The loss landscape of deep linear neural networks: A second-order analysis

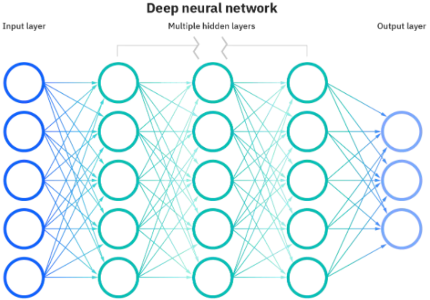
El Mehdi Achour[1] François Malgouyres[2] Sébastien Gerchinovitz[2]

[1]RWTH Aachen University
[2]Institut de Mathématiques de Toulouse ; UMR 5219
Université de Toulouse ; CNRS
UPS IMT F-31062 Toulouse Cedex 9, France

[2]Institut de Recherche Technologique Saint Exupéry, Toulouse, France

March 30th, 2025

Deep Learning

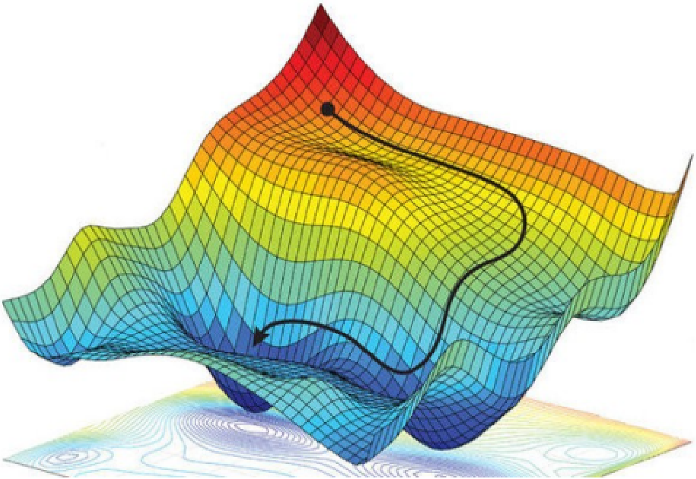


$$f_0(x) = x, \qquad f_h(x) = \sigma_h(W_h f_{h-1}(x) + b_h) \quad \forall h = 1..H$$

- Huge success in practice (e.g., Image Recognition, Natural Language Processing...)
- Not fully understood theoretically

The landscape problem

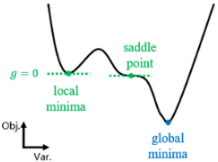
- Does a global minimizer exist?
- Do (stochastic) gradient-based algorithms converge?
- If so, what are the properties of the limit points?



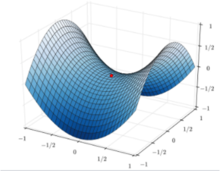
Critical points

For a smooth function L, we distinguish different types of critical points $\mathbf{W}$ (i.e. $\nabla L(\mathbf{W}) = 0$):

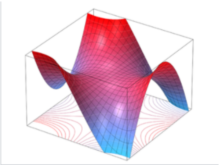
- Local (Global) minimizers/maximizers
- Saddle points: Neither a minimizer nor a maximizer:
    - Strict saddle points: $\sigma_{min}(\nabla^2 L(\mathbf{W})) < 0$
    - Non-strict saddle points: $\sigma_{min}(\nabla^2 L(\mathbf{W})) = 0$



(a) 3 types of critical points



(b) Strict saddle point



(c) Non-strict saddle point

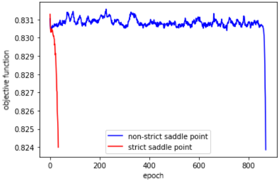
Gradient methods and saddle points

Non-convex optimization

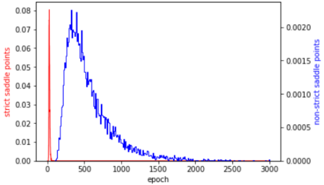
Under smoothness conditions,

- GD converges almost surely to a second-order critical point, therefore escaping strict saddle points. (Jin et al., 2017)
- Perturbed GD outputs with high probability an ε-second-order critical point in a polynomial time, therefore escaping strict saddle points (Lee et al., 2019).

Empirically for deep linear networks:



(a) Initialization close to a strict vs a non-strict saddle



(b) Histogram of escape epochs

Loss landscape of linear networks

- Multi-output regression problem, square loss
- Linear fully-connected neural network $\mathbf{W} = (W_H, \ldots, W_1)$
- $X \in \mathbb{R}^{d_x \times n}$ and $Y \in \mathbb{R}^{d_y \times n}$ the training data.
- $L(\mathbf{W}) = \sum_{i=1}^{n} \| W_H W_{H-1} \cdots W_2 W_1 x_i - y_i \|_2^2 = \| W_H \cdots W_1 X - Y \|_F^2$

Under weak assumptions on the data, we have (Baldi and Hornik 1989, Kawaguchi 2016):

Previous work

- Every critical point of L is a global minimizer or a saddle point.
- For shallow networks ($H = 2$), all the saddle points of L are strict.
- For deep networks ($H \geq 3$), $\mathbf{W} = 0$ is a non-strict saddle point of L.

Loss landscape of linear networks

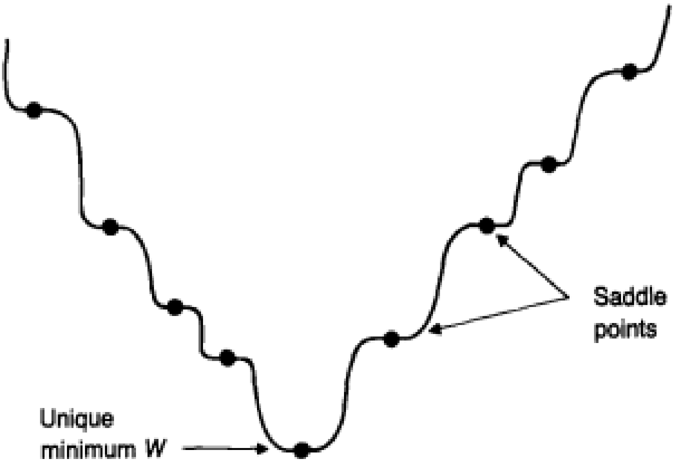
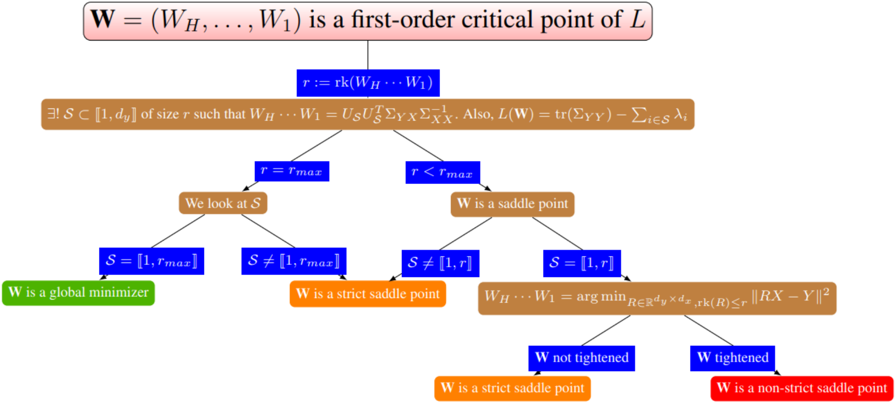


Figure: loss landscape of linear network (Baldi and Hornik'89)

Main theorem: Classification of critical points



Loss landscape of linear networks

Conclusion

- **Complete classification of critical points**: global minimizers; strict saddle points; non-strict saddle points.
- Non-strict saddle points are associated with rank-constrained minimizers of the problem.

Perspectives

- Characterize the non-strict saddles manifold and its attraction basin.
- How much time does it take to escape their vicinity?
- Characterize higher-order saddle points.
- Extend to nonlinear (or structured) networks.
- Design better algorithms.

References

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.*, 2(1):53–58, January 1989.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732, 2017.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 29:586–594, 2016.

Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical programming*, 176(1-2):311–337, 2019.