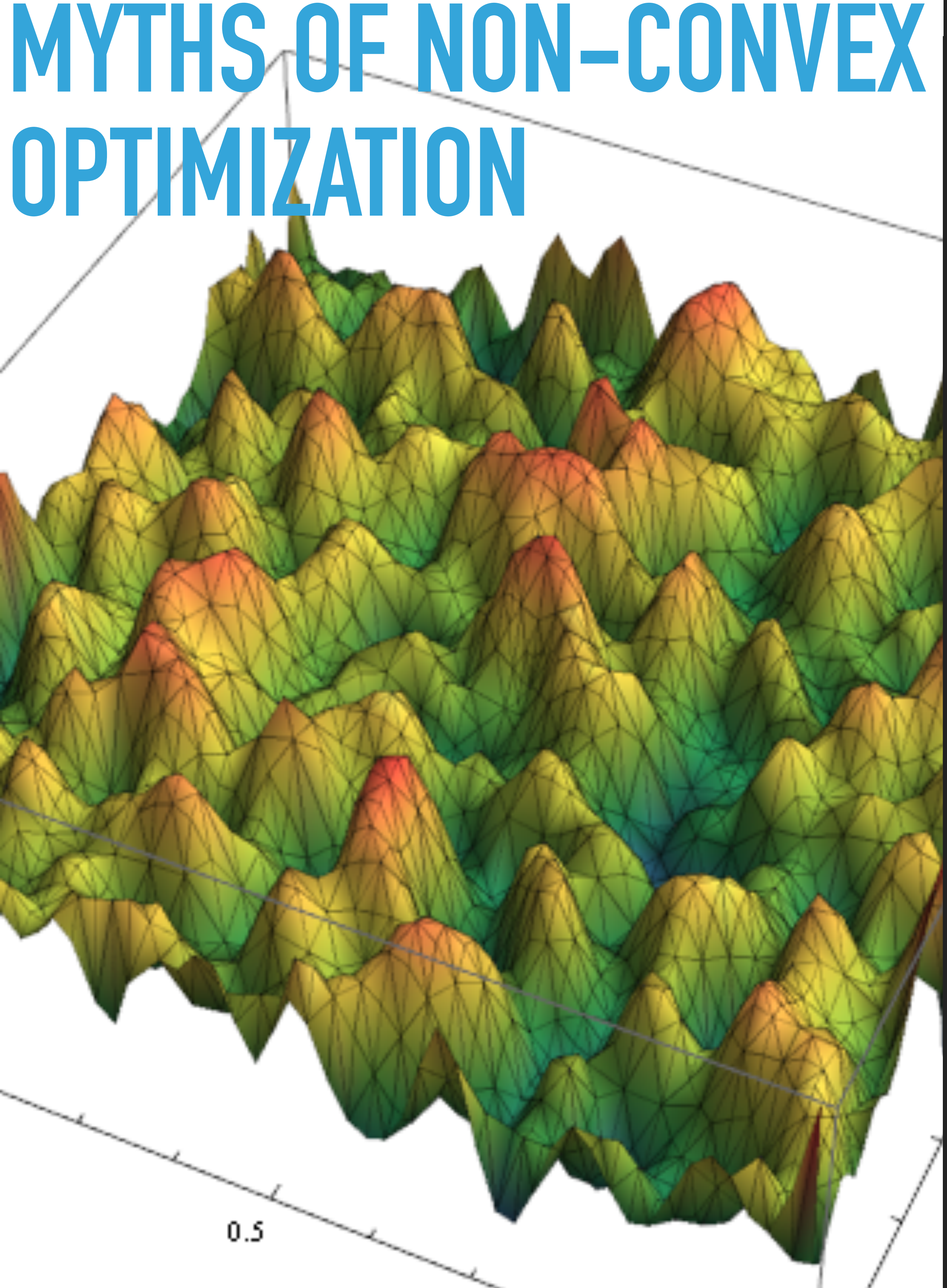
Yann N. Dauphin, Facebook AI Research

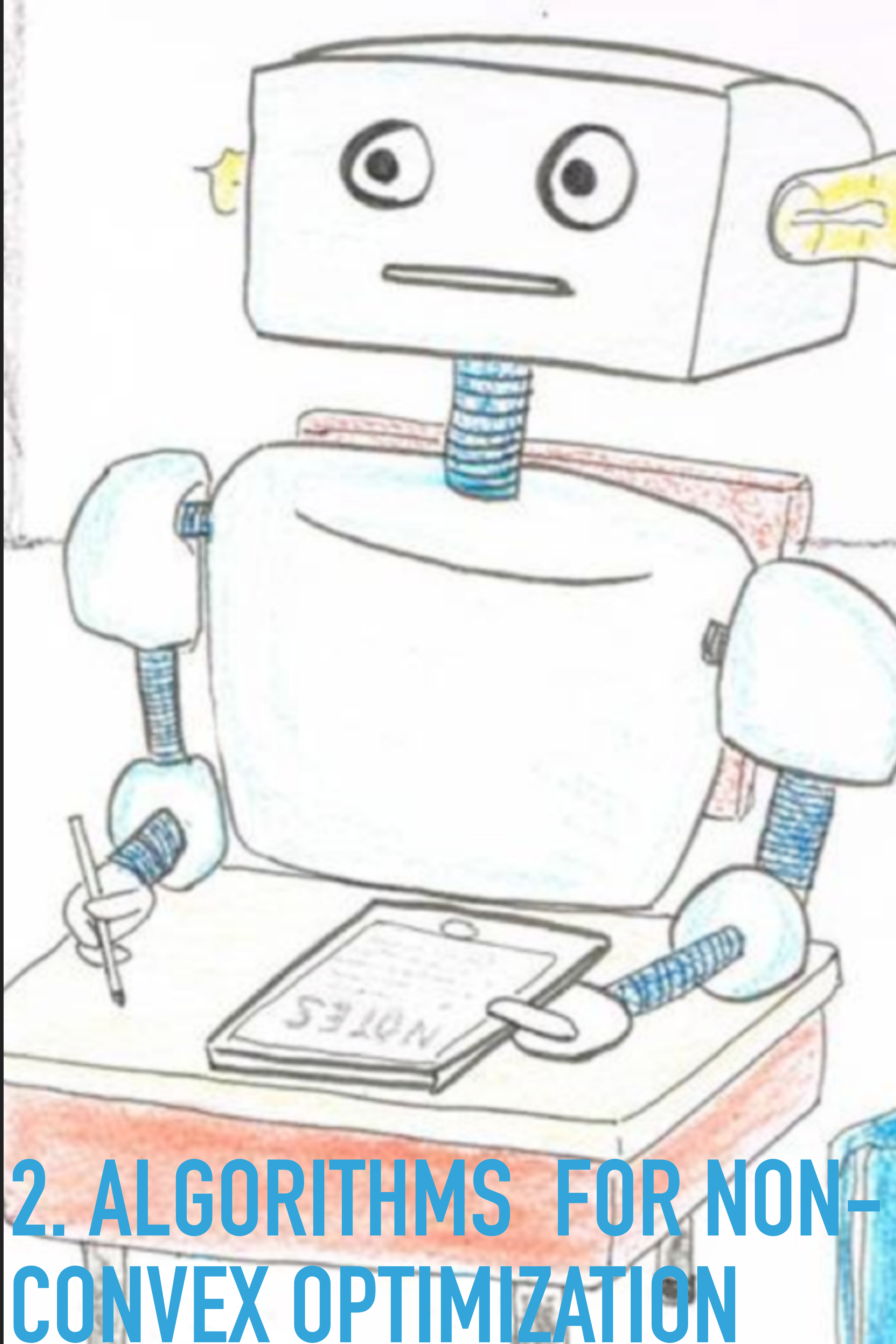
DISPELLING MYTHS AND GOING FORWARD

OPTIMIZING DEEP NETS

1. CHALLENGING THE MYTHS OF NON-CONVEX OPTIMIZATION



2. ALGORITHMS FOR NON-CONVEX OPTIMIZATION

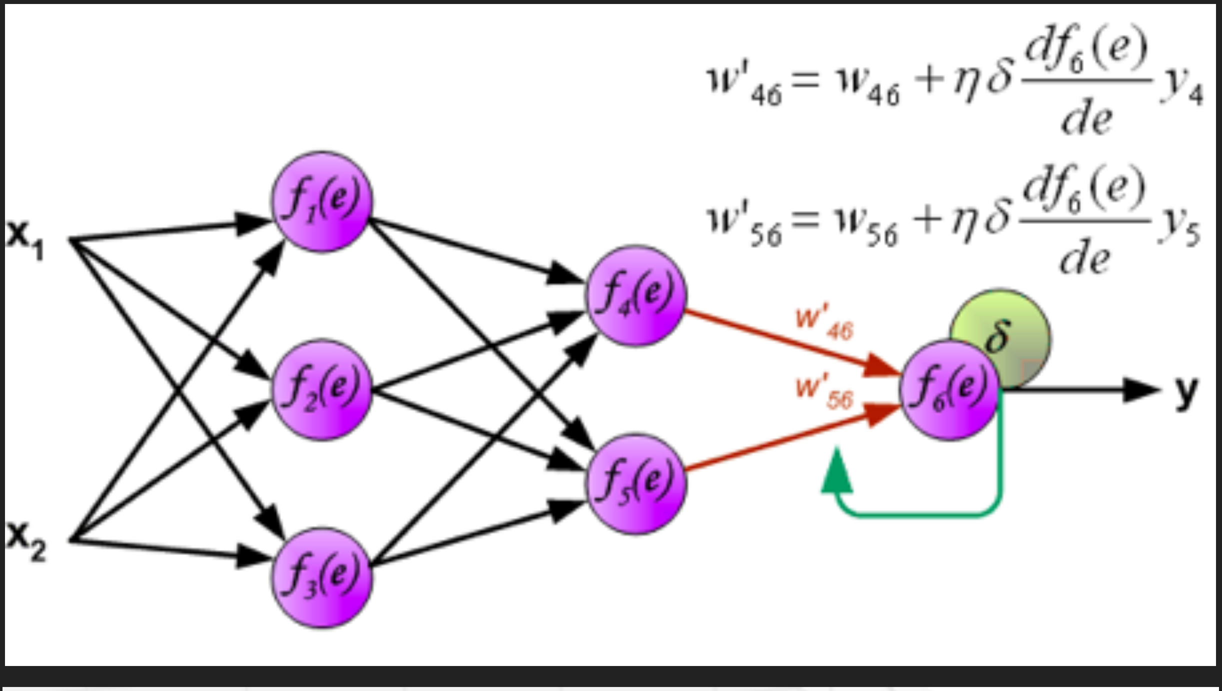


LEARNING DEEP NETS

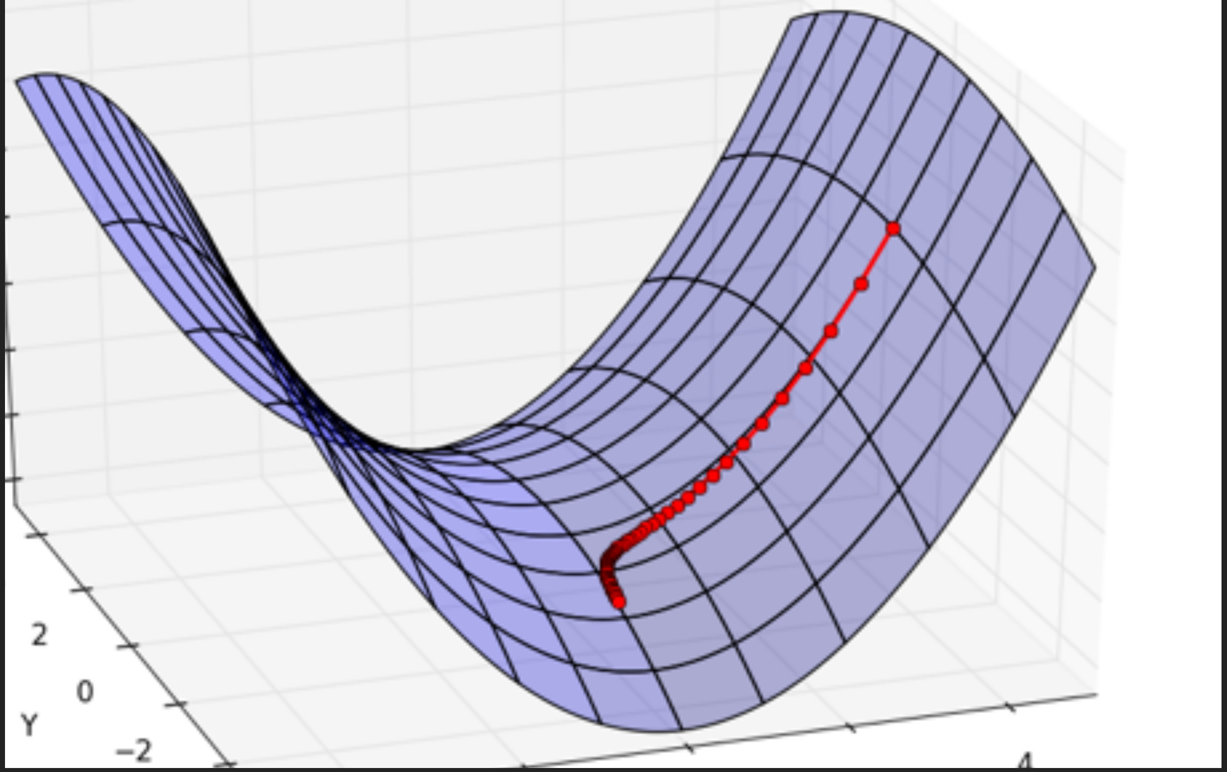
- The key idea of back-propagation was introduced by (Rumelhart et al, 1986).

- We consider the parameters as the coordinate of a point on a surface defined by the loss.

- Computing the gradient with the chain-rule tells us where to move in that space.

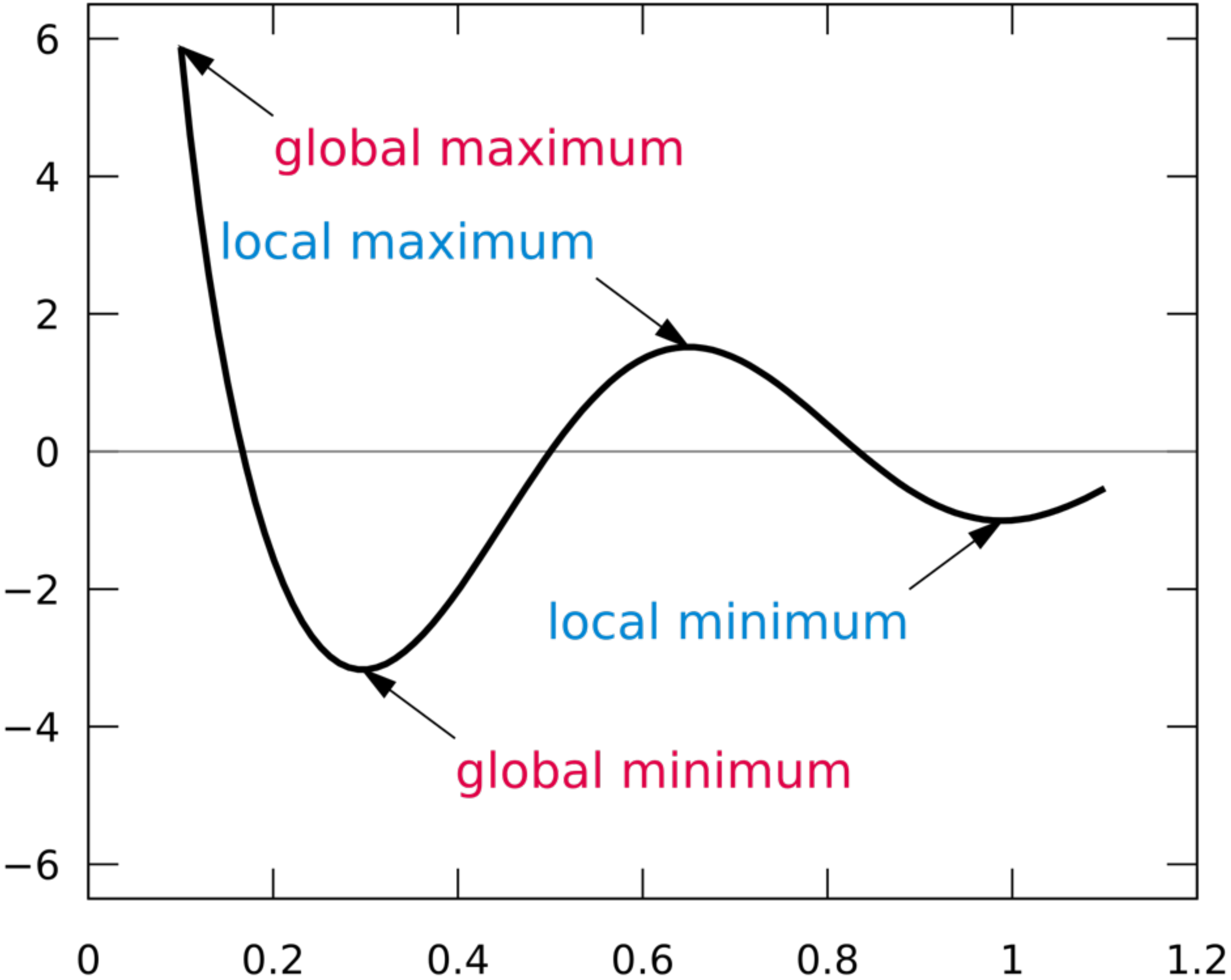


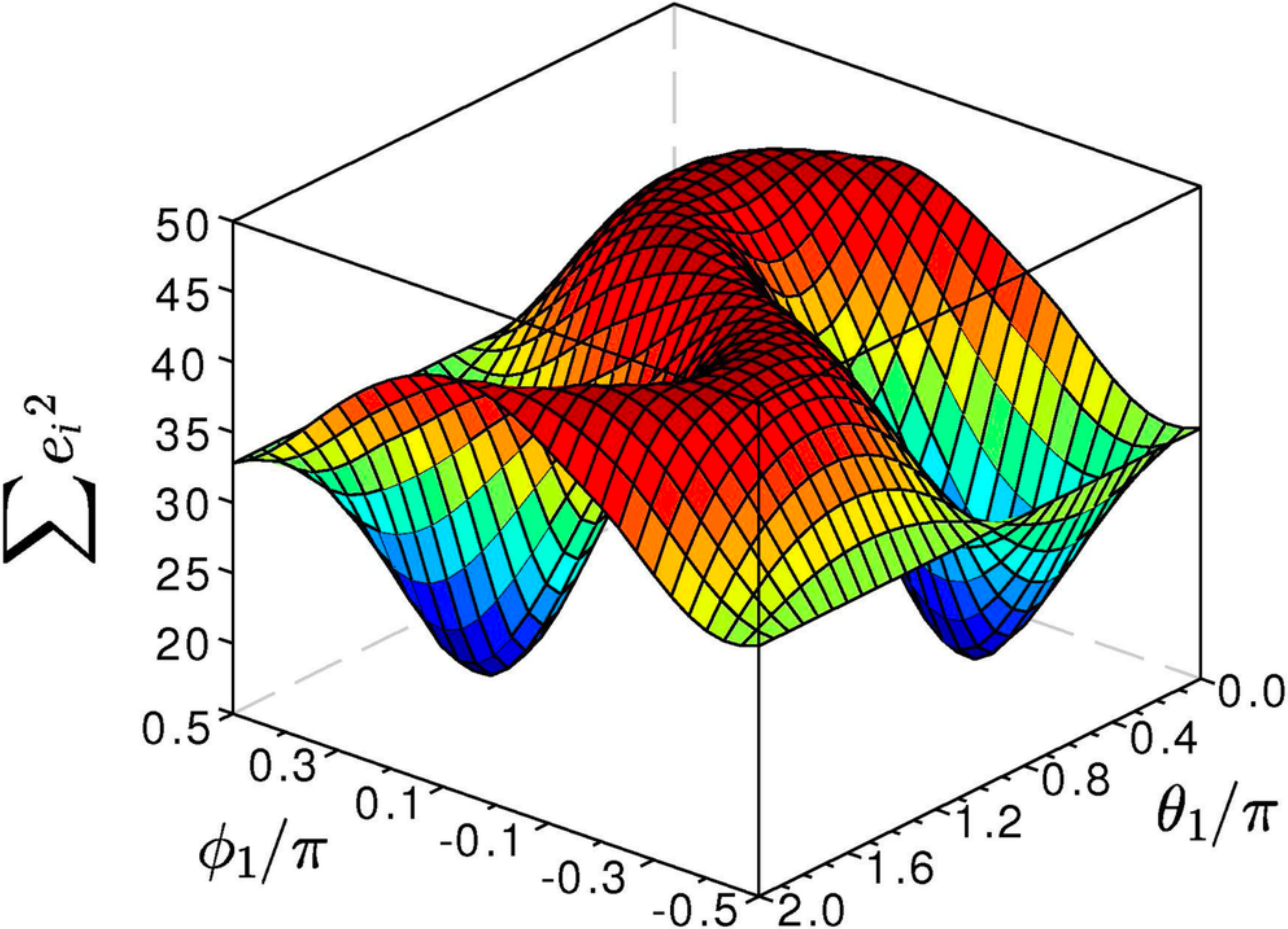
$w'_{46} = w_{46} + \eta\delta \frac{df_6(e)}{de} y_4$
$w'_{56} = w_{56} + \eta\delta \frac{df_6(e)}{de} y_5$
x_1
x_2
$f_1(e)$
$f_2(e)$
$f_3(e)$
$f_4(e)$
$f_5(e)$
$f_6(e)$
w'_{46}
w'_{56}
δ
y



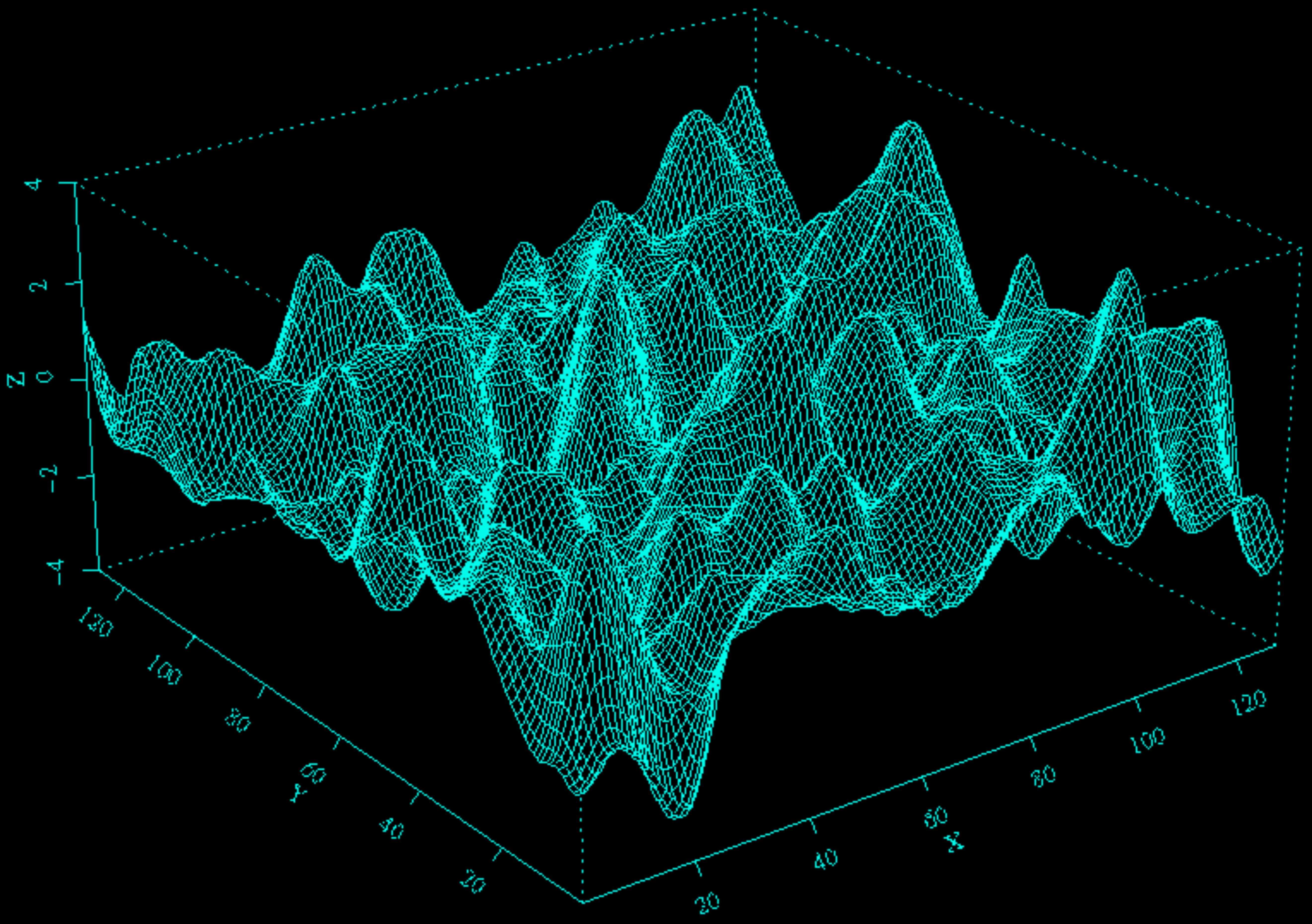
LOCAL MINIMA CAN GREATLY DOMINATE THE GLOBAL MINIMA IMPLYING A HIGH PROBABILITY FAILURE OF BACK-PROPAGATION

(Brady & Raghavan, 1989) on training 2D nets


6
4
2
0
−2
−4
−6
0
0.2
0.4
0.6
0.8
1
1.2
global maximum
local maximum
local minimum
global minimum

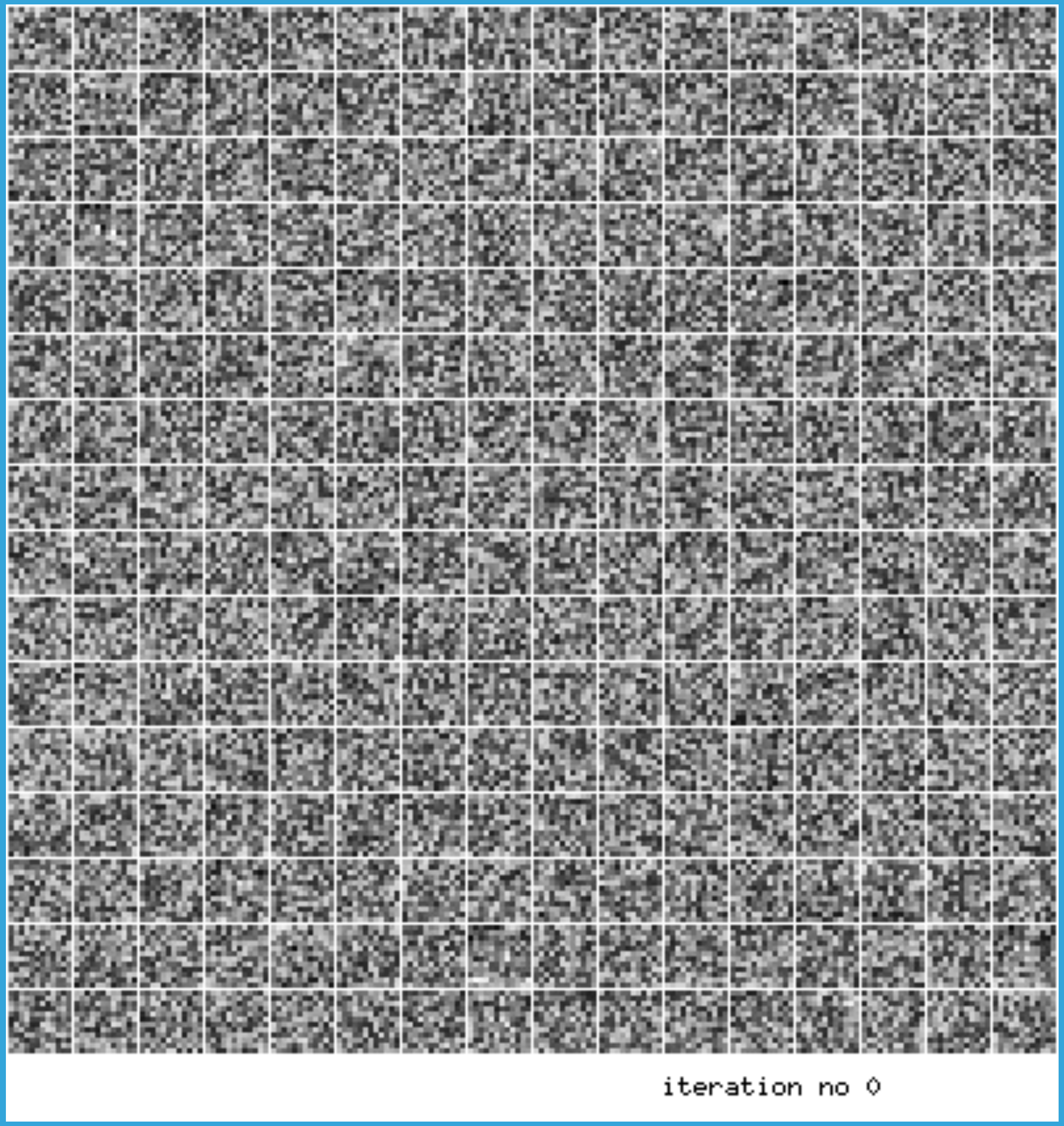

50
45
40
35
30
25
20
$\sum e_i^2$
0.5
0.3
0.1
-0.1
-0.3
-0.5
ϕ_1/π
0.0
0.4
0.8
1.2
1.6
2.0
θ_1/π

Matern(2,5)



TRAINING HUGE DEEP NETS MAY SEEM DOOMED

Training a in 2D seems impossible so how could we optimize 10^6 weights?



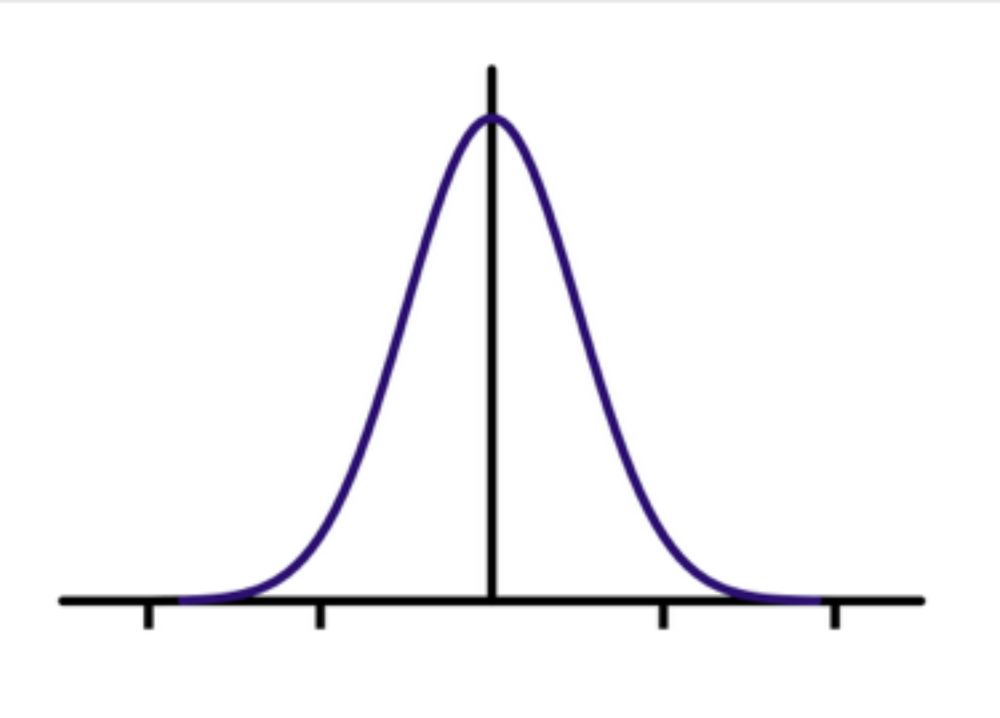
IT JUST WORKS!

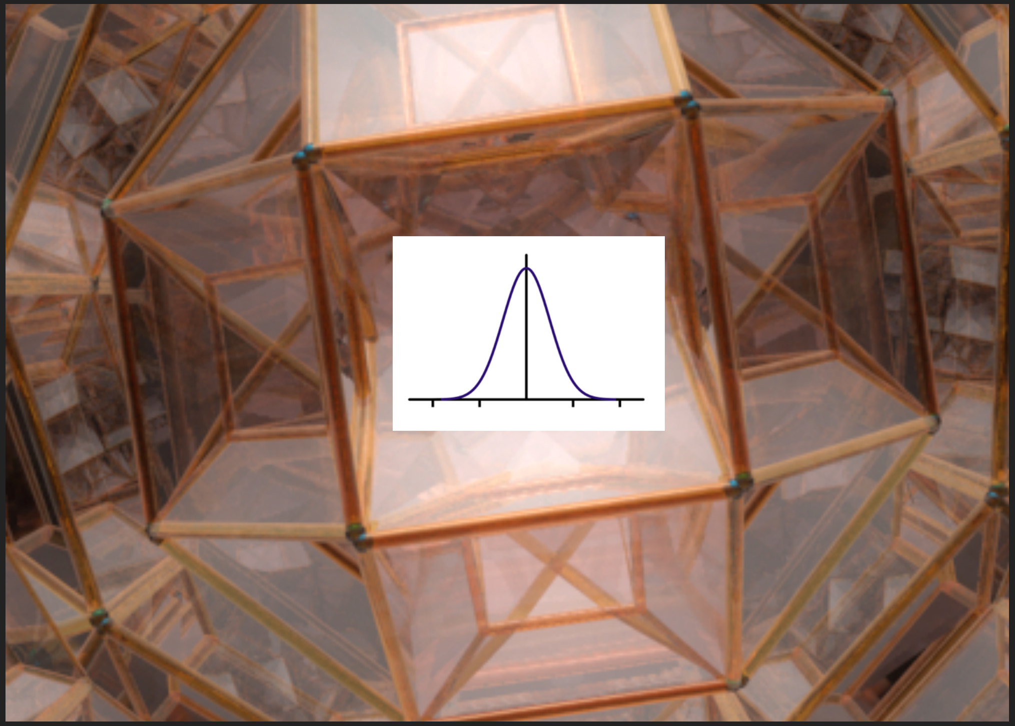
Why? Could our imagination is wrong?

2D IS VERY DIFFERENT FROM 10^6D

- We believe most of the mass of a Gaussian always lies near the mean
- This is true in low-dimension

- It is not always true (!)
- Most of the mass in high dimension lies at the edges of the distribution.





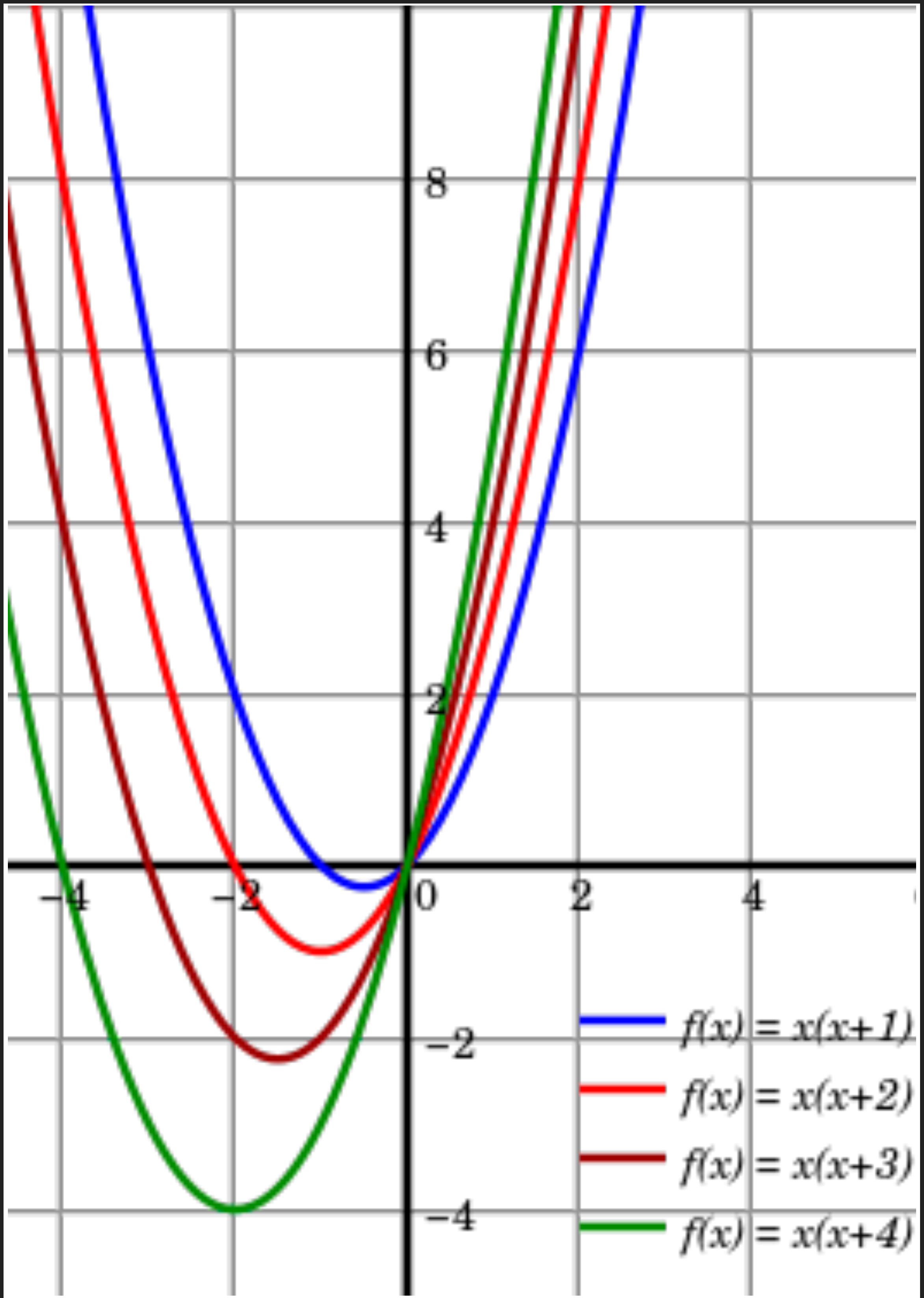
AN INTUITION FROM RANDOM QUADRATICS

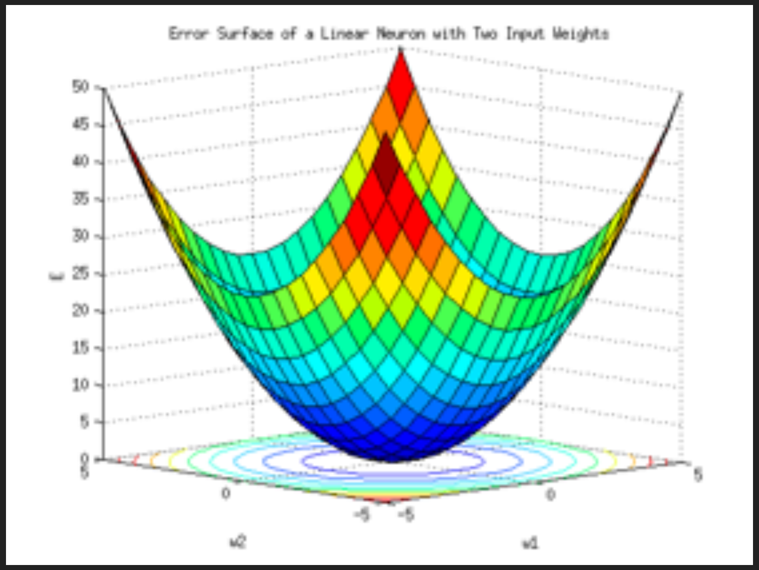
- Consider a random function

  $f(\theta) = \theta^T H \theta$

  where the Hessian **H** ~ N(μ, Σ).

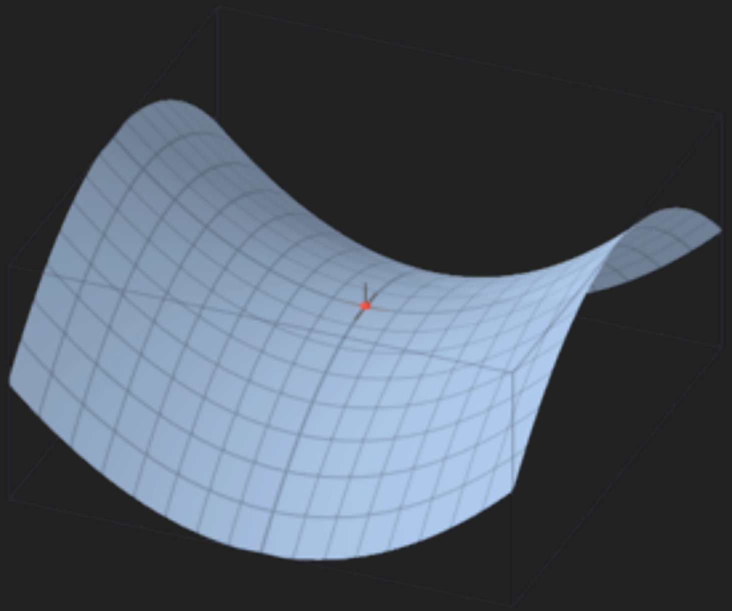
- The eigenvalues λ_i of the Hessian tell us what kind of critical point we have sampled.





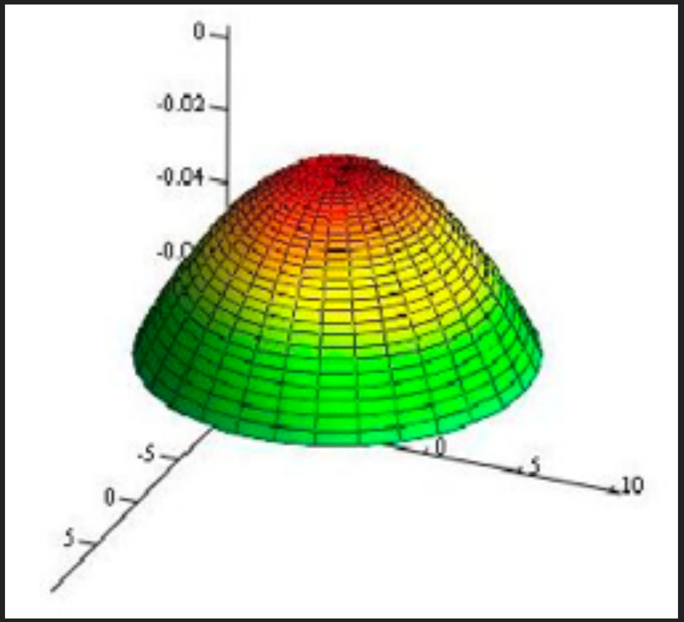
MINIMUM

Occur when the eigenvalues λ_i are all positive.



SADDLE POINT

Occur when there are positive and negative eigenvalues λ_i.

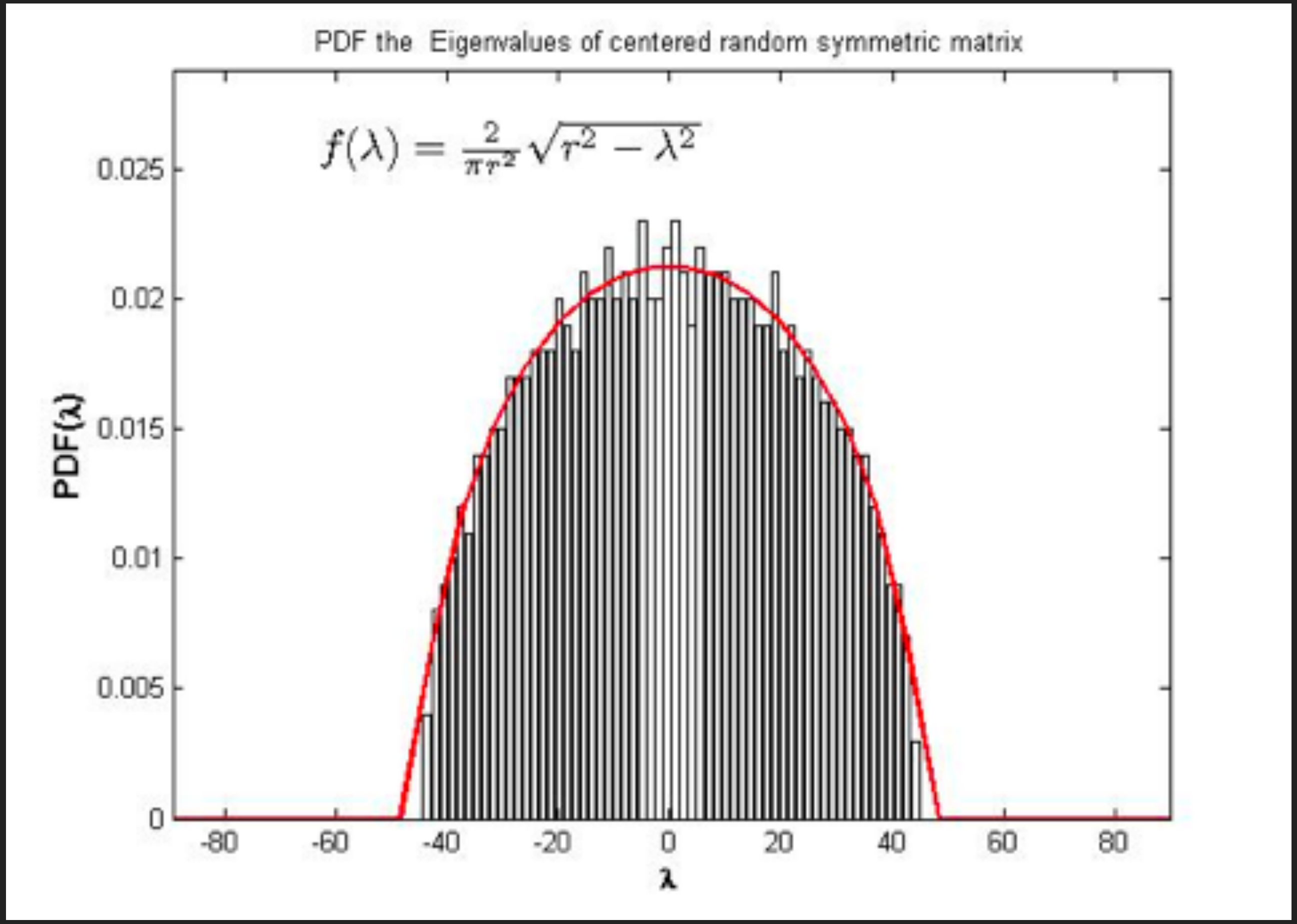


MAXIMUM

Occur when the eigenvalues λ_i are all negative.

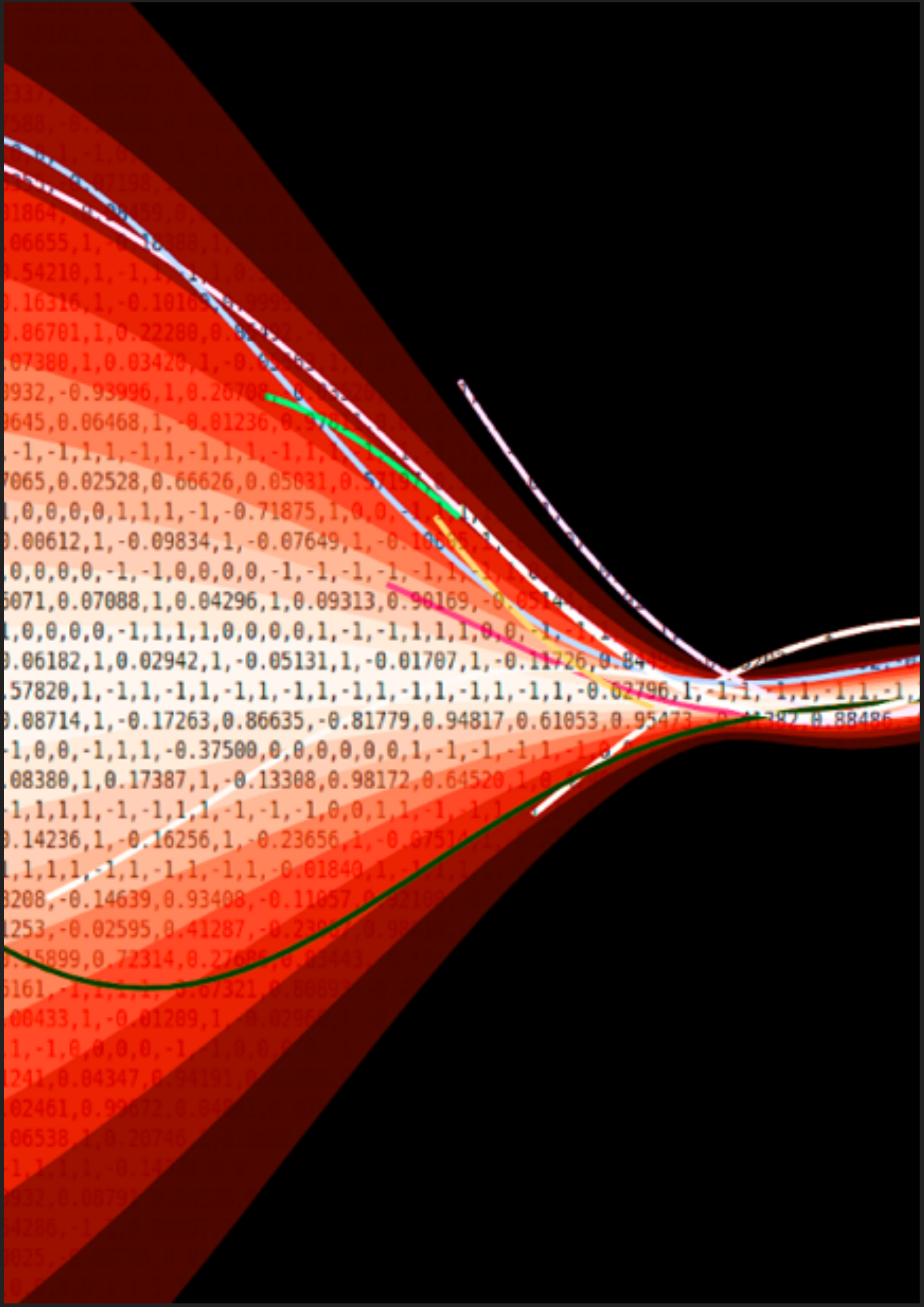
SEMI-CIRCULAR LAW AND COIN FLIPPING

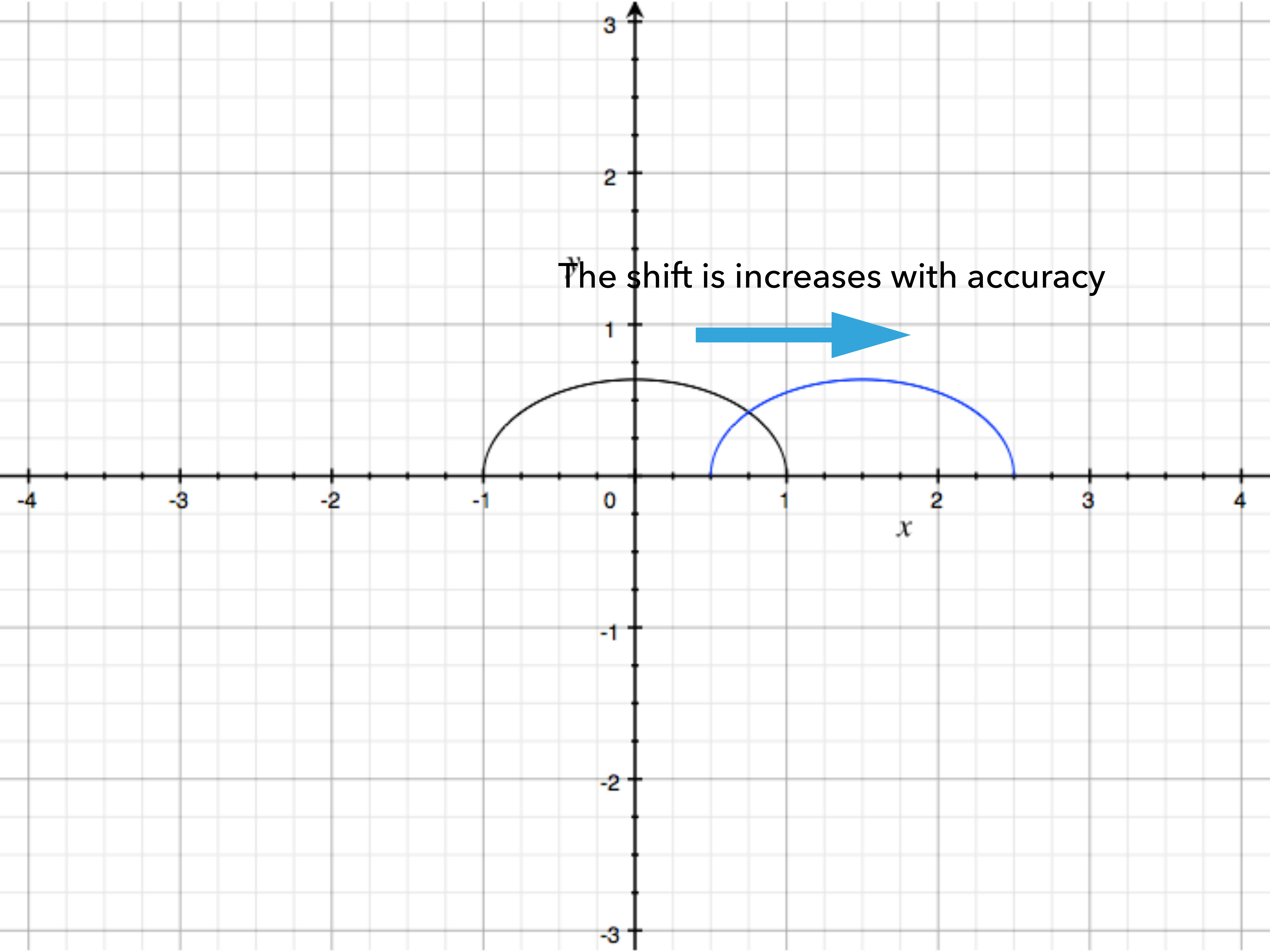
- The distribution of eigenvalues is given by the semi-circular law (Wigner, 1958).
- The sign of an eigenvalue is determined by a coin flip.
- The number of eigenvalues is the number of parameters.
- What is the likelihood of falling on heads 10^6 times in a row?



A MORE GENERAL CASE

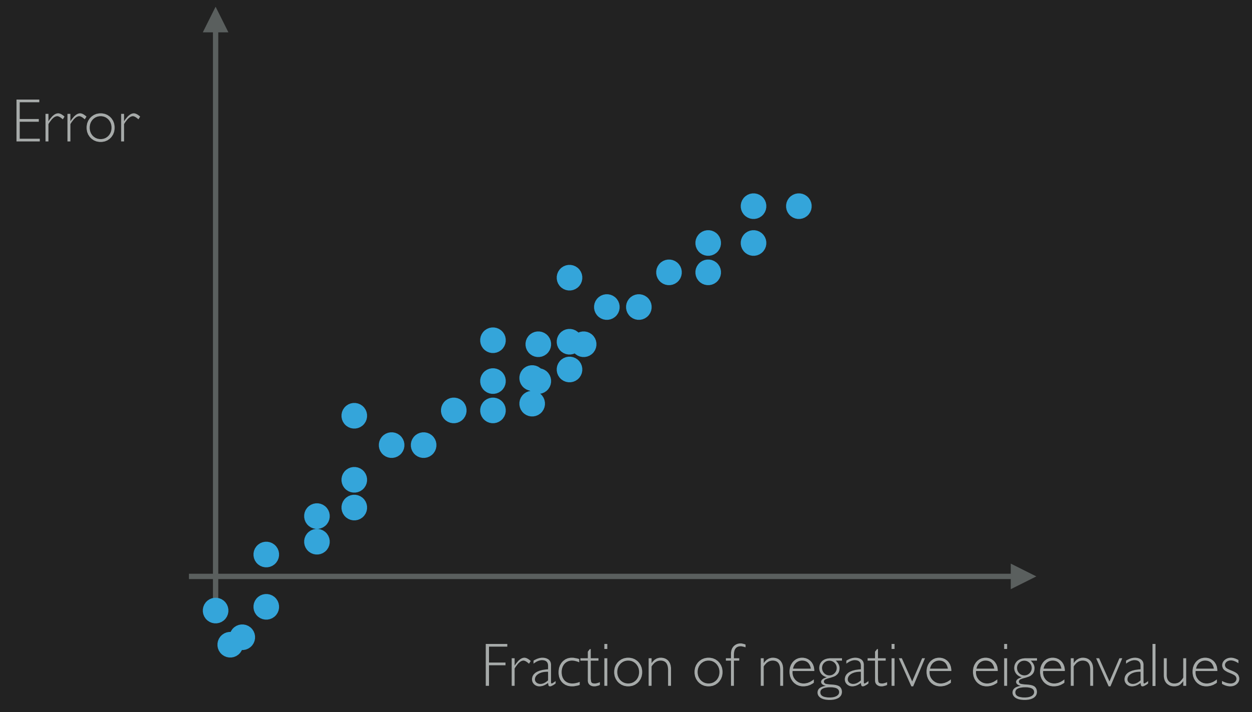
- Gaussian random fields can be seen as multi-dimensional Gaussian processes.
- They occur naturally in many applications due to the central limit theorem.
- In the context of statistical physics, (Bray & Dean, 2007) show that the critical points of these models follow the semi-circular rule shifted by the error ε.



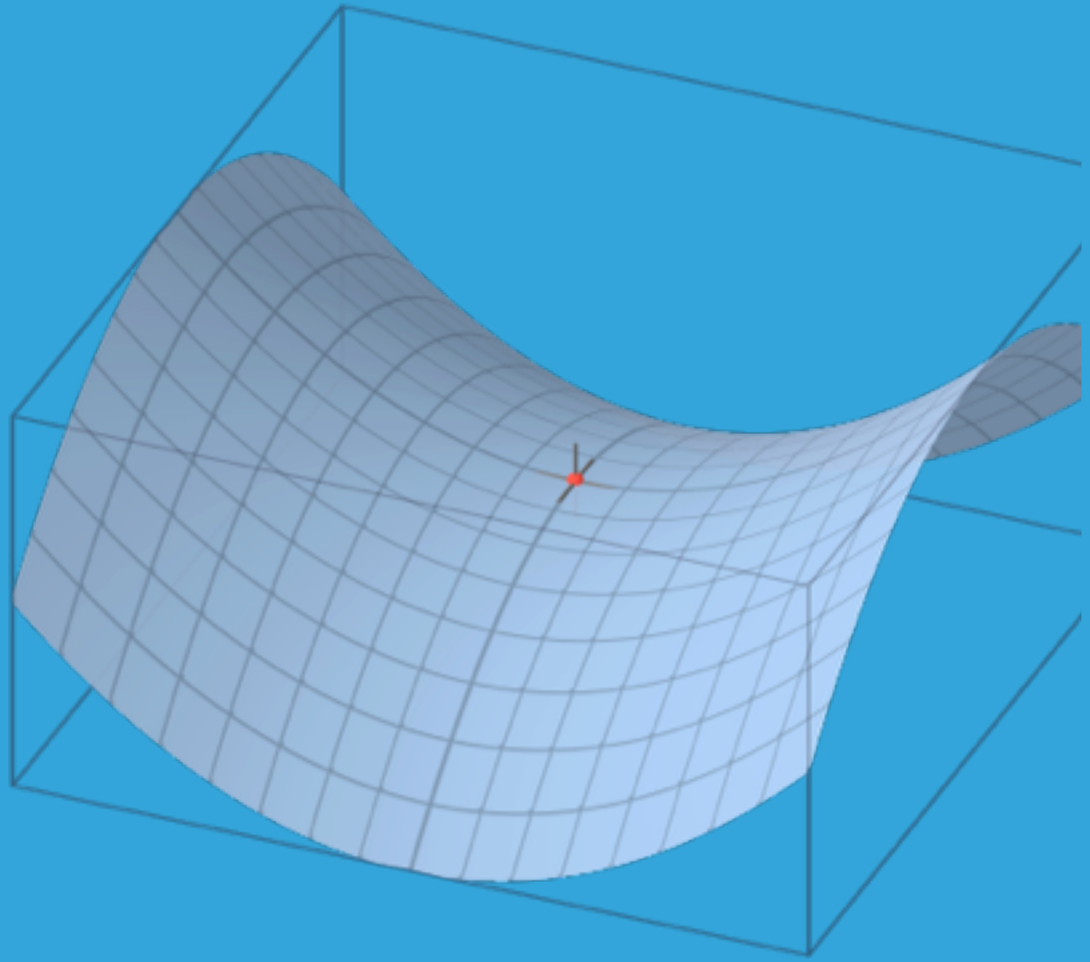


The shift is increases with accuracy

THE DISTRIBUTION OF CRITICAL POINTS

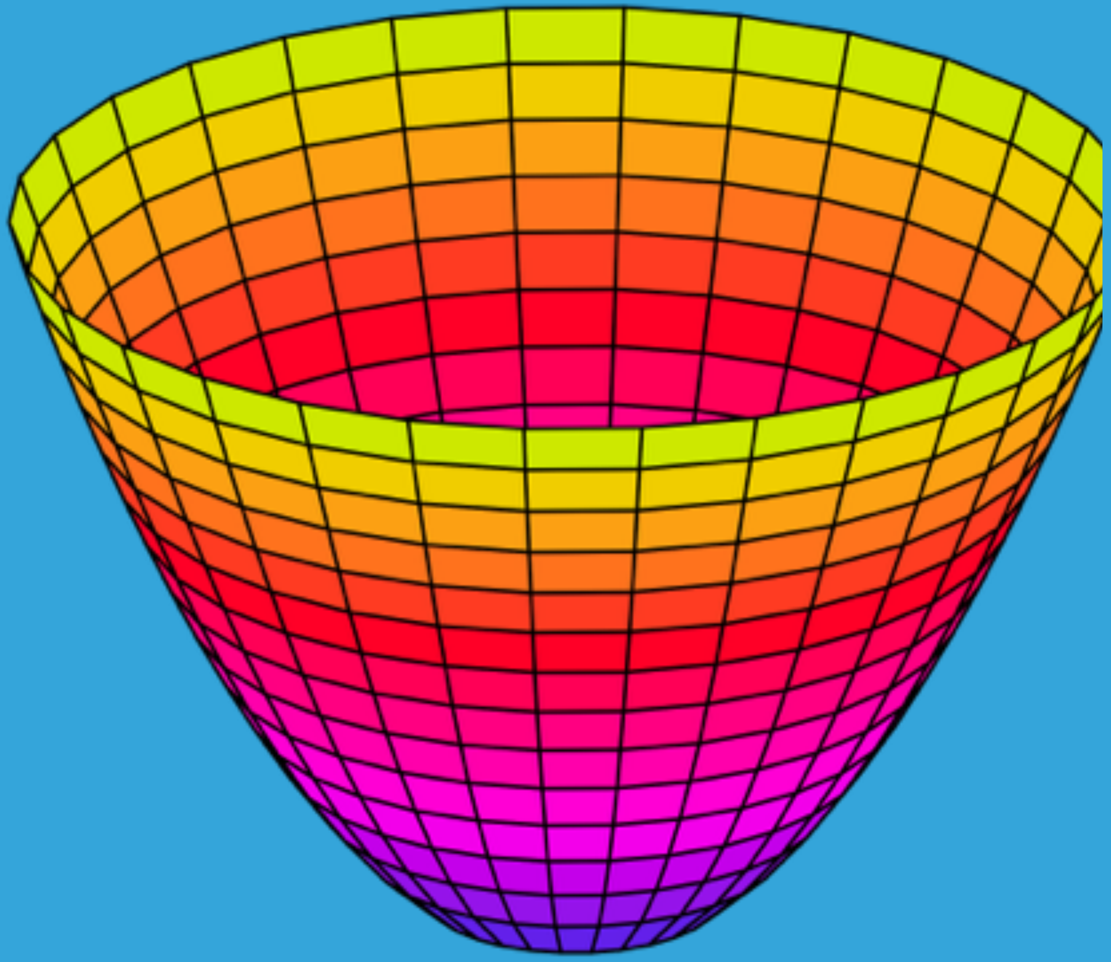


- (Bray and Dean, 2007) show critical points lie with high-probability on a curve in the space of error vs fraction of negative eigenvalues.



Observation 1

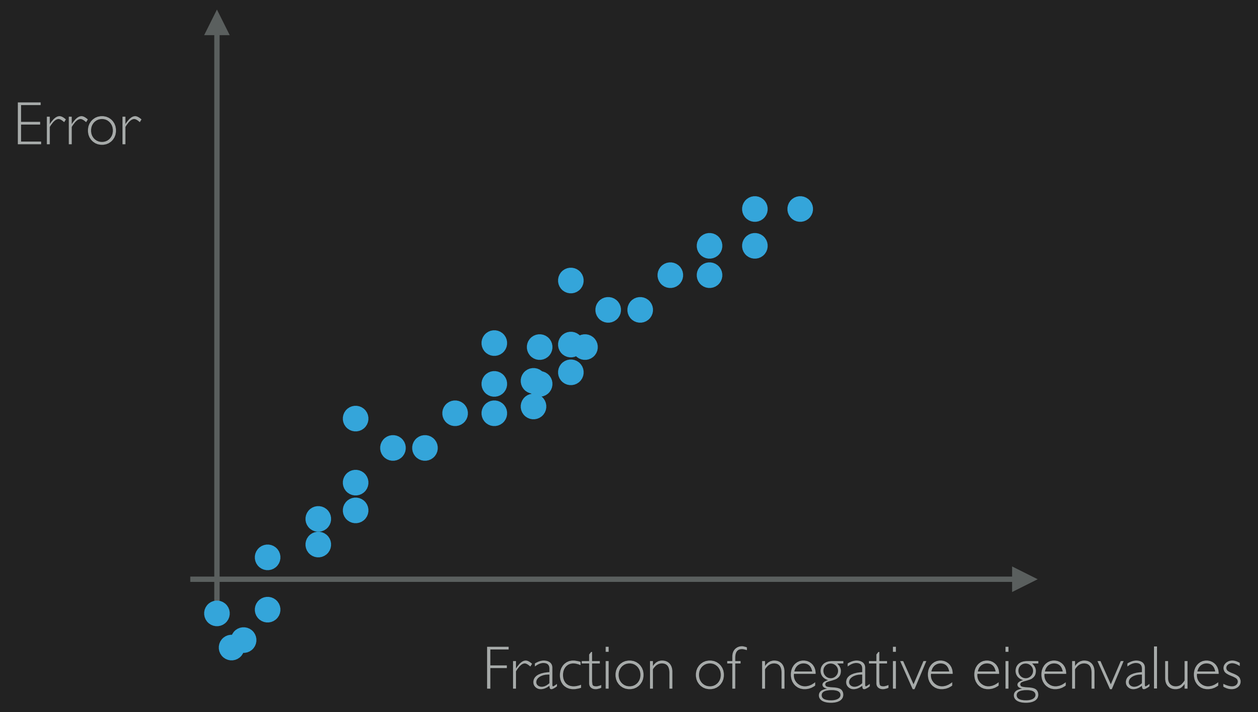
HIGH ERROR SOLUTIONS ARE LIKELY SADDLE POINTS



Observation 2

LOCAL MINIMA LIKELY HAVE NEAR OPTIMAL ERROR

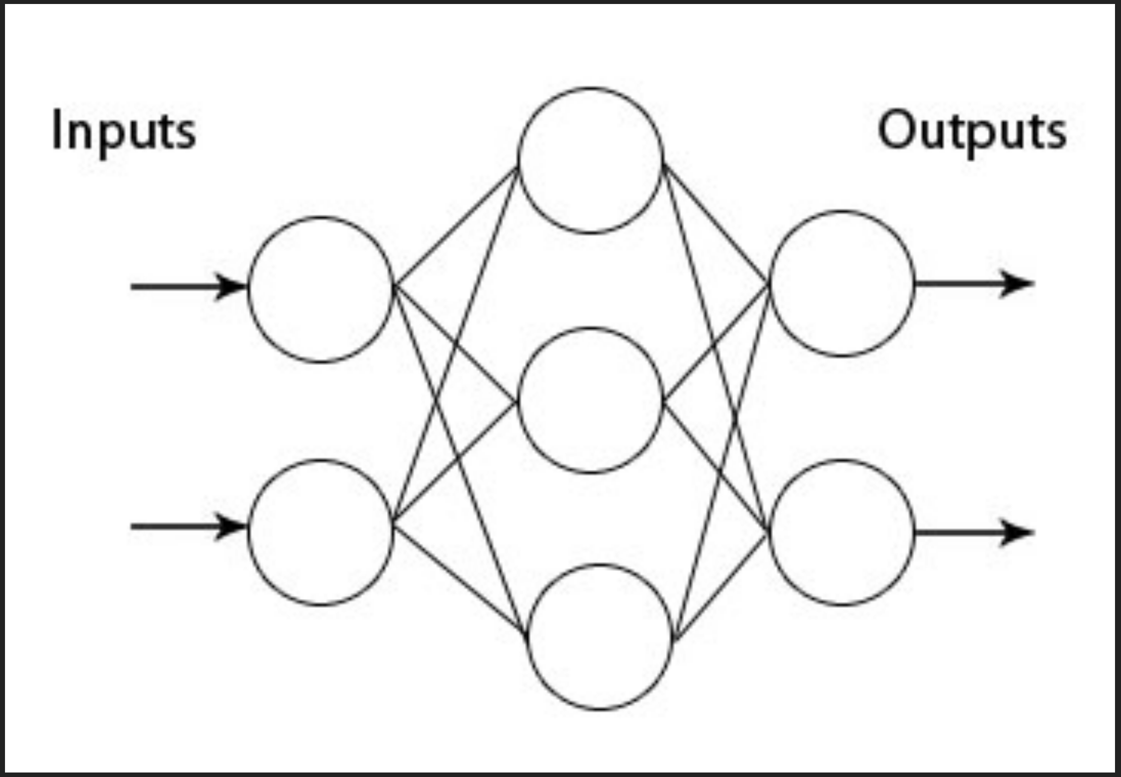
DO THESE RESULTS HOLD IN PRACTICE?



- It is not clear if neural nets exhibit this behavior in practice.
- Are the loss surfaces of neural nets similar to Gaussian random fields?

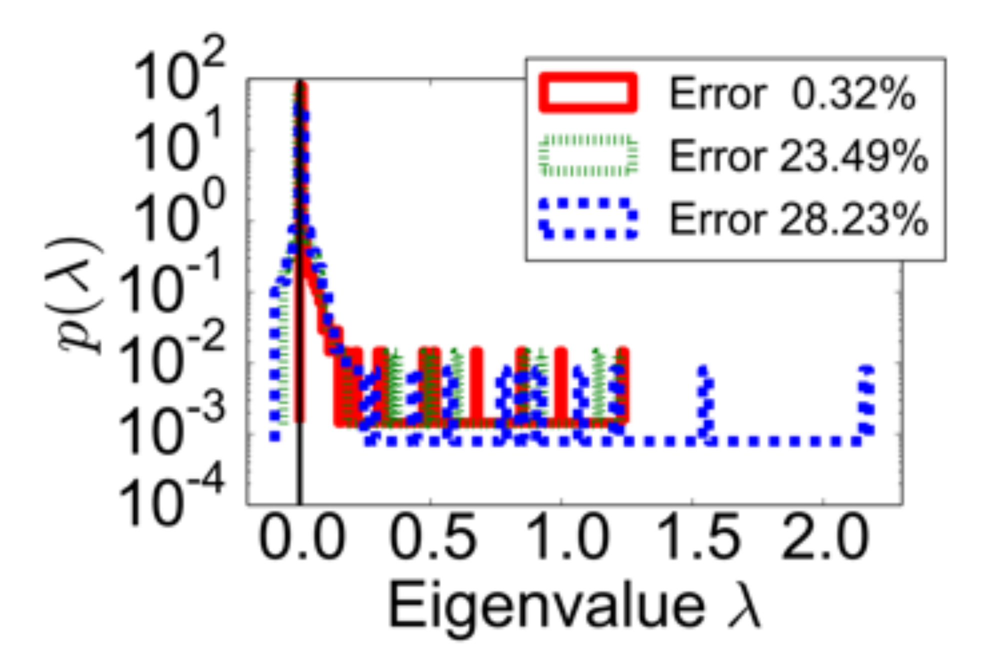
EXPERIMENTAL SETUP

- Does the Hessian follow Wigner's law in practice?
- We consider 1 hidden layer nets for object recognition with around 20k parameters.
- The datasets are MNIST and CIFAR resized to 10x10.
- This setup allows us to compute the Hessian exactly.

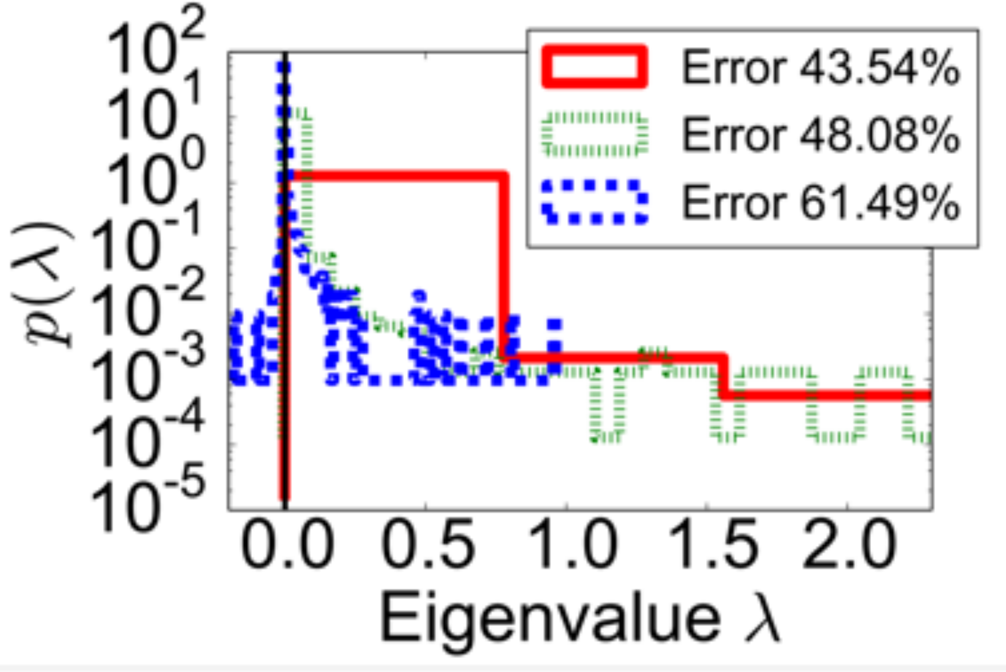


DO NEURAL NETS FOLLOW WIGNER'S LAW?

MNIST



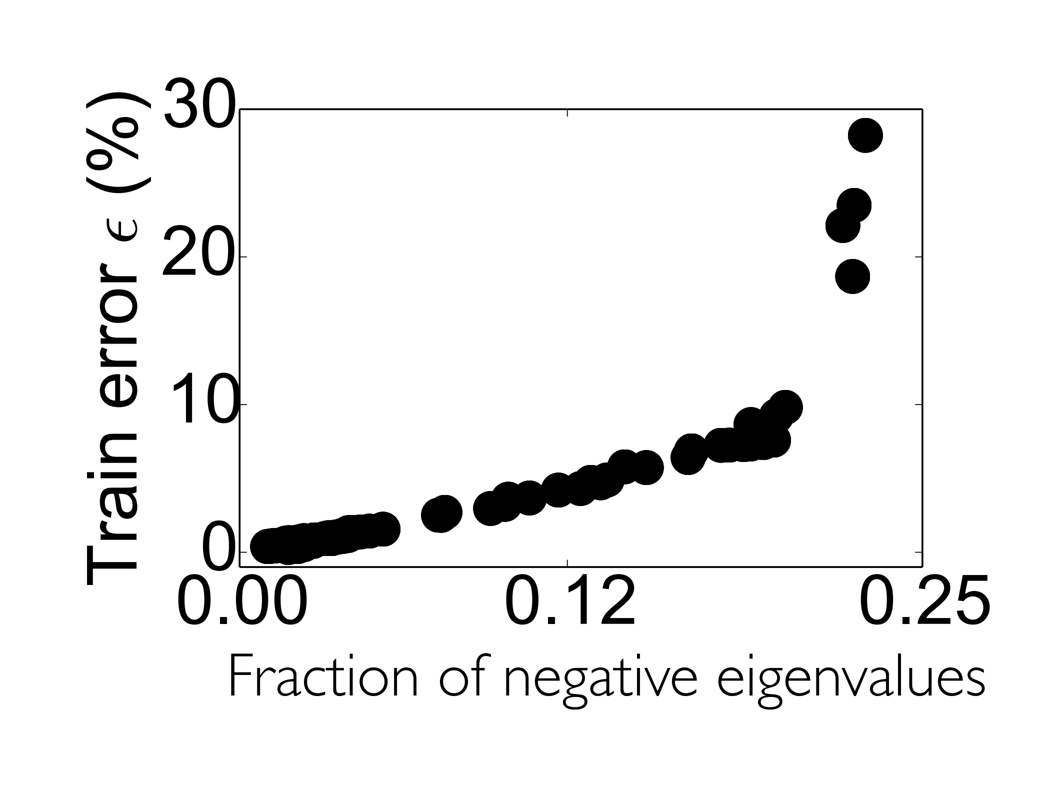
CIFAR



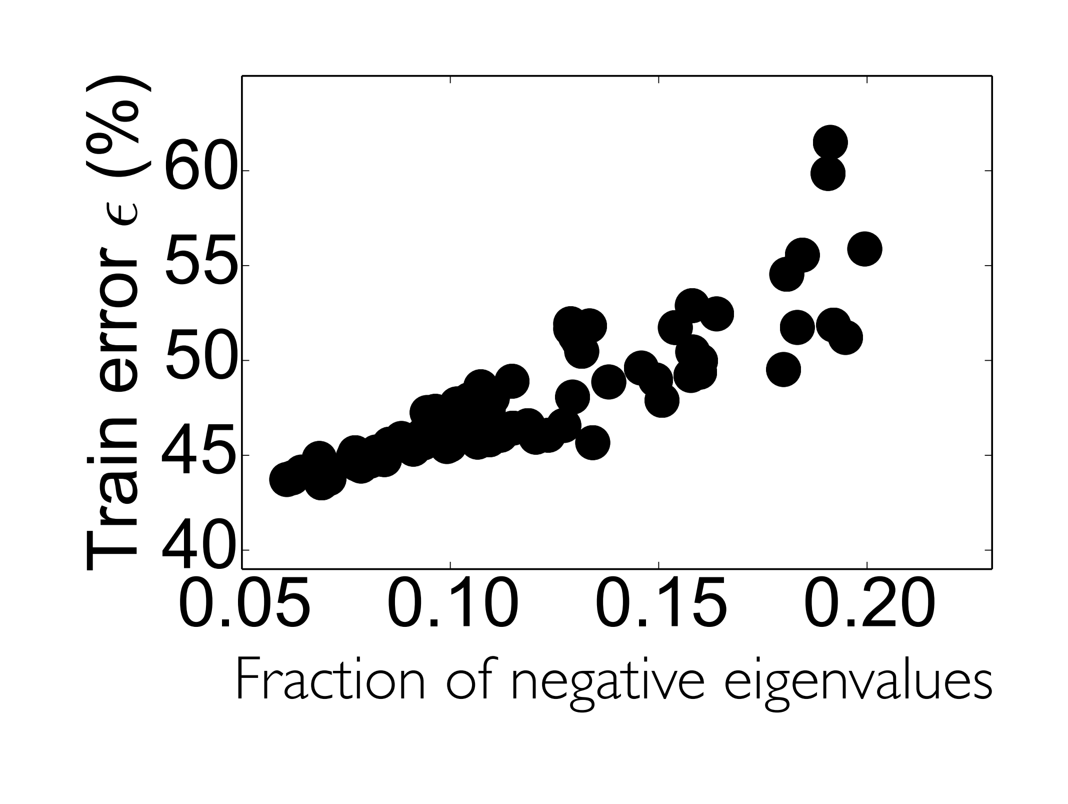
- The networks seem to loosely follow Wigner's law.
- The spectrum of eigenvalues shifts to the right as the error decreases.

IS THE DISTRIBUTION OF CRITICAL POINTS REGULAR?

MNIST



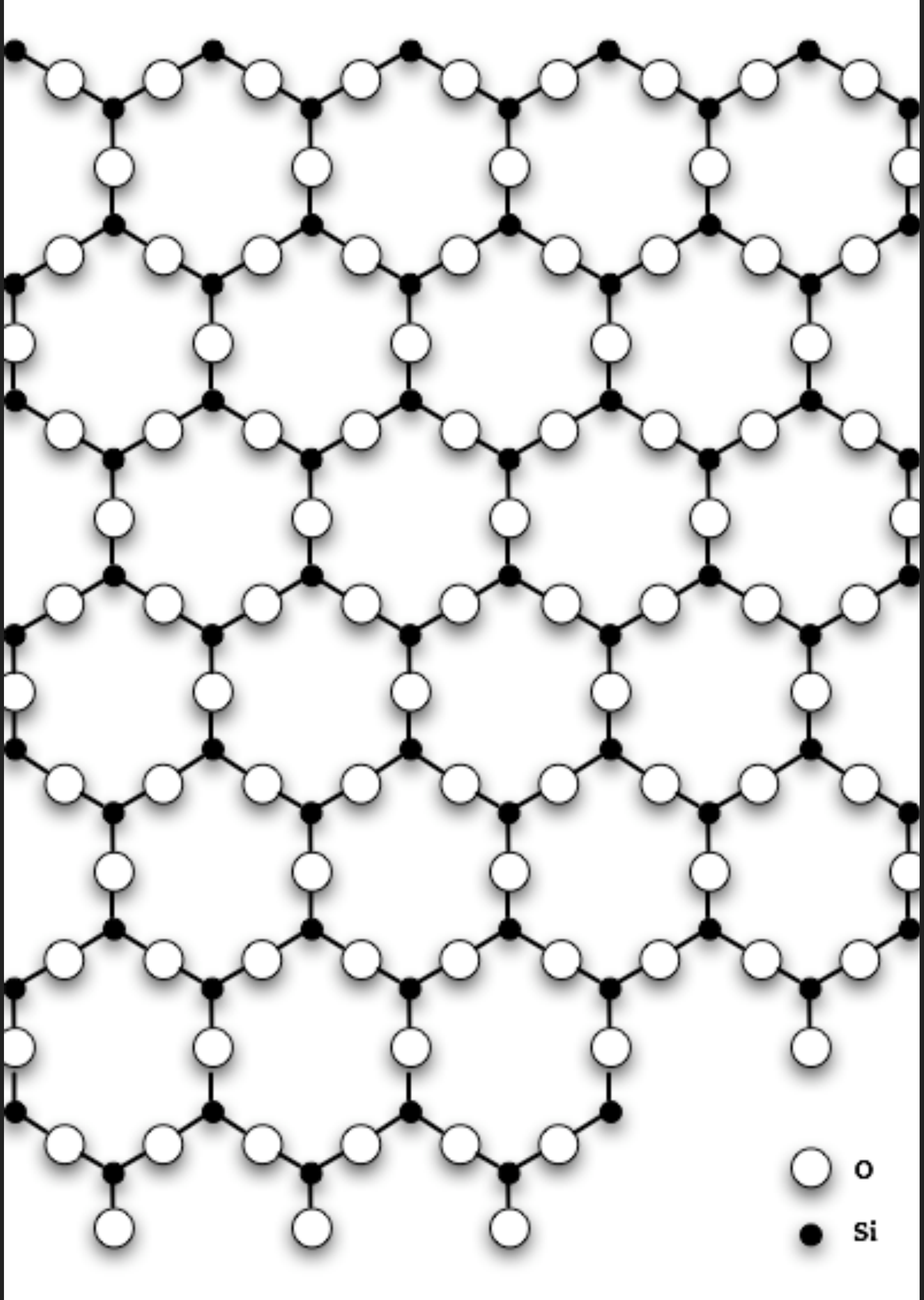
CIFAR



- The distribution exhibits a strong correlation between the error and the number of negative eigenvalues as caused by Wigner's law.
- The high error solutions are all saddle points leading to the near-optimum error as the index decreases.

NEURAL NETS AND SPIN GLASS

- (Choromanska et al, 2014) show that under some conditions rectified networks are a spin glass model.
- This explains the applicability of random matrix theory to neural nets.
- http://arxiv.org/abs/1412.0233

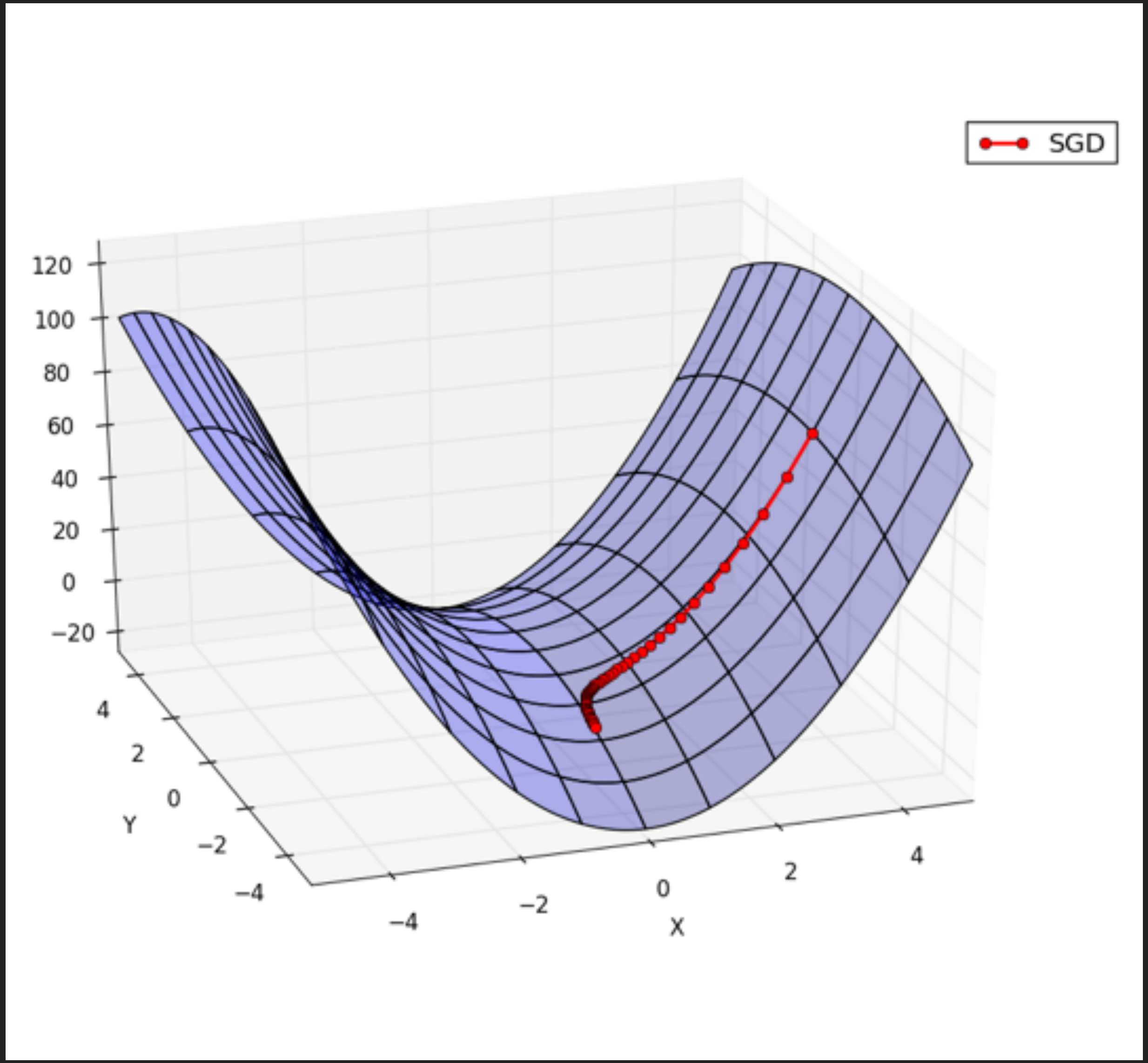


CONSEQUENCES OF THE PREVALENCE OF SADDLE POINTS

- Finding a local minimum is actually a desirable outcome for optimization.
- A local minimum can be found by following a sequence of saddle points.
- Do our optimizers behave correctly near saddle points?

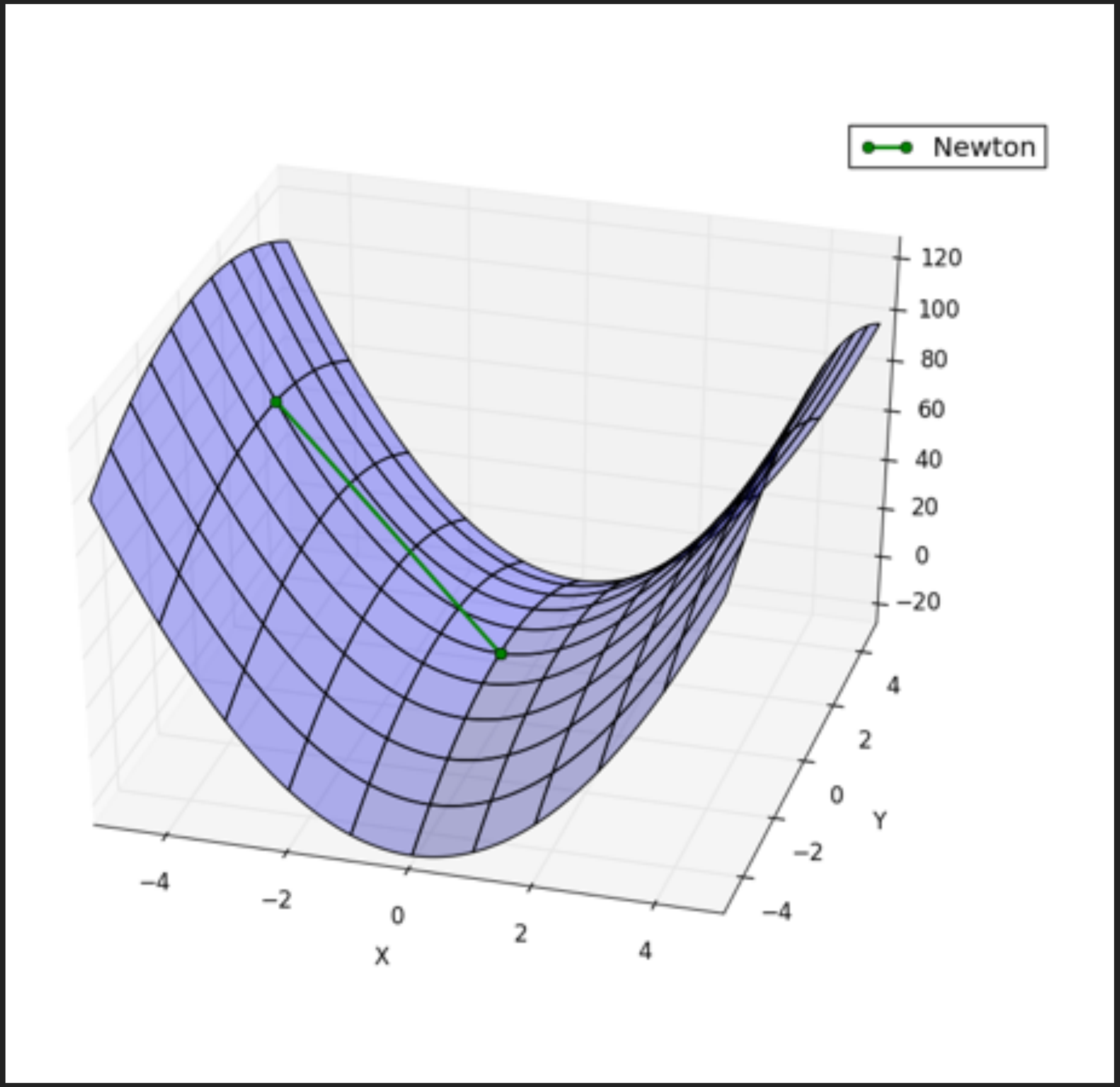
BACK-PROPAGATION

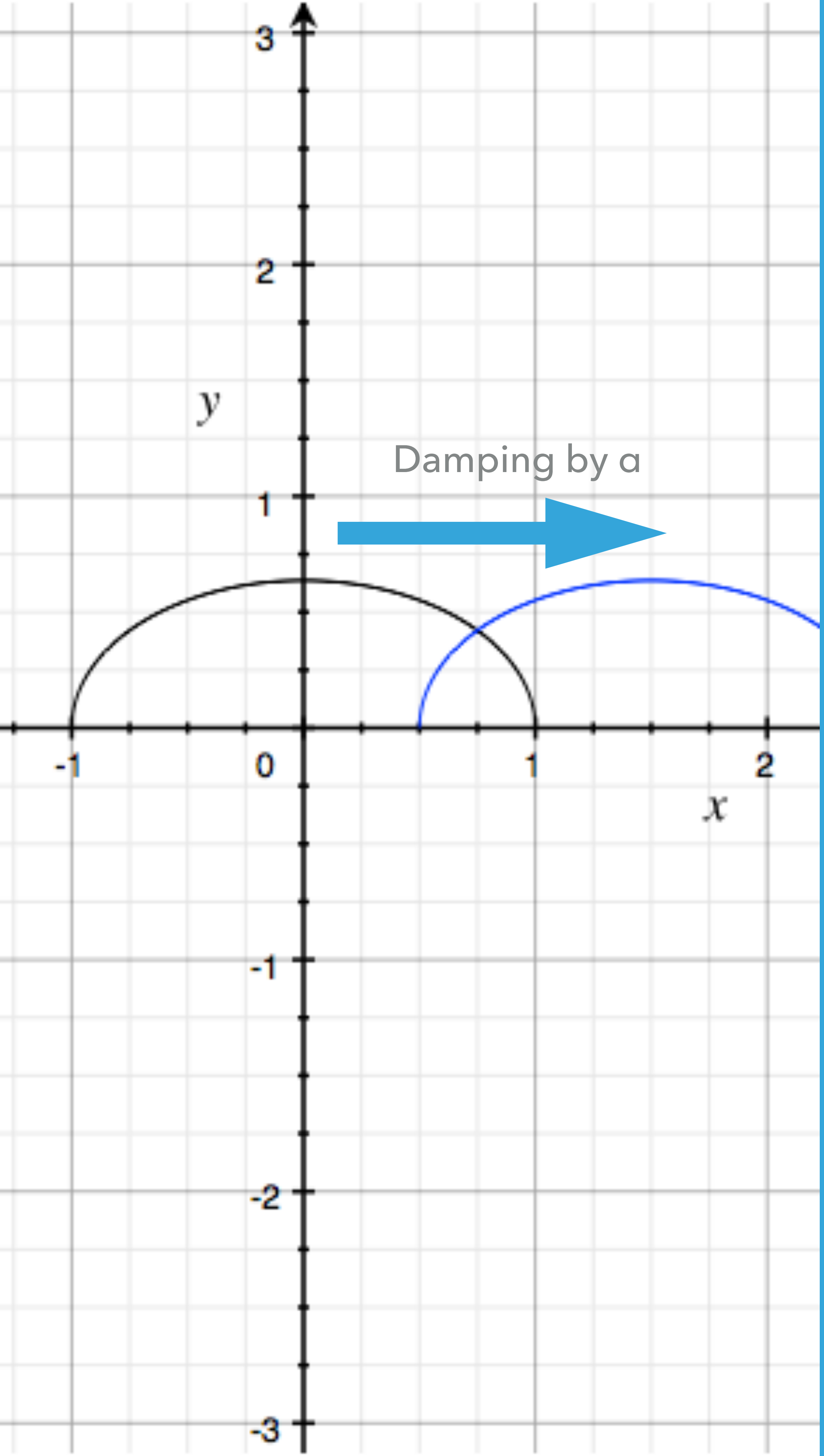
- Saddle points curves the trajectory of gradient descent.
- Gradient descent slows down near saddle points.
- Can second-order methods help us?



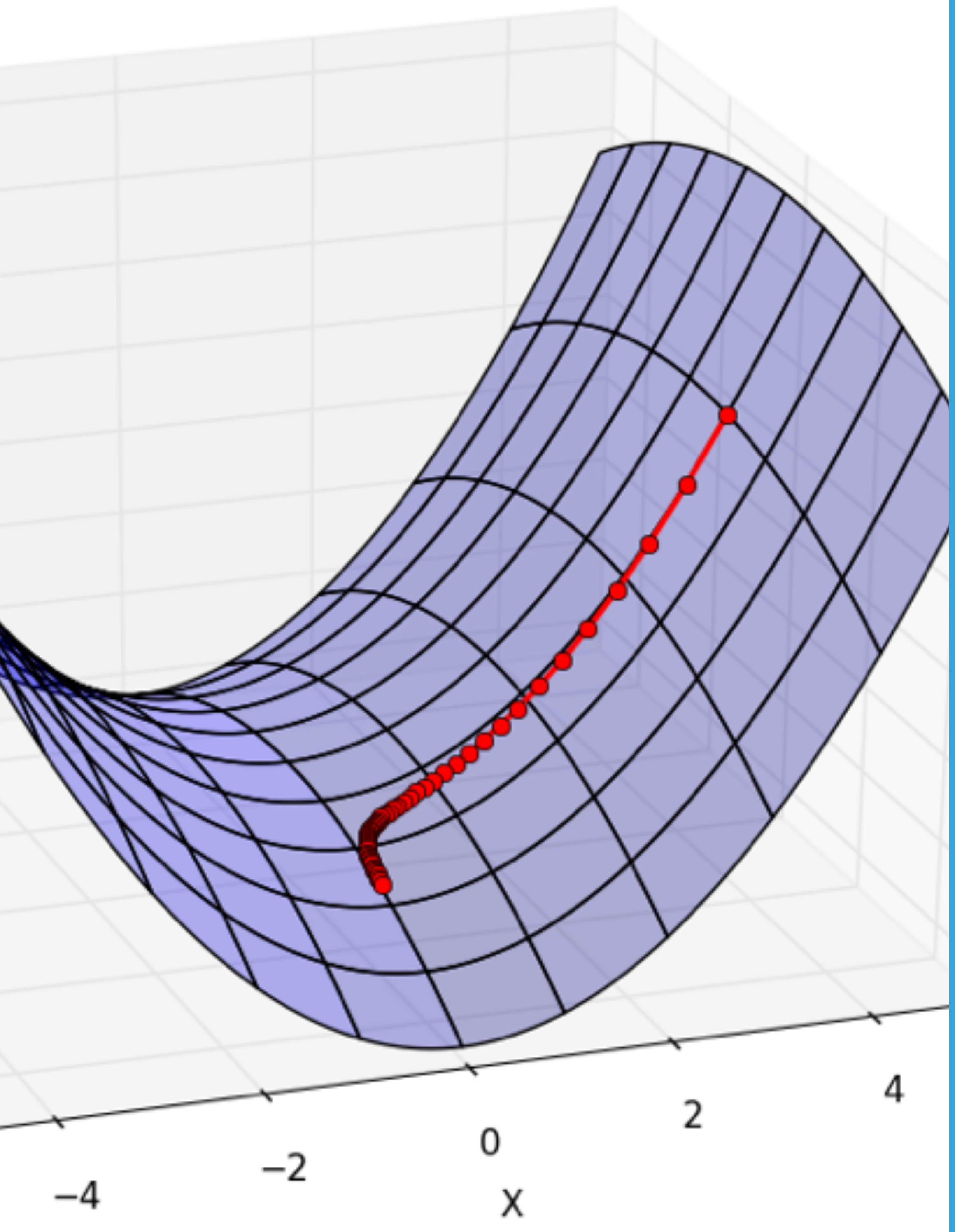
NEWTON METHOD

- The Newton method jumps directly to the saddle point.
- The Newton method seeks any critical points indiscriminately.
- The recommended solution is to damp the eigenvalues by a factor α such that we have H + αI





DAMPING OBFUSCATES NEGATIVE CURVATURE



DAMPING OBFUSCATES NEGATIVE CURVATURE

We need to properly deal with negative curvature.

PRECONDITIONING

- Preconditioning is a way to solve a problem by tackling an easier but equivalent problem.
- It is made by a change of variables

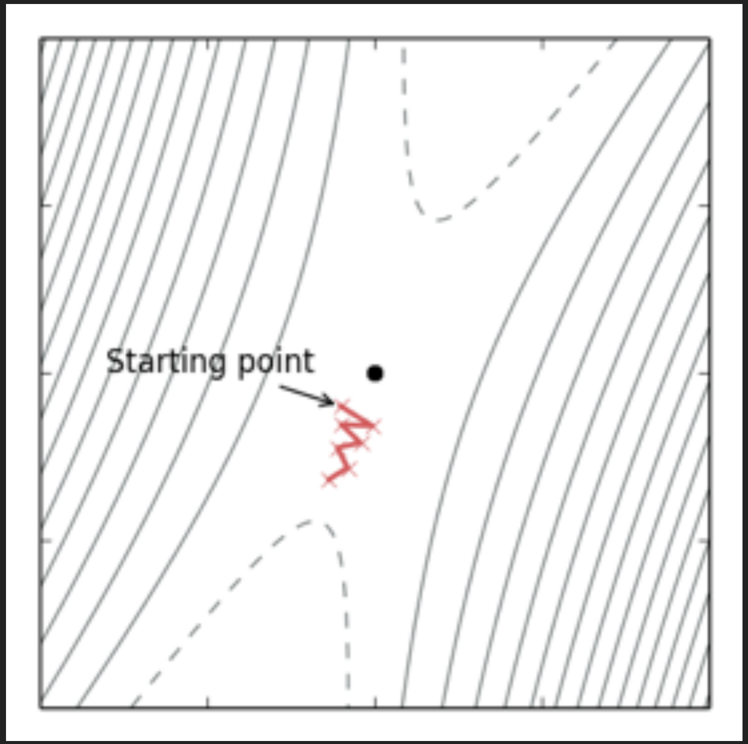
$$\hat{f}(\hat{\theta}) = f(\mathbf{D}^{-\frac{1}{2}}\hat{\theta}) = f(\theta)$$

which transforms the derivates

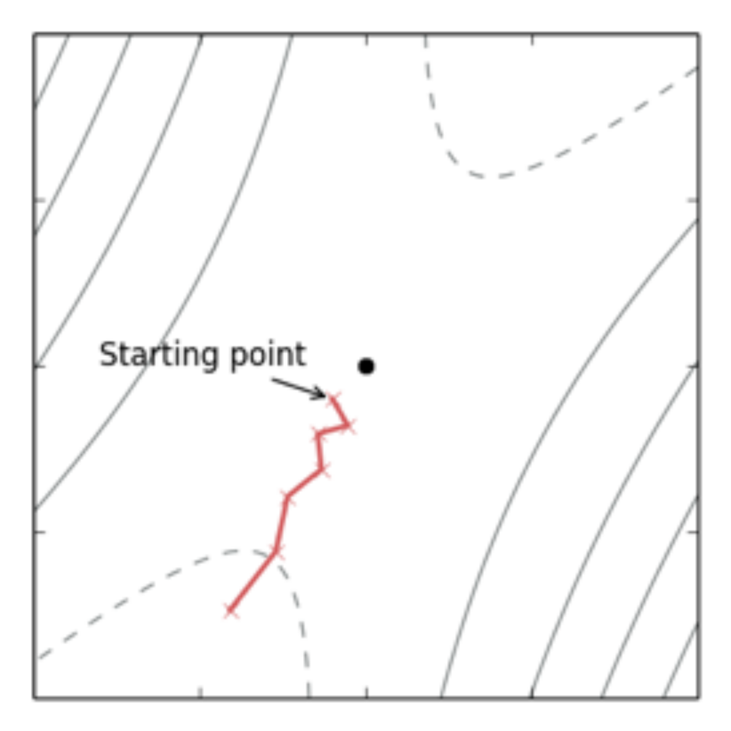
$$\nabla\hat{f}(\hat{\theta}) = \mathbf{D}^{-\frac{1}{2}}\nabla f(\theta)$$

$$\nabla^2\hat{f}(\hat{\theta}) = \mathbf{D}^{-\frac{1}{2}\top}\mathbf{H}\mathbf{D}^{-\frac{1}{2}} \text{ with } \mathbf{H} = \nabla^2 f(\theta)$$

Original



Preconditioned



PRECONDITIONING

- The trick is to choose D so that the preconditioned Hessian has less curvature

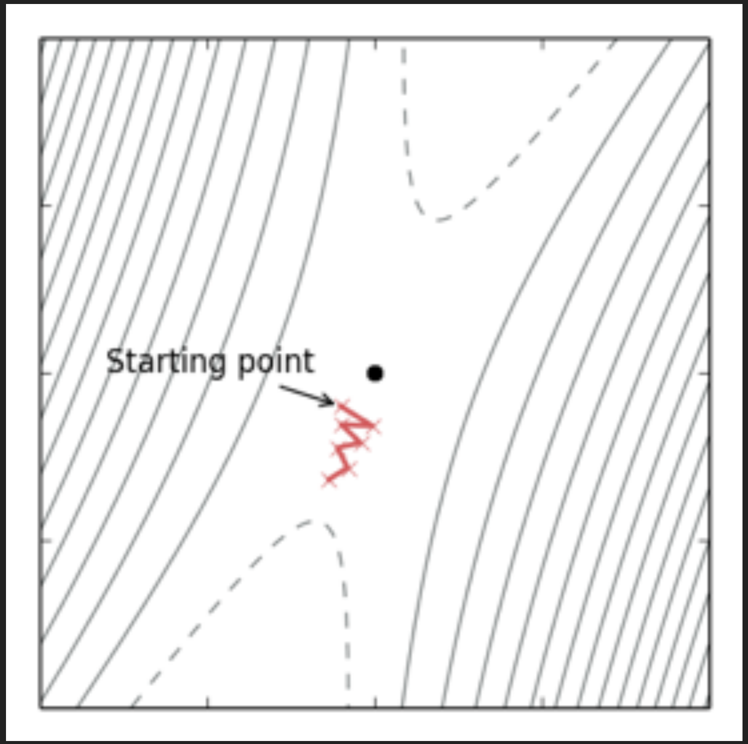
$$\nabla^2 \hat{f}(\hat{\theta}) = \mathbf{D}^{-\frac{1}{2}\top}\mathbf{H}\mathbf{D}^{-\frac{1}{2}}$$

- It is easier to make progress in each direction if they have the same curvature.

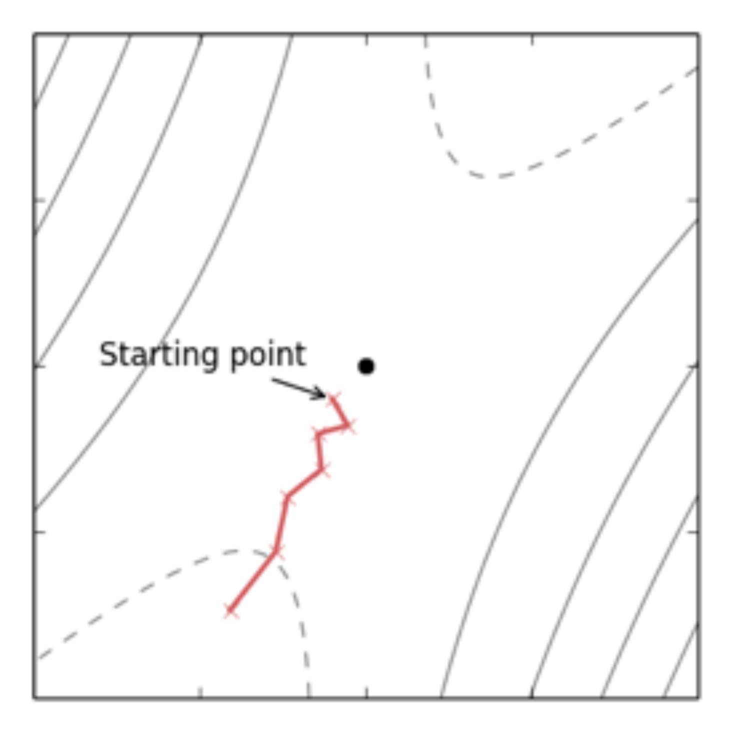
- The amount of curvature is measured by the condition number

$$\kappa(\mathbf{H}) = \frac{\sigma_{\max}(\mathbf{H})}{\sigma_{\min}(\mathbf{H})}$$

Original



Preconditioned



PRECONDITIONING

- The optimal choice to reduce the curvature would be H if it is positive definite.
- The issue is that it is to computationally intensive to store and invert the Hessian.
- Diagonal preconditioners are used for this reason.

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

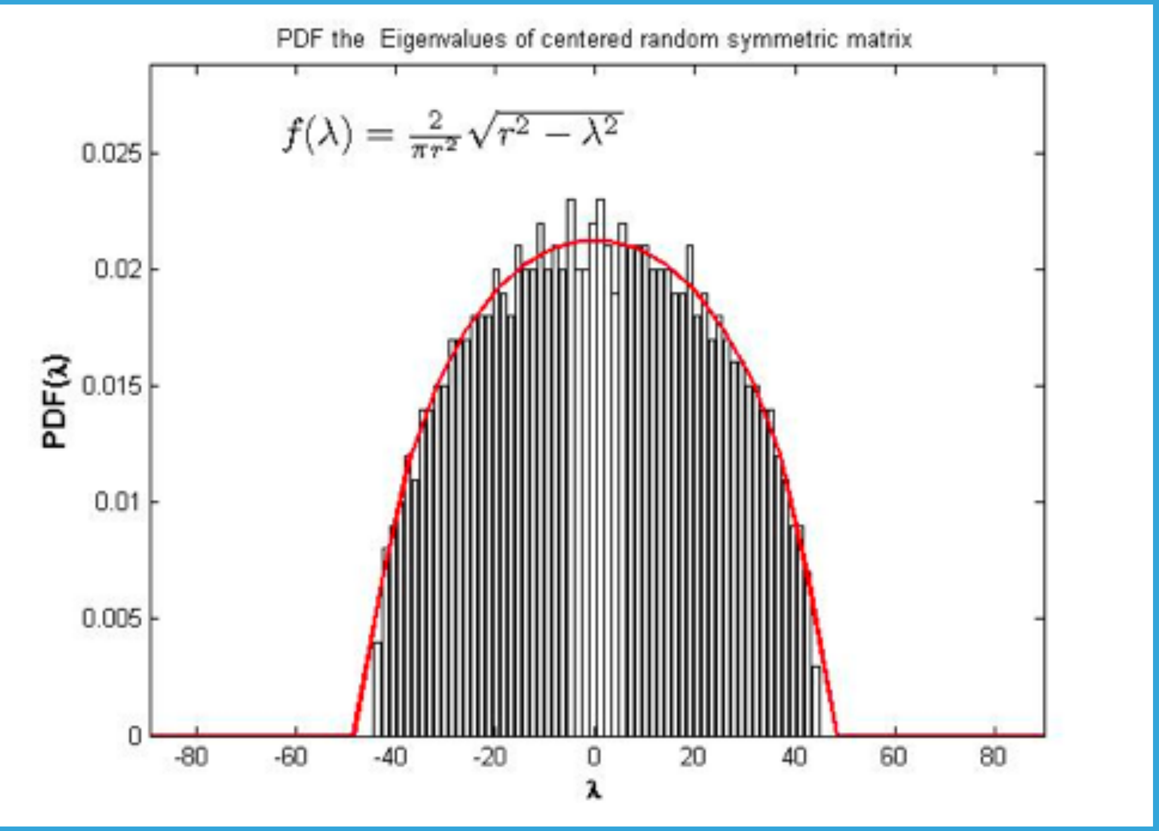
JACOBI

- The most common type of preconditioned is to use the diagonal of the Hessian.
- It effectively gives us an adaptive learning rate for each parameter based on the curvature

$$\theta_t = \theta_{t-1} - \eta \mathbf{D}^{-1} \nabla f(\theta)$$

- Jacobi does not work well for non-convex problems.

$$\mathbf{D} = \begin{pmatrix} |H_{11}| & 0 & 0 & \dots \\ 0 & |H_{22}| & 0 & \dots \\ 0 & 0 & |H_{33}| & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$



OPPOSING CURVATURES CANCEL

$$H_{ii} = \left|\sum_j \lambda_i \alpha_{ij}^2\right| \approx 0$$

SOLUTION

$$\cancel{\left|\sum_j \lambda_i \alpha_{ij}^2\right| \approx 0}$$

$$\sum_i \left|\lambda_i\right| \alpha_{ij}^2$$

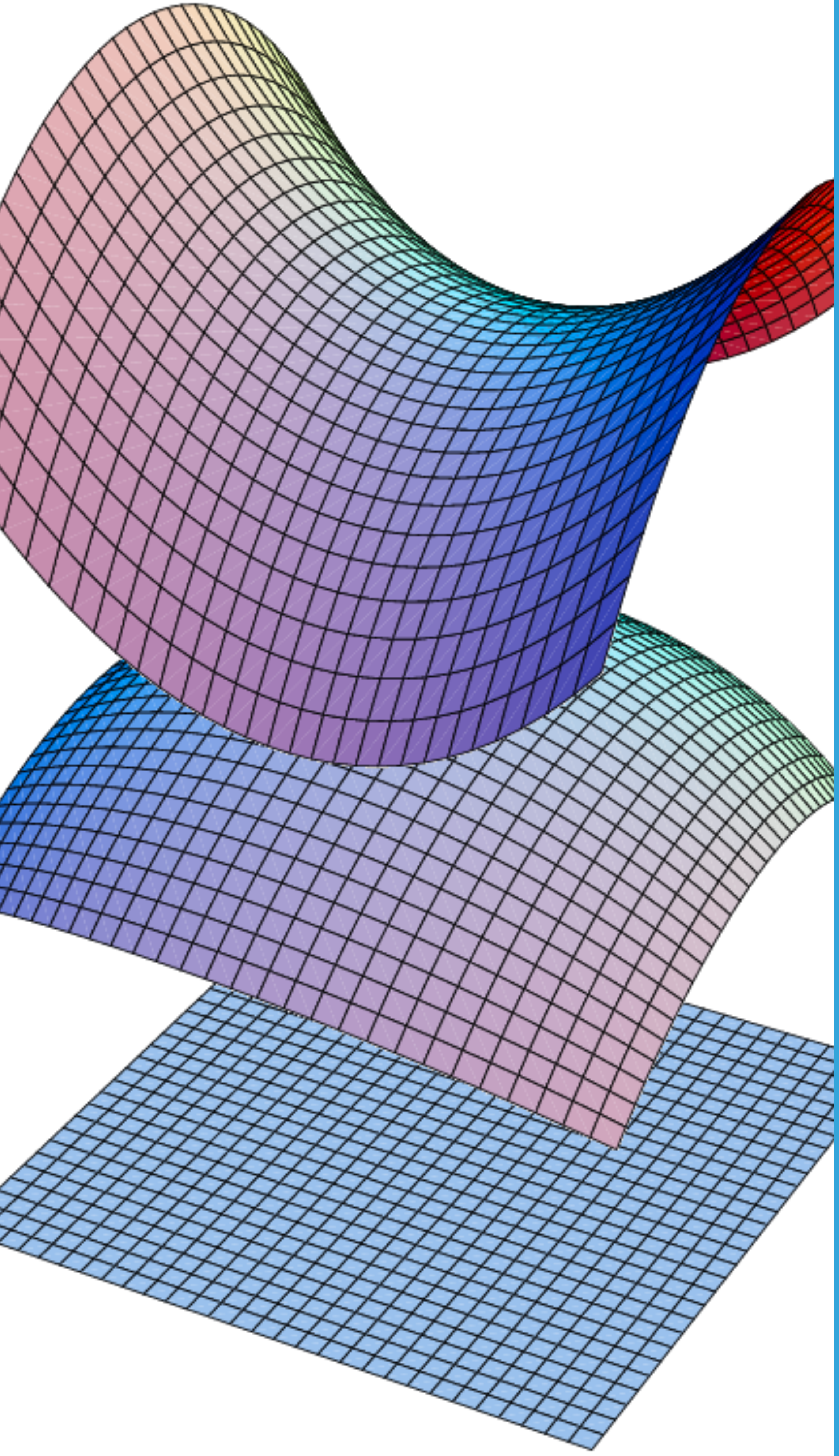
- Prevent the signs from cancelling by taking the absolute value.
- This solution is not tractable as it requires an eigen-decomposition.

EQUILIBRATION

- Equilibration is a technique developed in the mathematics community by (Sluis, 1969) that we rediscovered.
- Equilibration rescales each row by its norm.
- We are able to prove the new result that it reduces this upper bound of the condition number

$$\kappa(\mathbf{H}) < \frac{2}{|\det \mathbf{H}|} \left(\frac{||\mathbf{H}||_F}{\sqrt{N}} \right)^N$$

$$\mathbf{D} = \begin{pmatrix} ||H_1||^2 & 0 & 0 & \dots \\ 0 & ||H_2||^2 & 0 & \dots \\ 0 & 0 & ||H_3||^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$



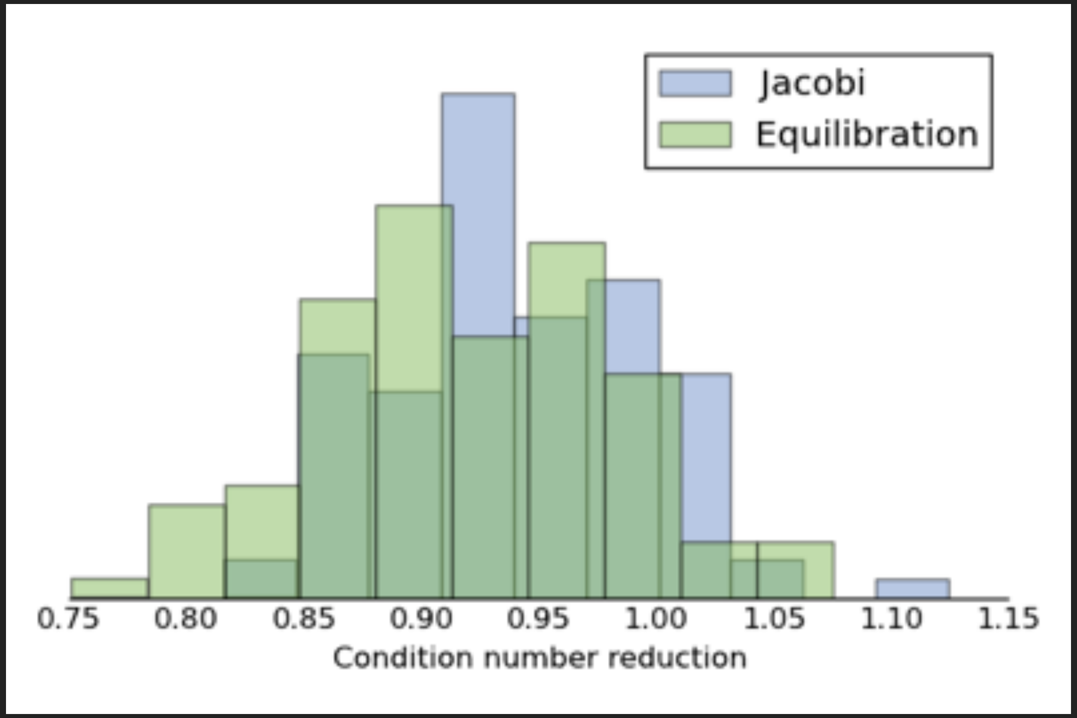
CURVATURE CANNOT DISAPPEAR

$$\|\mathbf{H}_i\|^2 = (\mathbf{H}^T\mathbf{H})_{ii} = \sum_j \lambda_i^2 \alpha_{ij}^2$$

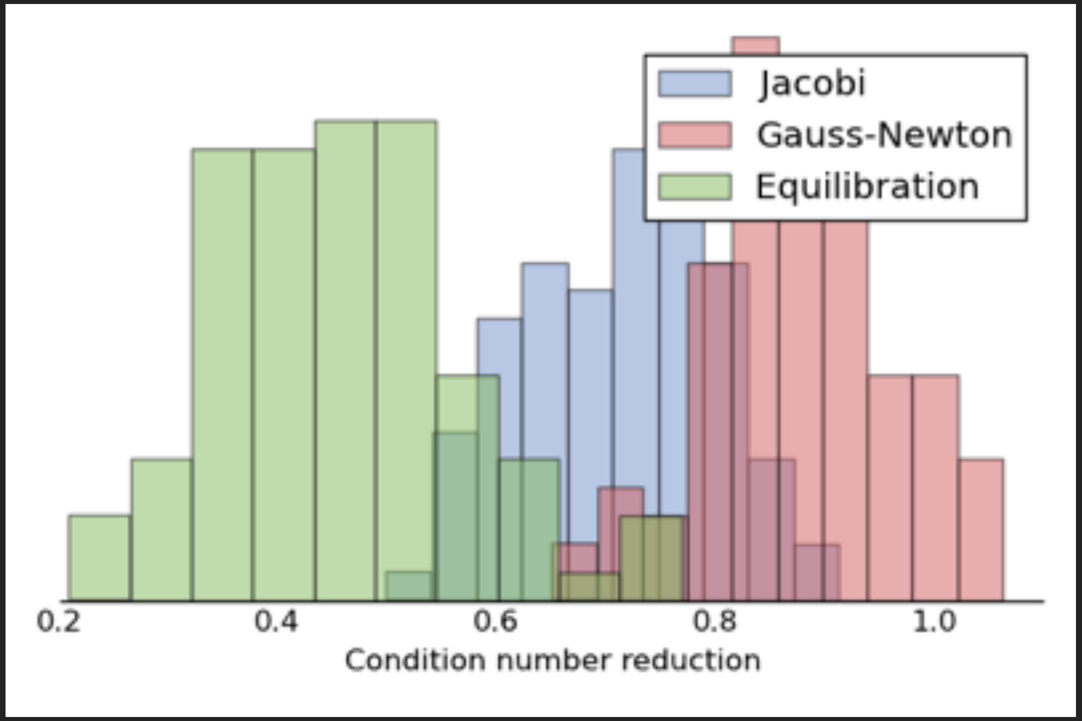
$$\frac{1}{\|\mathbf{H}_i\|} \leq \frac{1}{\sum_i |\lambda_i| \, \alpha_{ij}^2}$$

DOES THIS TRANSLATE IN PRACTICE?

Convex

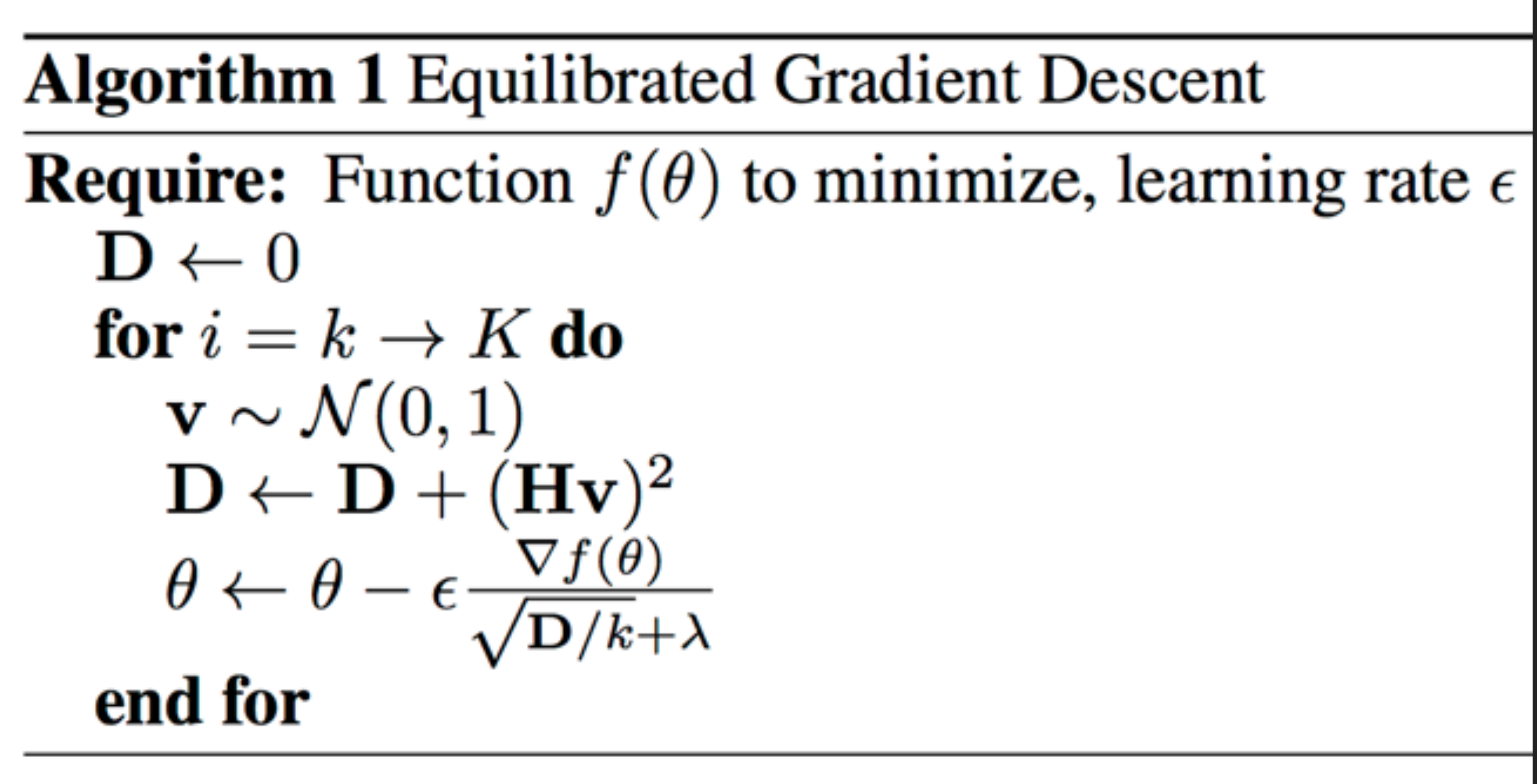


Non-Convex



- There is not much difference in performance in the convex case as there is no negative curvature.
- There is a sizable difference in the non-convex case.

IMPLEMENTATION

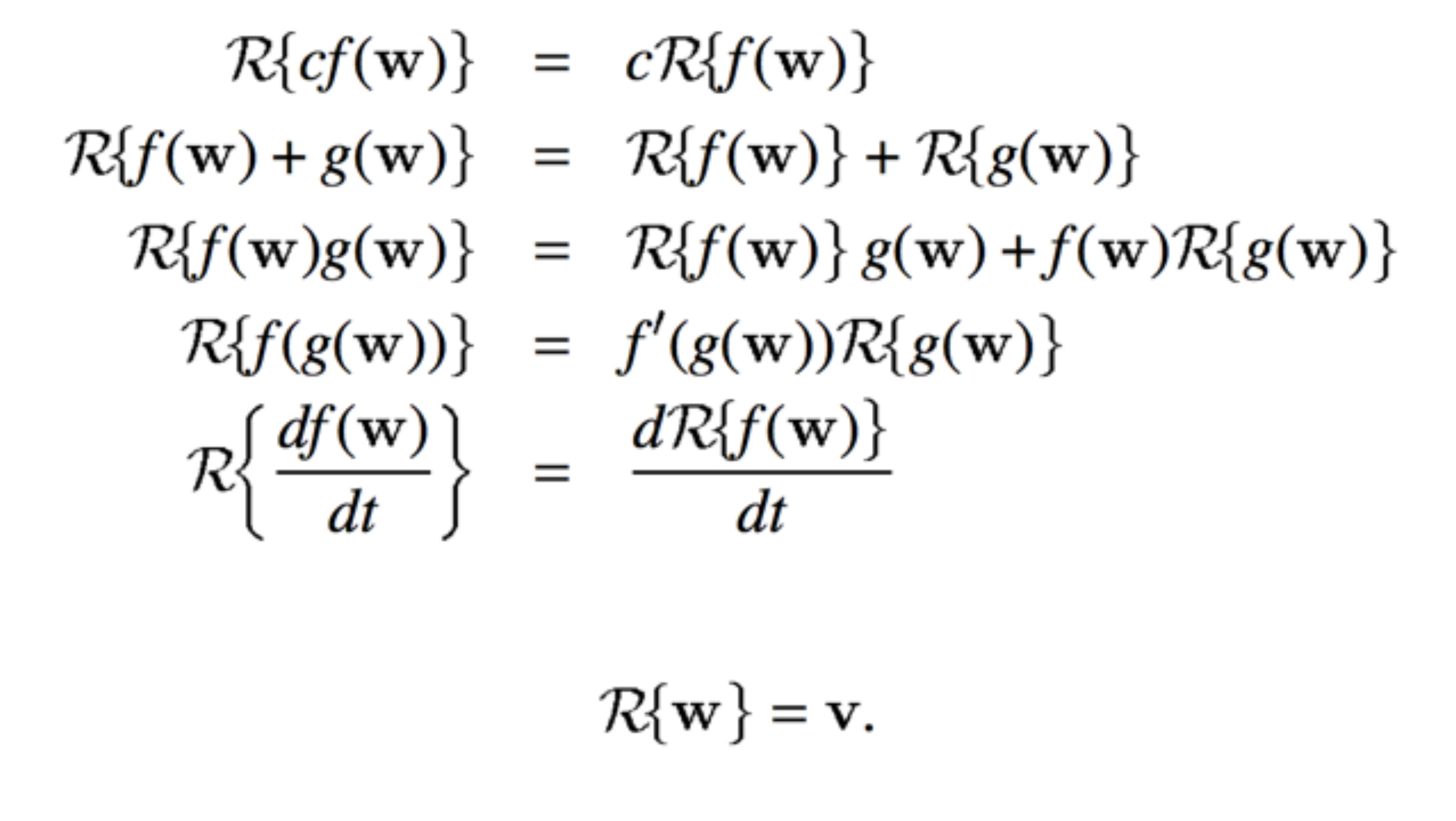


Algorithm 1 Equilibrated Gradient Descent

```
Require: Function $f(\theta)$ to minimize, learning rate $\epsilon$
  $\mathbf{D} \leftarrow 0$
  for $i = k \rightarrow K$ do
    $\mathbf{v} \sim \mathcal{N}(0, 1)$
    $\mathbf{D} \leftarrow \mathbf{D} + (\mathbf{H}\mathbf{v})^2$
    $\theta \leftarrow \theta - \epsilon \frac{\nabla f(\theta)}{\sqrt{\mathbf{D}/k + \lambda}}$
  end for
```

- D is an average leveraging the identity $||\mathbf{H}_{i,\cdot}||^2 = \mathrm{E}[(\mathbf{H}\mathbf{v})^2]$
- We can estimate the products Hv for the price of 2 gradients (Pearlmutter, 1994)

R-OPERATOR

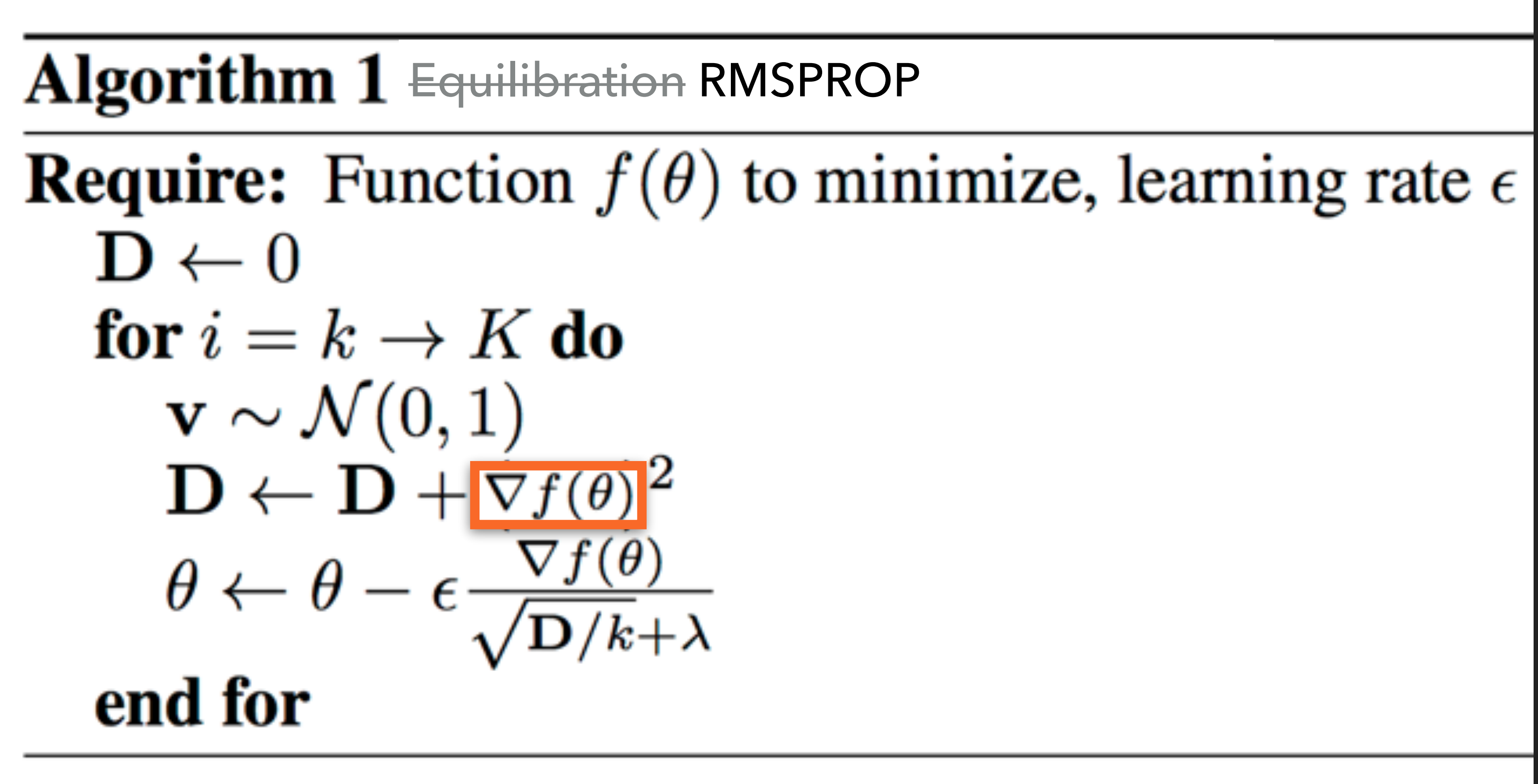


$$\begin{aligned}
\mathcal{R}\{cf(\mathbf{w})\} &= c\mathcal{R}\{f(\mathbf{w})\} \\
\mathcal{R}\{f(\mathbf{w})+g(\mathbf{w})\} &= \mathcal{R}\{f(\mathbf{w})\}+\mathcal{R}\{g(\mathbf{w})\} \\
\mathcal{R}\{f(\mathbf{w})g(\mathbf{w})\} &= \mathcal{R}\{f(\mathbf{w})\}\,g(\mathbf{w})+f(\mathbf{w})\mathcal{R}\{g(\mathbf{w})\} \\
\mathcal{R}\{f(g(\mathbf{w}))\} &= f'(g(\mathbf{w}))\mathcal{R}\{g(\mathbf{w})\} \\
\mathcal{R}\left\{\frac{df(\mathbf{w})}{dt}\right\} &= \frac{d\mathcal{R}\{f(\mathbf{w})\}}{dt}
\end{aligned}$$

$$\mathcal{R}\{\mathbf{w}\}=\mathbf{v}.$$

- The R-Operator is a set of rules to apply.
- These rule can be applied automatically, just like for differentiation.

IMPLEMENTATION RMSPROP (HINTON, 2014)



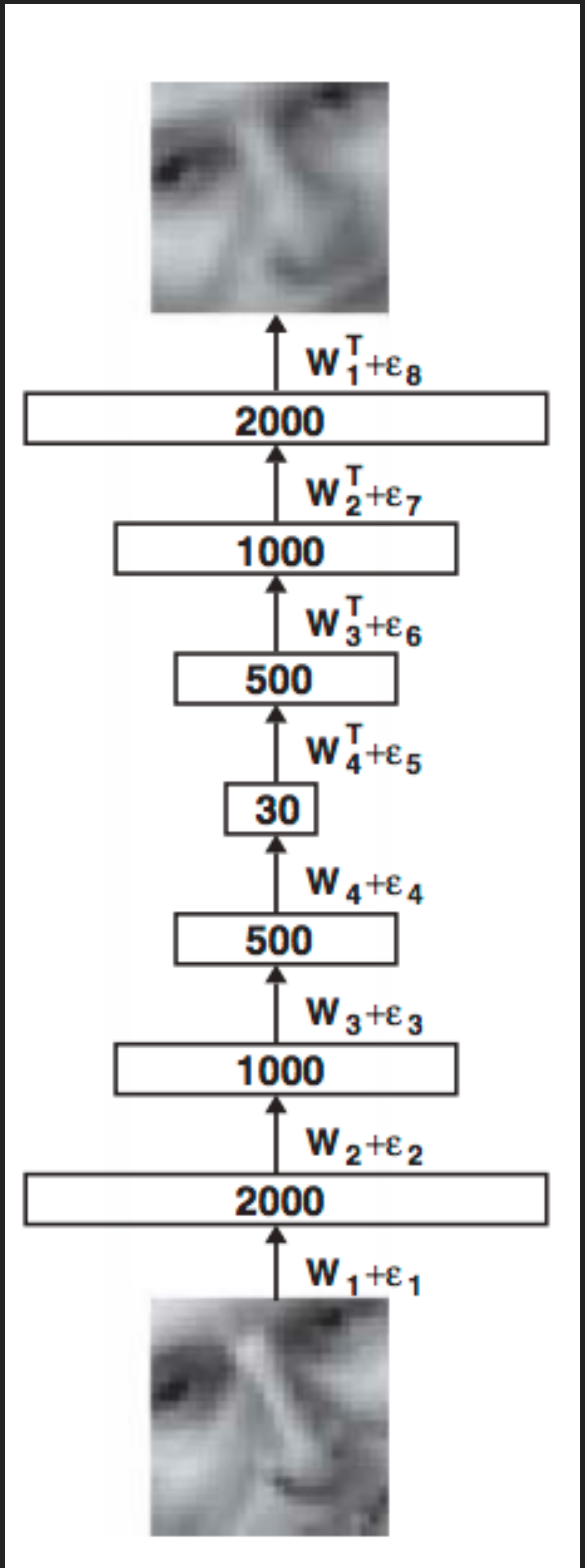
Algorithm 1 ~~Equilibration~~ RMSPROP

```
Require: Function $f(\theta)$ to minimize, learning rate $\epsilon$
  $\mathbf{D} \leftarrow 0$
  for $i = k \rightarrow K$ do
    $\mathbf{v} \sim \mathcal{N}(0, 1)$
    $\mathbf{D} \leftarrow \mathbf{D} + \boxed{\nabla f(\theta)}^2$
    $\theta \leftarrow \theta - \epsilon \frac{\nabla f(\theta)}{\sqrt{\mathbf{D}/k + \lambda}}$
  end for
```

- RMSPROP uses the approximation $\nabla f(\theta) \approx \mathbf{H}\Delta\theta$
- Then we recover a very biased form of equilibration with $\|\mathbf{H}_{i,\cdot}\|^2 = \mathrm{E}[(\mathbf{Hv})^2]$

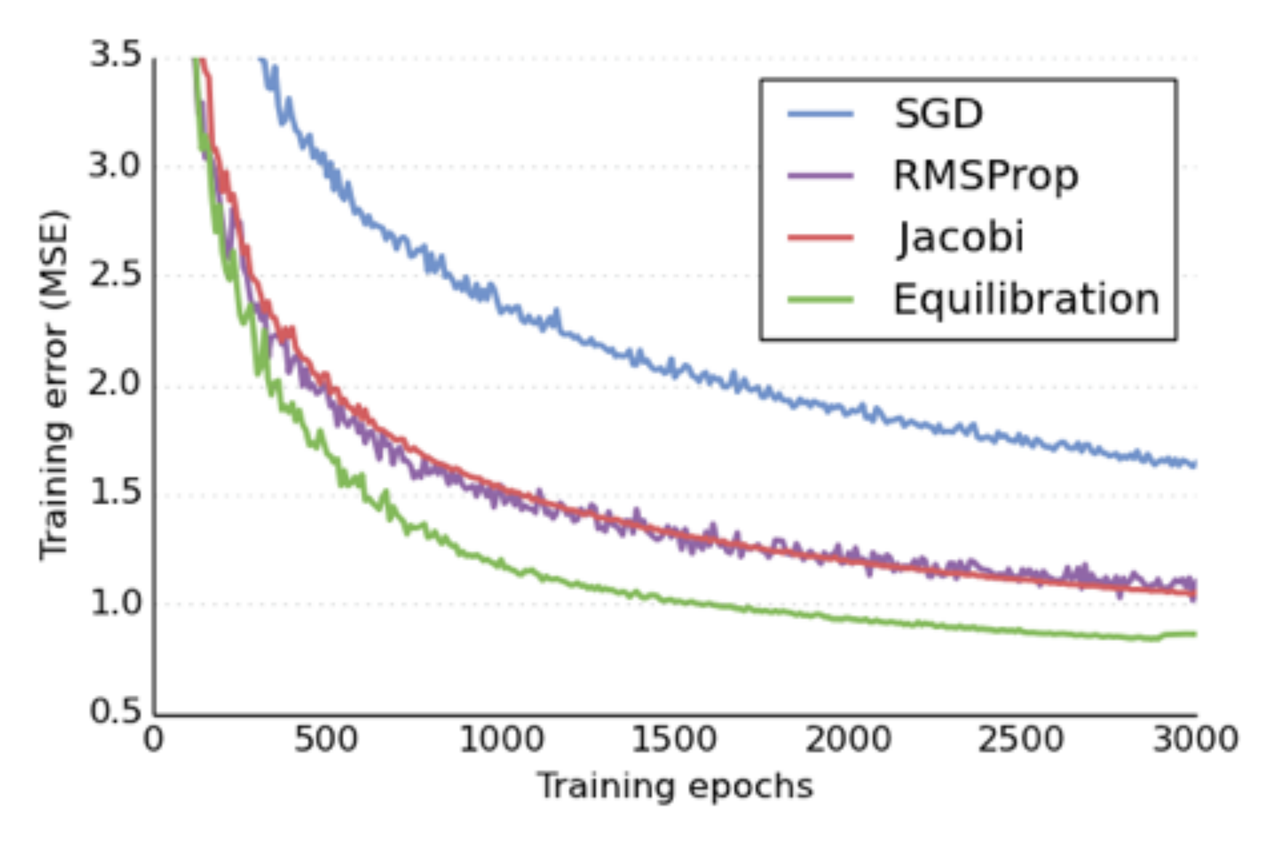
EXPERIMENTAL VALIDATION

- We compare RMSProp, Jacobi and equilibration on the task of training deep auto encoders following (Martens, 2010).
- We evaluate on MNIST and CURVES.
- The auto encoders have up to 10 layers and millions of parameters.

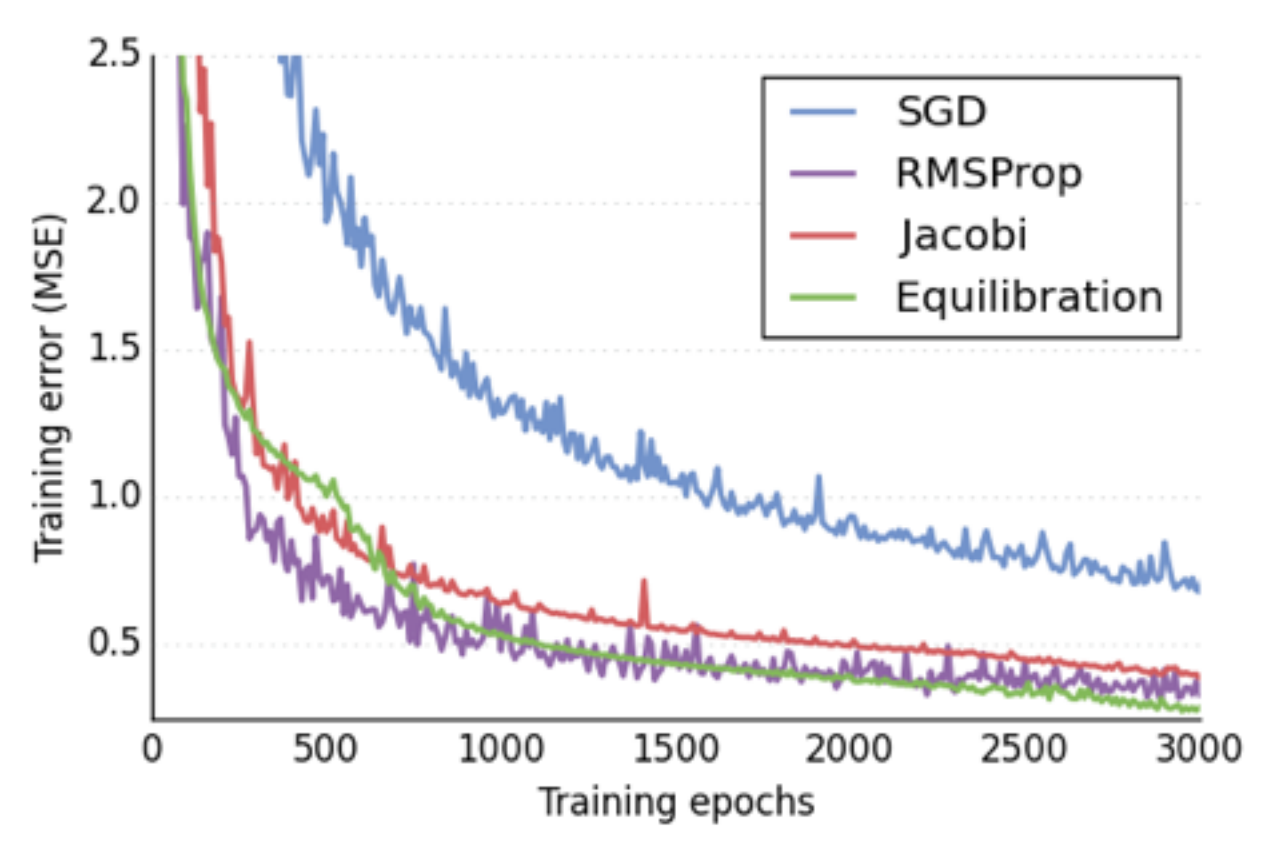


RESULTS

MNIST



CURVES



- All preconditioning methods perform better than simple SGD
- Equilibration performs better or at least as well as RMSProp
- Equilibration outperforms Jacobi

NEW DIRECTIONS

- Tensor methods (Janzamin et al, 2015)
- Graduated optimization (Hazan et al, 2015)
- Preconditioned Spectral Descent (Carlson et al, 2015)
- Stochastic Gradient Langevin Dynamics (Li et al, 2015)
- Debunking the myth of bad local minima is stimulating the field of non-convex optimization.

CONCLUSIONS

- High-dimensional loss surfaces do not suffer significantly from local minima.
- Non-convex optimization methods must appropriately handle negative curvature.
- RMSProp and equilibration can speed up SGD for non-convex problems by using the squared curvature.