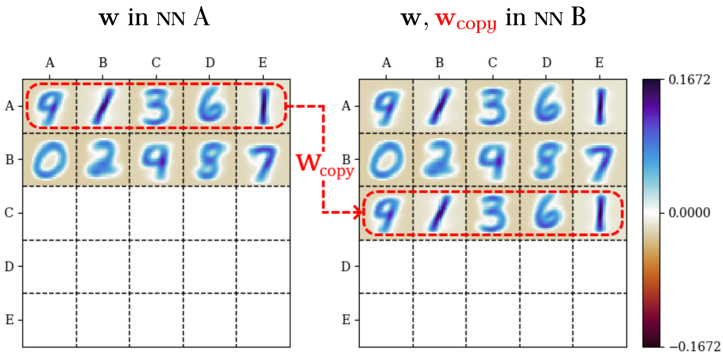
Memories



Fixed points

Let $\mathbf{w}_{\text{copy}}$ be a subset of the memories $\mathbf{w}$ of NN A, and let NN B have memories $\mathbf{w}, \mathbf{w}_{\text{copy}}$.

We show that, if $\mathbf{w}$ is a fixed point of $L(\mathbf{w})$ in NN A, then $\mathbf{w}, \mathbf{w}_{\text{copy}}$ is a fixed point of $L\left(\mathbf{w}, \mathbf{w}_{\text{copy}}\right)$ in NN B.



When β is large enough, $\mathbf{w}, \mathbf{w}_{\text{copy}}$ is unstable, i.e. a saddle.

Split Memories to Accelerate Training

Exploit saddles to accelerate training.

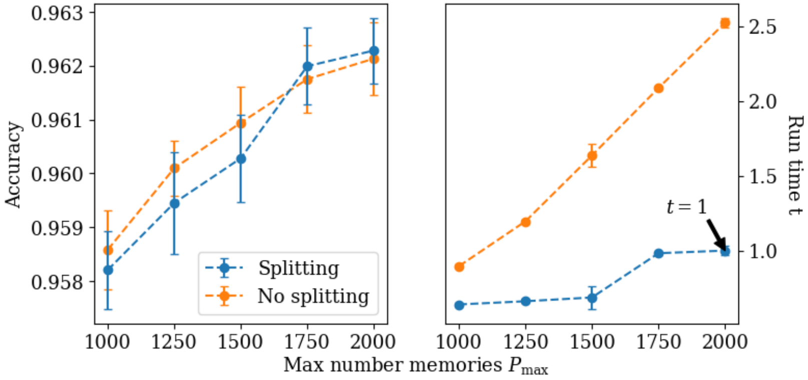
Splitting steepest descent [Wu et al., 2019, Wang et al., 2019]

1: Let P be the no. of memories $\mathbf{w}$
2: min $L(\mathbf{w})$
3: **while** $P_{\text{cur}} < P_{\text{max}}$ **do**
4: $\quad \mathbf{w} \leftarrow \mathbf{w}, \mathbf{w}_{\text{copy}}$
5: $\quad$ min $L(\mathbf{w})$
6: **end while**

While $\mathbf{w}, \mathbf{w}_{\text{copy}}$ is a saddle, each min step improves $L(\mathbf{w})$.

Split Memories to Accelerate Training

- Trains faster than using $P_{\max}$ memories from scratch.
- The speedup scales with $P_{\max}$.



Could be useful to train transformers [Ramsauer et al., 2020].

Summary

- We derive a dense associative memory (DAM) model amenable to statistical mechanics calculations.
- Show splitting memories transforms DAM minima into saddles.
- Exploit splitting to accelerate training by 2 times or more, which could be applied to transformers [Ramsauer et al., 2020].

Saddle points are often seen as a hindrance, but here they are useful. Perhaps they are misunderstood.

Summary

- We derive a dense associative memory (DAM) model amenable to statistical mechanics calculations.
- Show splitting memories transforms DAM minima into saddles.
- Exploit splitting to accelerate training by two times or more, which could be applied to transformers [Ramsauer et al., 2020].

The saddle points

↓

Bibliography I

Boukacem, N. E., Leary, A., Thériault, R., Gottlieb, F., Mani, M., and François, P. (2024).
Waddington landscape for prototype learning in generalized hopfield networks.
Phys. Rev. Res., 6:033098.

Choromanska, A., Henaff, M., Mathieu, M., Ben Arous, G., and LeCun, Y. (2015).
The Loss Surfaces of Multilayer Networks.
In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 192–204, San Diego, California, USA. PMLR.

Bibliography II

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014).
Identifying and attacking the saddle point problem in high-dimensional non-convex optimization.
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Krotov, D. and Hopfield, J. J. (2016).
Dense associative memory for pattern recognition.
Advances in neural information processing systems, 29.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998).
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324.

Bibliography III

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018).
Visualizing the loss landscape of neural nets.
In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. (2020).
Hopfield networks is all you need.
arXiv preprint arXiv:2008.02217.

Wang, D., Li, M., Wu, L., Chandra, V., and Liu, Q. (2019).
Energy-Aware Neural Architecture Optimization with Fast Splitting Steepest Descent.
arXiv e-prints, page arXiv:1910.03103.

Bibliography IV

Wu, L., Wang, D., and Liu, Q. (2019).
Splitting steepest descent for growing neural architectures.
In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhang, Y., Zhang, Z., Luo, T., and Xu, Z. J. (2021).
Embedding principle of loss landscape of deep neural networks.
In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14848–14859. Curran Associates, Inc.

Saddle-point equations

$$\bar{\mathbf{x}}_i^{\mu} = \frac{1}{P^*}\sum_{\gamma_*} \mathbf{x}_i^{\gamma_*}\sigma_\mu\left(\beta m^{\gamma_*} - s^{\gamma_*}\right)$$

$$\bar{\mathbf{y}}_y^{\gamma} = \frac{1}{P^*}\sum_{\gamma_*} \mathbf{y}_y^{\gamma_*}\sigma_\gamma\left(\beta m^{\gamma_*} - s^{\gamma_*}\right)$$

$$m^{\mu_*\mu} = \varsigma\left(2\beta\alpha\sqrt{\sum_i \left[\bar{\mathbf{x}}_i^{\mu}\right]^2}\right)\frac{\sum_i \mathbf{x}_i^{\mu_*}\bar{\mathbf{x}}_i^{\mu}}{\sqrt{\sum_i \left[\bar{\mathbf{x}}_i^{\mu}\right]^2}}$$

$$s^{\gamma_*\gamma} = -\sum_y \mathbf{y}_y^{\gamma_*}\log\left[\frac{\mathrm{C}\left(\gamma\right)\bar{\mathbf{y}}_y^{\gamma}}{\mathrm{P}\left(y\right)\sum_{y'}\bar{\mathbf{y}}_{y'}^{\gamma}}\right]$$

Additional plots