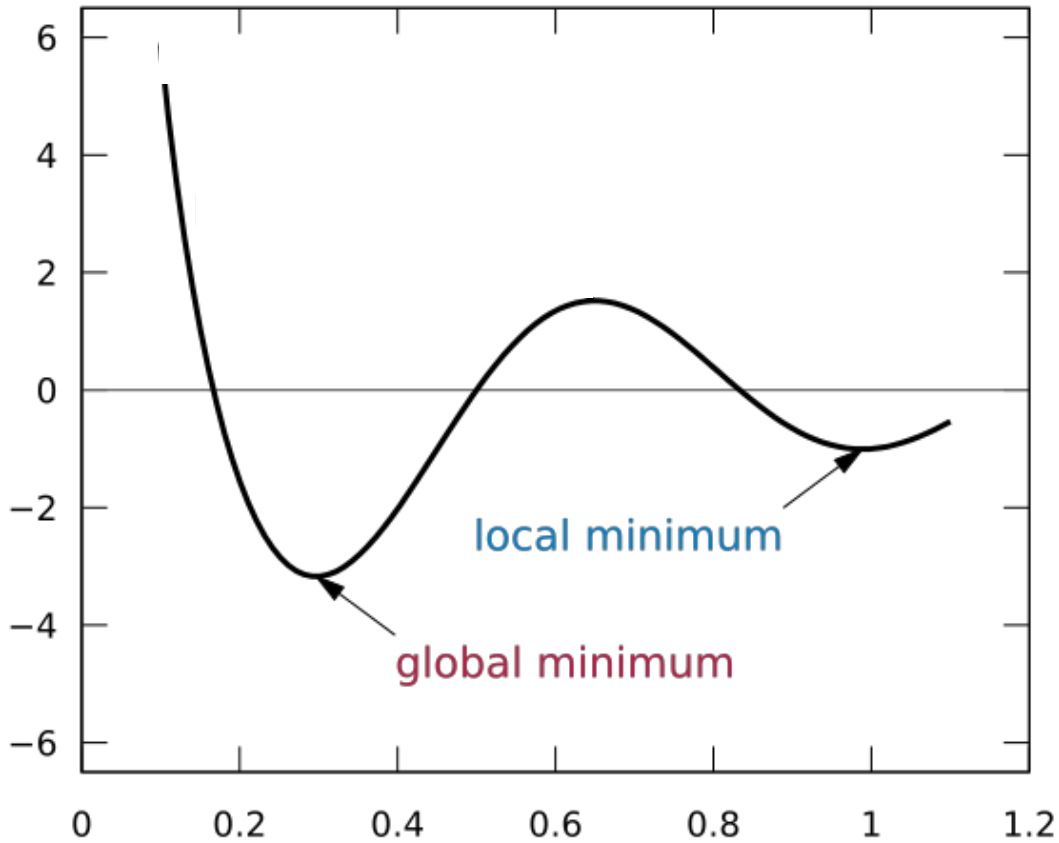


Convex Function vs. Nonconvex Function: A Little Bit Theory

Shusen Wang

Global Extremum vs. Local Extremum



Local Minimum of a function $f(\mathbf{w})$

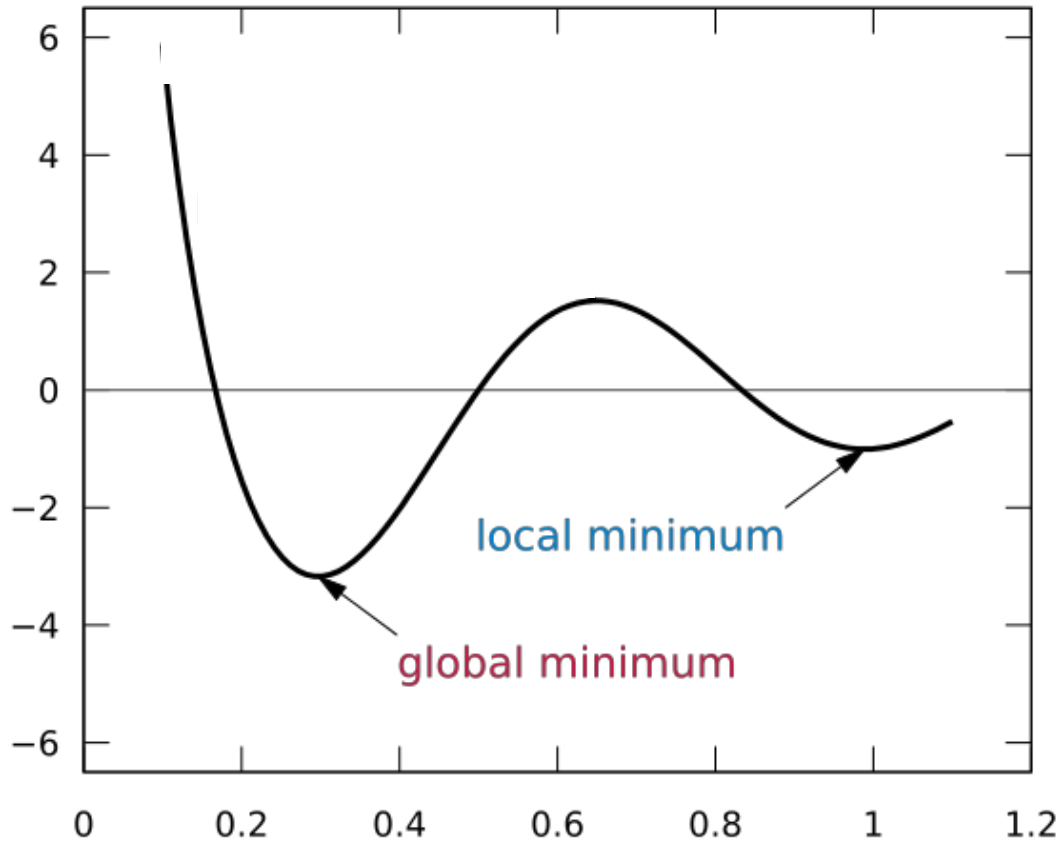
If $f(\mathbf{w}^*) \leq f(\mathbf{w})$ for all \mathbf{w} in a neighborhood of \mathbf{w}^* , then \mathbf{w}^* is a **local minimum** of f .

Global Minimum of a function $f(\mathbf{w})$

If $f(\mathbf{w}^*) \leq f(\mathbf{w})$ for all \mathbf{w} in the domain of f , then \mathbf{w}^* is a **global minimum** of f .

- A global minimum is a local minimum.
- Global minimum may not be unique.

Properties of Local Minimum



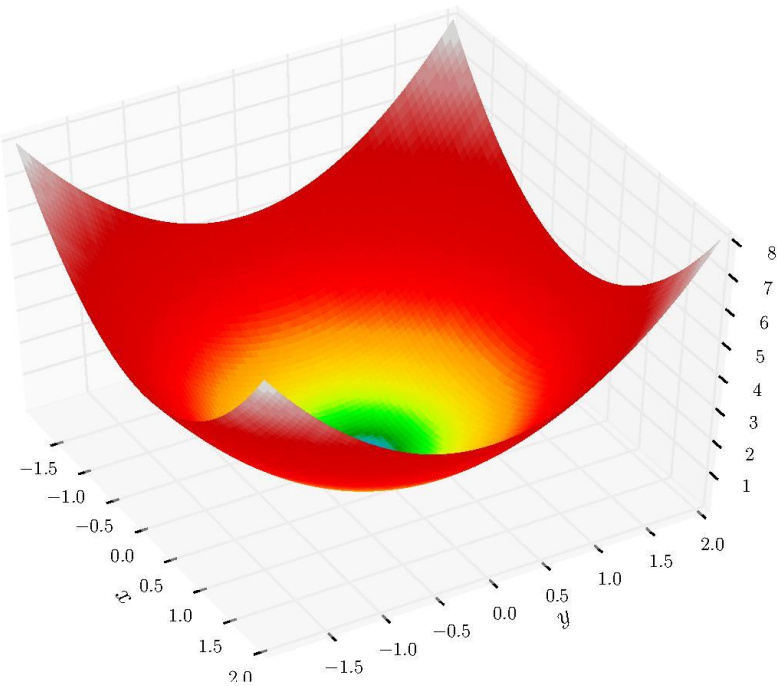
Assume f is defined on \mathbb{R}^d .

Properties of local minimum \mathbf{w}^* :

1. The gradient at \mathbf{w}^* , $\nabla f(\mathbf{w}^*) \in \mathbb{R}^d$, is all-zeros.
2. The Hessian matrix at \mathbf{w}^* , $\nabla^2 f(\mathbf{w}^*) \in \mathbb{R}^{d \times d}$, is positive semidefinite (i.e., all of its d eigenvalues are nonnegative.)

Convex Function

- **Convex function:** The line segment between any two points on the graph of the function lies above or on the graph

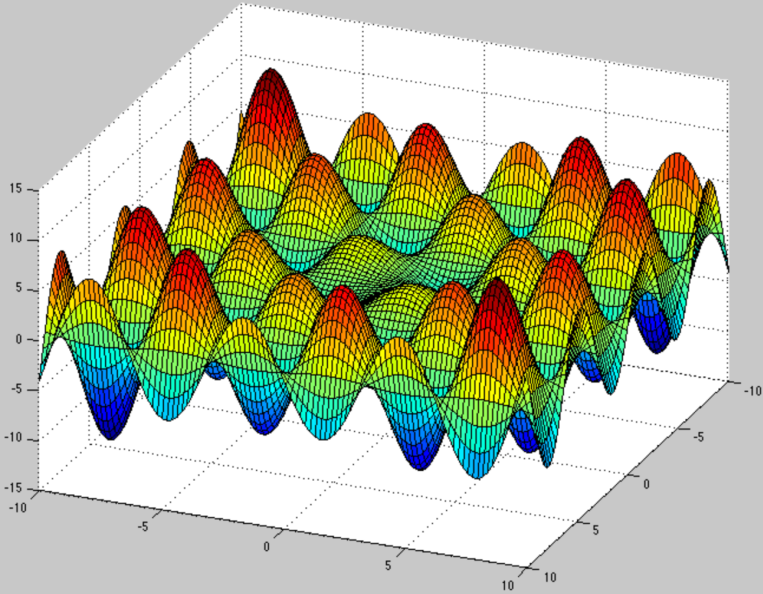


Graph of a convex function

Properties of a convex function f :

1. Local minimum = global minimum.
2. The Hessian matrix $\nabla^2 f(\mathbf{w})$ is positive semi-definite everywhere.
3. $\nabla f(\mathbf{w}^*) = \mathbf{0} \iff \mathbf{w}^*$ is a global minimum.

Nonconvex Function

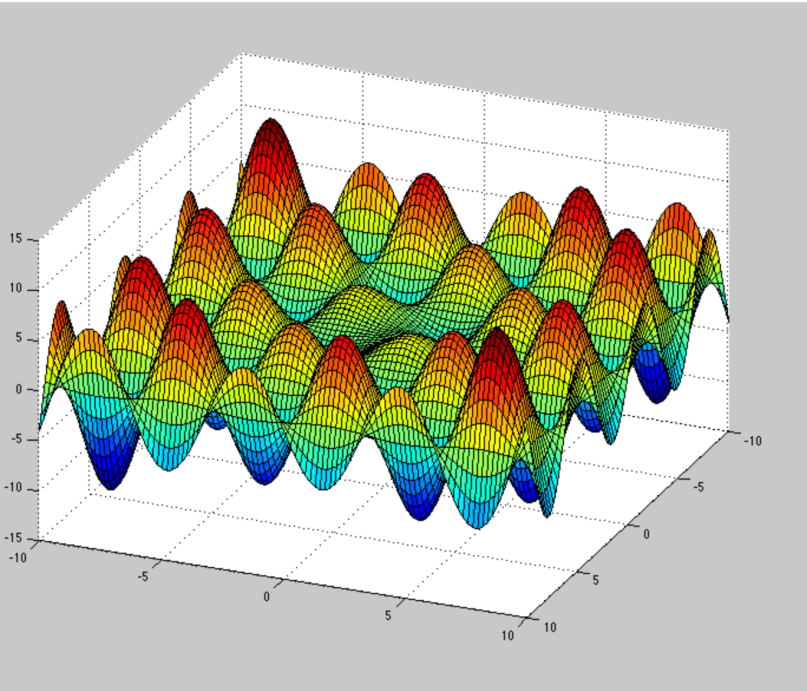


Graph of a nonconvex function

Properties:

1. Local minimum ~~is~~ global minimum.
2. The Hessian matrix $\nabla^2 f(\mathbf{w})$ is positive semi-definite ~~everywhere~~.
3. $\nabla f(\mathbf{w}^*) = \mathbf{0} \leftarrow \mathbf{w}^*$ is a global minimum.

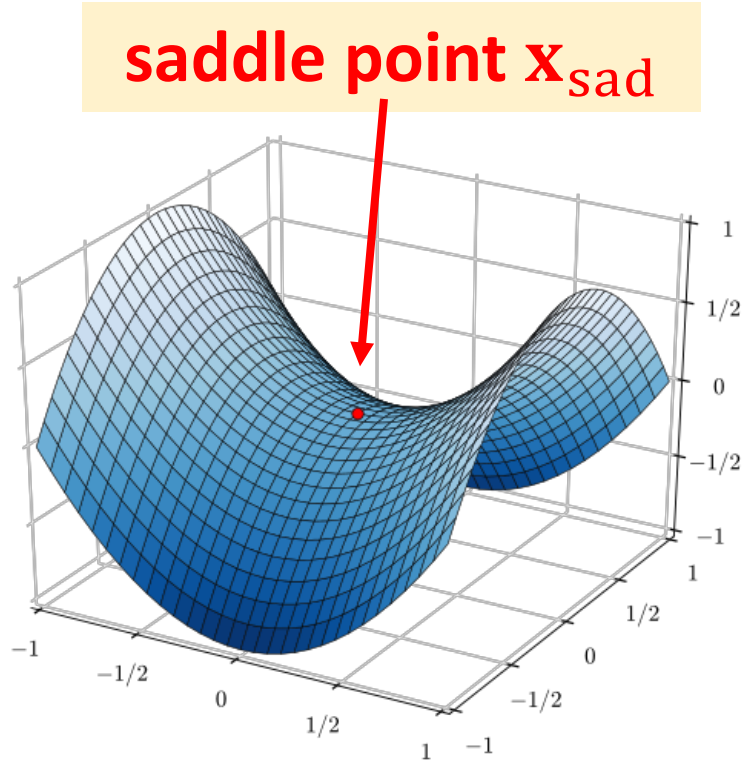
Global Minimum is Unlikely to Reach



- $\# \text{local minima} \gg \# \text{global minima}$.
- The final solution depends on the initialization.
- Reaching one of the global minima is very unlikely.

Graph of a nonconvex function

Saddle Point



Graph of a nonconvex function

Definition of saddle point:

1. The gradient of f at a saddle point is all-zeros: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
2. The Hessian matrix $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues..**

Saddle Point vs. Local Minimum

saddle point \mathbf{w}_{sad}

- Gradient: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues**.

local minimum \mathbf{w}^*

- Gradient: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}^*)$ does **not** have negative eigenvalues.

Saddle Point vs. Local Minimum

saddle point \mathbf{w}_{sad}

- Gradient: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues**.

local minimum \mathbf{w}^*

- Gradient: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}^*)$ does **not** have negative eigenvalues.

- Full gradient descent stops at either a **saddle point** or a **local minimum**.

Saddle Point vs. Local Minimum

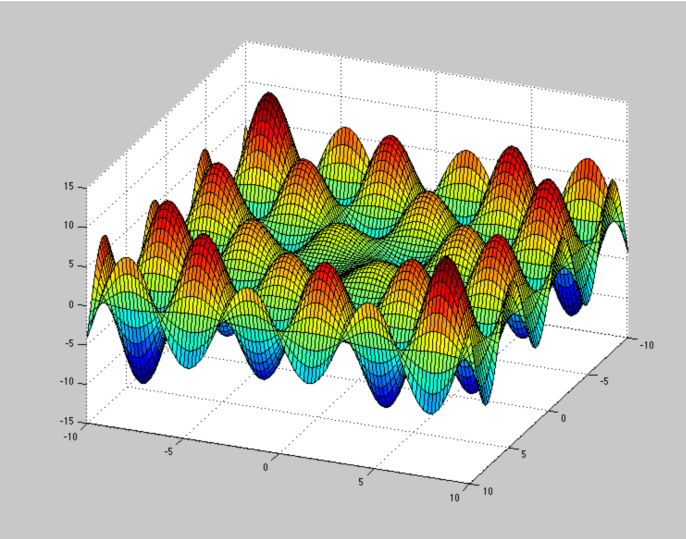
saddle point \mathbf{w}_{sad}

- Gradient: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues**.

local minimum \mathbf{w}^*

- Gradient: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}^*)$ does **not** have negative eigenvalues.

- Full gradient descent stops at either a **saddle point** or a **local minimum**.



- In 2D, **#saddle points** and **#local minimum** are comparable.
- It is not true in high-dim.

Saddle Point vs. Local Minimum

saddle point \mathbf{w}_{sad}

- Gradient: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues**.

local minimum \mathbf{w}^*

- Gradient: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}^*)$ does **not** have negative eigenvalues.

- Full gradient descent stops at either a saddle point or a local minimum.
- In high dim, #saddle points is much greater than #local minima.
 - The Hessian has d eigenvalues, each of which can be positive or negative.
 - $\Rightarrow 2^d$ combinations of positive and negative eigenvalues.
 - One out of the 2^d combinations corresponds to local minima.
 - $2^d - 2$ combinations corresponds to saddle points.

Saddle Point vs. Local Minimum

saddle point \mathbf{w}_{sad}

- Gradient: $\nabla f(\mathbf{w}_{\text{sad}}) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}_{\text{sad}})$ has **both positive and negative eigenvalues**.

local minimum \mathbf{w}^*

- Gradient: $\nabla f(\mathbf{w}^*) = \mathbf{0}$.
- Hessian: $\nabla^2 f(\mathbf{w}^*)$ does **not** have negative eigenvalues.

- Full gradient descent stops at either a **saddle point** or a **local minimum**.
- In high dim, the number of **saddle points** is much larger than **local minima**.
- If a neural net is optimized by the full gradient descent, it will converge to a **saddle point**.

Be Careful When Optimizing a Nonconvex Function

Be careful about the initialization!

- Bad initialization results in convergence to bad regions.
 - Because of the nonconvexity, global minimum cannot be attained.

Be Careful When Optimizing a Nonconvex Function

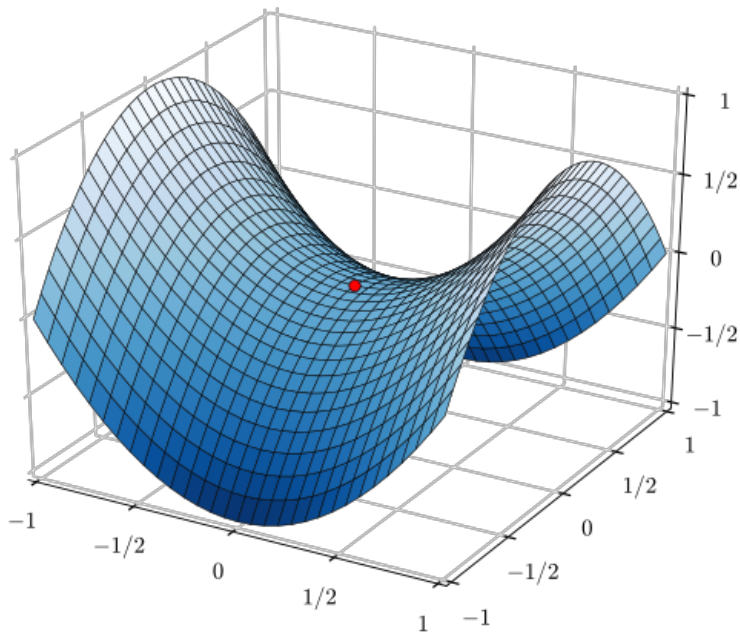
Be careful about the initialization!

- Bad initialization results in convergence to bad regions.
 - Because of the nonconvexity, global minimum cannot be attained.
- Rule of thumb :
 - The trainable parameters (e.g., the filters of ConvNet) are **randomly** initialized **with proper scaling**.
 - Bad scaling leads to terrible results.
 - All-zero and all-one initializations are bad ideas.
 - Pretrained parameters can be very good initialization.

Be Careful When Optimizing a Nonconvex Function

Be careful about the initialization!

Be careful about the optimization algorithm!



- Full gradient descent will be stuck in a saddle point.
 - Because the gradient is near zero when approaching the saddle point.
- Stochastic gradient descent (SGD) can escape the saddle points.
 - Because it is random and noisy.

Be Careful When Optimizing a Nonconvex Function

Be careful about the initialization!

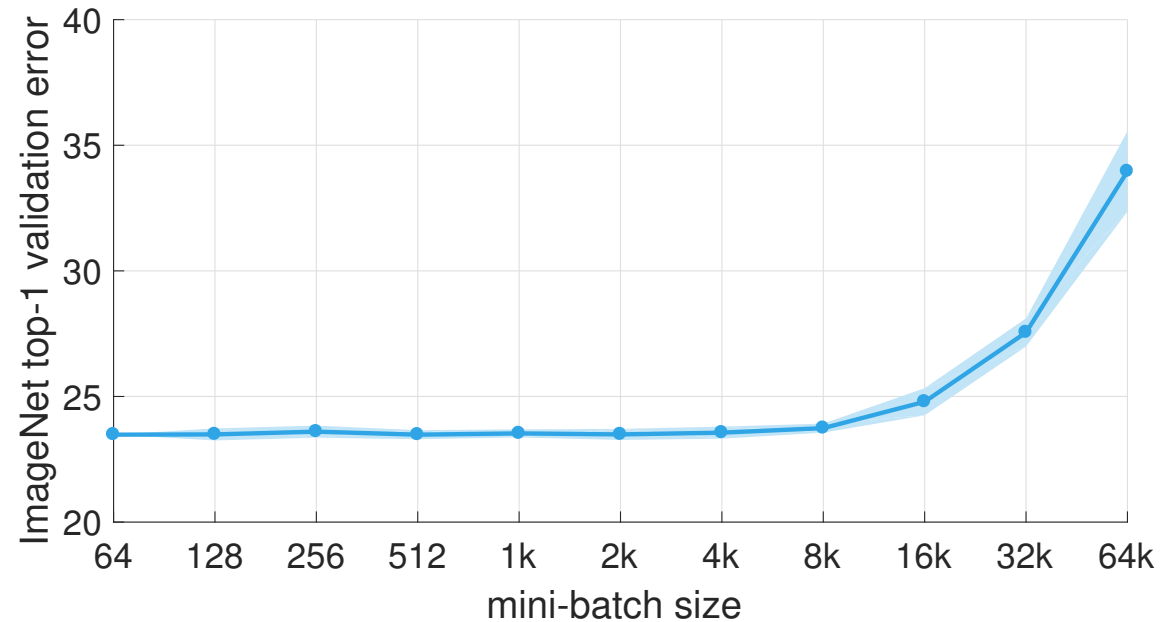
Be careful about the optimization algorithm!

Be careful about the batch size!

- For parallel computing with multiple GPUs, larger batch size → lower per-epoch runtime.
- Large batch size, e.g., $10K$, may result in bad generalization.

... More about the Batch Size

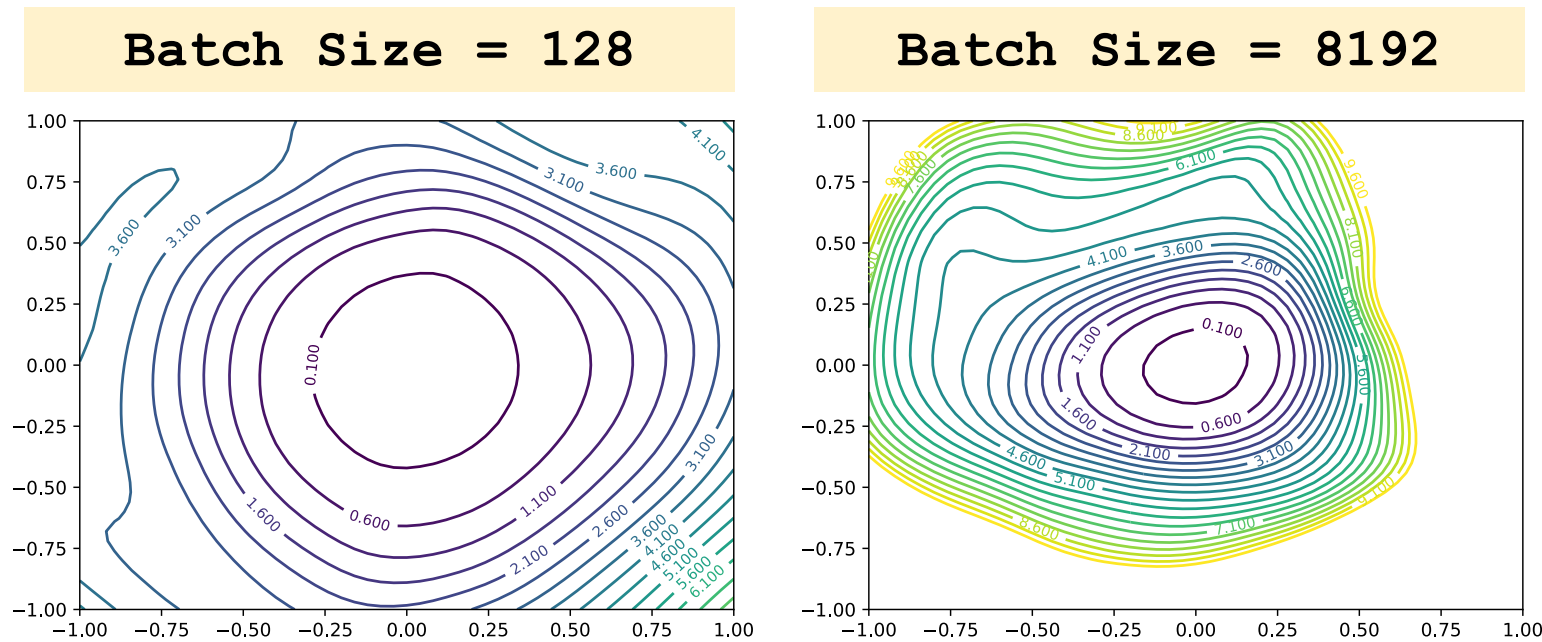
- Batch size larger than 8K results in poor generalization.
- Large batch size is good for time-efficiency.
- Lots of tricks are required in *large batch training*.



The figure is from the paper “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour”

... More about the Batch Size

- Researchers' conjecture:
 - Small batch size → flat local minima; Big batch size → shape local minima.
 - Flat local minima generalizes better (on the test set).



The figure is from paper <https://arxiv.org/abs/1712.09913>

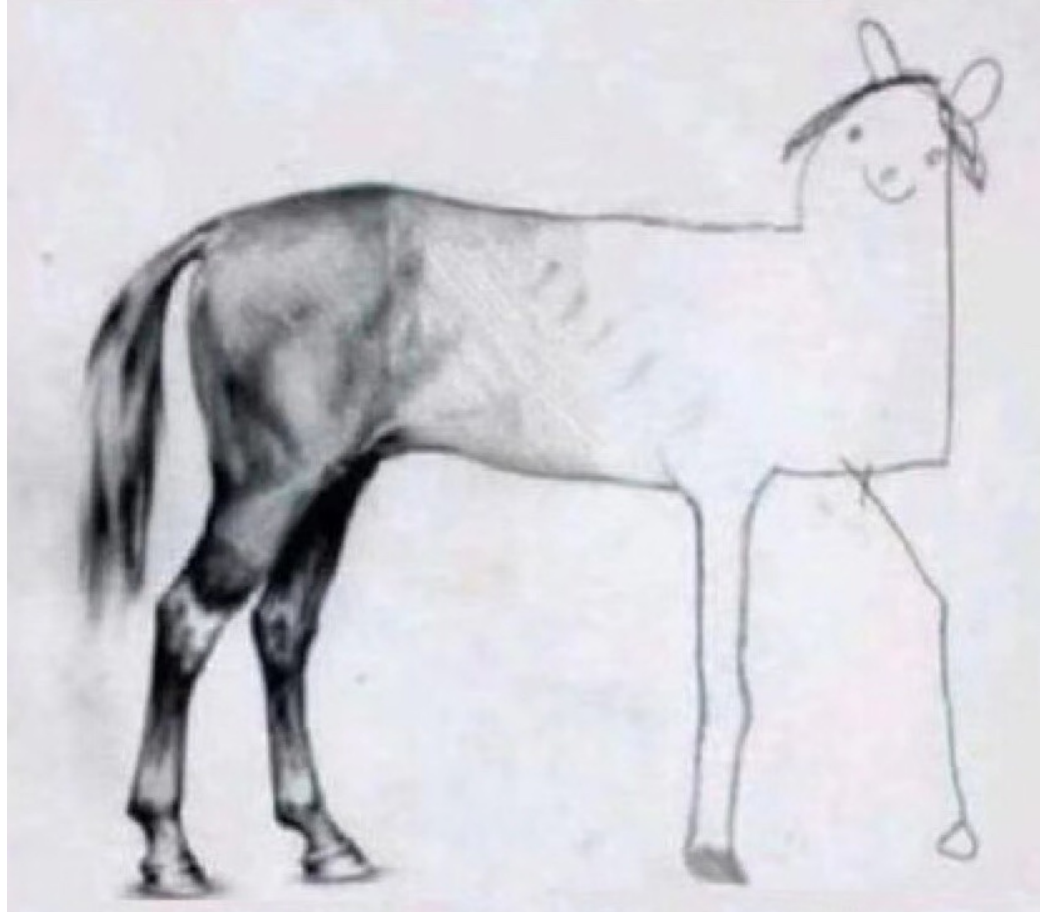
... More about the Batch Size

- There are papers supportive of small batch training, e.g., <https://arxiv.org/pdf/1804.07612.pdf>

The presented results confirm that using small batch sizes achieves the best training stability and generalization performance, for a given computational cost, across a wide range of experiments. In all cases the best results have been obtained with batch sizes $m = 32$ or smaller, often as small as $m = 2$ or $m = 4$.

Do Not Believe Deep Learning Theories Blindly

Empirical study



Explanations

Summary

- #global minima \ll #local minima \ll #saddle points.
- Full gradient descent converges to a saddle point.
- SGD converges to a local minimum.

Summary

- #global minima \ll #local minima \ll #saddle points.
- Full gradient descent converges to a saddle point.
- SGD converges to a local minimum.
- Initialization is crucial.
 - Proper scaling.
 - Pretrain.

Summary

- #global minima \ll #local minima \ll #saddle points.
- Full gradient descent converges to a saddle point.
- SGD converges to a local minimum.
- Initialization is crucial.
 - Proper scaling.
 - Pretrain.
- Batch size affects time efficiency and generalization.