



Peter Bartlett



Matus Telgarsky



Bin Yu

Reimagining Gradient Descent

Large Stepsize, Oscillation, Acceleration

Jingfeng Wu

Berkeley
UNIVERSITY OF CALIFORNIA

Gradient Descent

$$w_+ = w - \eta \cdot \nabla L(w)$$

stepsize / learning rate

How to imagine GD?



Cauchy, 1847

Gradient Descent

$$w_+ = w - \eta \cdot \nabla L(w)$$

stepsize / learning rate

How to imagine GD?

```
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
```



Cauchy, 1847

Gradient Descent

$$w_+ = w - \eta \cdot \nabla L(w)$$

stepsize / learning rate

How to imagine GD?

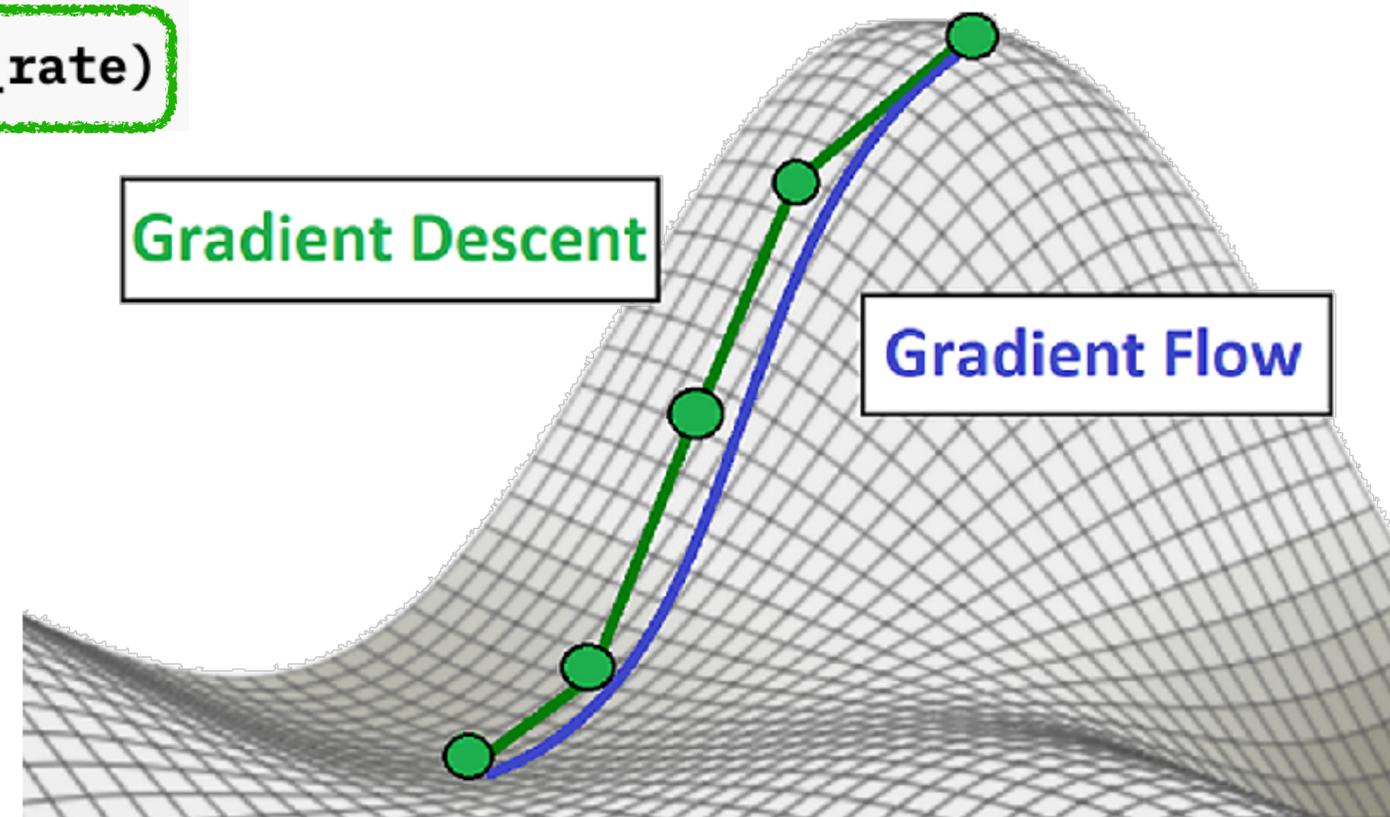
```
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
```

$$dw = - \nabla L(w) dt$$

picture credit to “off the convex path”
<http://www.offconvex.org/2022/01/06/gf-gd/>



Cauchy, 1847



1. data x , label y , weight w

2. compute loss

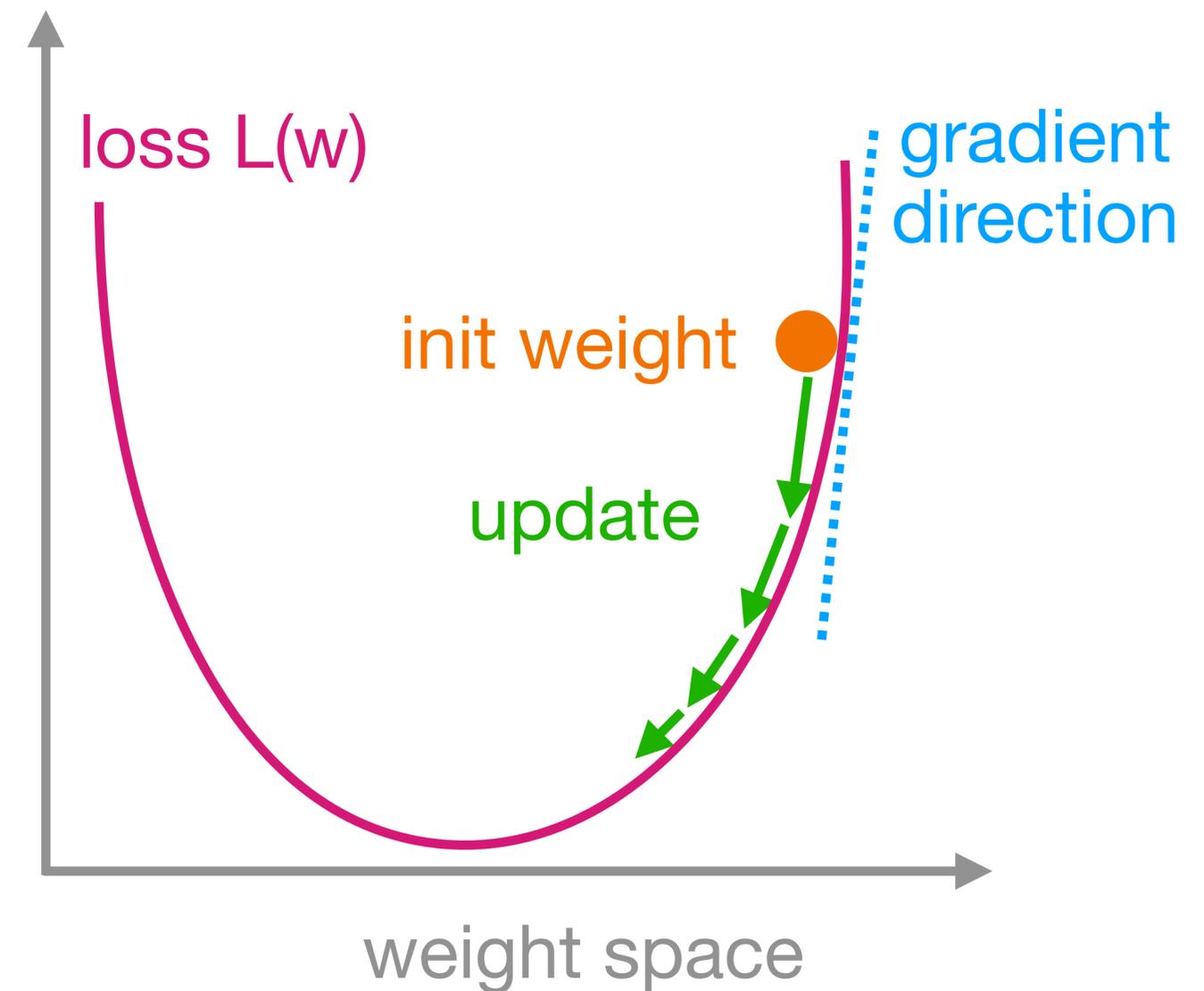
$$L(w) := \text{distance}(f_x(w), y)$$

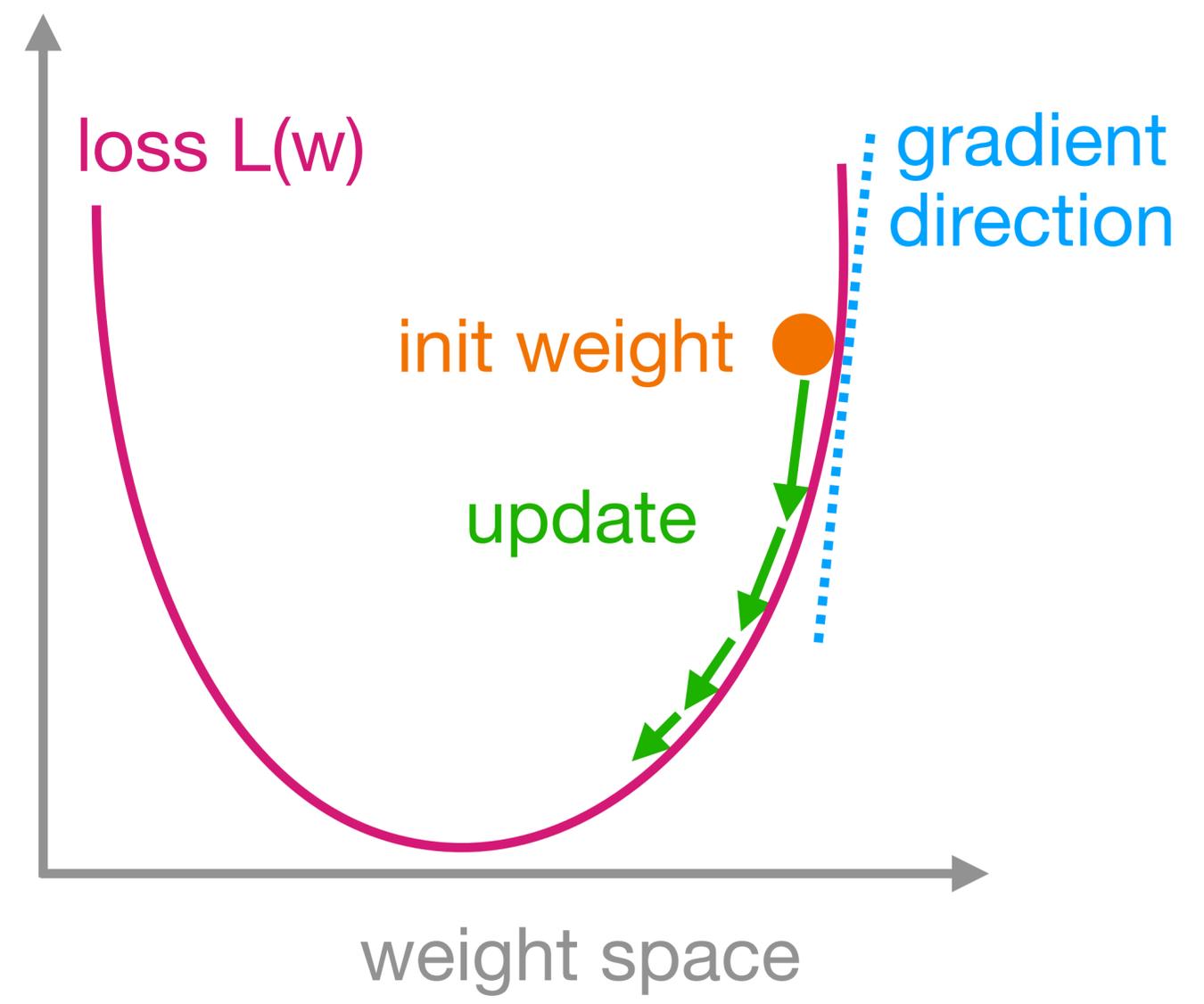
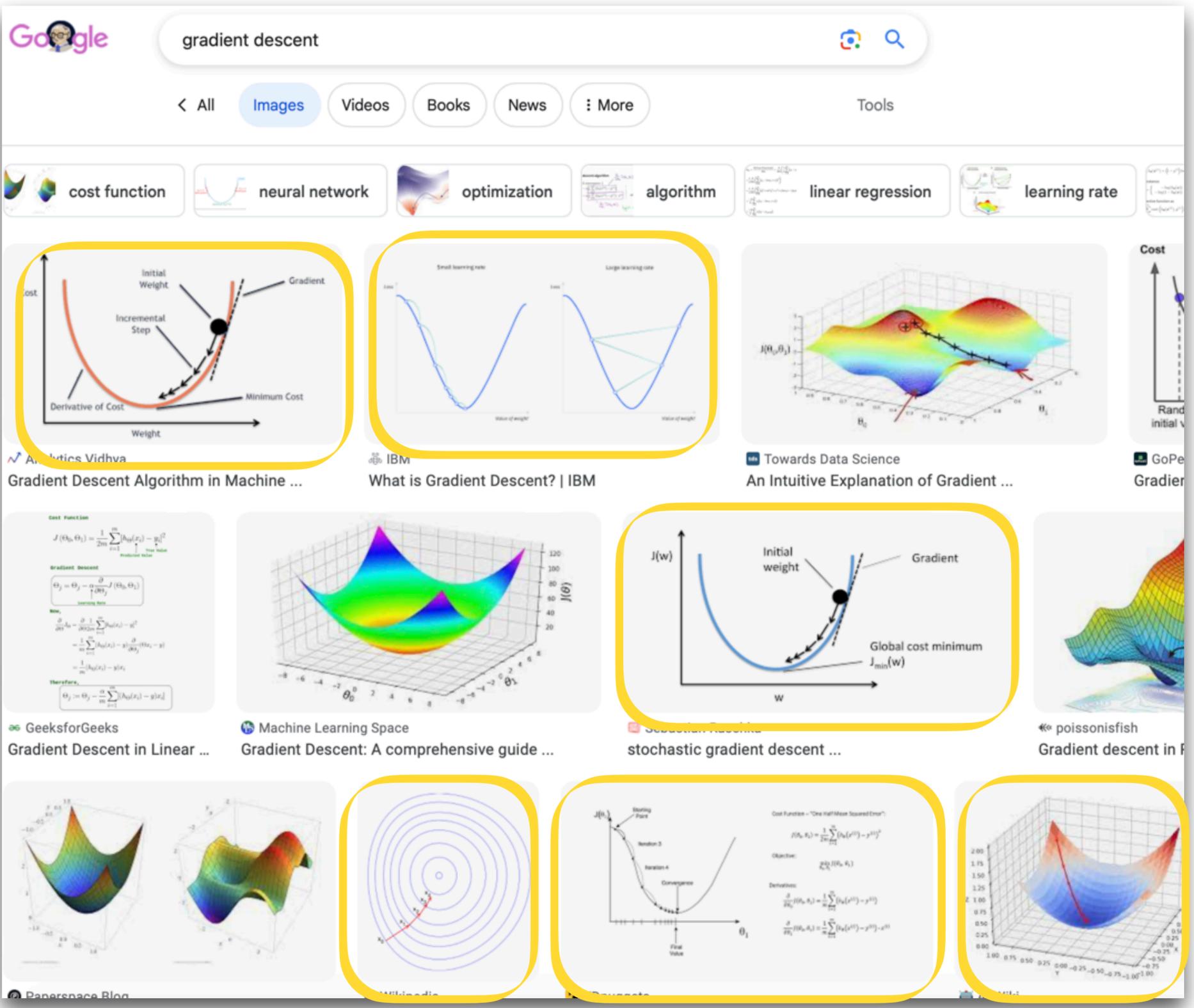
3. compute gradient

4. update w along the $-\nabla L(w)$

5. repeating 1 ~ 4

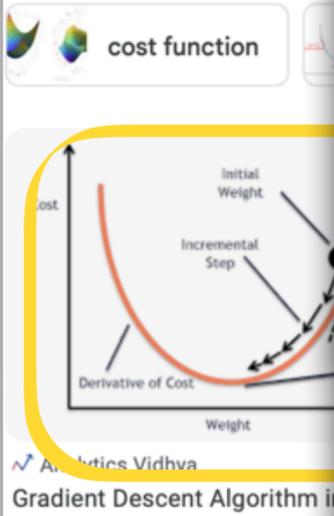
1. data x , label y , weight w
2. compute loss
 $L(w) := \text{distance}(f_x(w), y)$
3. compute gradient
4. update w along the $-\nabla L(w)$
5. repeating 1 ~ 4



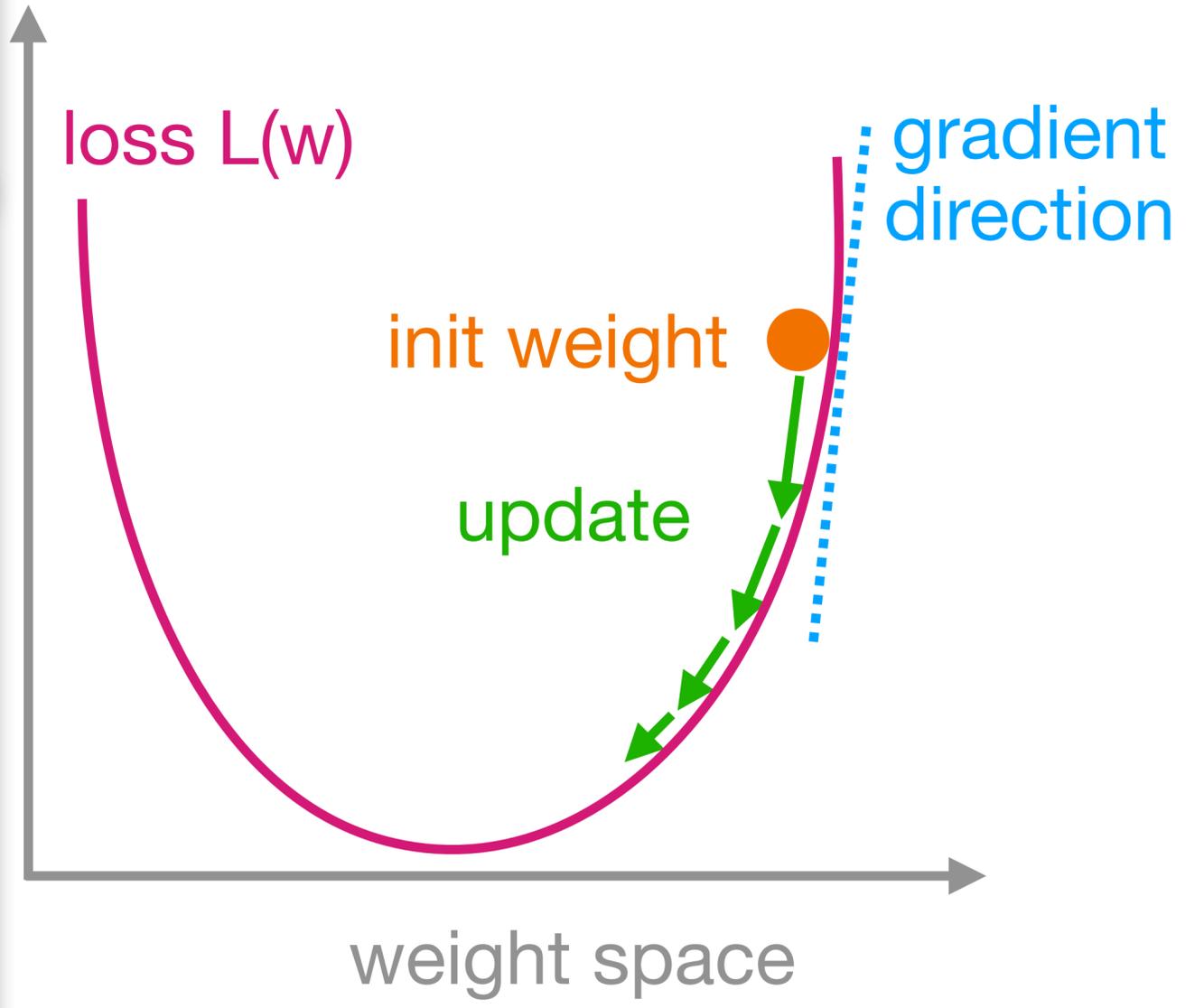
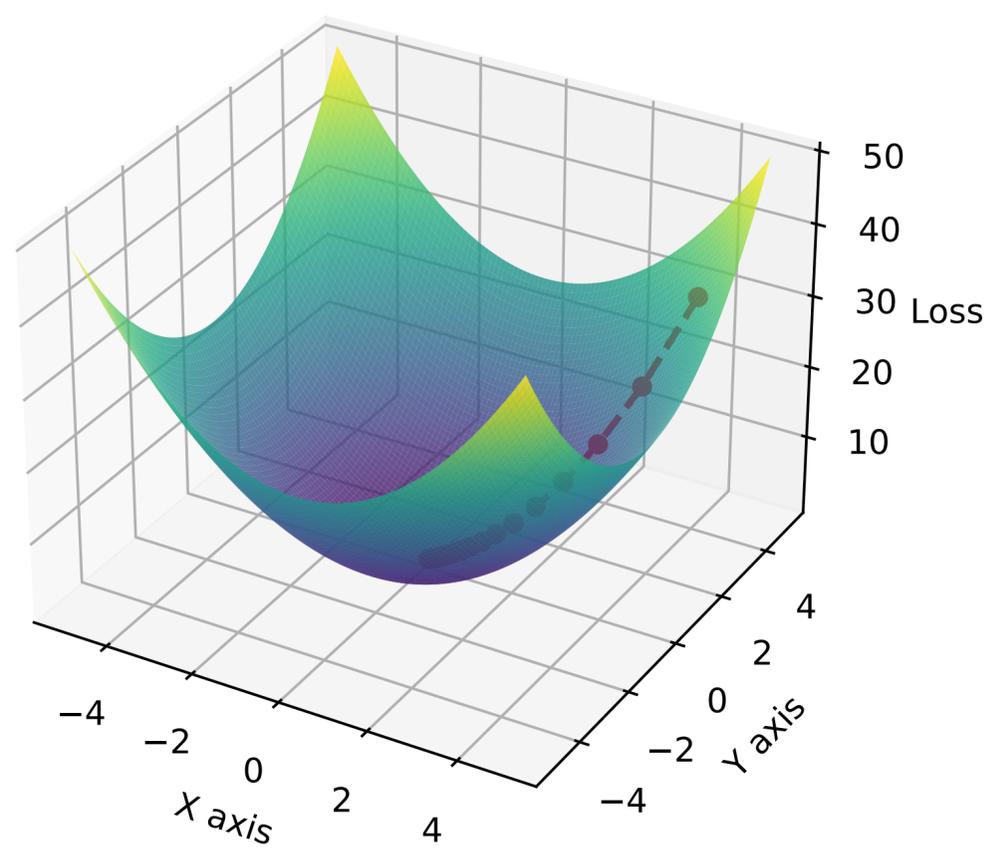


You
how to image gradient descent?

ChatGPT
If you're asking how to visualize or understand the concept of gradient descent in the context of image processing or more broadly, here are several ways to approach it:



3D Visualization of Gradient Descent



Cost Function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i)^2$$

Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Min.

$$\frac{\partial}{\partial \theta_0} J = \frac{\partial}{\partial \theta_0} \left[\frac{1}{2m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i)^2 \right]$$

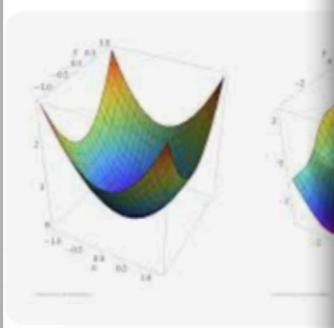
$$= \frac{1}{m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i) \frac{\partial}{\partial \theta_0} (h_{\theta_0, \theta_1}(x_i) - y_i)$$

$$= \frac{1}{m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i) x_i$$

Therefore,

$$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

GeeksforGeeks
Gradient Descent in Linear ...



Google gradient descent

Images Videos Books News More Tools

You how to image gradient descent?

ChatGPT If you're asking how to visualize or understand the concept of gradient descent in the context of image processing or more broadly, here are several ways to approach it:

3D Visualization of Gradient Descent

Cost Function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i)^2$$

Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Min.

$$\frac{\partial}{\partial \theta_0} J = \frac{\partial}{\partial \theta_0} \left(\frac{1}{2m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i)^2 \right)$$

$$= \frac{1}{m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i) \frac{\partial}{\partial \theta_0} (h_{\theta_0, \theta_1}(x_i) - y_i)$$

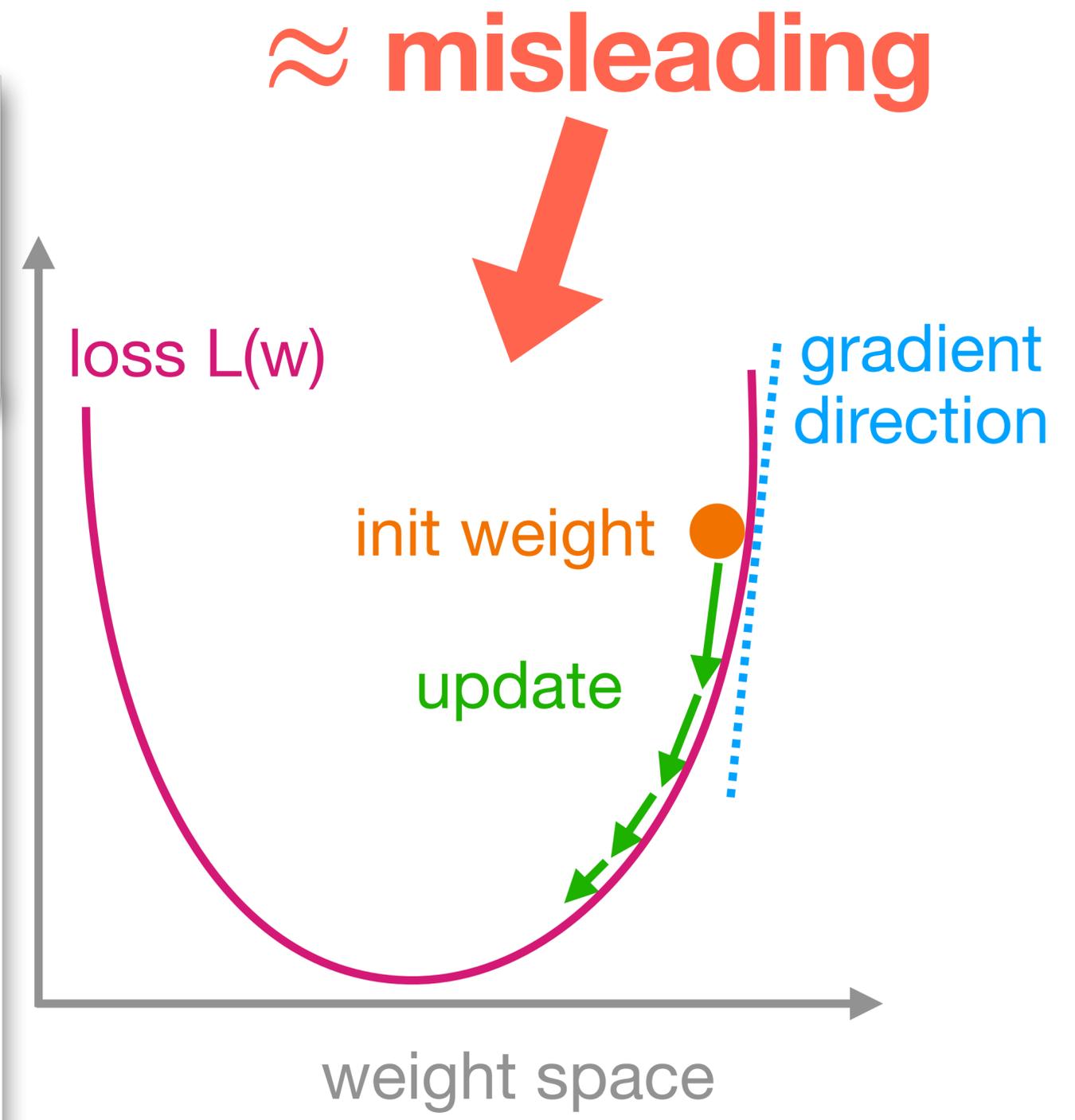
$$= \frac{1}{m} \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i) x_i$$

Therefore,

$$\theta_0 = \theta_0 - \alpha \sum_{i=1}^m (h_{\theta_0, \theta_1}(x_i) - y_i) x_i$$

GeeksforGeeks Gradient Descent in Linear ...

Paperspace Blog



$$L(w_+) = L(w - \eta \nabla L(w))$$

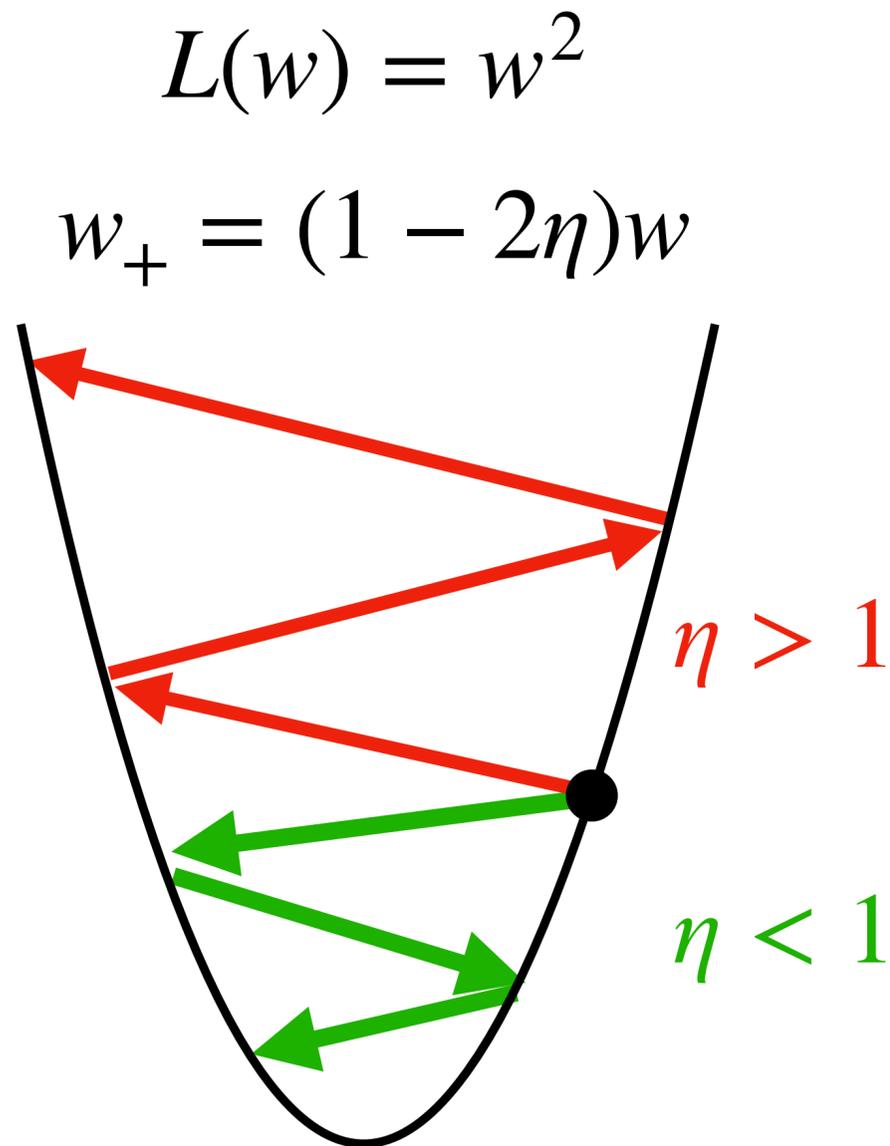
$$= L(w) - \eta \|\nabla L(w)\|^2 + \frac{\eta^2}{2} \nabla L(w)^\top \nabla^2 L(w) \nabla L(w) + \dots$$

$$\leq L(w) - \left(1 - \frac{\eta}{2} \|\nabla^2 L(w)\|_2\right) \cdot \eta \|\nabla L(w)\|^2 + \dots$$

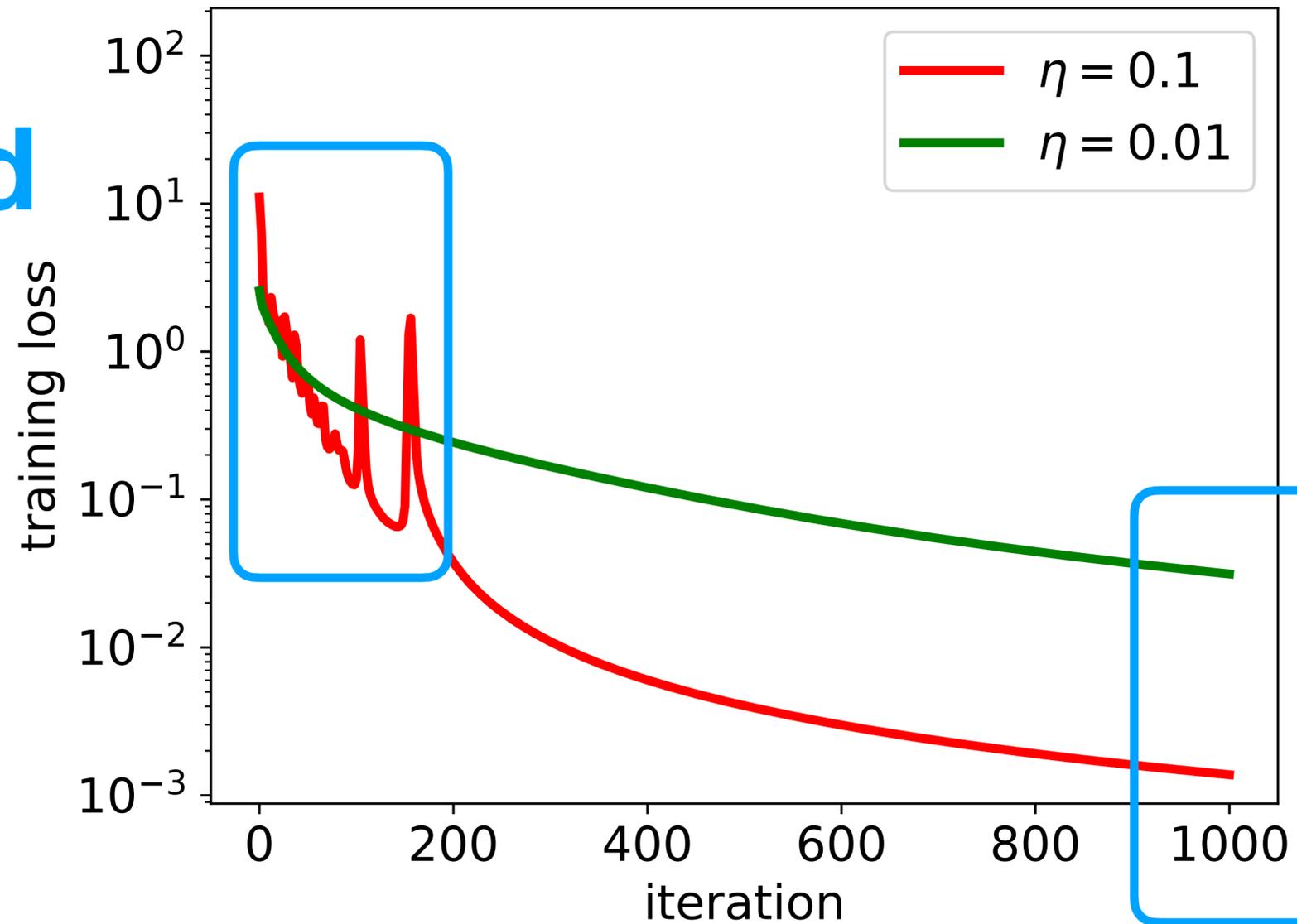
[descent lemma]

For **small** η , $L(w_t)$ decreases **monotonically**

For **large** η , $L(w_t)$ **diverges** for quadratics



**spikes
unexplained**



**larger η
smaller loss**

3-layer net + 1,000 samples from MNIST+ GD with const-stepsizes

“edge of stability”

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., & Talwalkar, A.
Gradient descent on neural networks typically occurs at the edge of stability. ICLR 2021

$$w_{t+1} = w_t - \eta \nabla L(w_t)$$

- Bounded feature, binary label

10 classes -> 2 classes

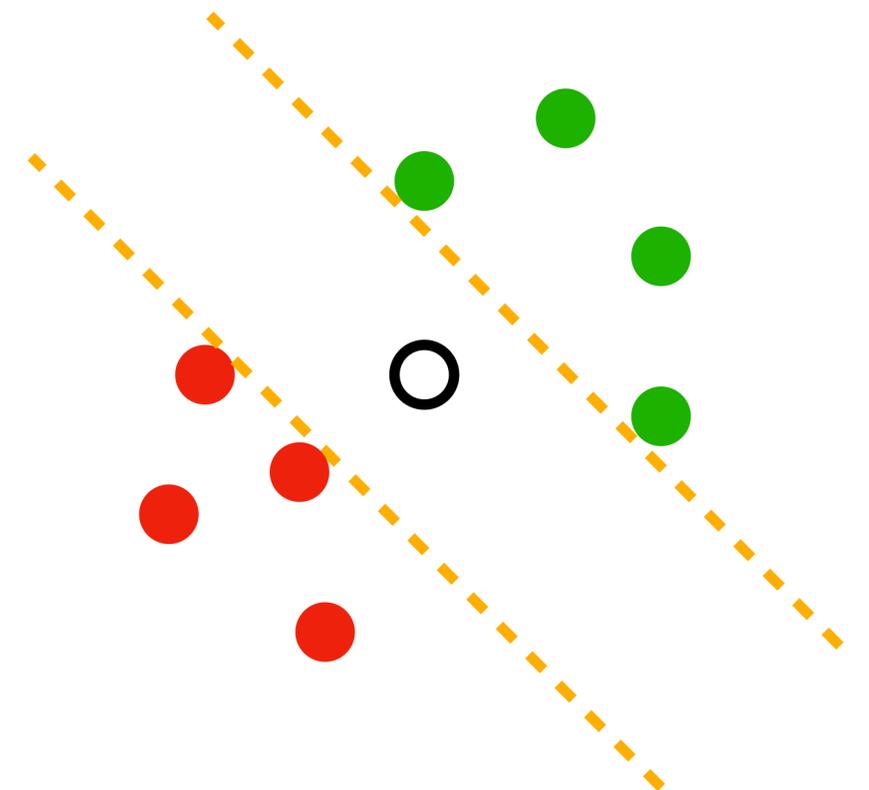
$$x_i \in \mathbb{R}^d, \|x_i\| \leq 1, y_i \in \{\pm 1\}, i = 1, \dots, n$$

- Linear classification

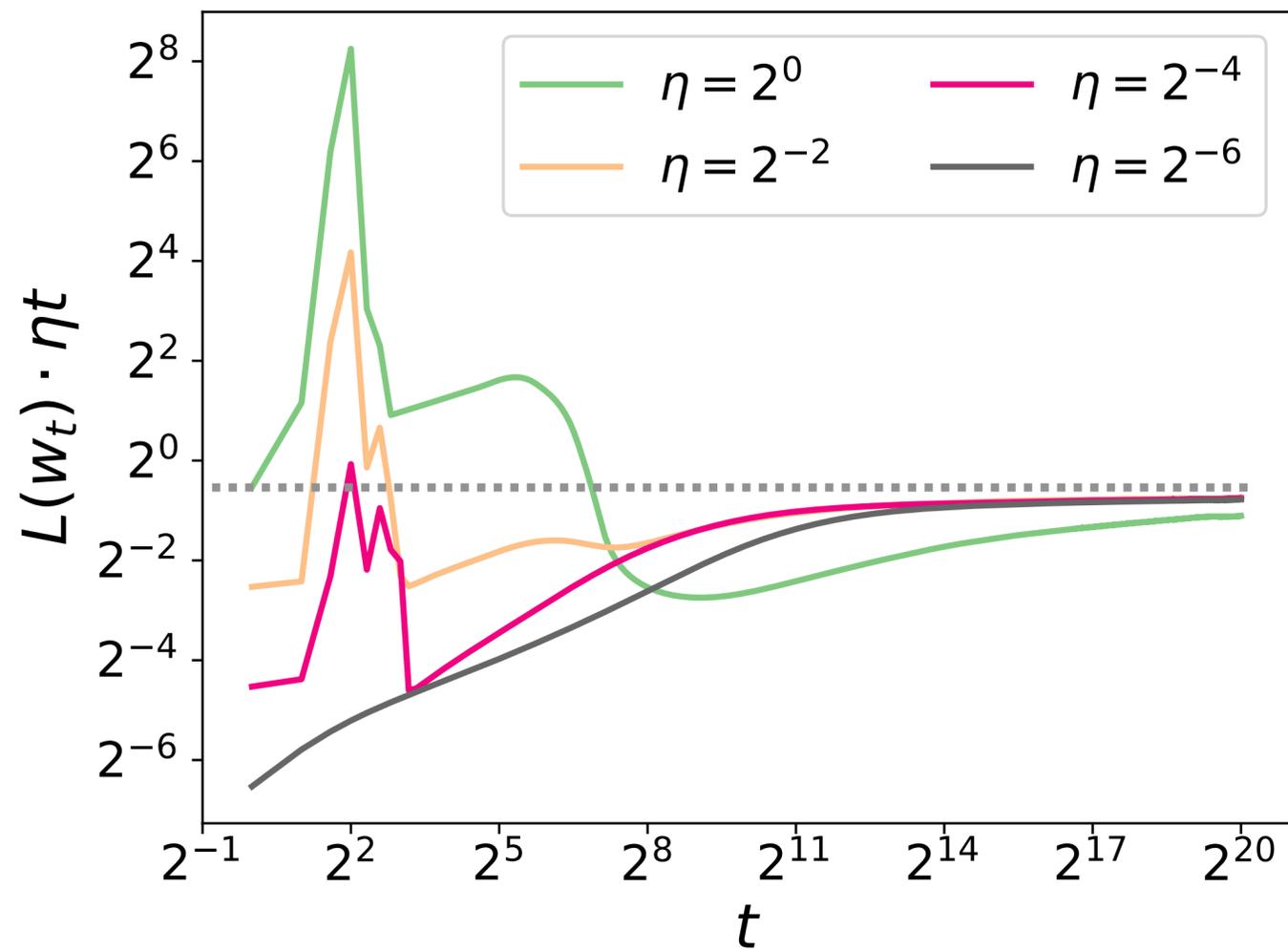
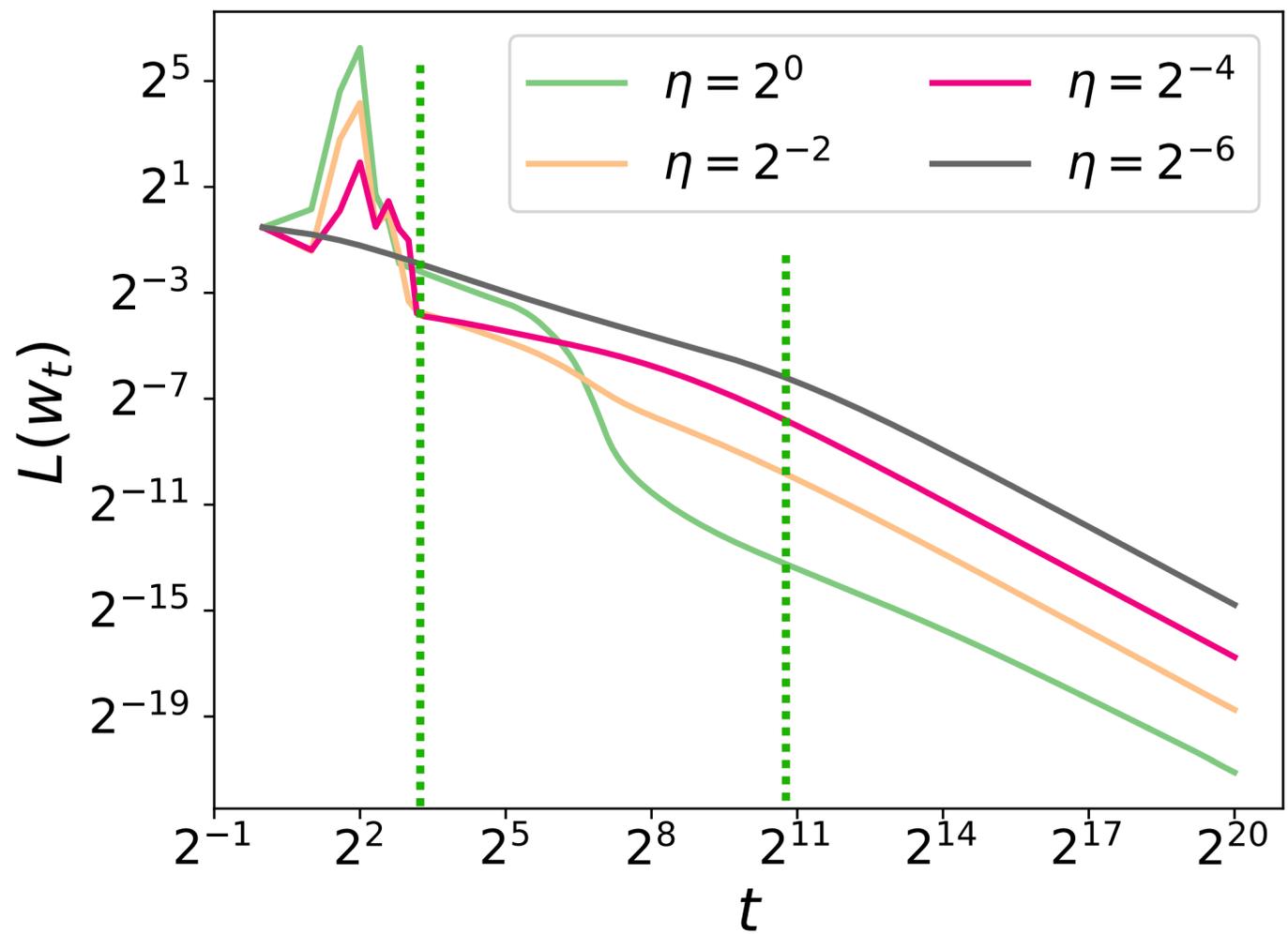
NN -> linear model (w/o bias)

$$L(w) := \frac{1}{n} \sum_i \ln(1 + \exp(-y_i x_i^\top w))$$

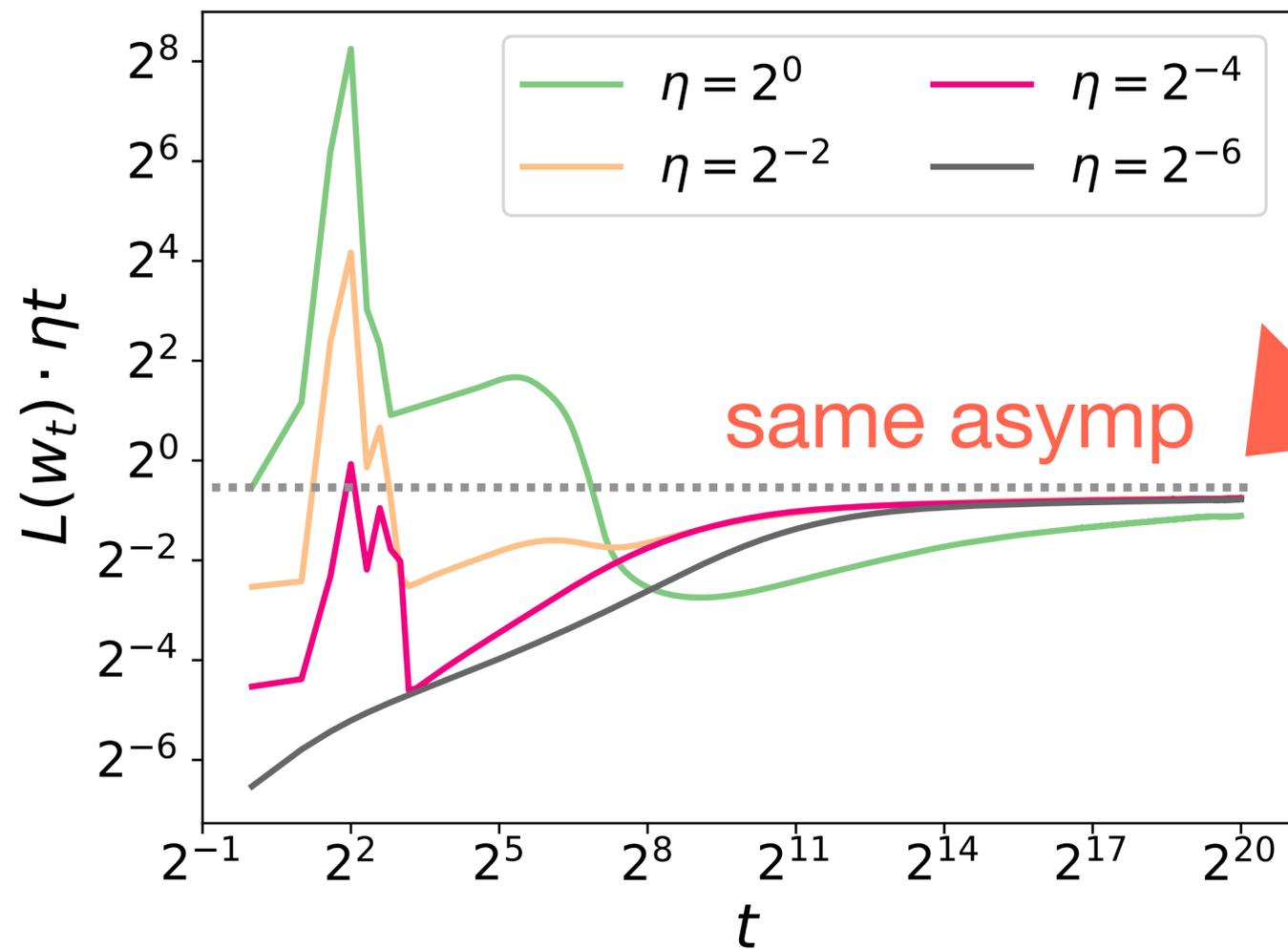
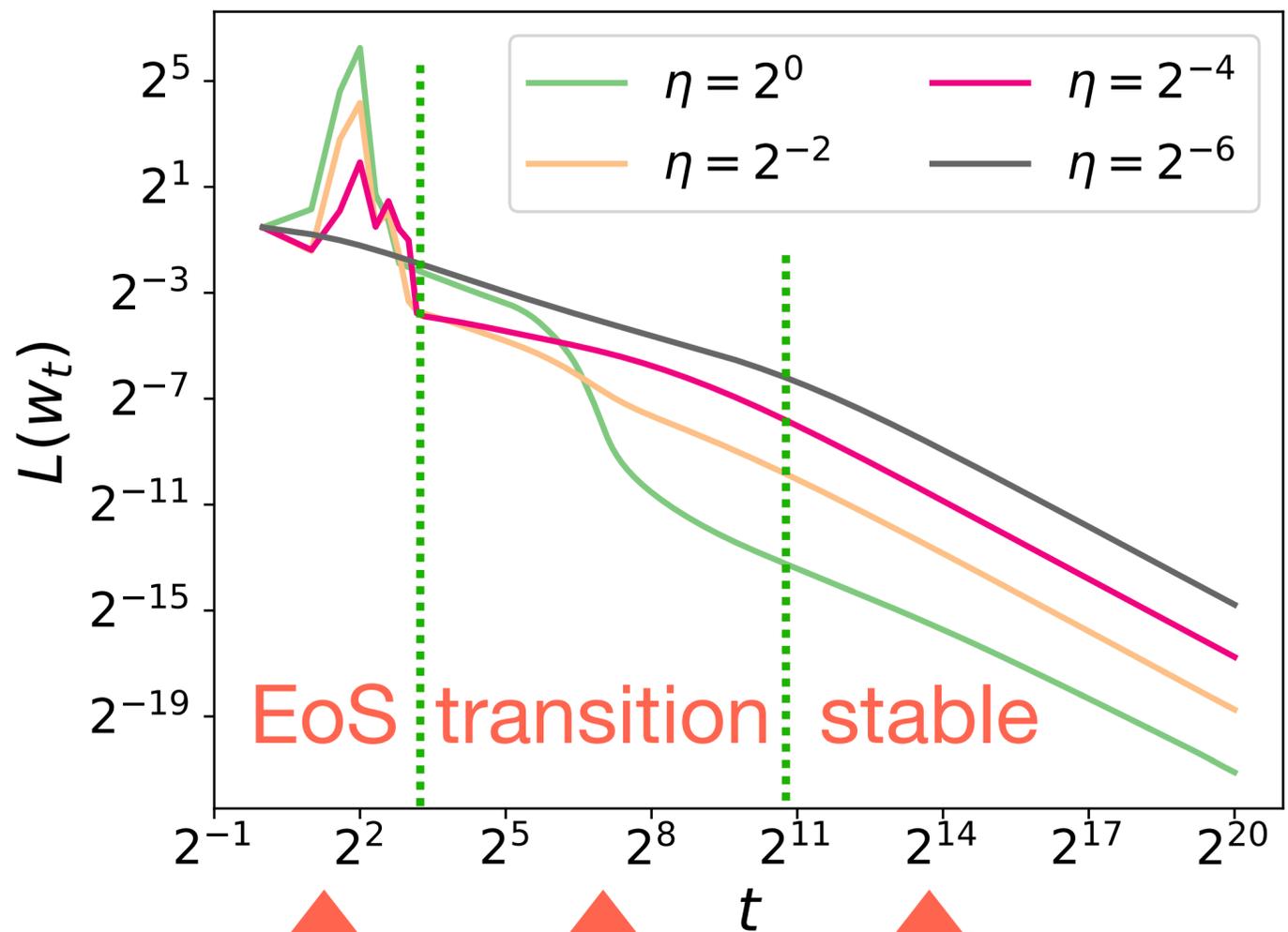
logistic regression, for now :)



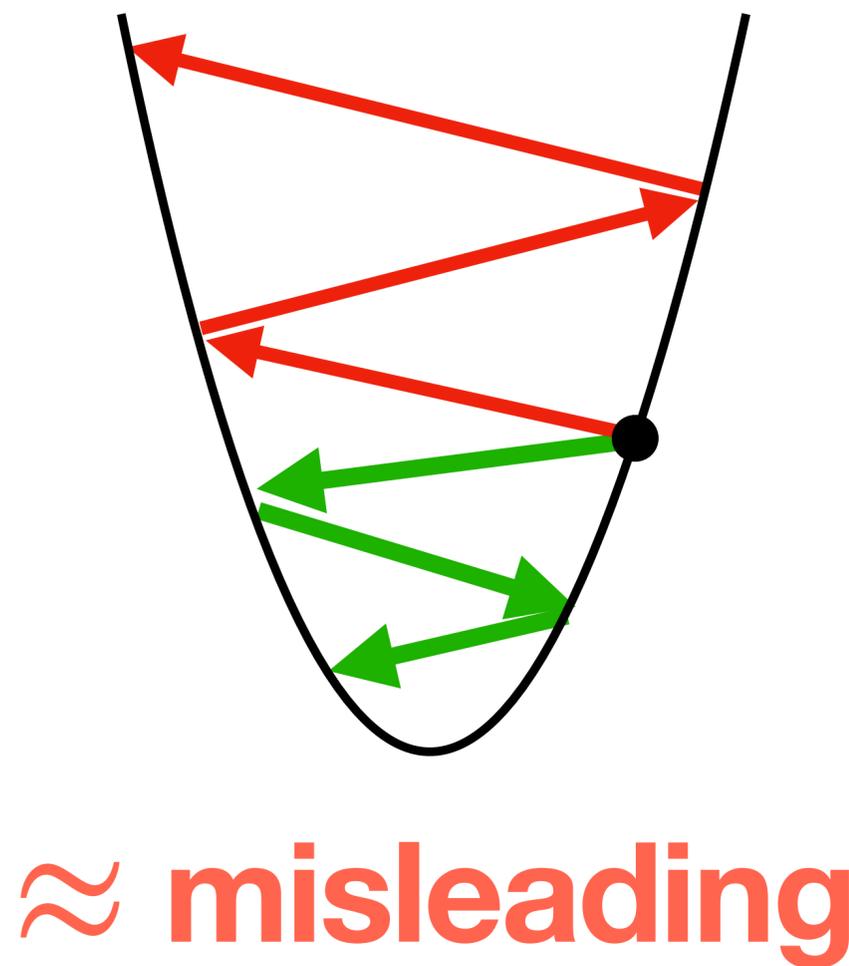
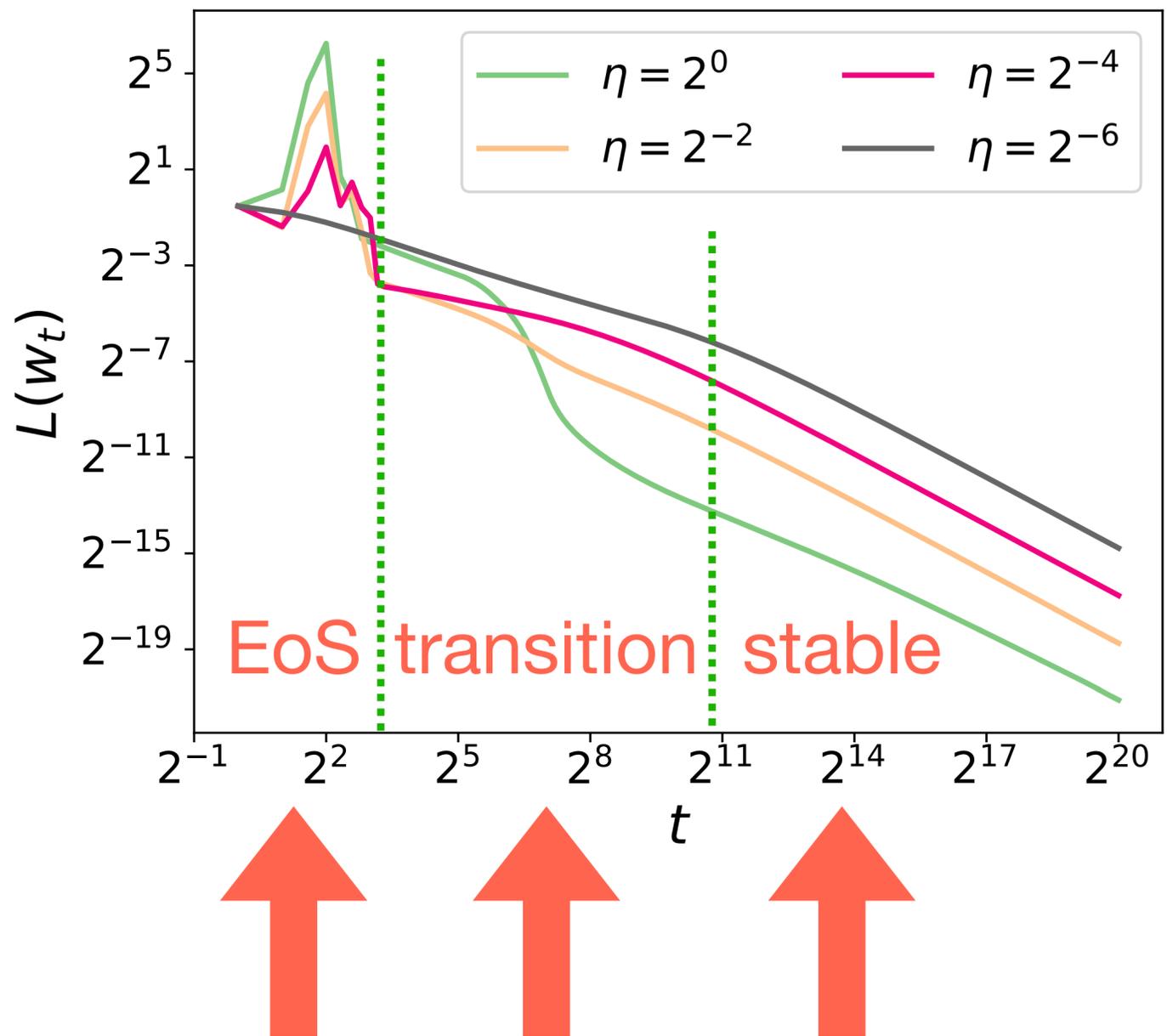
Logistic regression, 1000 samples from MNIST "0" or "8"



Logistic regression, 1000 samples from MNIST "0" or "8"



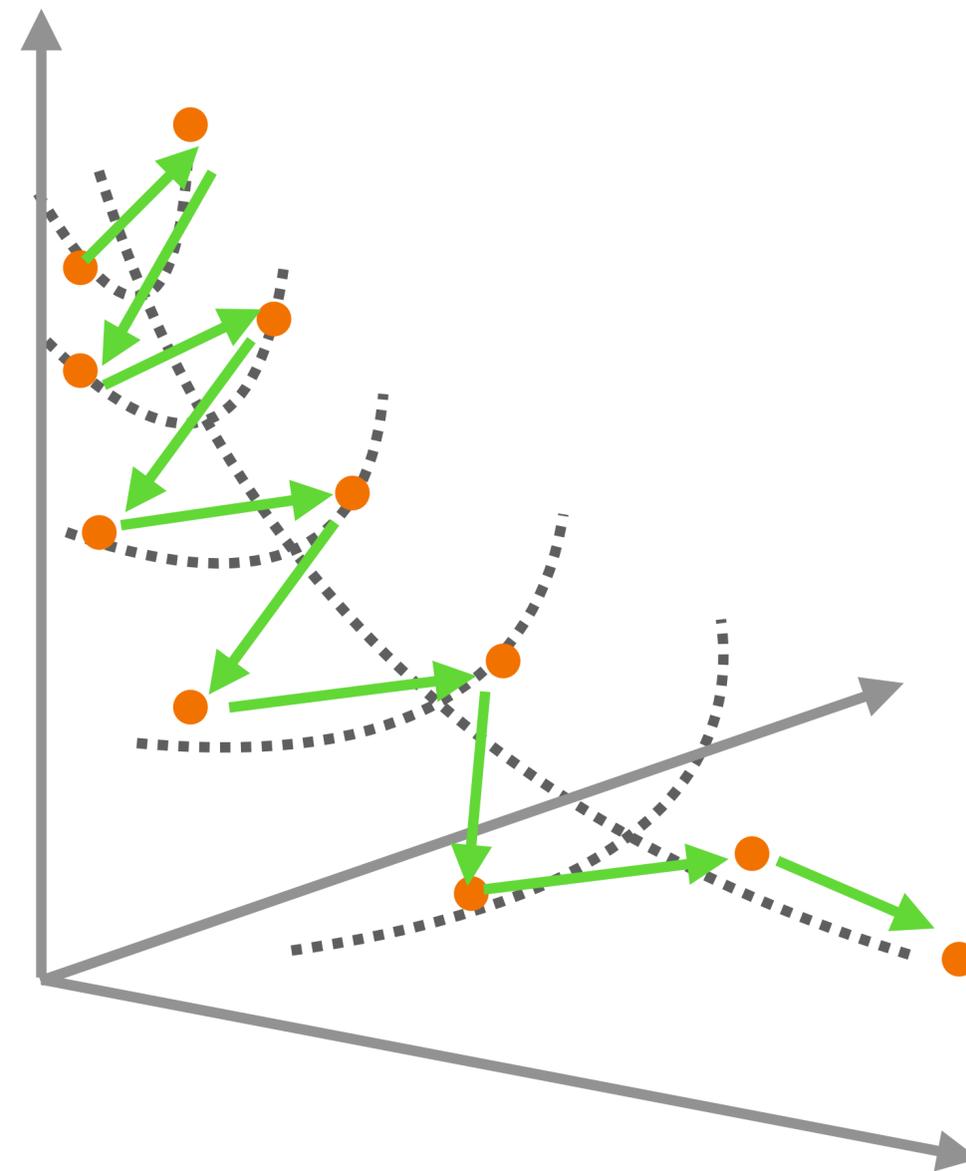
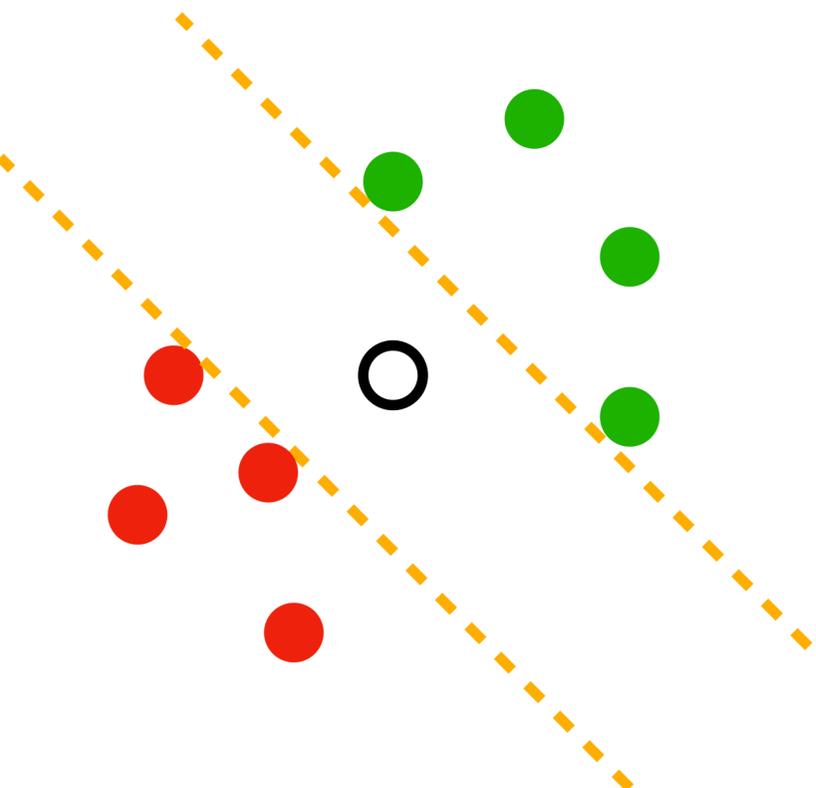
Logistic regression, 1000 samples from MNIST "0" or "8"



$$L(w) := \hat{\mathbb{E}} \ln(1 + \exp(-yx^\top w)), y \in \{\pm 1\}$$

Assume: \exists vector w_*
such that $yx^\top w_* > \gamma > 0$

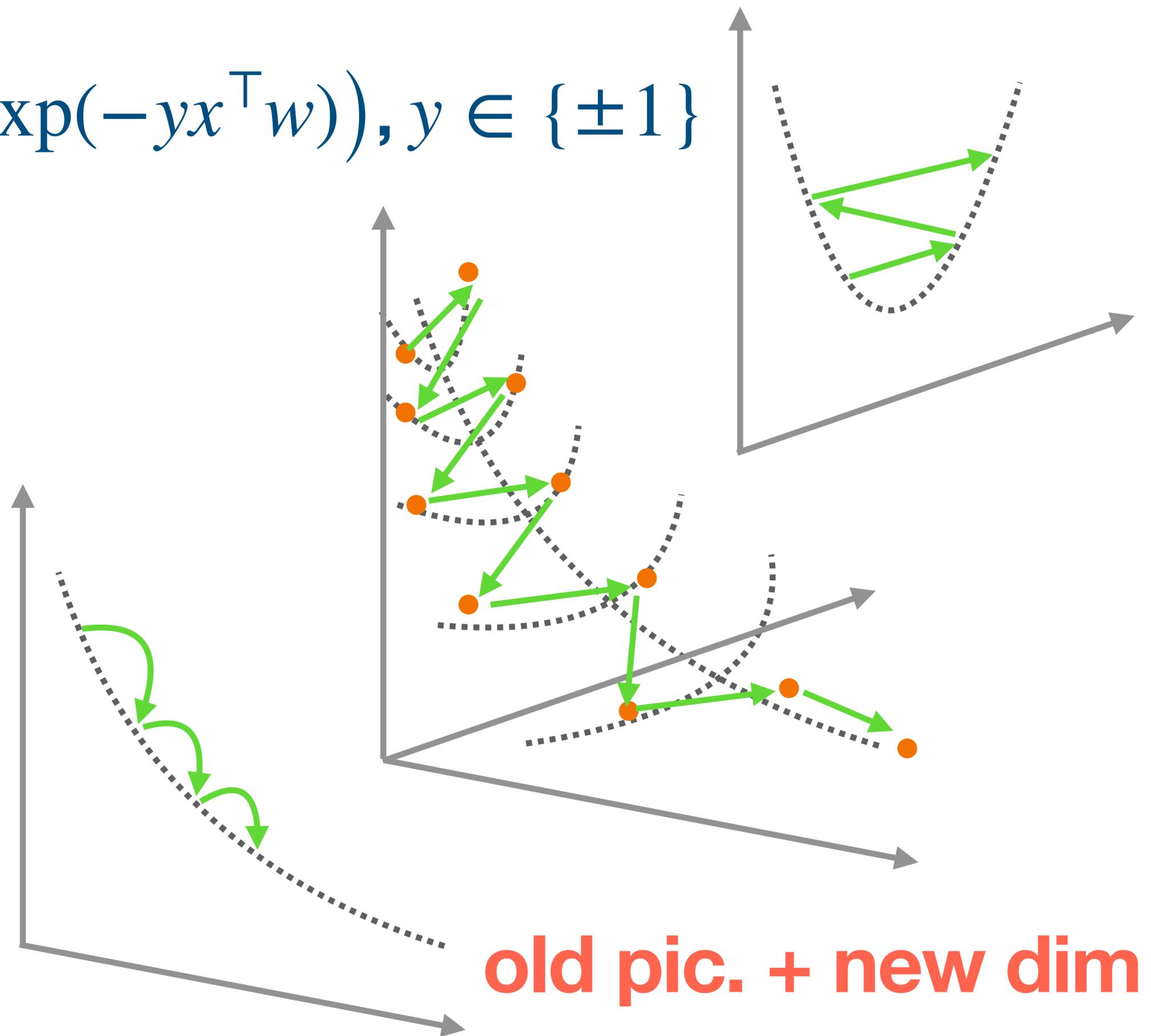
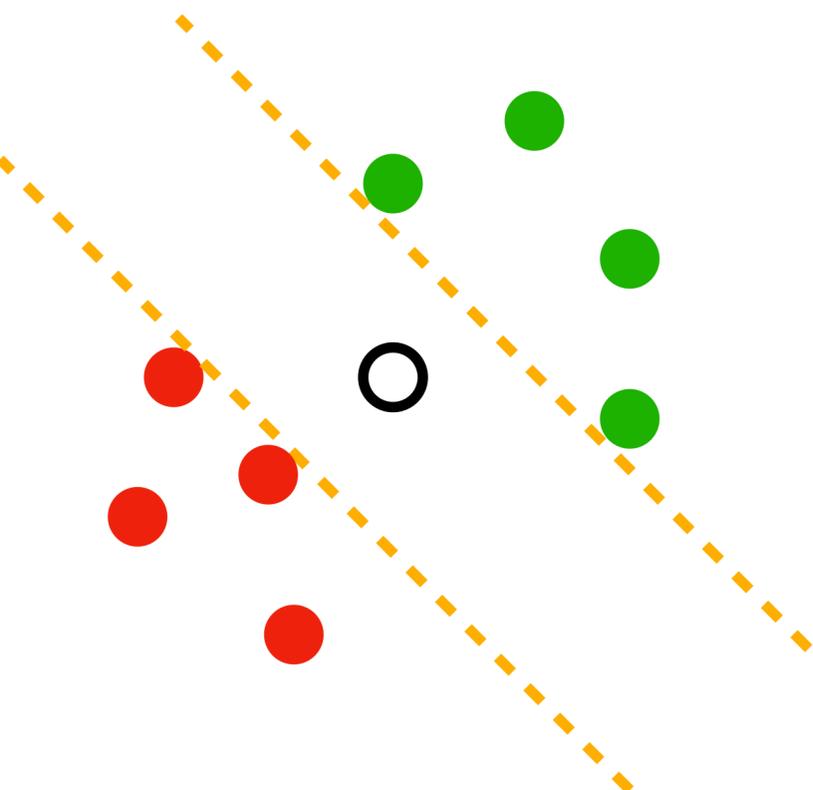
$\rightarrow L(\lambda w_*) \rightarrow 0$
 $\lambda \rightarrow +\infty$



$$L(w) := \hat{\mathbb{E}} \ln(1 + \exp(-yx^\top w)), y \in \{\pm 1\}$$

Assume: \exists vector w_*
such that $yx^\top w_* > \gamma > 0$

$\rightarrow L(\lambda w_*) \rightarrow 0$
 $\lambda \rightarrow +\infty$

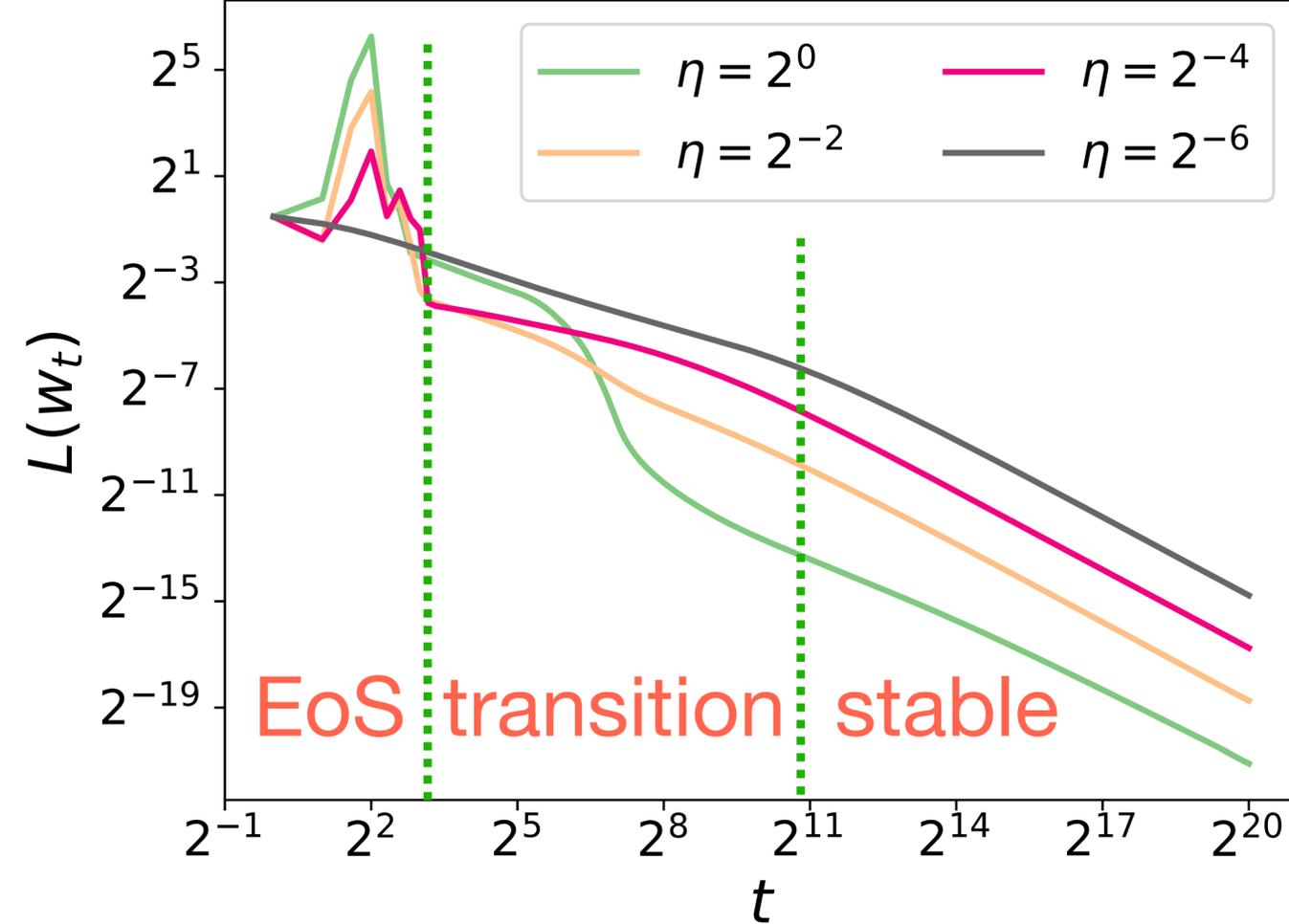


Theorem ...for every η ...

- **EoS phase.** For every t

EoS may not happen

$$\frac{1}{t} \sum_{k=0}^{t-1} L(w_k) \leq \tilde{O}\left(\frac{1 + \eta^2}{\eta t}\right)$$



- **Stable phase.** If $L(w_s) \leq 1/\eta$ for some s , then $L(w_{s+t}) \downarrow$ for $t \geq 0$ and

$$L(w_{s+t}) \leq \tilde{O}\left(\frac{F(w_s)}{\eta t}\right), \quad F(w_s) := \hat{\mathbb{E}} \exp(-yx^\top w)$$

$\tilde{O}(1/\eta t)$
nearly sharp
“flow rate”

- **Phase transition.** We have $L(w_s) \leq 1/\eta$ and $F(w_s) \leq 1$ for

$$s \leq \tau := \Theta\left(\max\{\eta, n, n/\eta \ln(n/\eta)\}\right) \quad \text{EoS lasts} \leq O(\eta) \text{ steps}$$

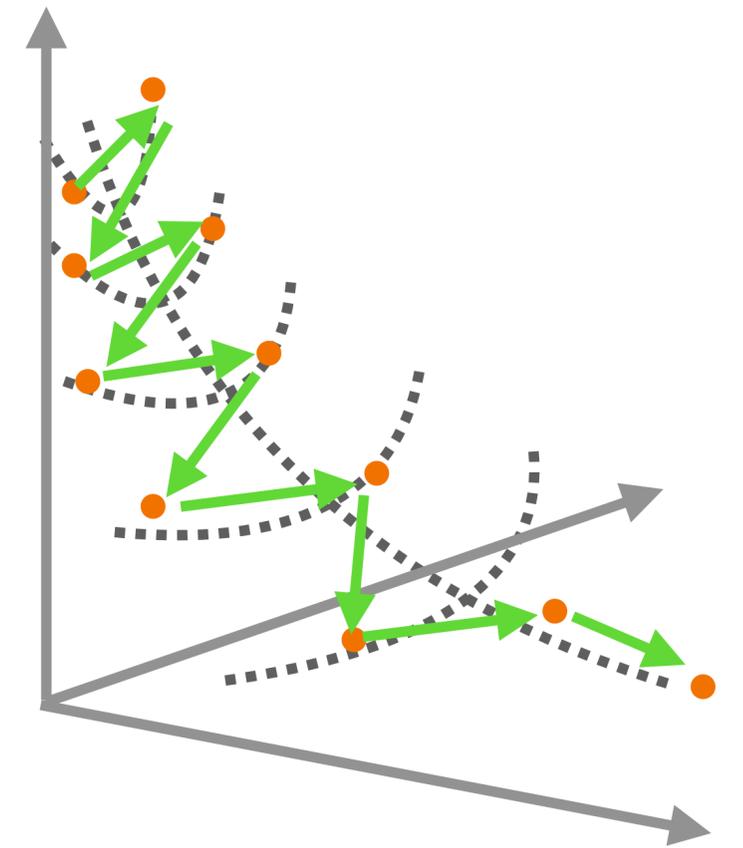
Reimagine GD $w_{t+1} = w_t - \eta \nabla L(w_t)$

1. Asymptotic $\tilde{O}(1/\eta t)$ for **every** η (beyond $1/\text{smoothness}$)
2. Larger $\eta \Rightarrow$ smaller const factor, but longer EoS
3. Given #steps $T \geq \Omega(n)$, if choose $\eta = \Theta(T)$, then

$$\tau \leq T/2 \text{ and } L(w_T) \leq \tilde{O}(1/T^2)$$

**“acceleration” by EoS
w/o momentum or varying stepsizes**

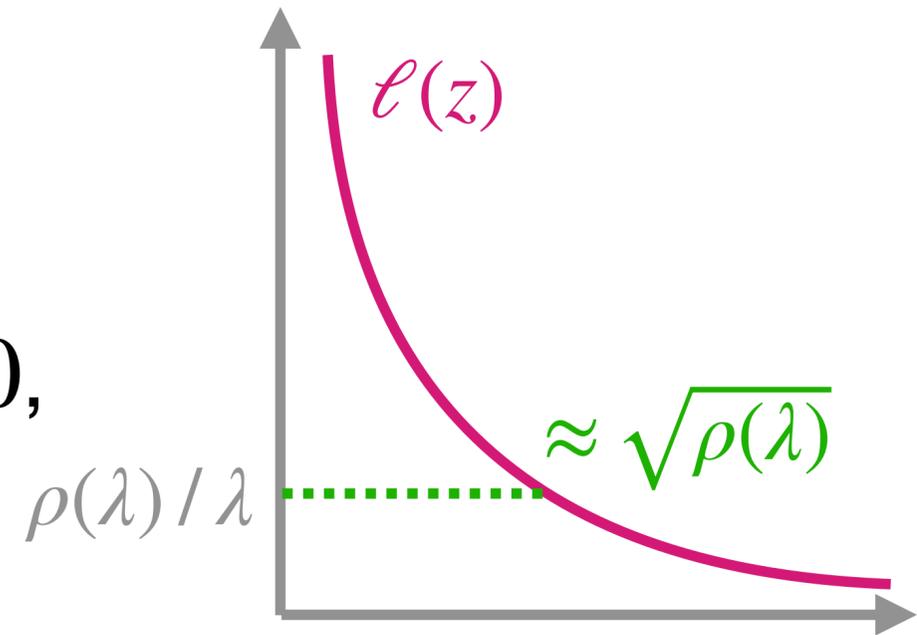
4. **Theorem.** In general, if not enter EoS, then $L(w_T) \geq \Omega(1/T)$



A general loss function $\ell : \mathbb{R} \rightarrow \mathbb{R}_+$

A. **Regularity.** Assume ℓ is \mathcal{C}^2 , convex, \downarrow , and $\ell(+\infty) = 0$,

$$\text{define } \rho(\lambda) := \min_{z \in \mathbb{R}} \lambda \ell(z) + z^2, \quad \lambda \geq 1$$



B. **Lipschitzness.** Assume $g(\cdot) := |\ell'(\cdot)| \leq C_g$ **w/o B, GD may diverge**

C. **Self-boundedness.** Assume $g(\cdot) \leq C_\beta \ell(\cdot)$ and **=> stable phase**

$$\ell(z) \leq \ell(x) + \ell'(z-x) + C_\beta g(x)(z-x)^2, \text{ for } |z-x| \leq 1$$

D. **Exp-tail.** Assume $\ell(\cdot) \leq C_e g(\cdot)$ **=> better transition time**

1. **Logistic loss**, $\ell(z) = \ln(1 + \exp(-z))$, satisfies **A-D**, with

$$\rho(\lambda) \leq 1 + \ln^2(\lambda)$$

2. **Flattened exp loss**, $\ell(z) = \begin{cases} e^{-az} & z \geq 0, \\ 1 - az & z < 0, \end{cases}$ satisfies **A-D**, with

$$\rho(\lambda) \leq 1 + \ln^2(\lambda)/a^2$$

3. **Flattened poly loss**, $\ell(z) = \begin{cases} (1 + z)^{-a} & z \geq 0, \\ 1 - az & z < 0, \end{cases}$ satisfies **A-C**, with

$$\rho(\lambda) \leq 2\lambda^{2/(a+2)}$$

“flatten” is crucial, ensuring Lipschitzness

$$w_{t+1} = w_t - \eta \nabla L(w_t)$$

Assume: NTK init, $w_0 \sim \mathcal{N}(0, I_{md})$

- $x_i \in \mathbb{R}^d$, $\|x_i\| \leq 1$, $y_i \in \{\pm 1\}$, $i = 1, \dots, n$

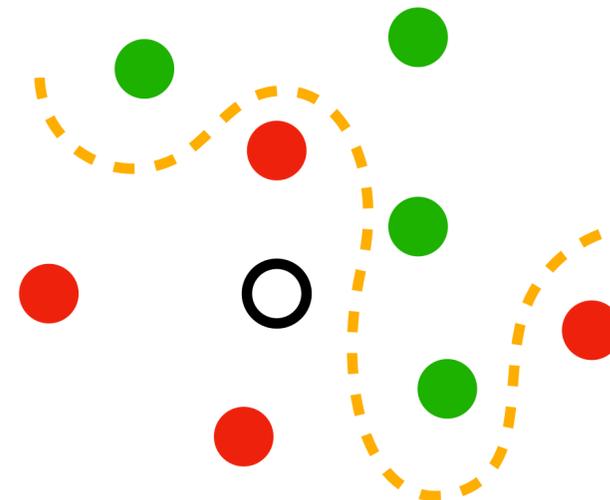
- Neural network under loss function ℓ

$(a_s)_{s=1}^m$ random from $\{\pm 1\}$

$$L(w) := \hat{\mathbb{E}} \ell(y f_x(w)), \quad f_x(w) := \frac{1}{\sqrt{m}} \sum_{s=1}^m a_s \max\{x^\top w^{(s)}, 0\}, \quad w \in \mathbb{R}^{md}$$

- “Separable” in NTK RKHS

1. weaker than “linearly separable”
2. always true for large m



Theorem

NTK for any stepsize

Assume ℓ satisfies A-B. Fix T , assume $m \geq \Omega(R^2)$ for $R := \Theta(\sqrt{\rho(\eta T)} + \eta)$. Then

- **Lazy training.** For $t \leq T$, we have $\|w_t - w_0\| \leq R$

- **EoS phase.** For $t \leq T$, we have $\frac{1}{t} \sum_{k=0}^{t-1} L(w_k) \leq O\left(\frac{\rho(\eta t) + \eta^2}{\eta t}\right)$

- **Stable phase.** Assume ℓ also satisfies C. If $L(w_s) \leq \Theta(1/(\eta + n))$ for some s , then

$$L(w_{s+t}) \downarrow \text{ and } L(w_{t+s}) \leq O\left(\frac{\rho(\eta t)}{\eta t}\right), \quad s + t \leq T$$

- **Phase transition.** We have $L(w_s) \leq \Theta(1/(\eta + n))$ for $s \leq \tau$, where

$$\tau := \Theta\left(\max\{\psi^{-1}(\eta + n), \eta(\eta + n)\}\right), \quad \psi(\lambda) := \lambda/\psi(\lambda)$$

or $\tau := \Theta(\max\{\eta, n \ln(n)\})$ if ℓ also satisfies D

loss function	logistic / flattened exponential		flattened polynomial of degree a		
degree condition	N/A		$a > 0$	$0 < a \leq 1$	$a > 1$
$\rho(\lambda)$	$\Theta(\ln^2(\lambda))$		$\Theta(\lambda^{\frac{2}{a+2}})$		
stepsize η	1	$\Theta(T)$	1	$\Theta(T^{\frac{a}{2}})$	$\Theta(T^{\frac{1}{2}})$
width m	$\Omega(\ln^2(T))$	$\Omega(T^2)$	$\Omega(T^{\frac{2}{a+2}})$	$\Omega(T)$	$\Omega(T)$
phase transition time s	N/A	$\leq T/2$	N/A	$\leq T/2$	$\leq T/2$
loss $L(\mathbf{w}_T)$	$\mathcal{O}(\ln^2(T)/T)$	$\mathcal{O}(\ln^2(T)/T^2)$	$\mathcal{O}(T^{\frac{-a}{a+2}})$	$\mathcal{O}(T^{-\frac{a}{2}})$	$\mathcal{O}(T^{\frac{-3a}{2a+4}})$

Large stepsize exits NTK faster, exponentially

Contribution

New mental picture of GD

1. flow rate for any stepsize
2. EoS => acceleration
3. versatile techniques
 - other loss functions
 - neural networks (NTK)
 - SGD (w/ caveats, see the paper)

