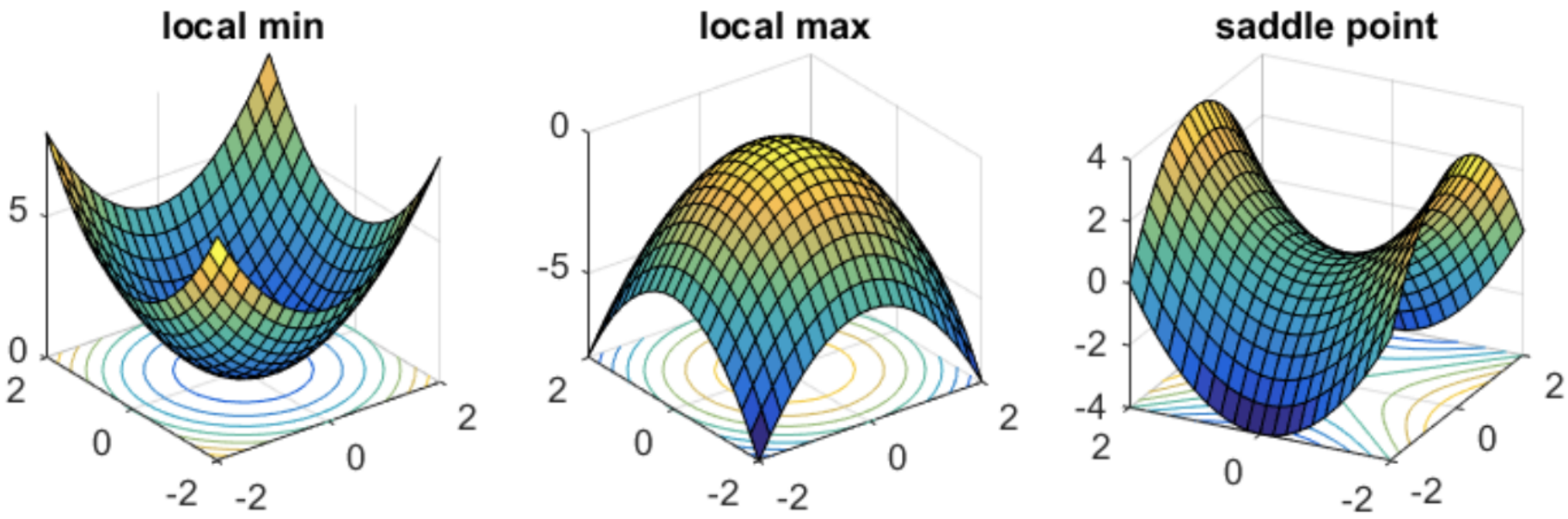
Lecture 13 CS 165

Nonconvex Optimization

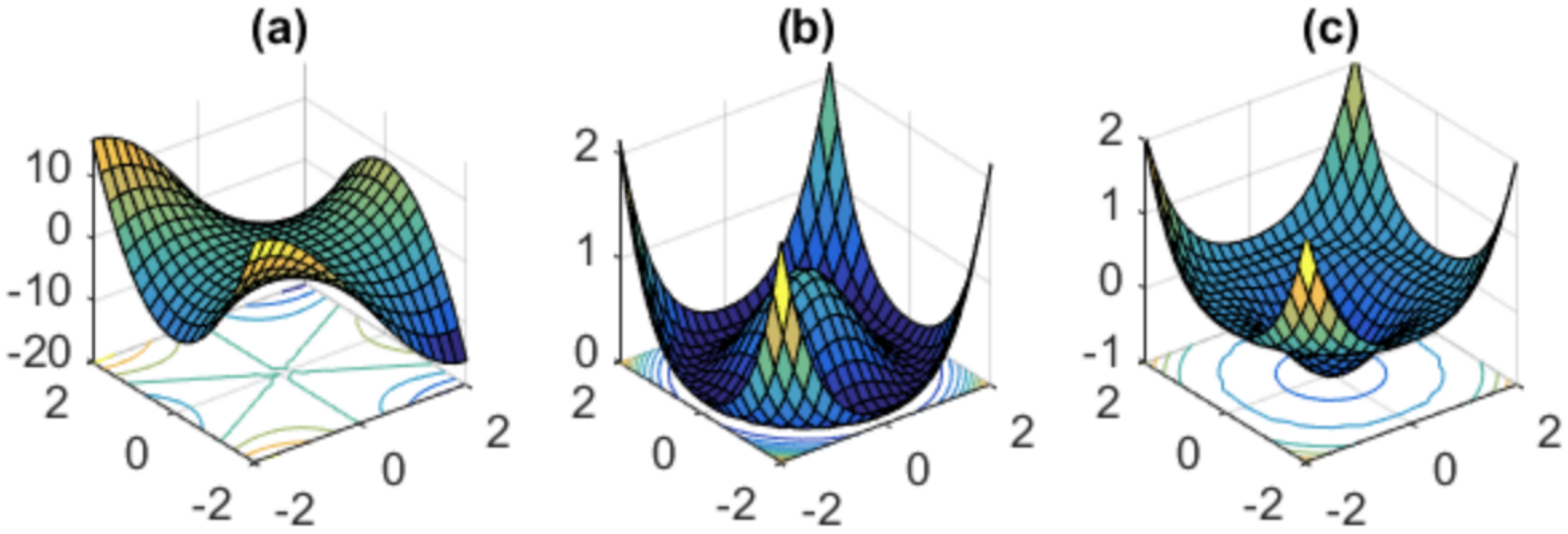
Non-convex optimization

- All loss-functions that are not convex: not very informative.
- Global optimality: too strong. Weaker notions of optimality?
- What is a saddle point?

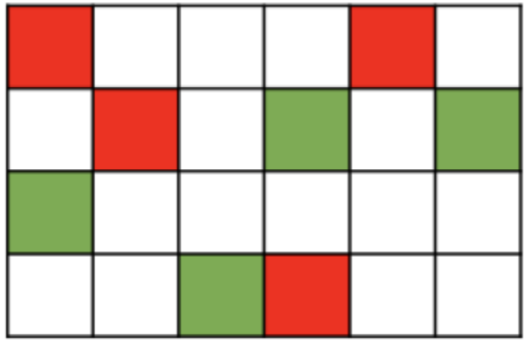
Different kinds of critical/stationary points



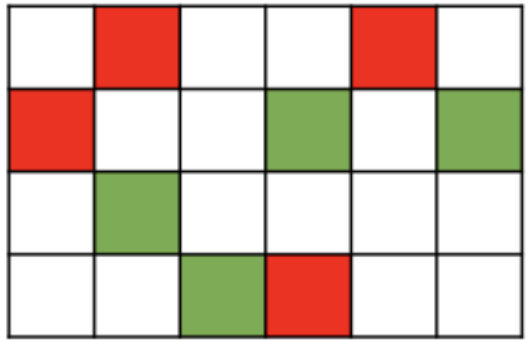
Higher order saddle points



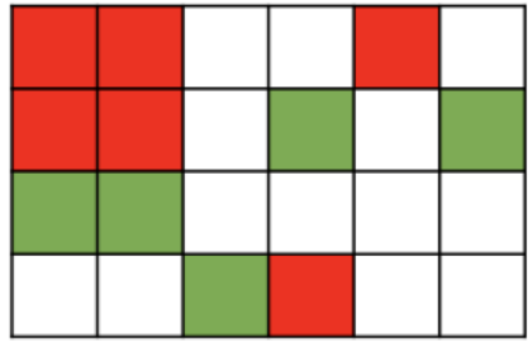
When can saddles arise: Symmetries



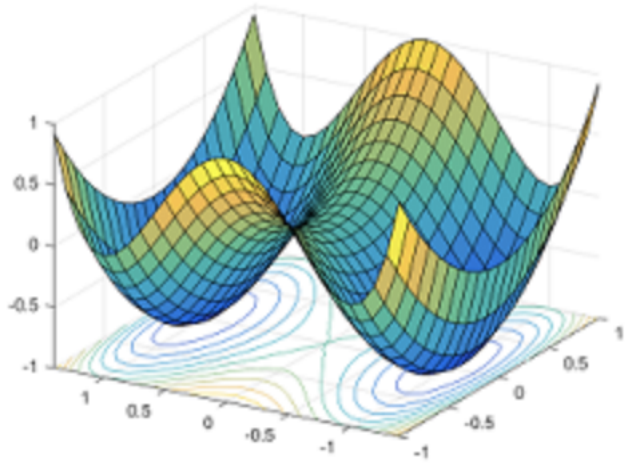
Optimal Solution



Equivalent Solution

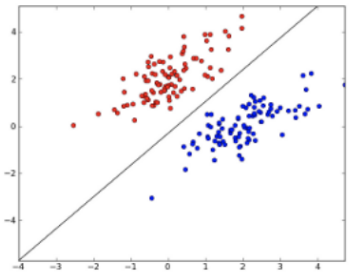


Not optimal

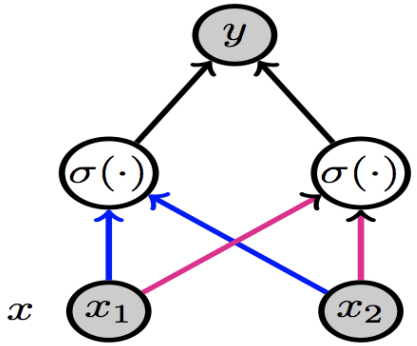


Higher order saddles: Over-parameterization

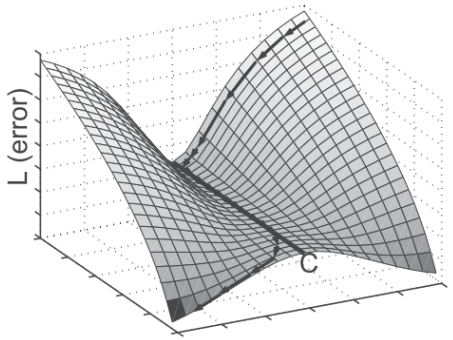
Training Data



Neural Network



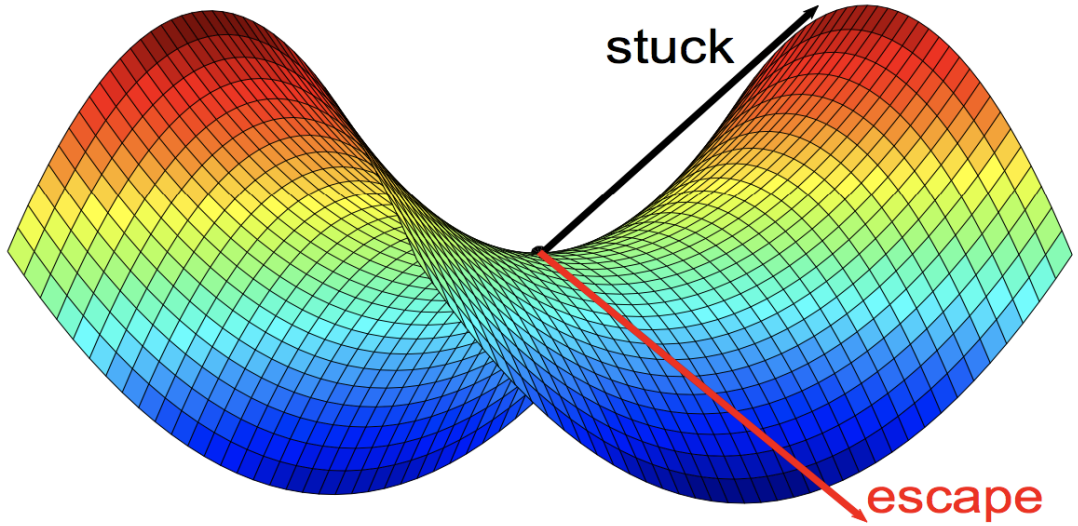
Learning Trajectory



- Overspecification: More capacity (neurons) than required.
- Each critical point in smaller network (over which function is realized) generates a set of critical points in larger network (degeneracy)
- Even global optima in smaller network can generate local optima and saddle points in larger network.
- A long time spent in the manifold of singularities.

How to escape saddle points?

- What happens to Gradient descent near saddles?
- What happens to Newton's method near saddles?
- Solution: Cubic regularization: Second order optimization.
- What about first order methods?



Stochastic Gradient Descent

- Why is it so popular?
- Why is it so special?
- What guarantees?

Stochastic Gradient Descent

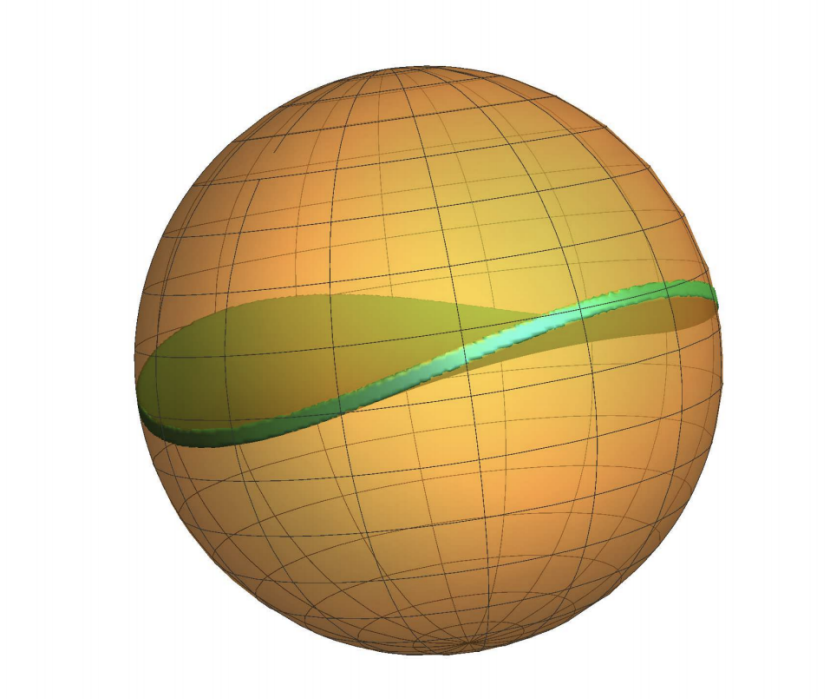


Figure 1: Pertubation ball in 3D and "thin pancake" shape stuck region

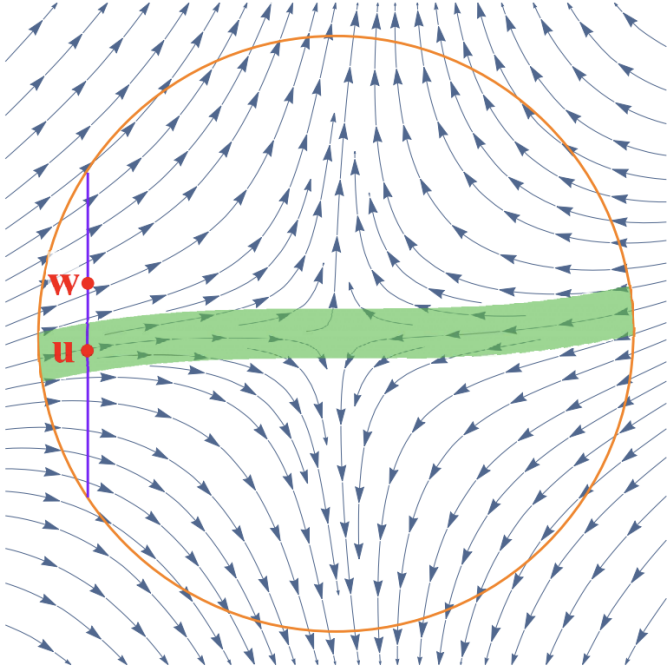


Figure 2: Pertubation ball in 2D and "narrow band" stuck region under gradient flow

Optimization in ML

- Optimizer used as a black-box in ML. We expect it to work perfectly.
- In many settings it works, e.g. over-parameterized neural networks. But extensive hyper-parameter magic (which no one talks about).
- SGD is the workhorse. Noise helps (for regularization). Second order methods are not yet scalable and don't seem to help.
- In many cases, we don't want it to work well.
  - We want it to avoid certain solutions, even globally optimal solutions. E.g. in learn multi-lingual embeddings.
  - Initialize in the hope of nudging to desirable solutions. But if a perfect optimizer, this should not matter.
  - Early stopping: We don't want it to fit loss function too perfectly, when there is not enough regularization. (Generalization, wait for a few more lectures).

References

- ICML 2016 tutorial on non-convex optimization: http://tensorlab.cms.caltech.edu/users/anima/slides/icml2016-tutorial.pdf
- “Cubic regularization of Newton method and its global performance” Nestorov, Polyak.
- Rong Ge blog https://www.offconvex.org/2017/07/19/saddle-efficiency/
- “How to Escape Saddle Points Efficiently” Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, Michael Jordan https://arxiv.org/pdf/1703.00887.pdf
- Hal Daume’s blog https://nlpers.blogspot.com/2016/05/a-bad-optimizer-is-not-good-thing.html
- *Optional follow-up reading:*
- Early stopping in kernel methods https://papers.nips.cc/paper/7187-early-stopping-for-kernel-boosting-algorithms-a-general-analysis-with-localized-complexities.pdf
- Escaping saddles in constrained optimization: https://arxiv.org/pdf/1809.02162.pdf
- Escaping higher order saddles: http://www.offconvex.org/2016/03/22/saddlepoints/
- K Fukumizu, S Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. 2000.