Non-Convex Optimization in Machine Learning

Jan Mrkos

AIC

The Plan

1. Introduction
2. Non convexity
3. (Some) optimization approaches
4. Speed and stuff?

Neural net – universal approximation

Theorem (1989):

- Let $\sigma(.)$ be bounded, monotonically increasing and continuous,
- Let $\mathrm{C}(I_m)$ be space of continuous functions over unit interval in R^m.

Then for any $f \in C(I_m)$ and $\epsilon > 0$ **exists**:

$$F(x) = \sum_{i=1,\dots N} v_i \, \sigma(w_i^T x + b_i)$$

Such that $|F(x) - f(x)| < \epsilon$

Neural net – universal approximation

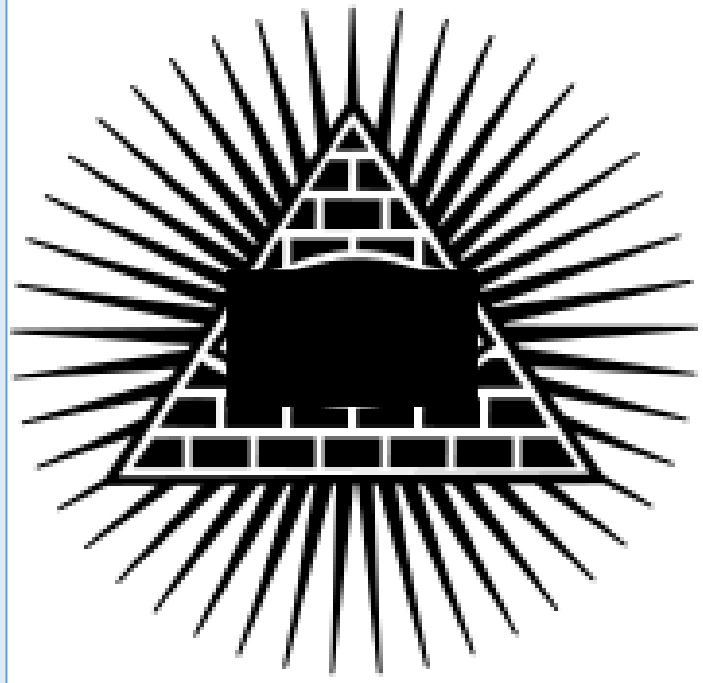
Theorem (1989):

- Let $\sigma(.)$ be bounded, monotonically increasing and continuous,
- Let $\mathrm{C}(I_m)$ be space of continuous functions over unit interval in R^m.

Then for any $f \in C(I_m)$ and $\epsilon > 0$ **exists**:

$$F(x) = \sum_{i=1,\dots N} v_i \, \sigma(w_i^T x + b_i)$$

Such that $|F(x) - f(x)| < \epsilon$



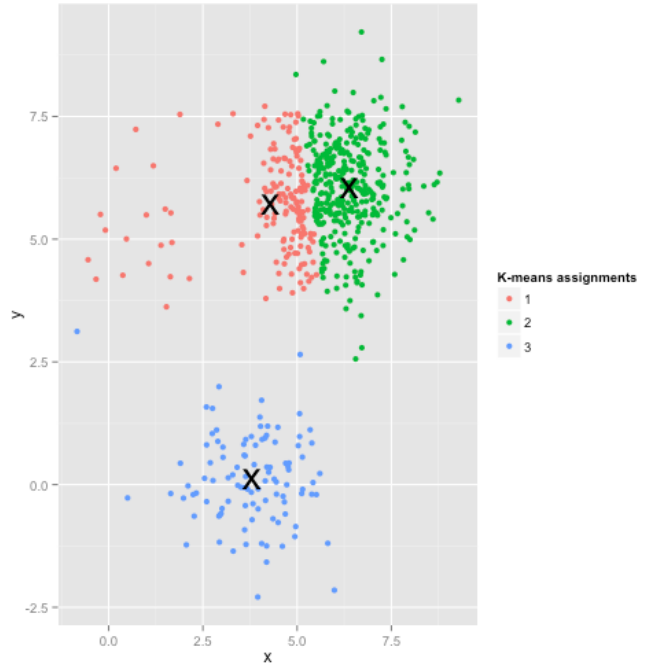
But training Neural Networks turns out to be NP hard


WINTER IS COMING
quickmeme.com

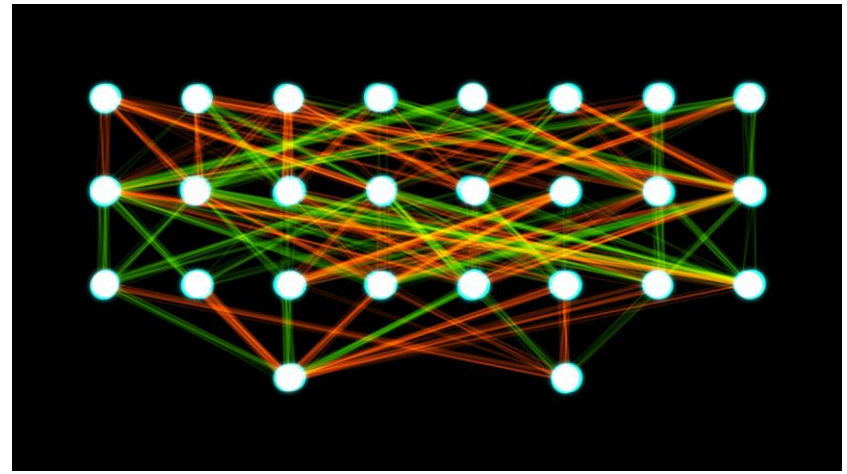
* Reza Zadeh, https://www.oreilly.com/ideas/the-hard-thing-about-deep-learning

ML and Optimization

- Unsupervised learning
  - Clustering - minimizing within cluster sum of squares



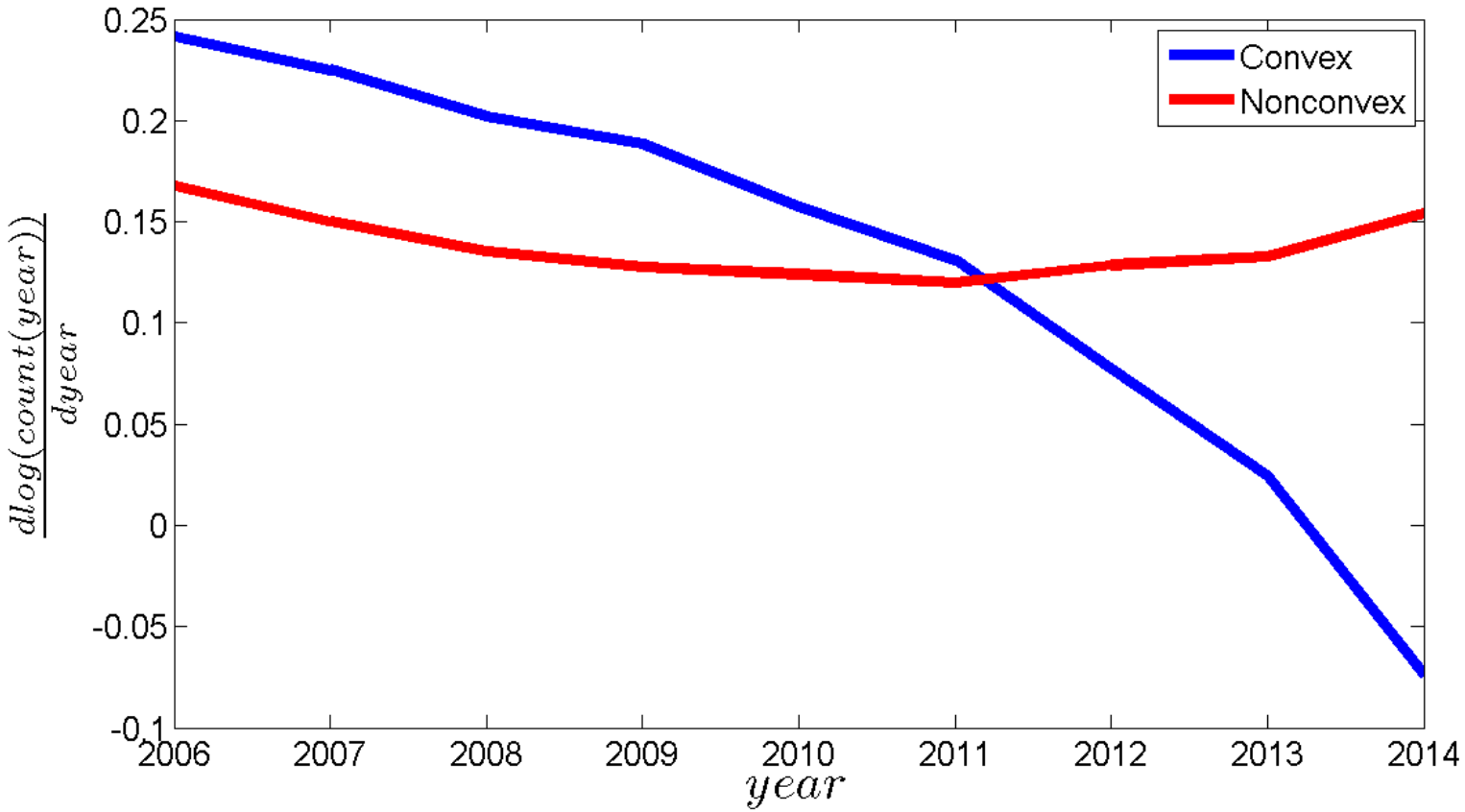
- Supervised learning
  - Neural nets - minimizing loss / generalization error



The Plan

1. Introduction
2. **Non convexity**
3. (Some) optimization approaches
4. Speed and stuff?

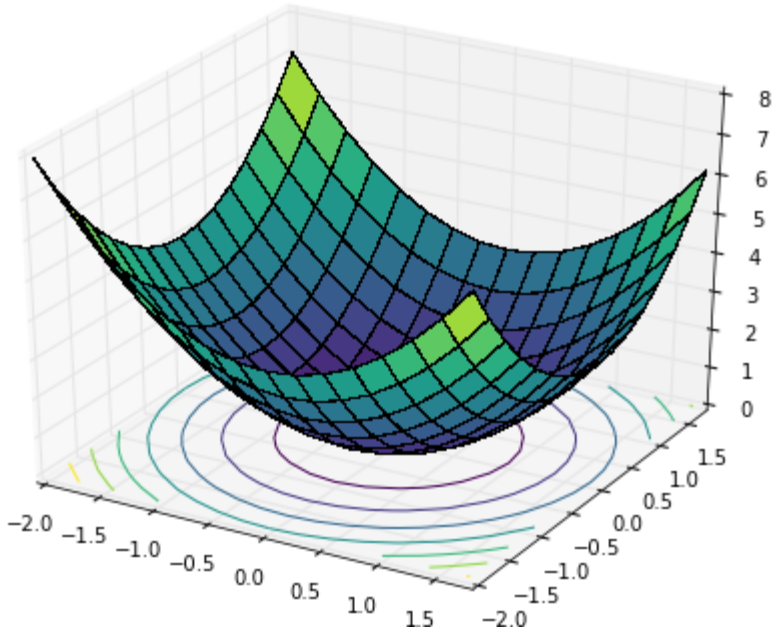
Part 2: Convex vs. Non-convex



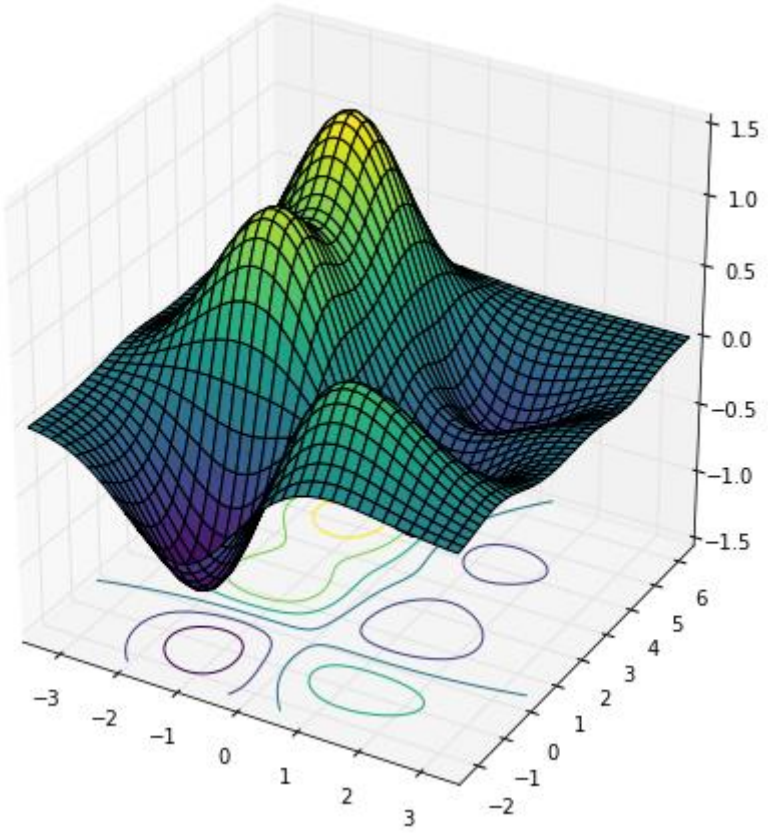
Source: https://www.facebook.com/nonconvex

Convex vs. Non-convex

- Convex ⊂ Non-convex



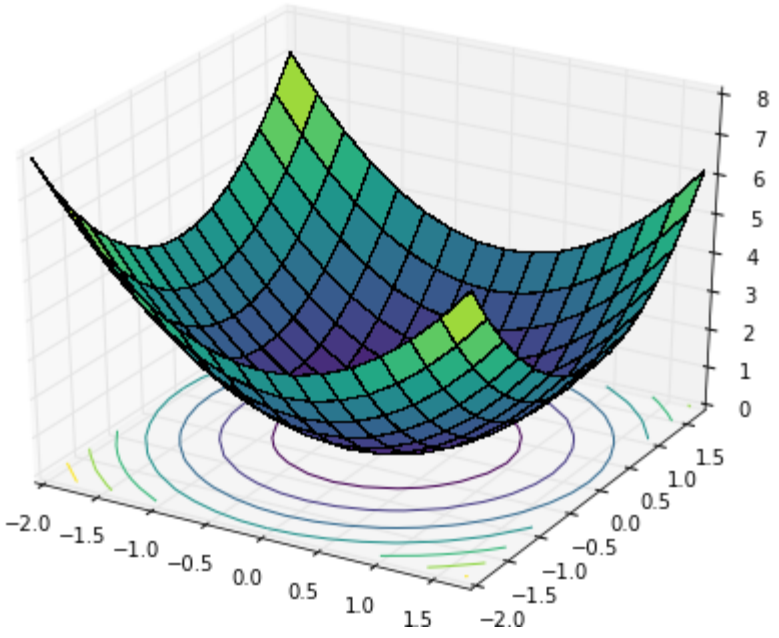
- One global minimum

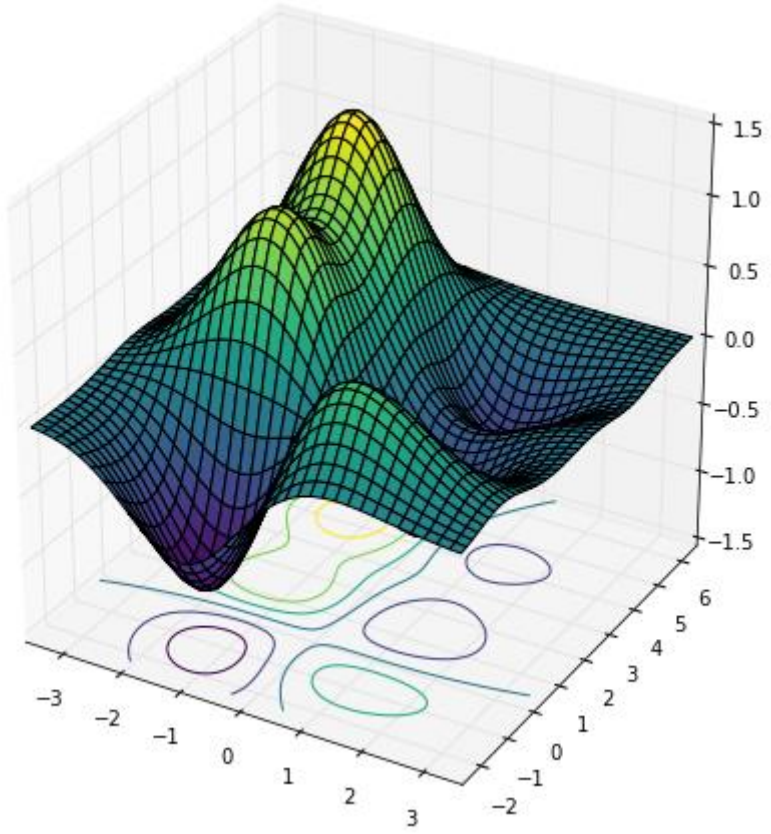


- Multiple local optima

Convex vs. Non-convex

- Convex ⊂ Non-convex

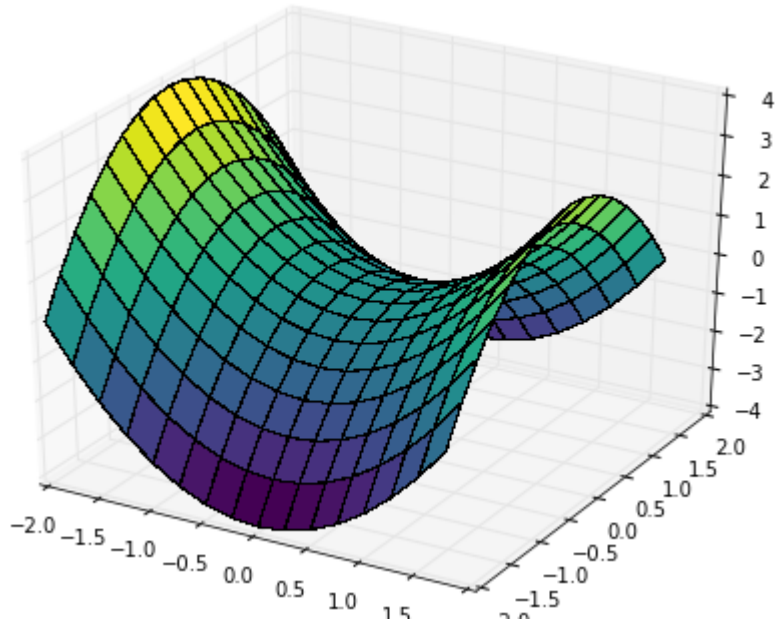


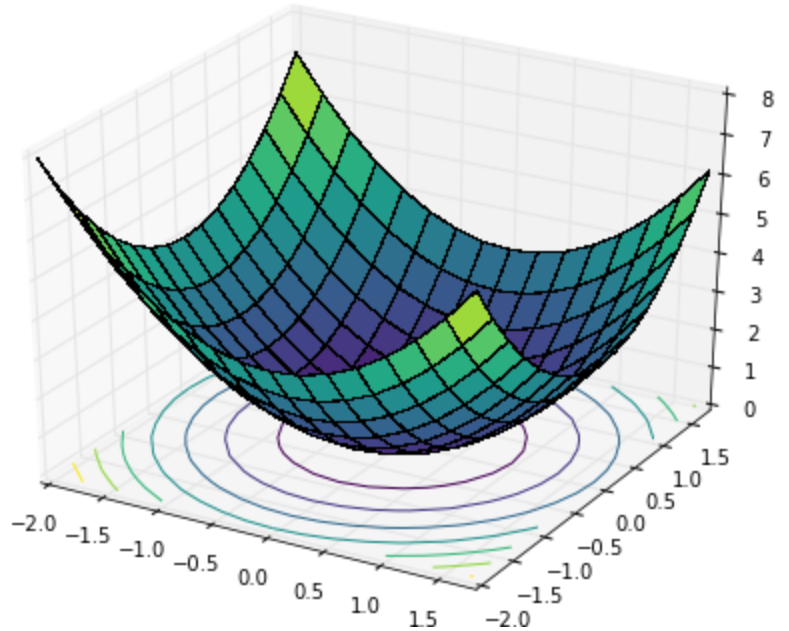


- How do you find the global optima?

Non – convex Optimization

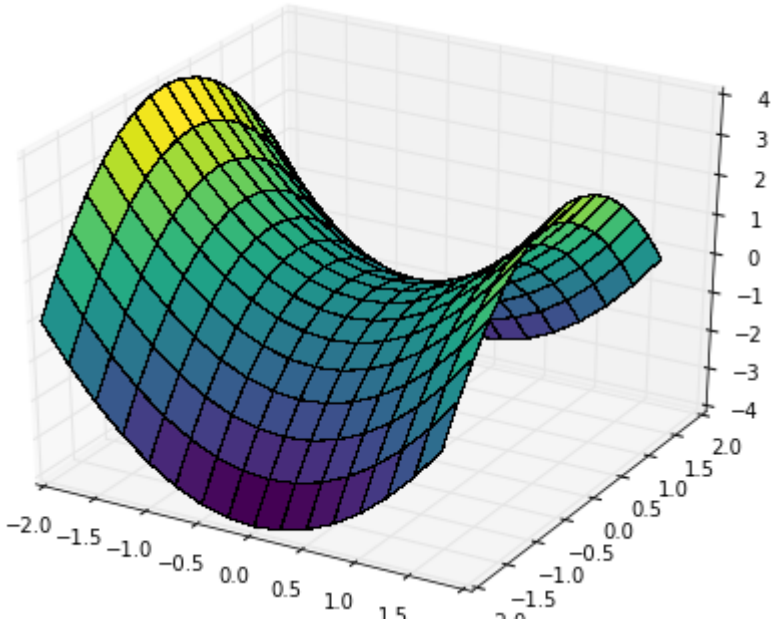
- Critical point $\{x : \nabla_x f(x) = 0\}$





Saddle points

- How many saddle points are there?



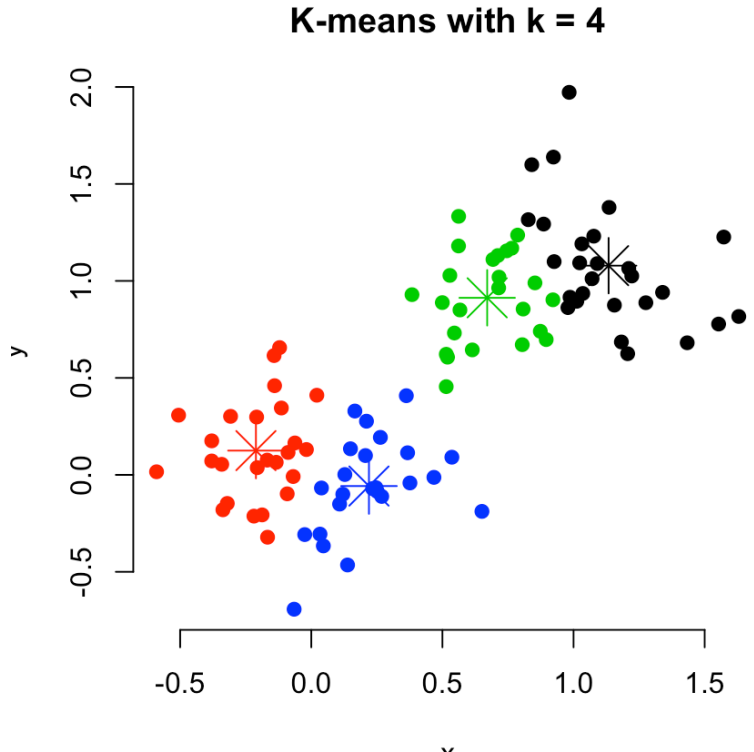
- Curse of dimensionality – for general problems, number of critical points increases exponentially with the dimension*

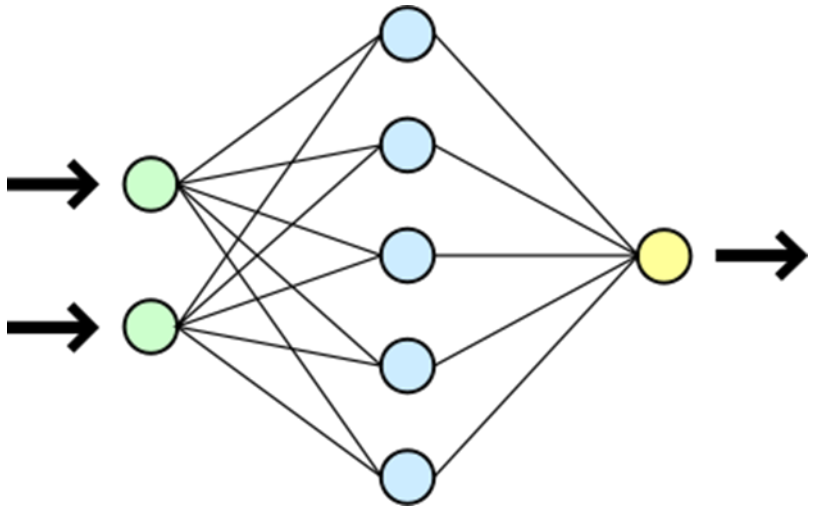
*Dauphin et al. - Identifying and attacking the saddle point problem in high-dimensional non-convex optimization

Saddle points in ML problems

- Loss f invariant to permutations:

$$f(x_1, x_2, \dots x_k) = f(x_2, x_1, \dots x_k), x_i \in R^n$$

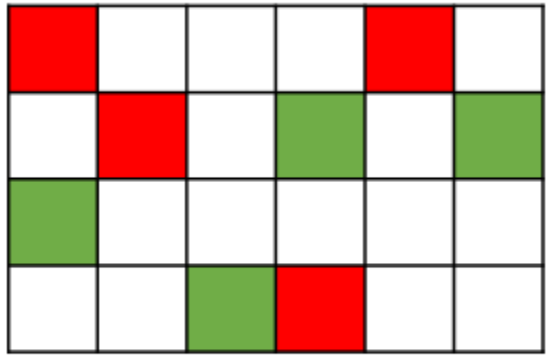




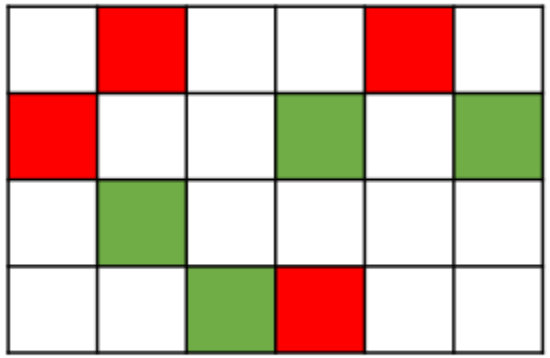
k - hidden neurons with n=2 weights

Saddle points in ML problems

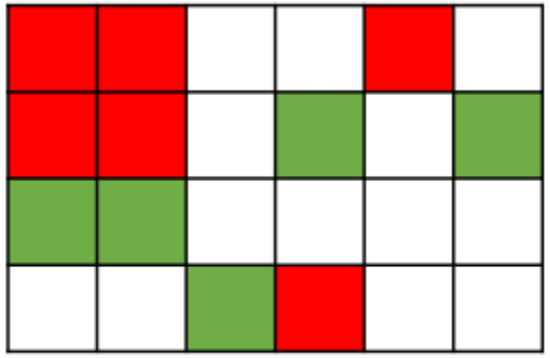
- For convex function, average of optimal solutions -> optimal
- Non convex function:



Optimal Solution



Equivalent Solution



Not optimal

- Average of optimal solutions: 0 gradient, but not optimal -> saddle point.

Image from http://www.offconvex.org/2016/03/22/saddlepoints/

Saddle points in ML problems

- Some invariances in NN:
  - Neuron permutations
  - Layer scaling
- -> Number of critical points grows exponentially with number of neurons! – Universal approximation, but No free lunch

Saddle points in ML problems

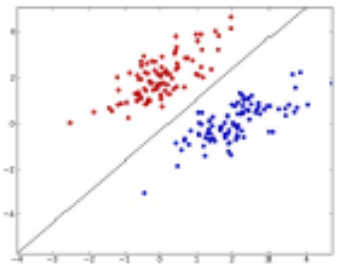
- Some invariances in NN:
  - Neuron permutations
  - Layer scaling
- -> Number of critical points grows exponentially with number of neurons! – Universal approximation, but No free lunch
- Overspecified models (such as for dropout)*:
  - Critical points in smaller networks create critical points in bigger nets
  - Even global minimum can generate saddle points

*Fukumizu, Amari: Local minima and plateaus in hierarchical structures of multilayer perceptrons

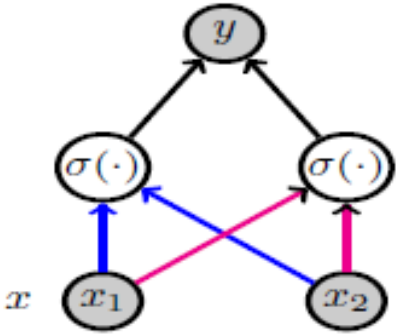
Saddle points in ML problems

- Overspecified models:
  - Critical points in smaller networks create critical points in bigger nets
  - Even global minimum can generate saddle points
- EXAMPLE*:

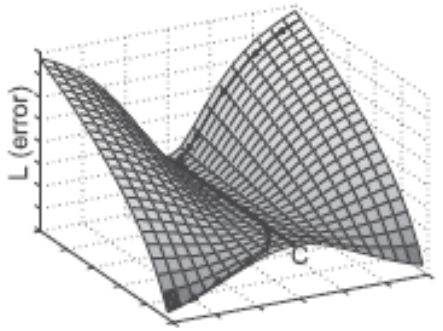
Training Data



Neural Network



Learning Trajectory



* Anandkumar: Nonconvex optimization: Challenges and Recent Successes - ICML2016 Tutorial

The Plan

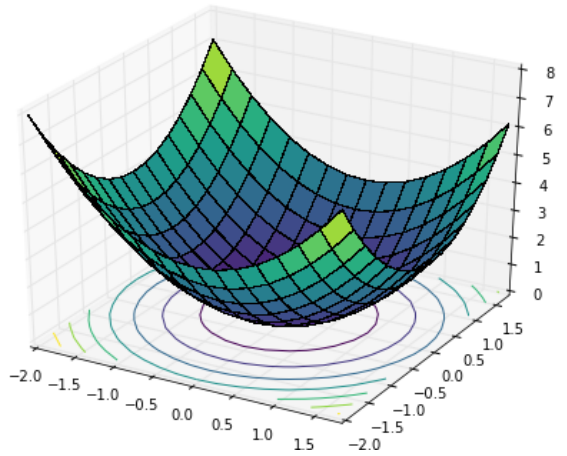
1. Introduction
2. Non convexity
3. **(Some) optimization approaches**
4. Speed and stuff?

Part 3: Learning with saddle points

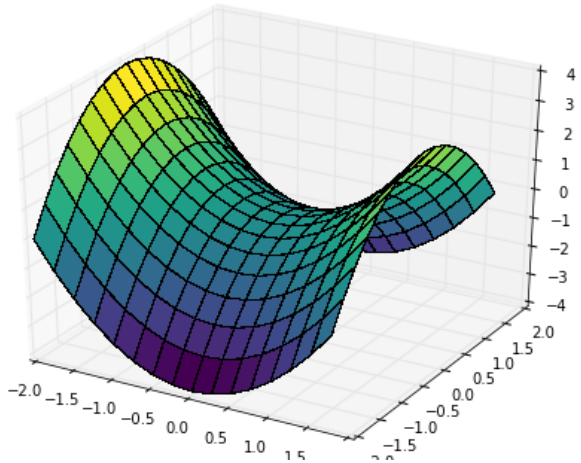
- Saddle points lead to slow learning (long time spent in saddle manifold)
- How to deal with saddles?

Saddle point or local optima?

- Function to optimize $f(x)$, with critical points $\nabla_x f(x) = 0$



Minimum

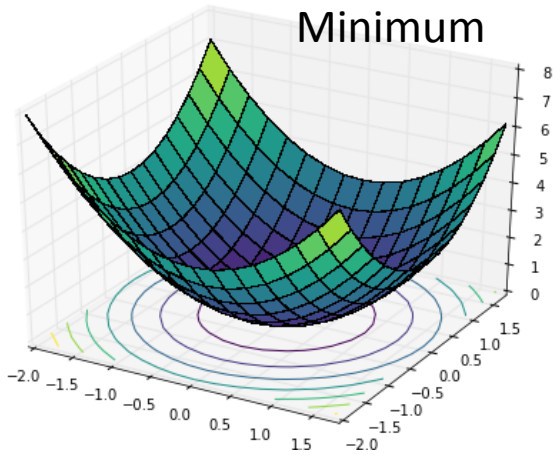


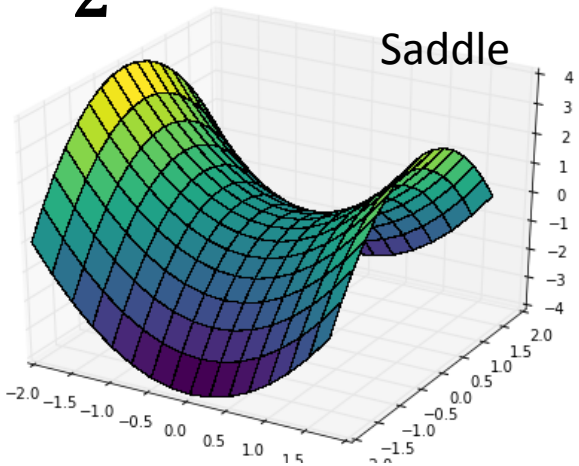
Saddle

$$\boldsymbol{f(y) \approx f(x) + \langle \nabla_x f(x), y - x \rangle + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)}$$

Saddle point or local optima?

$$\boldsymbol{f(y) \approx f(x) + \langle \nabla_x f(x), y - x \rangle + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)}$$

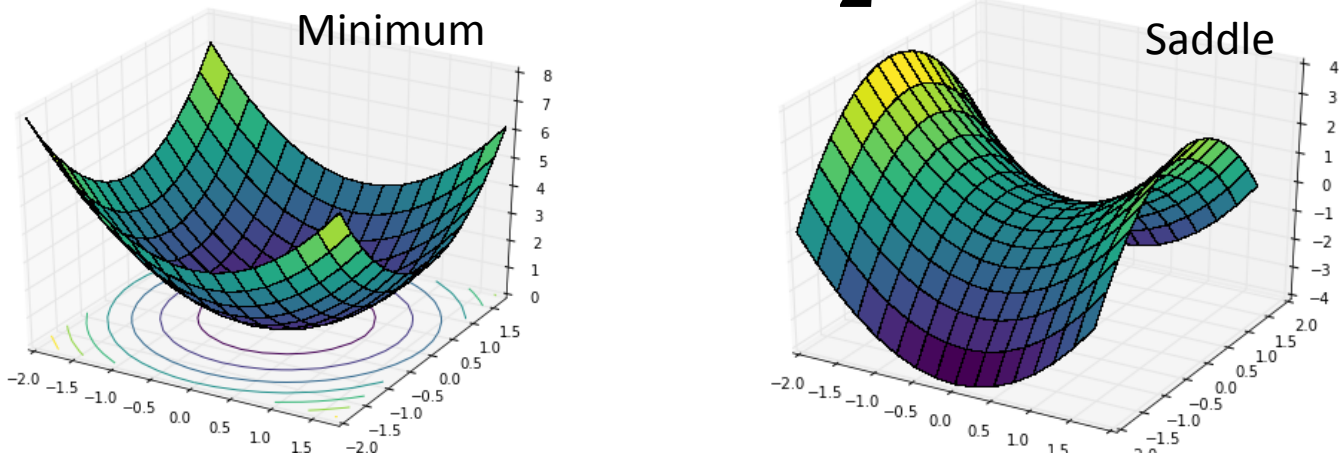




- **Local minima:** critical point with $\nabla^2 f(x) > 0$
- **Non degenerate saddle point:** critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues
- Indeterminate critical points points: zeros as eigenvalues.
- Anyway, why are critical points also called stationary?

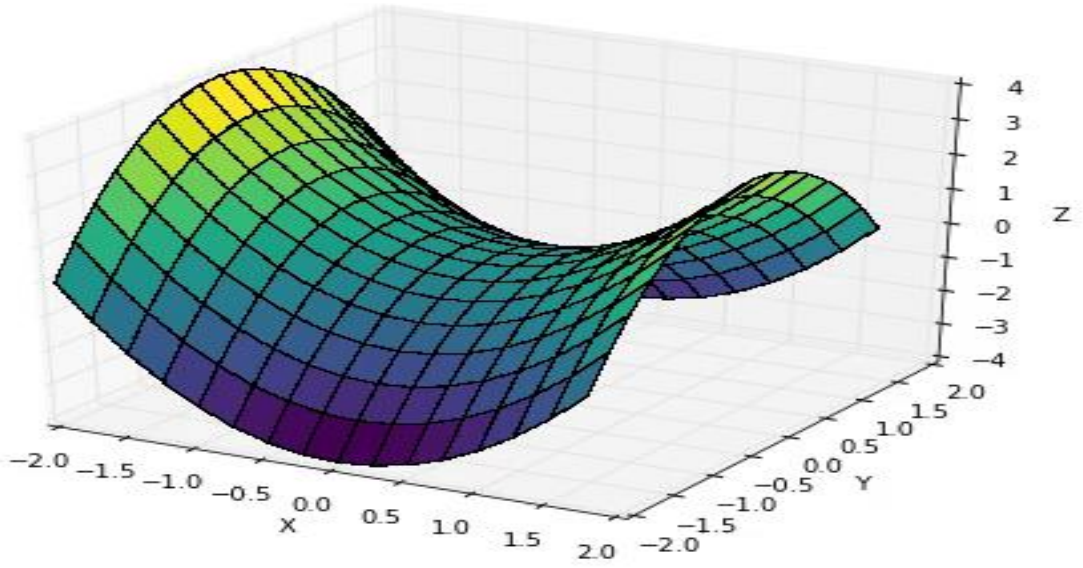
Saddle point or local optima?

$$f(y) \approx f(x) + \langle \nabla_x f(x), y - x \rangle + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$



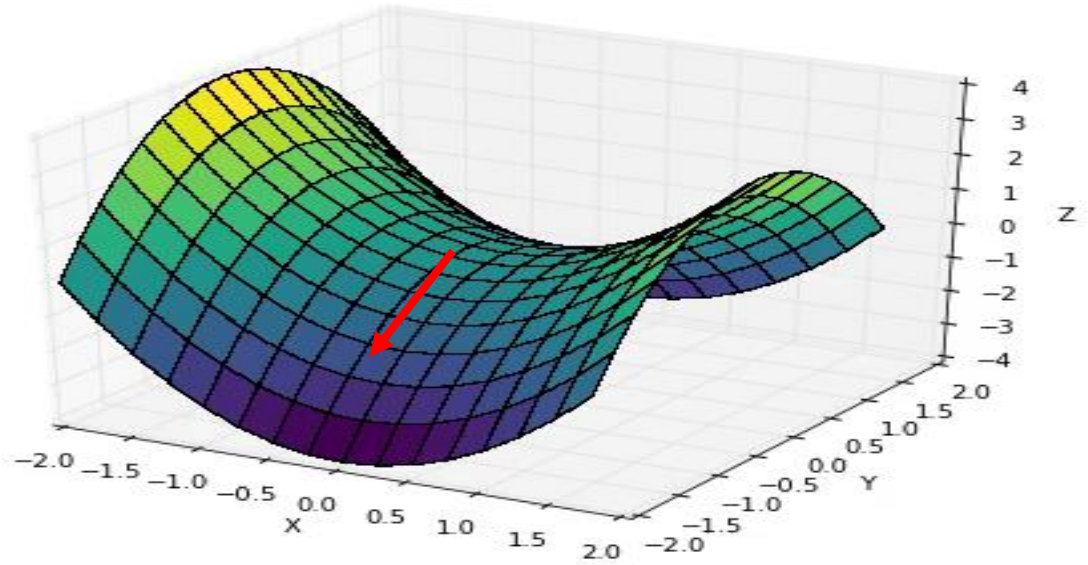
- **Local minima:** critical point with $\nabla^2 f(x) > 0$
- **Non degenerate saddle point:** critical point where $\nabla^2 f(x)$ has strictly positive and negative eigenvalues
- Indeterminate critical points points: zeros as eigenvalues.
- Anyway, why are critical points also called stationary?

Gradient descent



$$x_n = x_{n-1} - \mu \nabla_x f(x)$$

Gradient descent



- Gets stuck at the saddle. But for non degenerate saddle, there is u such that $u^T \nabla^2 f(x) u < 0$
- Negative eigen-value -> direction of escape
- We need a second order method.

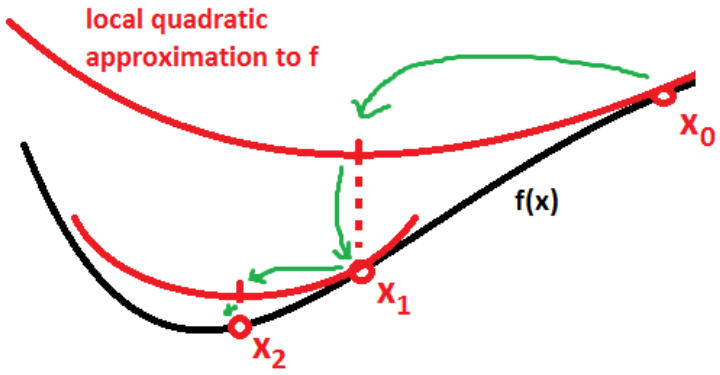
Newton method

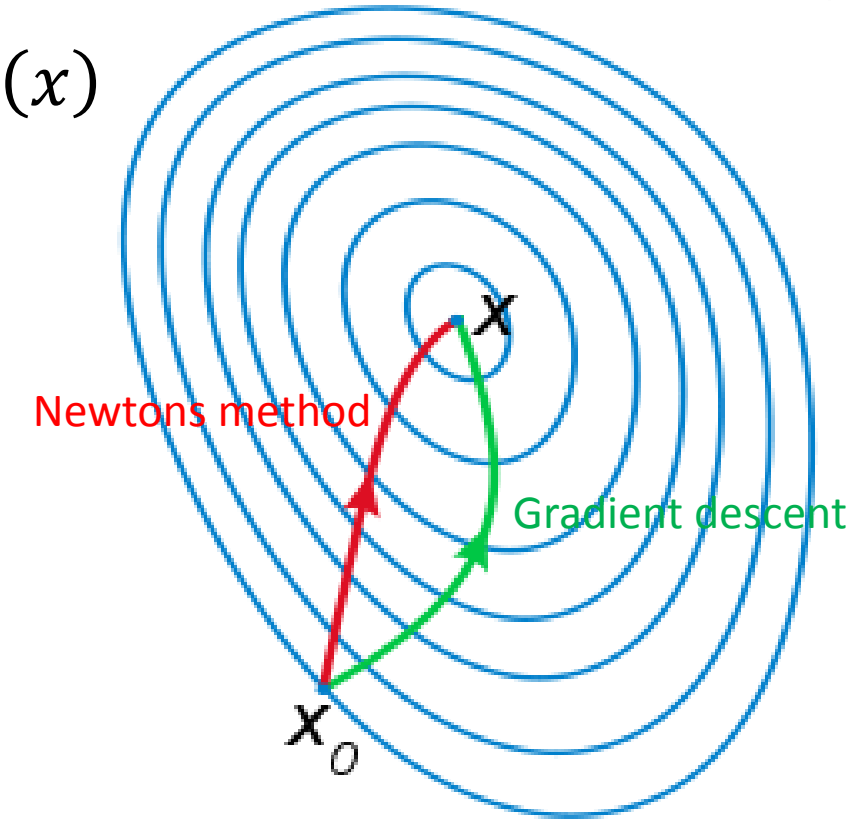
Well known second order method:

$$x_n = x_{n-1} - H(x)^{-1} \nabla_x f(x)$$

Where $H(x) = \nabla^2 f(x)$

- Faster convergence than gradient descent
- Great for convex problems





Images:
left: Anandkumar: Nonconvex optimization: Challenges and Recent Successes - ICML2016 Tutorial
right: wikipedia

Newton method - example

Local quadratic approximation

- Assume x^* non-degenerate saddle (negative eigenvalues in Hessian) and let Δv_i change along eigenvector v_i:

$$f(x^* + \Delta x) \approx f(x^*) + \frac{1}{2}\sum_i \lambda_i \Delta v_i^2$$

Gradient Descent

- Very slow in the neighborhood of x^* due to small gradient
- Moves "down" due to negative values of λ_i

Newtons Method

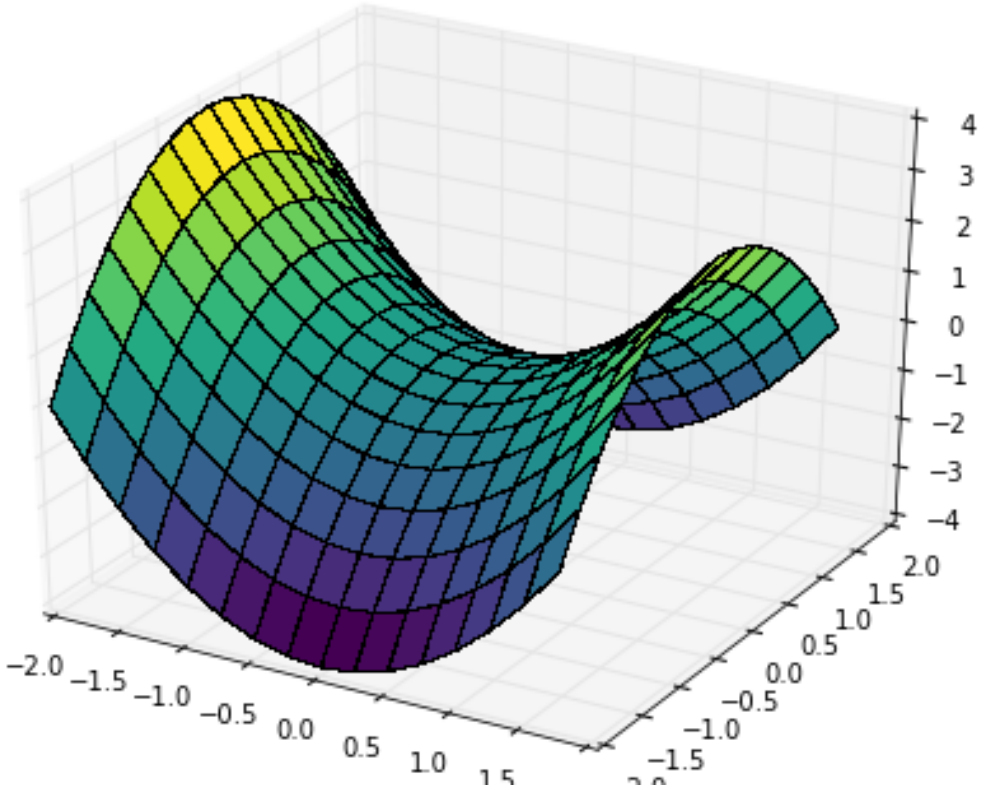
- Multiplies each eigendirection with λ_i^{-1}. This gives better convergence, but also reverses the sign of "down" going directions.

i.e. converges on the saddle point

Escaping saddle point

Strategy

- Use gradient descent when gradient large
- When gradient becomes small, move along eigenvector $\lambda_i < 0$ to escape



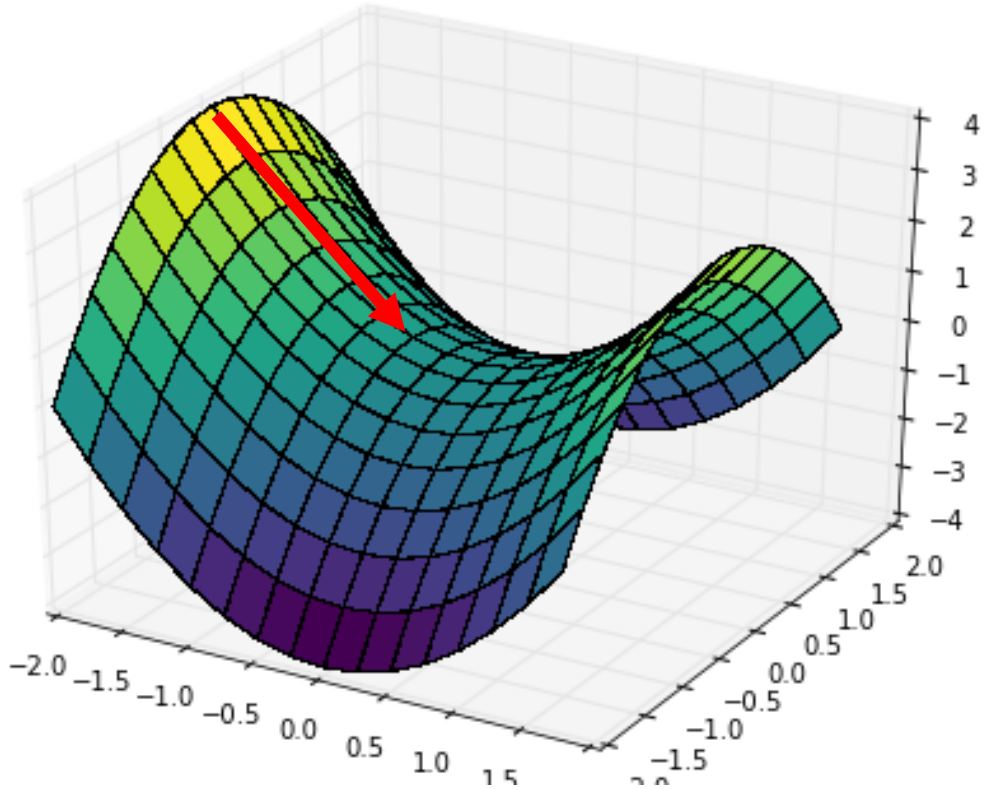
Sophisticated alternative:

Cubic regularization of Newton method

Escaping saddle point

Strategy

- Use gradient descent when gradient large
- When gradient becomes small, move along eigenvector $\lambda_i < 0$ to escape



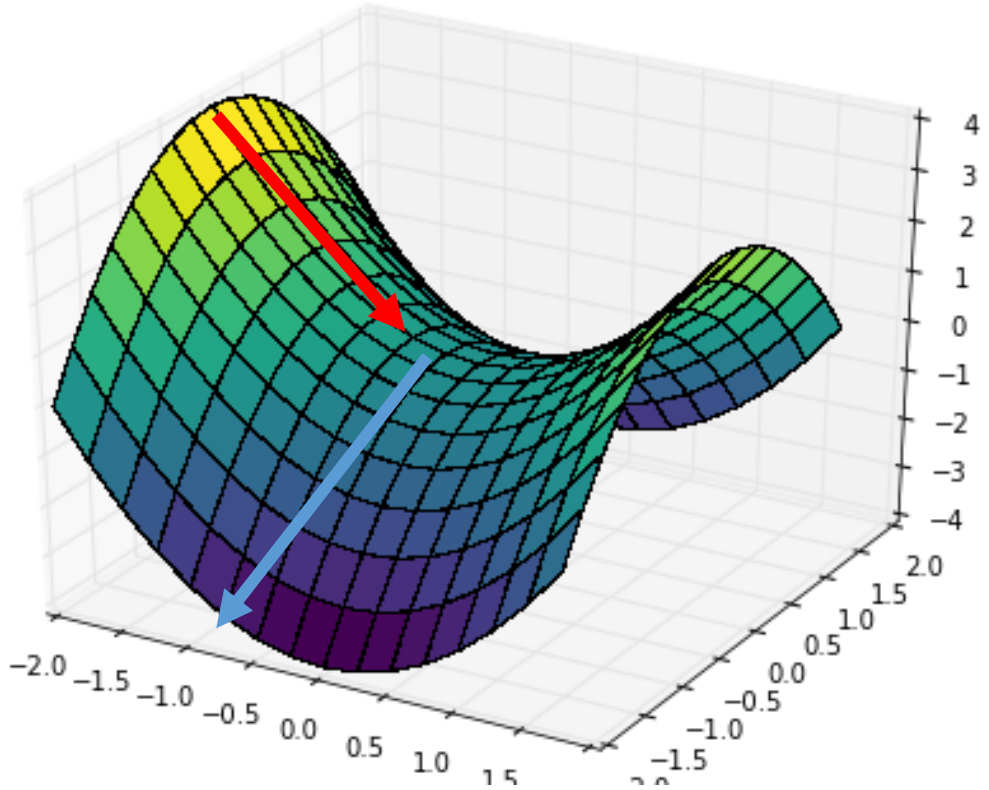
Sophisticated alternative:

Cubic regularization of Newton method

Escaping saddle point

Strategy

- Use gradient descent when gradient large
- When gradient becomes small, move along eigenvector $\lambda_i < 0$ to escape



Sophisticated alternative:

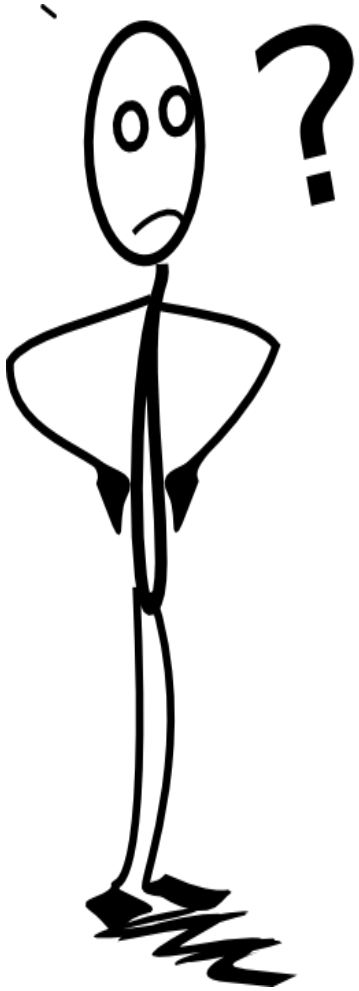
Cubic regularization of Newton method

The Plan

1. Introduction
2. Non convexity
3. (Some) optimization approaches
4. **Speed and stuff?**

Part 4: How fast can we converge to local min?

- Generally, hard to say



Class of strict saddle functions

A function $f(x)$ is strict saddle if all points x satisfy at least one of the following

1. Gradient $\nabla f(x)$ is large.
2. Hessian $\nabla^2 f(x)$ has a negative eigenvalue that is bounded away from 0.
3. Point x is near a local minimum.

Class of strict saddle functions

A function $f(x)$ is strict saddle if all points x satisfy at least one of the following

1. Gradient $\nabla f(x)$ is large.
2. Hessian $\nabla^2 f(x)$ has a negative eigenvalue that is bounded away from 0.
3. Point x is near a local minimum.

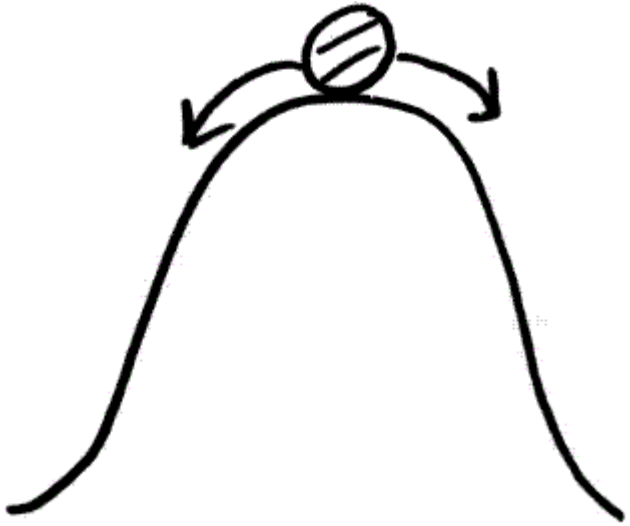
- For strict saddle function, cubic regularization finds local minimum in polynomial time.

Simple is better

- Second order algorithms are powerful, but computing second derivative is expensive

Observation:

- Non degenerate saddle points are unstable



Noisy gradient descent will not get stuck!

$$x_n = x_{n-1} - \mu \nabla_x f(x) + \epsilon$$

No wonder stochastic gradient for NN works!

Noisy gradient descent - speed

Result 1 (Ge 2015):
Noisy gradient descent can find a local minimum of strict saddle functions in polynomial time.

Noisy gradient descent - speed

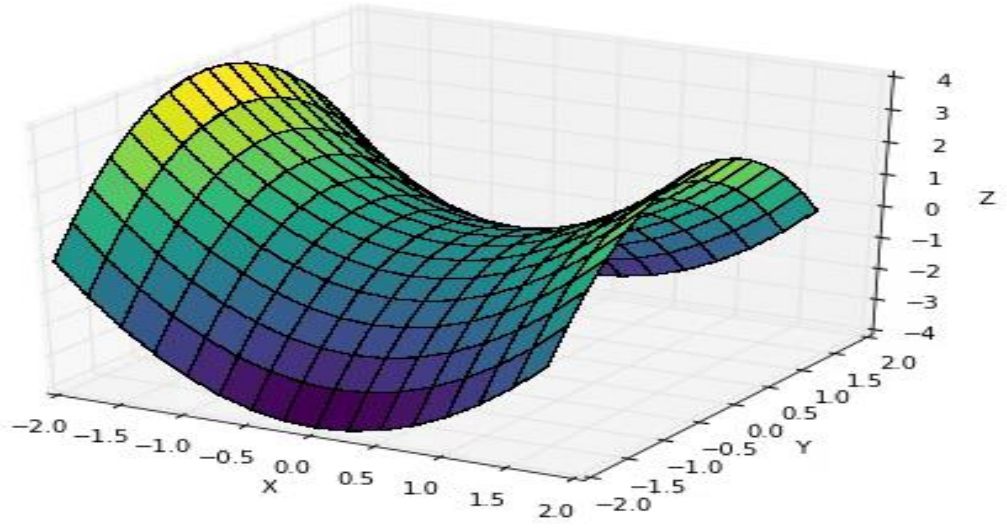
Result 1 (Ge 2015):
Noisy gradient descent can find a local minimum of strict saddle functions in polynomial time.

However, polynomial dependency on dimension and smallest eigenvalue of hessian is high and not practical.

Gradient descent still slow around saddle points.

Gradient descent not that bad?

Result 2 (Lee et al. 2016): For strict saddle functions, set of points from which gradient descent will converge to saddle is measure 0.



Gradient descent not that bad?

Result 2 (Lee et al. 2016): For strict saddle functions, set of points from which gradient descent will converge to saddle is measure 0.

Other possible saddle points

- So we know nothing.
- Also, other saddle points:

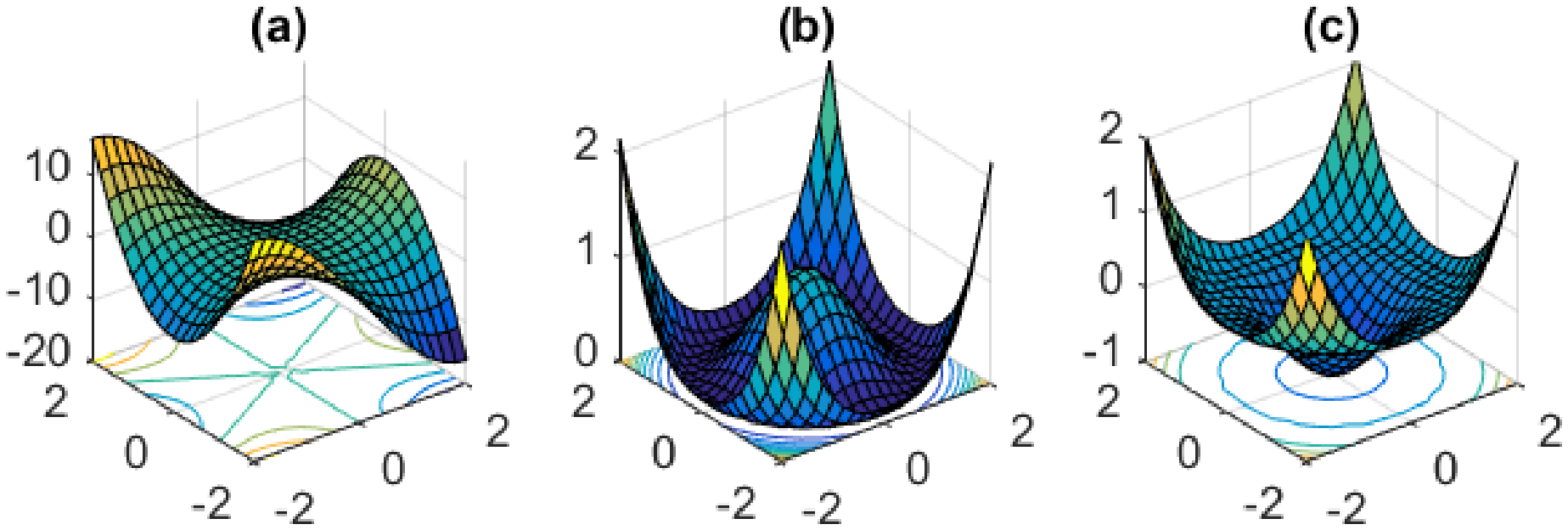


Image: Rong Ge: http://www.offconvex.org/2016/03/22/saddlepoints/

Summary

- In ML non convex problems, critical points are ubiquitous.
  - Symmetries and overspecification make stuff even worse
- Saddles make convergence slower
- Escaping saddles – noisy gradient descent and second order methods
- Time complexity estimates for some problems. General case for higher order (>3) saddle points still NP-hard.

Open questions

- How do we make optimization algorithms that work even when there are degenerate saddle points?
- Are there other guarantees for time complexity outside strict saddle functions?
- What are the saddle structures of popular problems, e.g. deep learning?
- Can noisy SGD escape higher order saddle points in bounded time?
- Noisy SGD has worse scaling with dimension vs. second order methods. Can this be improved?

Resources

- ICML turial on non-convex optimization: http://newport.eecs.uci.edu/anandkumar/ICML2016tutorial.html
- Result 1:
  - http://www.offconvex.org/2016/03/22/saddlepoints/
  - https://arxiv.org/abs/1503.02101
- Result 2:
  - http://www.offconvex.org/2016/03/24/saddles-again/
  - https://arxiv.org/abs/1602.04915
- https://www.oreilly.com/ideas/the-hard-thing-about-deep-learning
- Convexity and deep NN: https://rinuboney.github.io/2015/10/18/theoretical-motivations-deep-learning.html
- Nice paper: Netorov, Polyak, 2006, Cubic regularization of Newton method and its global performance
- Plots and animations in matplotlib