Lecture 2

HW0 is due 10:59pm Friday.

Optimization

1. Implicit Regularization of Gradient Descent.
2. Stochastic Gradient Descent.
3. Momentum.

Last time: Gradient Descent converges for linear regression.

$$\eta < \frac{1}{\lambda_{max}(X^T X)}$$

Key takeaway: Geometric properties of the loss matter, and dictate choice of learning rate.

Regularization + Ridge : Why all the singular values + spread of singular values matter.

$$\vec{w}_* = (X^T X + \lambda I)^{-1} X^T \vec{y} = X^T (XX^T + \lambda I)^{-1} \cdot \vec{y}$$

$$X = U \Sigma V^T$$

$$\vec{w}_* = (V \Sigma^T \cancel{U^T} U \Sigma V^T + \lambda I)^{-1} V \Sigma^T U^T \cdot \vec{y}$$

$$= V (\Sigma^T \Sigma + \lambda I)^{-1} \cancel{V^T V} \Sigma^T U^T \vec{y}$$

$$= V \underbrace{(\Sigma^T \Sigma + \lambda I)^{-1}} \Sigma^T U^T \vec{y} \cdot$$

$$= V \begin{bmatrix} \sigma_1^2+\lambda & & 0 \\ & \sigma_2^2+\lambda & \\ & & \ddots \\ 0 & & \lambda \lambda \lambda \end{bmatrix}^{-1} \Sigma^T U^T \vec{y} \cdot$$

$$\vec{w}_* = \sum_{i=1}^{n} \vec{v}_i \boxed{\frac{\sigma_i}{\sigma_i^2+\lambda}} \vec{u}_i^T \cdot \vec{y}$$

$\frac{\sigma_i}{\sigma_i^2+\lambda}$ if $\lambda \gg \sigma_i$ $\to 0$ as $\lambda \to \infty$

$\frac{\sigma_i}{\sigma_i^2+\lambda}$ $\to$ $\sigma_i \gg \lambda$ $\frac{1}{\sigma_i}$

GD update: $\vec{w}_{t+1} = \vec{w}_t - 2\eta X^T (X\vec{w}_t - \vec{y})$.

residual

always remains in the span of the data.

GD update for ridge

$$\vec{w}_{t+1} = (1 - 2\eta\lambda)\vec{w}_t - 2\eta \cdot (X^T)(X\vec{w}_t - \vec{y})$$

weight decay.

Implicit Regularization

$$V^T\vec{w}_{t+1} = V^T\vec{w}_t - 2\eta V^T (V\Sigma^T U^T)(X\vec{w}_t - \vec{y})$$

$$\vec{\tilde{w}}_{t+1} = \vec{\tilde{w}}_t - 2\eta \Sigma^T (U^T U \Sigma V^T \vec{w}_t - U^T\vec{y})$$

$$= \vec{\tilde{w}}_t - 2\eta \Sigma^T (\Sigma \vec{\tilde{w}}_t - U^T\vec{y})$$

$$\tilde{w}_{t+1,i} = \tilde{w}_{t,i} - 2\eta \cdot (\sigma_i^2 \tilde{w}_{t,i} - \sigma_i \tilde{y}_i)$$

$$= \tilde{w}_{t,i} - 2\eta \cdot \sigma_i (\sigma_i \tilde{w}_{t,i} - \tilde{y}_i)$$

"Early Stopping"

Implicit Regularization.

Ridge → λ is large → we don't have large components of directions of small singular values in Ridge solution.

GD → early stopping → also does not move (or moves very little) in directions of small singular values.

Stochastic Gradient Descent

- GD is great for convex functions.
- Neural network fitting requires optimization over non-convex landscapes
- GD is also expensive.

SGD

- Adds some noise to gradient direction. In expectation, still descends.

$$\min_{\theta} f(\theta)$$

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{1}{B} \sum_{i=1}^{B} \nabla f_i(\theta_t)$$

B: batch size
"mini-batch"

where $f(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$

Gradient descent is "full-batch"

$$E[\nabla f_i(\theta)] = \frac{1}{n} \cdot \sum_{i=1}^{n} \nabla f_i(\theta) = \nabla f(\theta)$$

draw i uniformly iid

- Many standard loss functions have this form.

  e.g. squared loss functions:

$$\underset{\vec{x}}{\operatorname{argmin}} \|A\vec{x}-\vec{y}\|_2^2 = \underset{\vec{x}}{\operatorname{argmin}} \frac{1}{m}\|A\vec{x}-\vec{y}\|_2^2 =$$

$$= \underset{\vec{x}}{\operatorname{argmin}} \frac{1}{m}\sum_{i=1}^{m}(\vec{a}_i^T\vec{x}-y_i)^2.$$

  $(\vec{a}_i^T\vec{x}-y_i)^2$: loss on individual data point.

- Though noisy, improves fit to subset of data each step.
- Computationally less expensive
- Noise helps us (can help us) escape local minima and saddle points.

Question: Does it converge?

Optimization : Must use a learning-rate schedule with decreasing step size for SGD

In practice: can see it converge with constant step size.

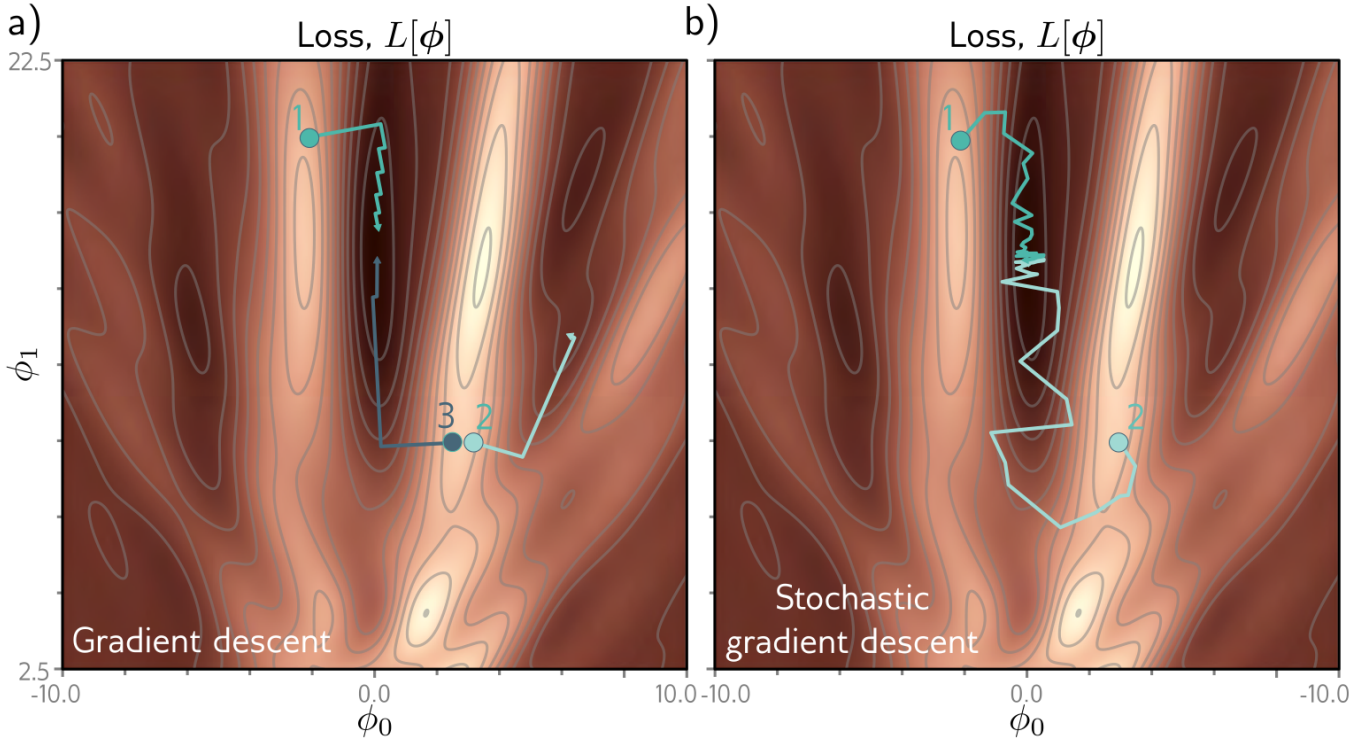


Figure 6.5 Gradient descent vs. stochastic gradient descent. a) Gradient descent with line search. As long as the gradient descent algorithm is initialized in the right "valley" of the loss function (e.g., points 1 and 3), the parameter estimate will move steadily toward the global minimum. However, if it is initialized outside this valley (e.g., point 2), it will descend toward one of the local minima. b) Stochastic gradient descent adds noise to the optimization process, so it is possible to escape from the wrong valley (e.g., point 2) and still reach the global minimum.

Prince 2025.

Momentum:

Gradient Descent.

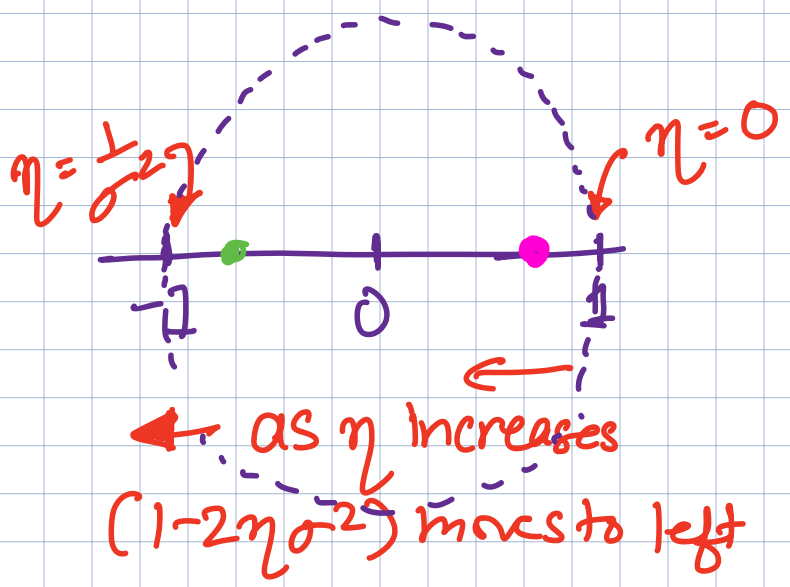
Scalar case: $\sigma w = y$

(i.e. on direction of SVD basis)

$$w_{t+1} = w_t + \eta \cdot 2\sigma(y - \sigma w_t).$$

$$= (1 - 2\eta\sigma^2) w_t + 2\eta \cdot \sigma \cdot y.$$

$$\Rightarrow \eta < \frac{1}{\sigma^2_{max}}$$

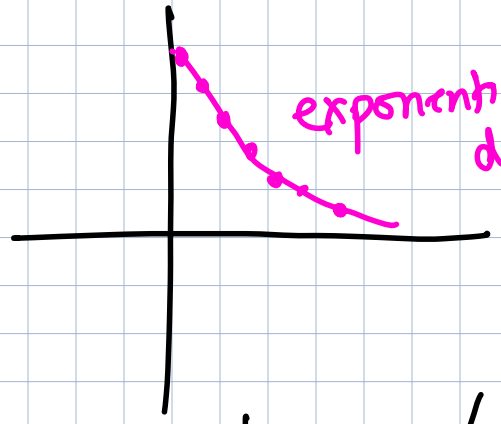


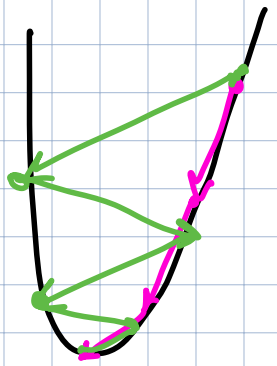
i.e. $\min_w \|y - \sigma w\|_2^2 = f(w)$

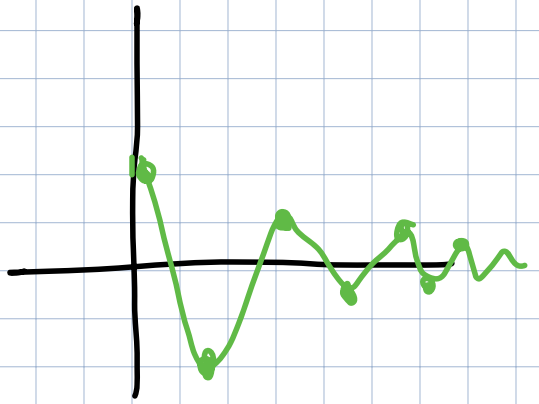
$$\frac{d}{dw} f(w) = -2\sigma(y - \sigma w)$$

GD learning rate gated by max singular value.

So smaller SVs may still converge slowly.







So how can we smooth out the oscillations?

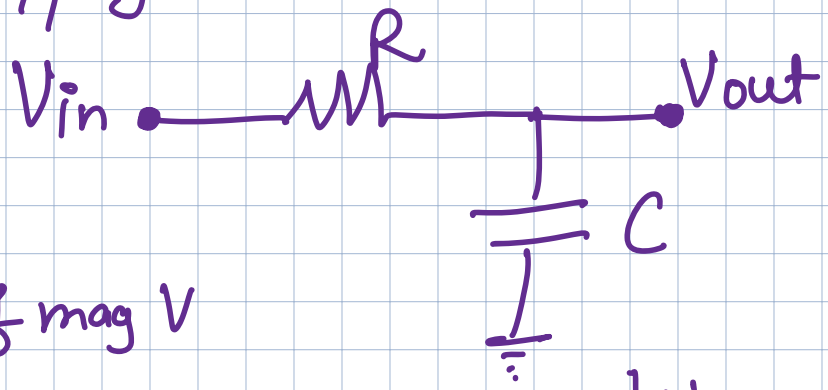
→ Averaging, Low-pass filtering

One strategy: $\frac{1}{t}\sum_{i=1}^{t} \nabla f(w_i)$ → straight average.

→ Requires lots of memory ... have to remember all the past gradients.

→ In physics / state-space systems a state can be used for averaging.

A circuit / physics intuition:



Low-pass filter

If V_{in} = step of mag V

$$V_{out} = V\left(1 - e^{-\frac{1}{RC}t}\right)$$

Momentum : Another tweak on Gradient Descent. || How to make optimization go faster?

Add memory.

$$\vec{w}_{k+1} = \vec{w}_k - \eta \cdot \vec{z}_{k+1}$$

$$\vec{z}_{k+1} = \beta \cdot \vec{z}_k + (1-\beta) \nabla f(\vec{w}_k), \quad z_0 = 0. \quad \beta \in [0,1]$$

If $\beta = 0$, $z_{k+1} = \nabla f(w_k)$. i.e same old gradient descent.

$$z_1 = (1-\beta) \nabla f(w_0)$$

$$z_2 = \beta \cdot (1-\beta) \cdot \nabla f(w_0) + (1-\beta) \nabla f(w_1).$$

$$z_3 = \beta^2 \cdot (1-\beta) \cdot \nabla f(w_0) + \beta \cdot (1-\beta) \nabla f(w_1) + (1-\beta) \nabla f(w_2).$$

$\vdots$

Add history of the gradients. Exponentially decaying weight on history.

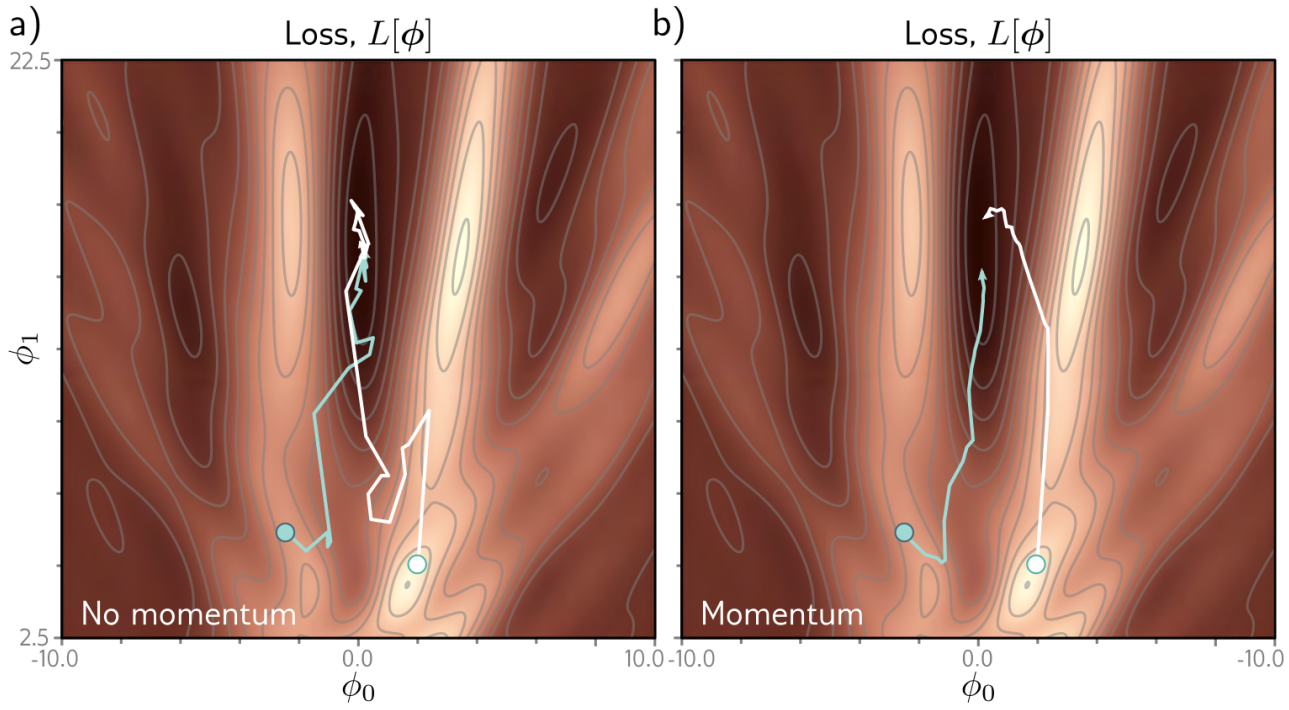
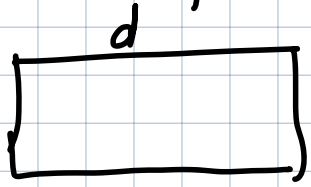


Figure 6.7 Stochastic gradient descent with momentum. a) Regular stochastic descent takes a very indirect path toward the minimum. b) With a momentum term, the change at the current step is a weighted combination of the previous change and the gradient computed from the batch. This smooths out the trajectory and increases the speed of convergence.

SGD Convergence

SGD can provably converge even with constant stepsize if loss goes to zero.

$X\vec{w} = \vec{y}$. $X \in \mathbb{R}^{n \times d}$ full-row rank.



$d > n$. Under-determined.

Let $\vec{w}_*$ be the min-norm solution.

$$\vec{w}_* = X^T(XX^T)^{-1}\vec{y}.$$

We do SGD, batch size 1. $\vec{w}_0 = 0$.

$$X(\underbrace{\vec{w} - \vec{w}^*}_{\vec{q}}) = \vec{y} - \vec{y} = 0$$

(error)

$X\vec{q} = 0$ is equivalent equation with $\vec{q}_0 = \vec{w}_0 - \vec{w}_* = -\vec{w}_*$

We know the SVD dictates which directions SGD/GD moves in.

$$X\vec{q} = U\Sigma V^T\vec{q}$$

$$= U\begin{bmatrix}\underbrace{\Sigma_1}_{n\times n} & \underbrace{0}_{n\times(d-n)}\end{bmatrix}V^T\vec{q}$$

$$= \begin{bmatrix}\tilde{X} & 0\end{bmatrix}\vec{z}$$

$V^T\vec{q} = \vec{z}$

$U\Sigma_1 = \tilde{X}$

$$\vec{z}_0 = V^T\vec{q}_0 = V^T(-\vec{w}_*)$$

$$= -\begin{bmatrix}\Sigma_1^{-1}U^T\vec{y} \\ 0\end{bmatrix}\} d-n \text{ zero entries.}$$

So what matters is

$$\underbrace{\tilde{X}}_{n \times n} \vec{z} = 0$$

ith row of this is the same as ith row of $X\vec{q}$

(we have only dropped zeros)

Now GD.

$$\vec{z}_{t+1} = \vec{z}_t - 2\eta \cdot \tilde{X}^T \tilde{X} \vec{z}_t$$

$$\text{Loss} = \|\tilde{X}\vec{z}\|_2^2$$

$$\nabla \text{Loss} = 2\tilde{X}^T\tilde{X}\vec{z}.$$

$$\tilde{X}^T\tilde{X} = \Sigma_1^T U^T U \Sigma_1 = \Sigma_1^T \Sigma_1$$

For SGD with batch size 1, this is.

$$\vec{z}_{t+1} = \vec{z}_t - 2\eta \cdot \tilde{x}_{I_t} \tilde{x}_{I_t}^T \vec{z}_t$$

point sampled at time I_t.

For this SGD update can we argue that

$$\text{Loss} = \|\tilde{X}\vec{z}\|_2^2 = \vec{z}^T\tilde{X}^T\tilde{X}\vec{z}$$ goes to zero?

How to show? "Lyapunov function"

Argue that it decreases in expectation on every step and is non-negative.

Furthermore $\mathcal{L}(\vec{z})=0 \Rightarrow \vec{z}=0$. Since $\tilde{X}$ is full rank.

Further,

$$E[\mathcal{L}(\vec{z}_{t+1})] \leq E[\mathcal{L}(\vec{z}_t)] + C\cdot\mathcal{L}(\vec{z}_t))$$

continues downward pressure. to guarantee convergence.

So how can we connect $\mathcal{L}(\vec{z}_{t+1})$ to $\mathcal{L}(\vec{z}_t)$?

Write $\mathcal{L}(\vec{z}_{t+1}) = \mathcal{L}(\vec{z}_t + (\vec{z}_{t+1}-\vec{z}_t))$

Algebra

$$= \mathcal{L}(\vec{z}_t) + A + B.$$

$$A = 2\vec{z}_t^T\tilde{X}^T\tilde{X}(\vec{z}_{t+1}-\vec{z}_t)$$ Linear in update

$$B = (\vec{z}_{t+1}-\vec{z}_t)^T\tilde{X}^T\tilde{X}(\vec{z}_{t+1}-\vec{z}_t)$$ Quadratic in update

Can we show a contraction?

$$\boxed{E[A \mid \vec{z}_t] \le -c\eta \cdot L(\vec{z}_t)}$$

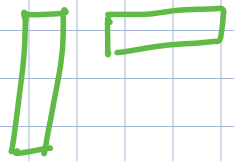
Recall only randomness here comes from SGD.

$$E[A \mid \vec{z}_t] = 2\vec{z}_t^T \tilde{X}^T \tilde{X}\, E\left[\vec{z}_{t+1} - \vec{z}_t \mid \vec{z}_t\right]$$

$$= 2\vec{z}_t^T \tilde{X}^T \tilde{X}(-2\eta) E\left[\tilde{x}_{I_t} \tilde{x}_{I_t}^T\right] \cdot \vec{z}_t$$

$$E\left[\tilde{x}_{I_t} \tilde{x}_{I_t}^T\right] = \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T = \frac{1}{n} \tilde{X}^T \tilde{X}$$

Recall $\tilde{X} = \begin{bmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_n^T \end{bmatrix}$



$$\Rightarrow E[A \mid \vec{z}_t] = -\frac{4\eta}{n} \cdot \vec{z}_t^T \tilde{X}^T \tilde{X} \boxed{\tilde{X}^T \tilde{X} \cdot \vec{z}_t} \rightarrow \|\tilde{X}^T \tilde{X} \vec{z}_t\|_2^2$$

$\rightarrow \sigma_{min}^2 \|\tilde{X} \vec{z}_t\|_2^2$

min scaling is σ_{min}^2

$$\le -\frac{4\eta}{n} \cdot \sigma_{min}^2 \cdot \vec{z}_t^T \tilde{X}^T \tilde{X} \vec{z}_t$$

$$= -\frac{4\eta}{n} \cdot \sigma_{min}^2 L(\vec{z}_t)$$

Now for the B term, quadratic in $(\vec{z}_{t+1} - \vec{z}_t)$.

$$E[B|\vec{z}_t] \leq c_2 \cdot \eta \cdot \mathcal{L}(\vec{z}_t)$$

$$E[B|\tilde{w}_t] = E\left[(\vec{z}_{t+1} - \vec{z}_t)^T \tilde{x}^T \tilde{x} (\vec{z}_{t+1} - \vec{z}_t)\right]$$

$\tilde{x}^T \tilde{x}$: max scaling.

$$\leq 4\eta^2 \sigma_{max}^2 \vec{z}_t^T E\left[\tilde{x}_{I_t} \tilde{x}_{I_t}^T \tilde{x}_{I_t} \tilde{x}_{I_t}^T\right] \vec{z}_t$$

$\|\vec{x}_{I_t}\|^2 \leq \rho^2$

$\vec{z}_{t+1} - \vec{z}_t$

$= -2\eta \tilde{x}_{I_t} \tilde{x}_{I_t}^T \vec{z}_t$

$-2\eta \tilde{x}_{I_t} \tilde{x}_{I_t}^T \vec{z}_t$

$$\leq \frac{4\eta^2}{n} \sigma_{max}^2 \cdot \rho^2 \mathcal{L}(\tilde{w}_t)$$

Putting it all together.

$$E[L(\vec{z}_{t+1}) \mid z_t] \leq \underbrace{(1 - c_1\eta + c_2\eta^2)}\, L(\vec{z}_t).$$

smaller than 1 if η is small enough.

Takeaway:

- Interesting dependance on norms of data points.
- Decoupling to SVD helps us see what is going on.
- SGD shares many properties of GD.
  e.g. can only move in span of data.