# *Efficient* Online Learning, Deterministic, and Stochastic Optimization

Yoram Singer
Machine Intelligence Center
Google Research

CS6780: Advanced Machine Learning, Cornell

# High Dimensional Sparse Data

- To predict the MID of an entity a large number of boolean predicates are built and combined

- Most predicates evaluate to be false for most examples

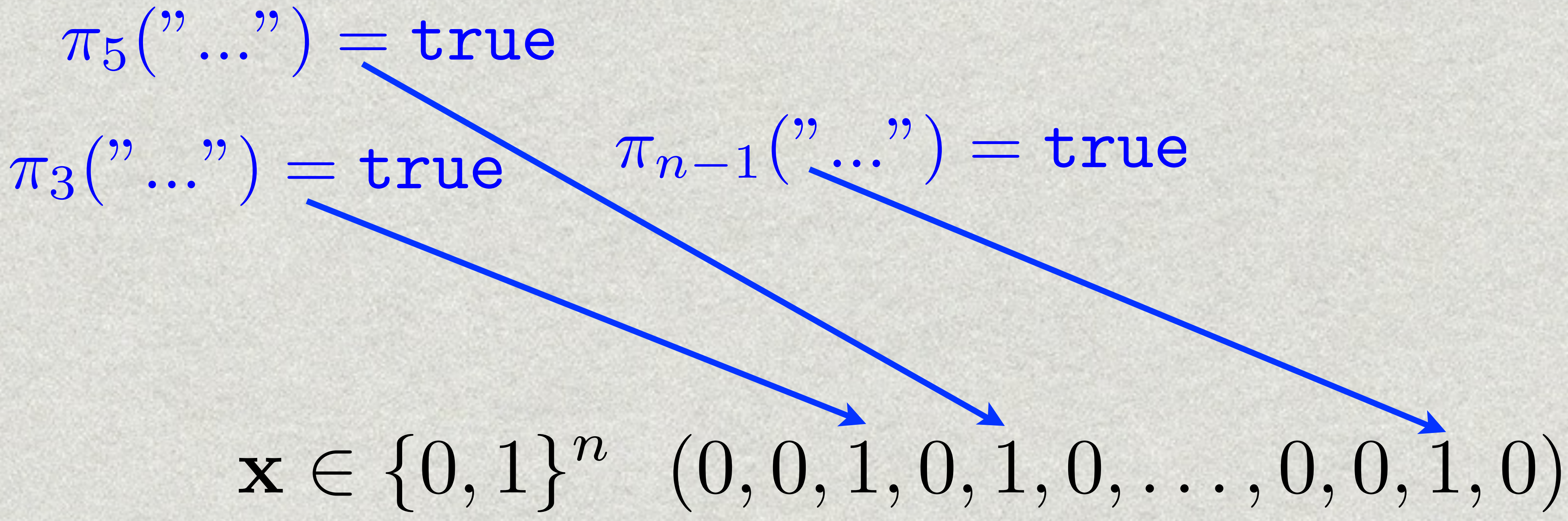
- Example: [$\omega_t$ = President-Name] & [$\omega_{t+1}$=“White-House”]

# High Dimensional Sparse Data

- To predict the MID of an entity a large number of boolean predicates are built and combined

- Most predicates evaluate to be false for most examples

- Example: [$\omega_t$ = President-Name] & [$\omega_{t+1}$=“White-House”]



$\pi_5(\text{"..."}) = \texttt{true}$

$\pi_3(\text{"..."}) = \texttt{true}$

$\pi_{n-1}(\text{"..."}) = \texttt{true}$

$\mathbf{x} \in \{0,1\}^n \quad (0, 0, 1, 0, 1, 0, \ldots, 0, 0, 1, 0)$

# High Dimensional Sparse Data

- To predict the MID of an entity a large number of boolean predicates are built and combined
- Most predicates evaluate to be false for most examples
- Example: [$\omega_t$ = President-Name] & [$\omega_{t+1}$=“White-House”]

How to use the predicates in order to make accurate predictions ?

?

$$\mathbf{x} \in \{0,1\}^n \quad (0,0,1,0,1,0,\ldots,0,0,1,0)$$

# High Dimensional Sparse Data

- To predict the MID of an entity a large number of boolean predicates are built and combined
- Most predicates evaluate to be false for most examples
- Example: [$\omega_t$ = President-Name] & [$\omega_{t+1}$=“White-House”]

Which predicates are “important” for performing accurate predictions?

$$\mathbf{x} \in \{0,1\}^n \quad (0,0,1,0,\overset{?}{1},0,\ldots,0,0,1,0)$$

# Setting in a Picture

**INSTANTIATED PREDICATES == FEATURES**

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| Example $\mathbf{x_1}$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $\mathbf{x_2}$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $\mathbf{x_7}$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

(e.g. MID="Obama")

| Target **Y** |
|---|
| 0 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |

# Setting in a Picture

**INSTANTIATED PREDICATES == FEATURES**

(e.g. MID="Obama")

|  | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | Target **Y** |
|---|---|---|---|---|---|---|---|---|---|
| Example $x_1$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $x_2$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
|  | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| $x_7$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |

**FREQUENT BUT NON-INFORMATIVE**

# Setting in a Picture

**INSTANTIATED PREDICATES == FEATURES**

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| Example $x_1$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $x_2$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $x_7$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

(e.g. MID="Obama")

| Target **Y** |
|---|
| 0 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |

**FREQUENT BUT NON-INFORMATIVE**

**INFREQUENT YET INFORMATIVE**

# Setting in a Picture

**INSTANTIATED PREDICATES == FEATURES**

|  | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| Example $x_1$ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $x_2$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|  | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
|  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
|  | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| $x_7$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

(e.g. MID="Obama")

| Target **Y** |
|---|
| 0 |
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |

**FREQUENT BUT NON-INFORMATIVE**

**INFREQUENT YET INFORMATIVE**

**TRUE ALSO FOR REAL VALUED FEATURES**

# Challenges

# Challenges

- Large amounts of high dimensional sparse data:
    - Computational time should scale with #“1” features
    - Cannot process entire dataset “all at once”

# Challenges

- Large amounts of high dimensional sparse data:
    - Computational time should scale with #“1” features
    - Cannot process entire dataset “all at once”
- Many *frequent* features are *irrelevant*

# Challenges

- Large amounts of high dimensional sparse data:
  - Computational time should scale with #“1” features
  - Cannot process entire dataset “all at once”
- Many *frequent* features are *irrelevant*
- Some of the *infrequent* features are *highly relevant*

# Challenges

- Large amounts of high dimensional sparse data:
  - Computational time should scale with #“1” features
  - Cannot process entire dataset “all at once”
- Many *frequent* features are *irrelevant*
- Some of the *infrequent* features are *highly relevant*
- Need to learn relatively compact models:
  - Training can use lots of (distributed) memory & CPUs
  - Serving (testing) is performed on many more instances than training and often should be

# Outline

- Brief reminder:
  linear models, empirical loss, regularization
- Convexity, Smoothness, and $L_1$ regularization
- Gradients & Subgradients for loss minimization
- Gradient Descent & Stochastic Gradient Methods
- Proximal view of GD & SG
- *Fobos: dimension efficient* proximal method
- *AdaGrad*: *feature efficient* adaptive proximal method

# Elementary Start: Linear Models

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance **x** | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ |

×

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Weights **w** | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ |

Prediction $$\hat{y} = \mathbf{w} \cdot \mathbf{x} = \sum_{j=1}^{n} w_j x_j$$

True Target $y \quad \Rightarrow \quad \ell(y, \hat{y})$ (loss function)

# Elementary Start: Linear Models

Instance **X**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ |

×

Weights **W**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ |

Prediction $$\hat{y} = \mathbf{w} \cdot \mathbf{x} = \sum_{j=1}^{n} w_j x_j$$

True Target $y \quad \Rightarrow \quad \ell(y, \hat{y})$ (loss function)

Example of losses

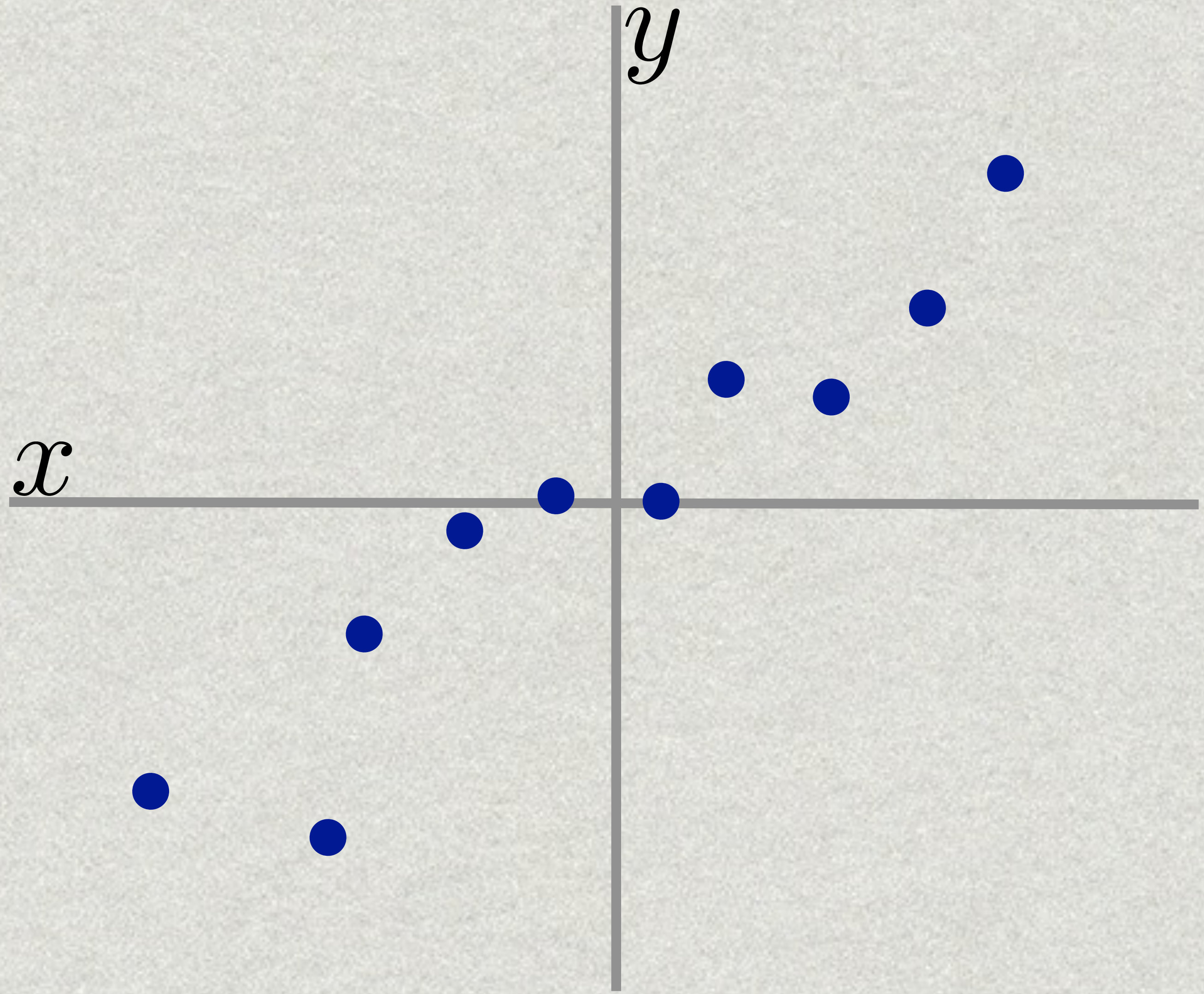
$$\ell(y, \hat{y}) = (y - \hat{y})^2$$

squared error

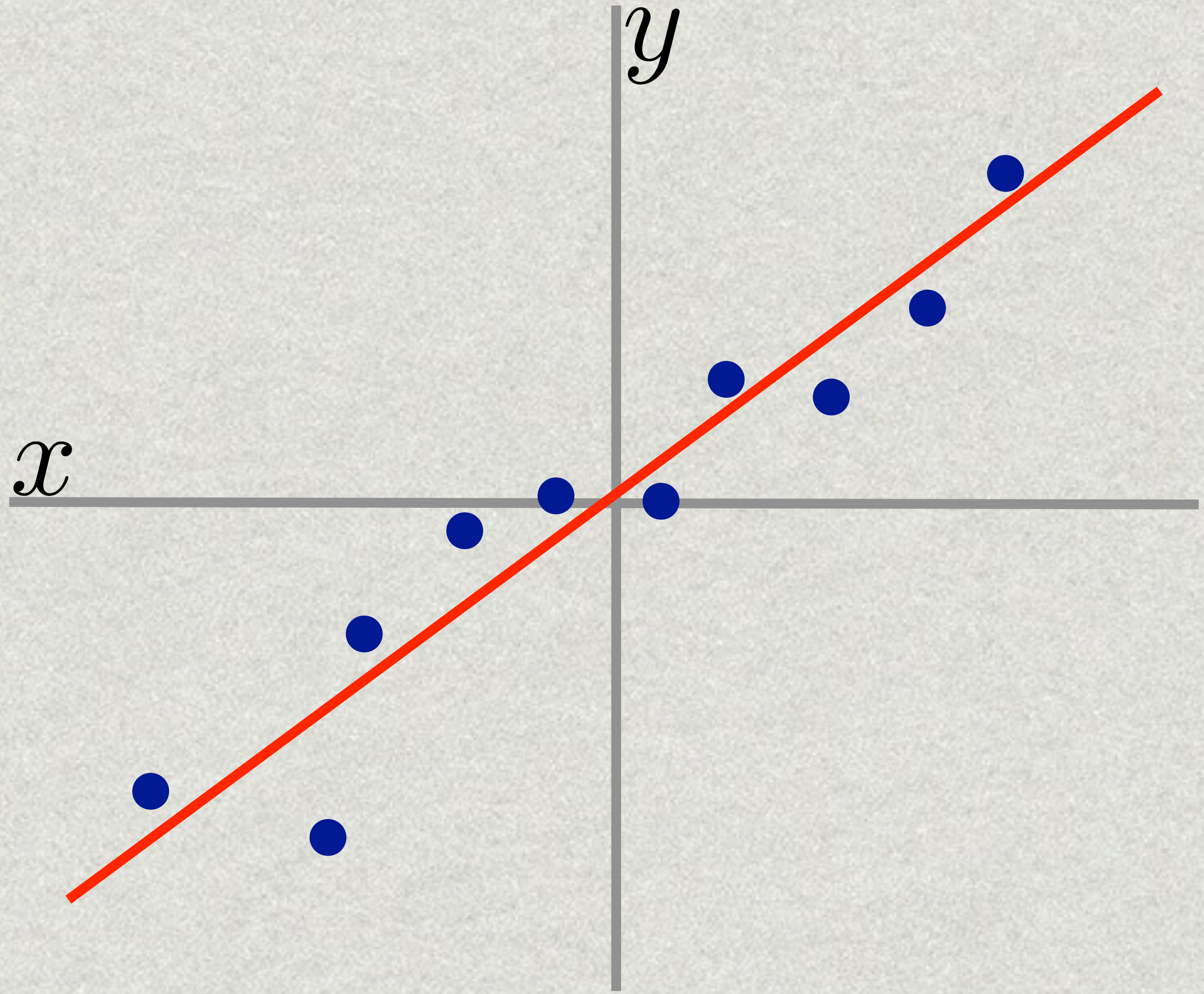
$$\ell(y, \hat{y}) = e^{-y\hat{y}}$$

exponential loss

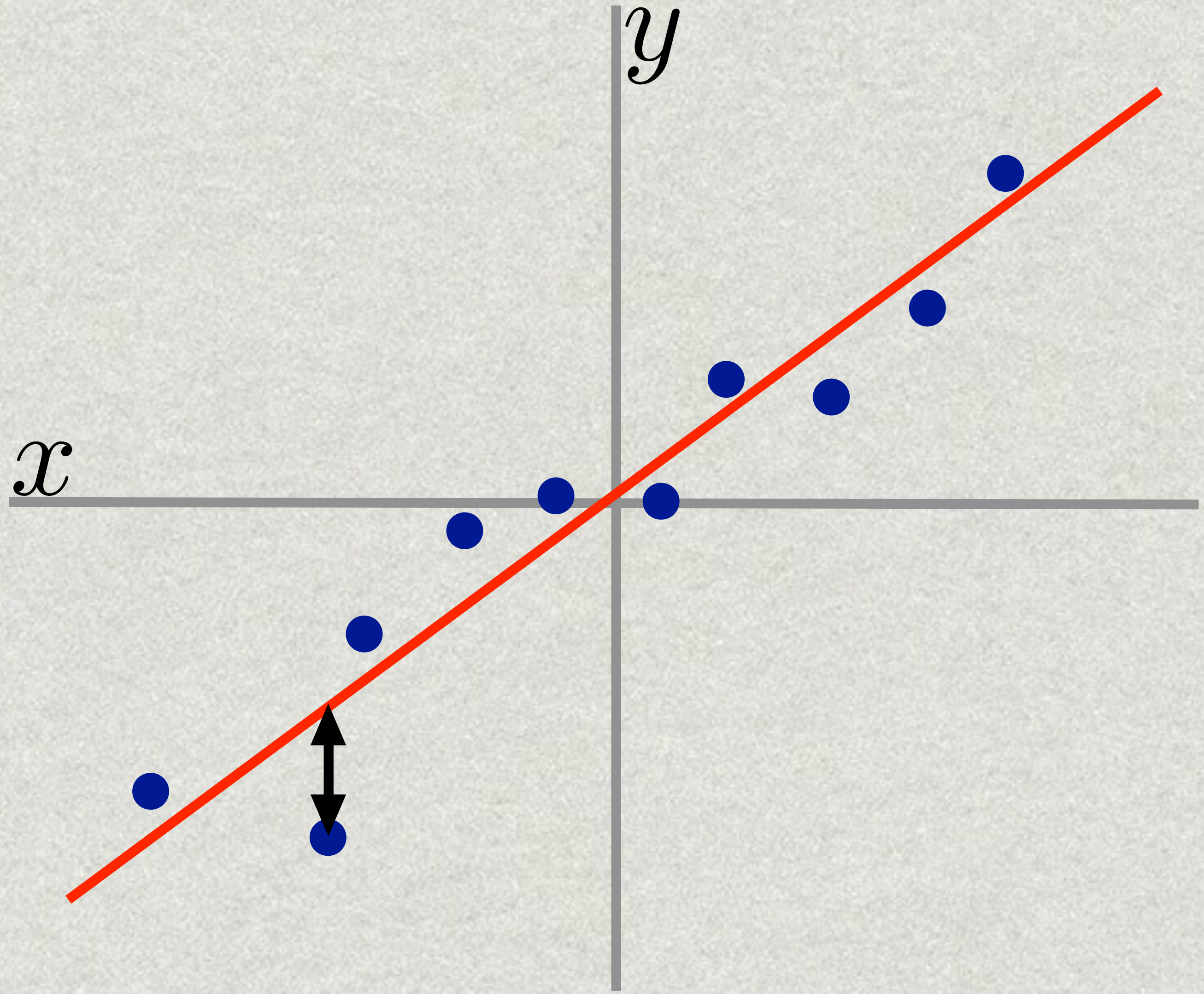
# Empirical Loss (Linear Regression)



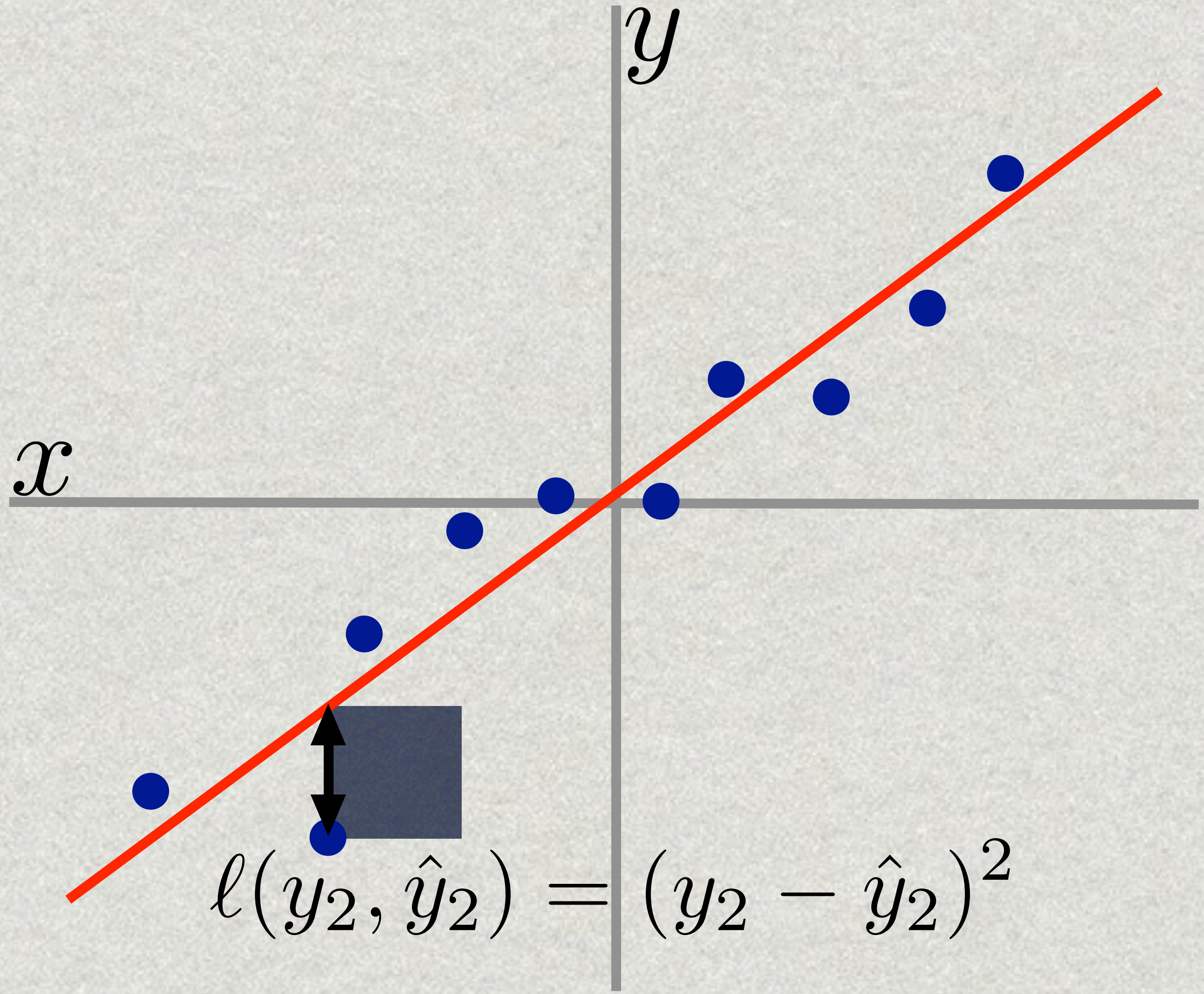
# Empirical Loss (Linear Regression)



# Empirical Loss (Linear Regression)

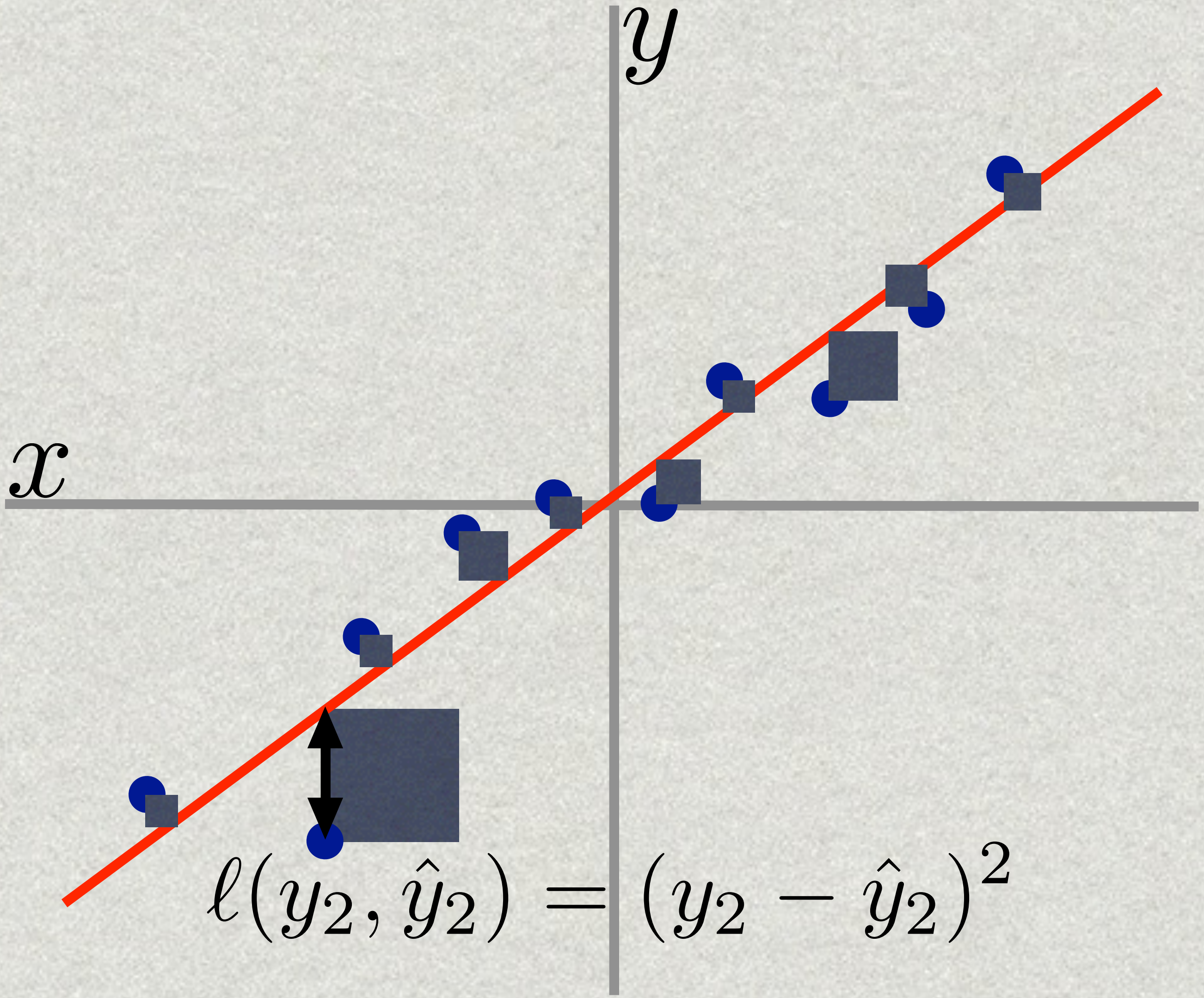


# Empirical Loss (Linear Regression)



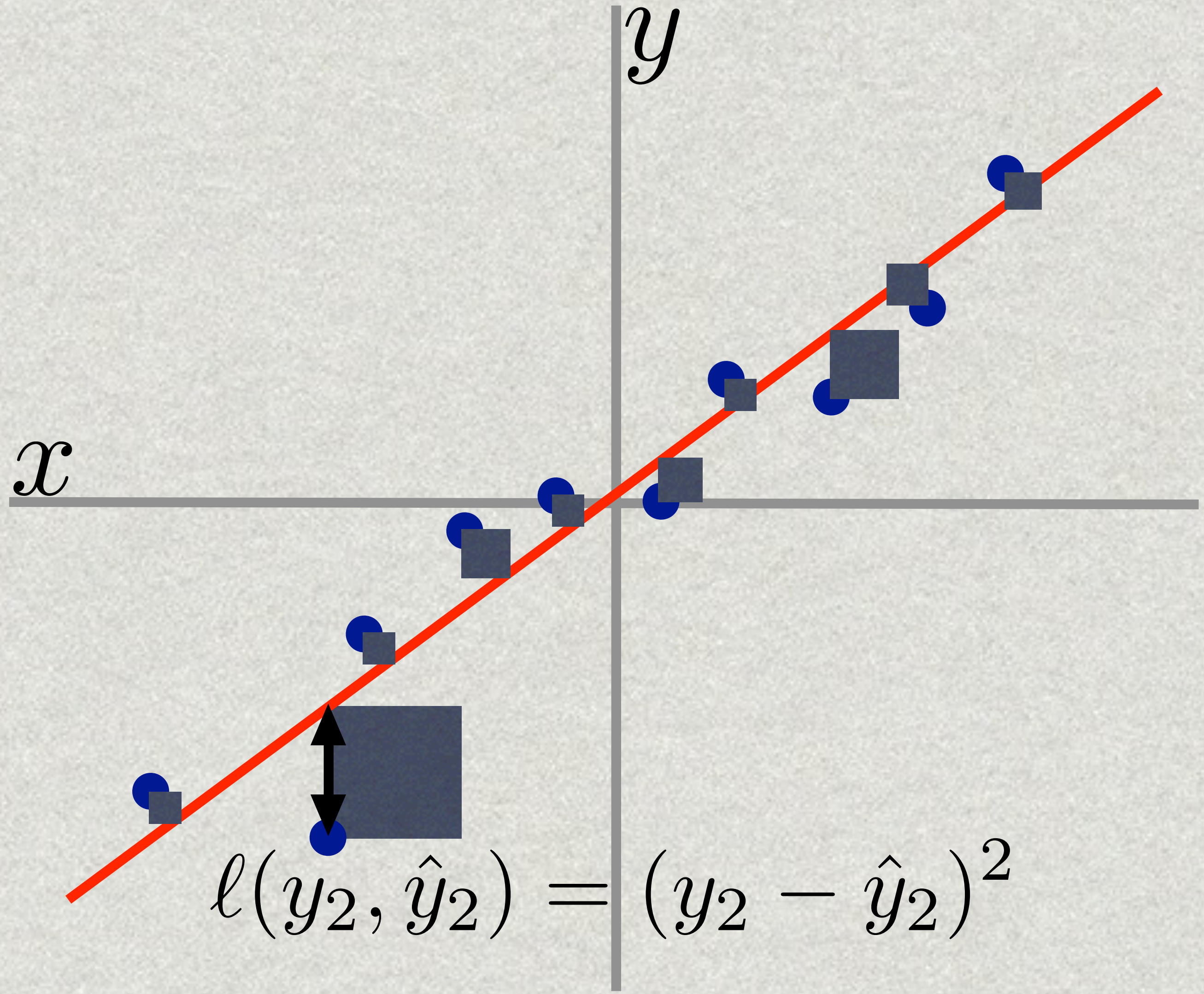
$$\ell(y_2, \hat{y}_2) = (y_2 - \hat{y}_2)^2$$

# Empirical Loss (Linear Regression)



$$\ell(y_2, \hat{y}_2) = (y_2 - \hat{y}_2)^2$$

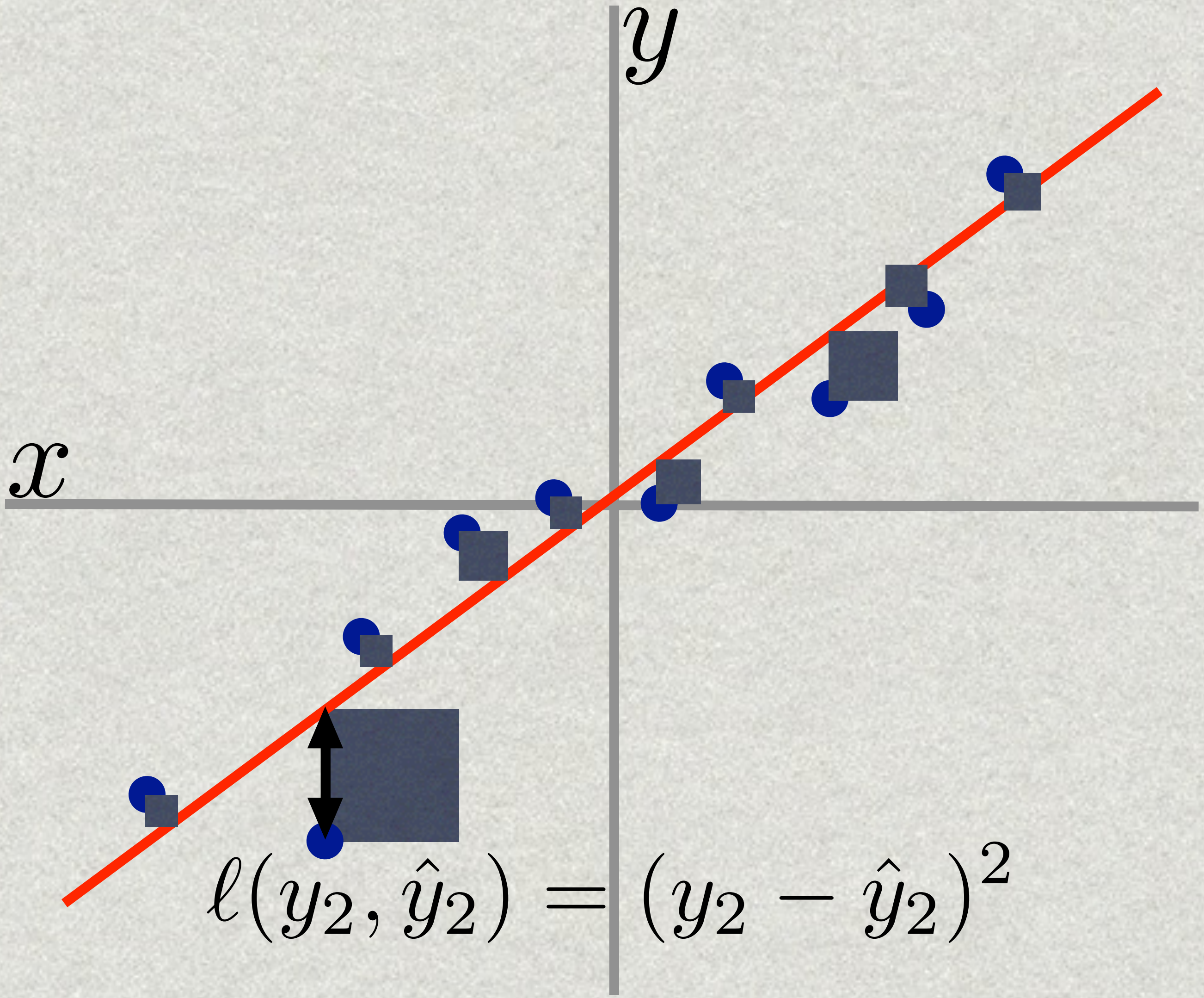
# Empirical Loss (Linear Regression)



$$\ell(y_2, \hat{y}_2) = (y_2 - \hat{y}_2)^2$$

$$\frac{1}{10}\left(\blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare + \blacksquare\right)$$

# Empirical Loss (Linear Regression)



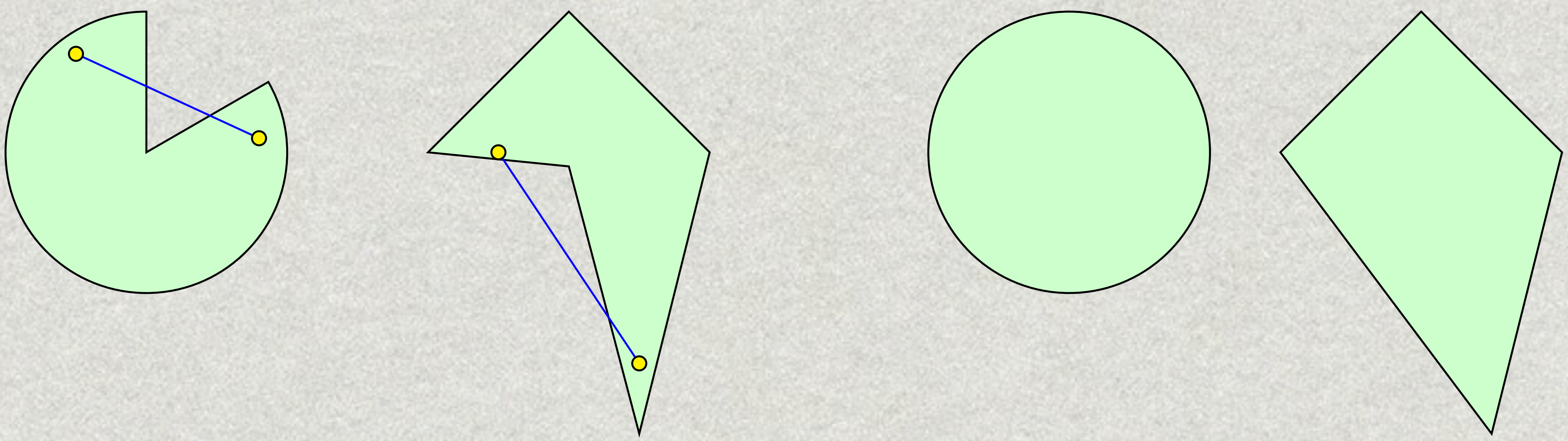
$$\ell(y_2, \hat{y}_2) = (y_2 - \hat{y}_2)^2$$

Empirical Loss

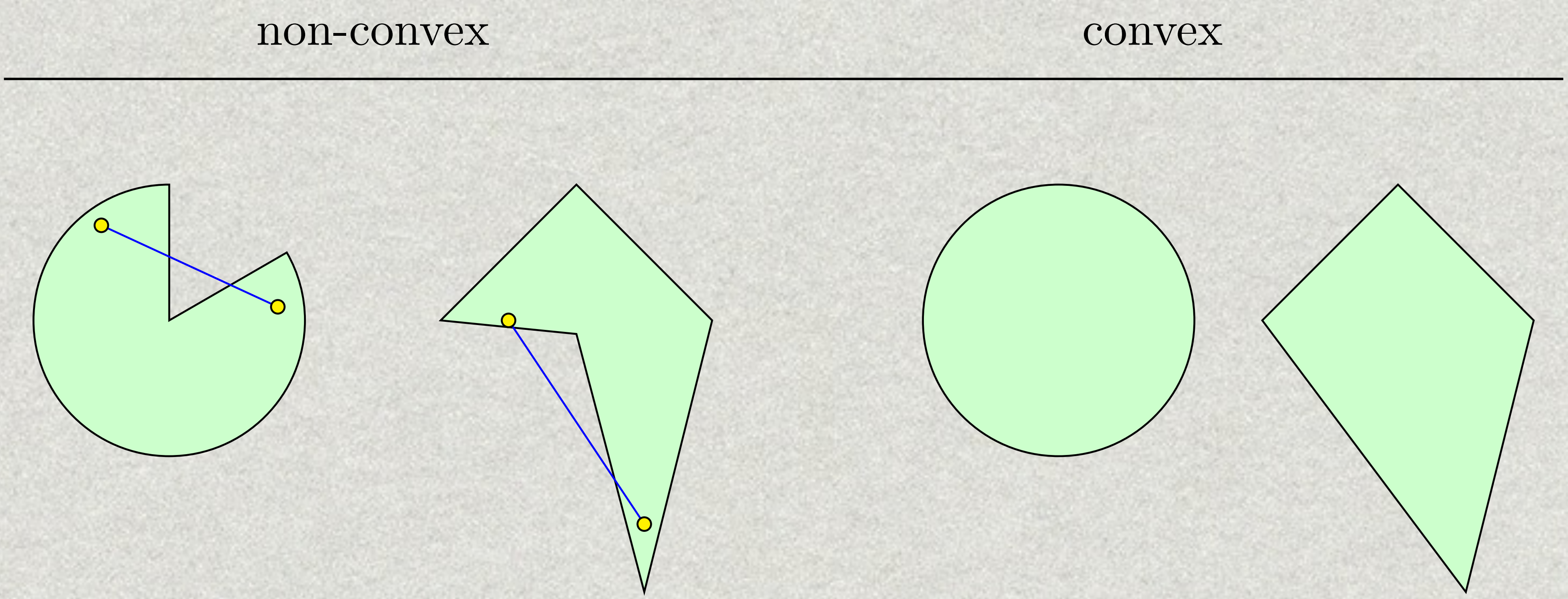
$$L(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, \mathbf{w} \cdot \mathbf{x}_i)$$

# Convex Losses

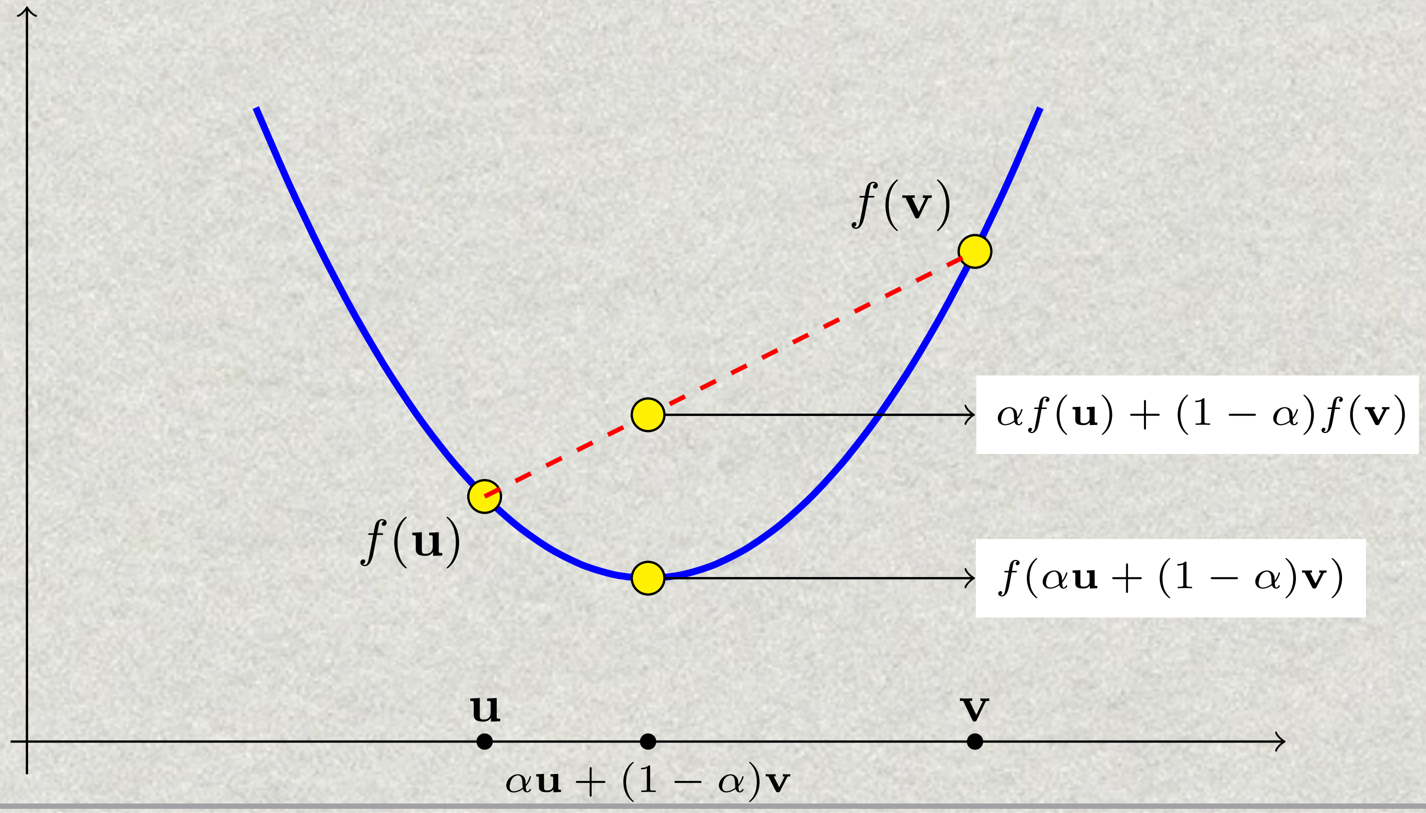
non-convex convex



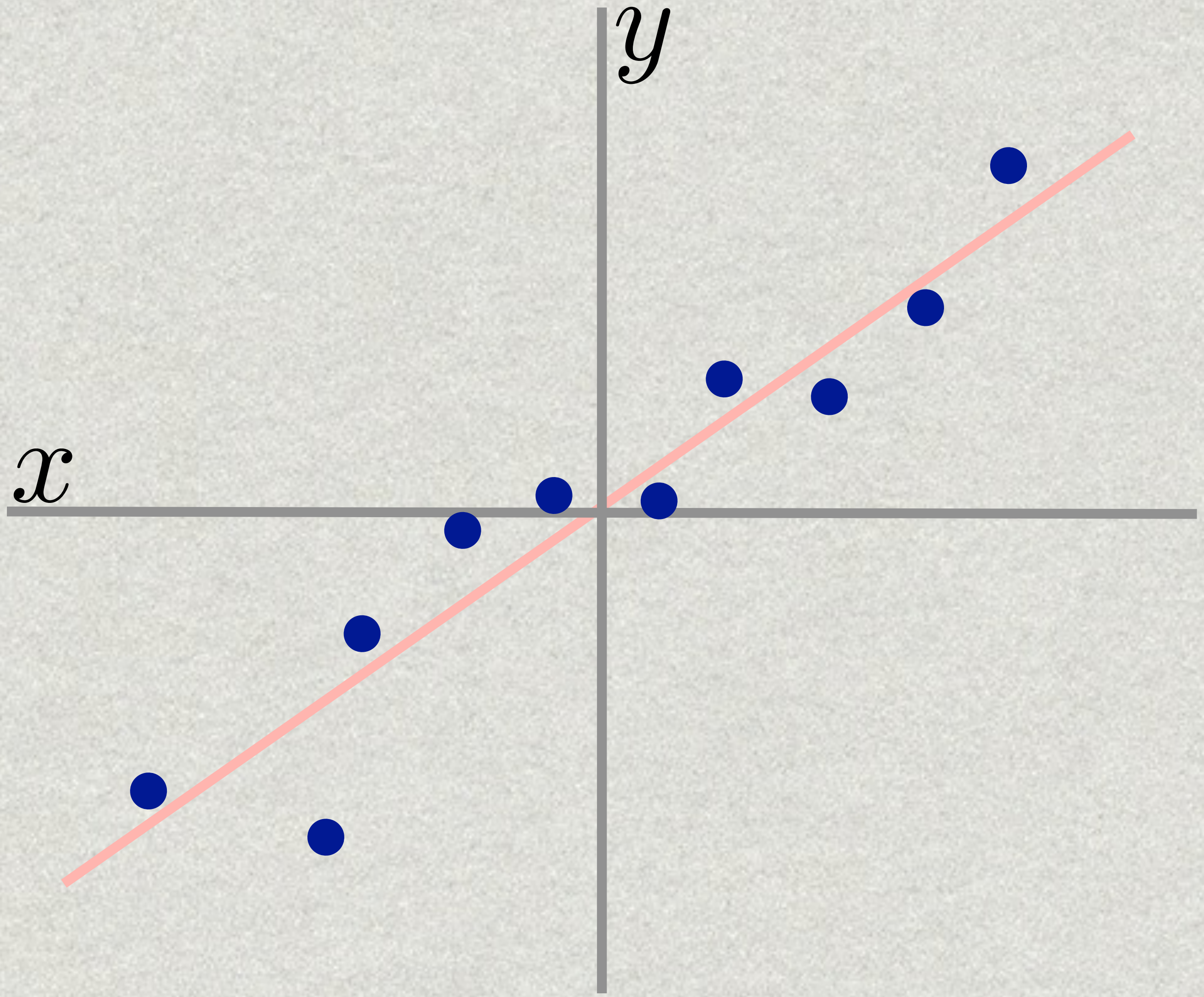
# Convex Losses



$$f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v}) \;\leq\; \alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v})$$

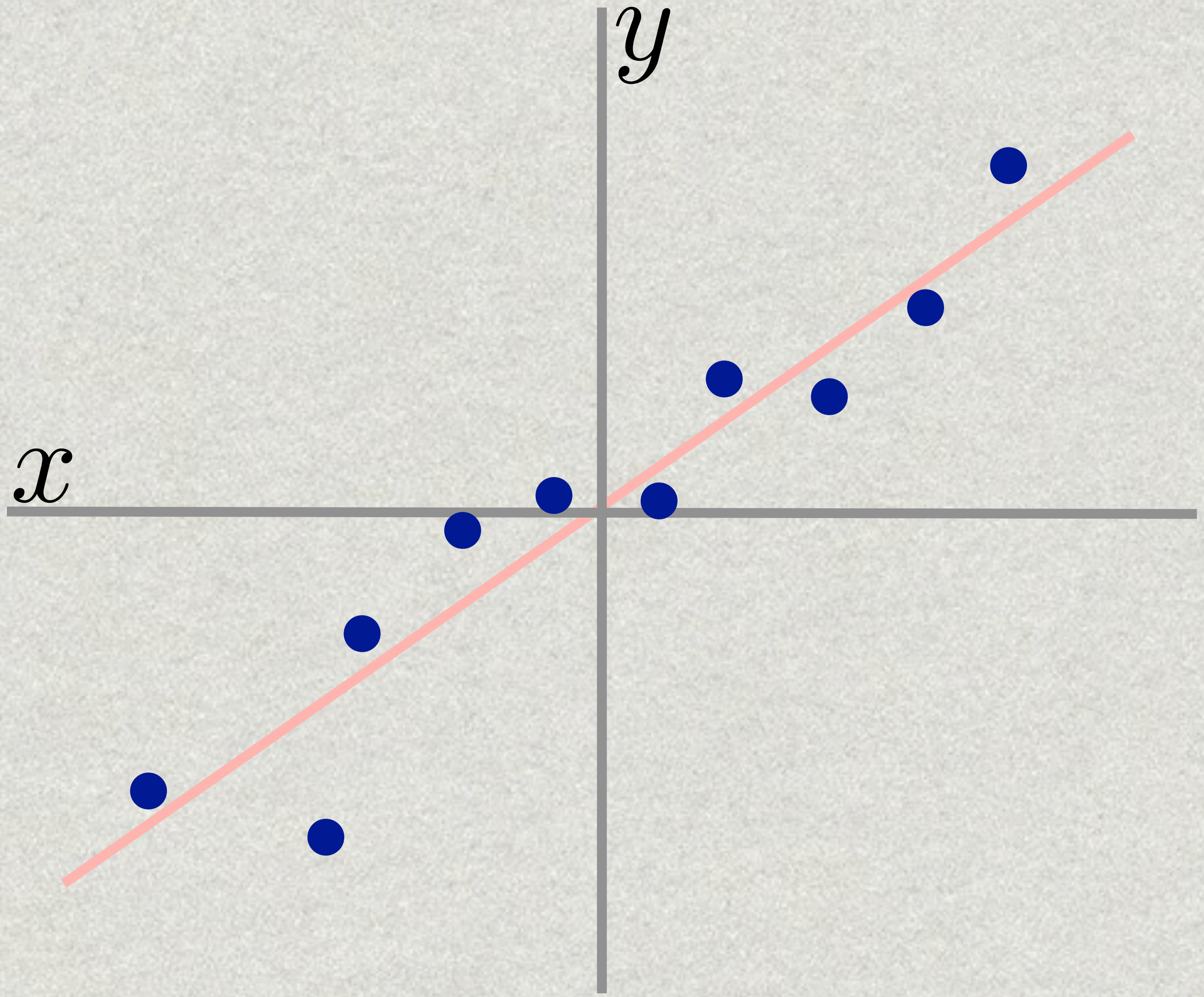

$f(\mathbf{v})$
$\alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v})$
$f(\mathbf{u})$
$f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v})$
$\mathbf{u}$
$\mathbf{v}$
$\alpha\mathbf{u} + (1-\alpha)\mathbf{v}$

# Overfitting



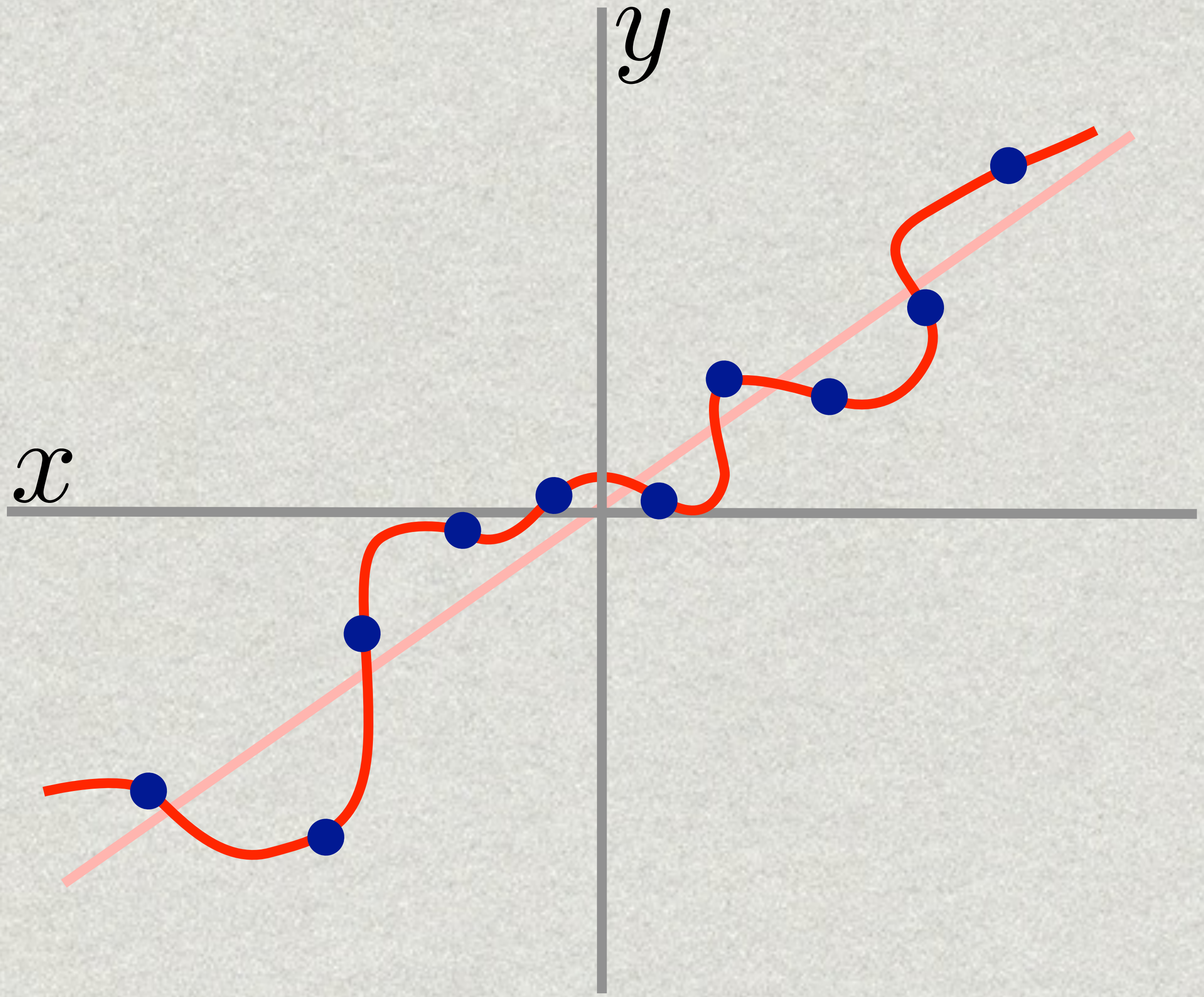
# Overfitting

Suppose we were able to fit a spline function to the data

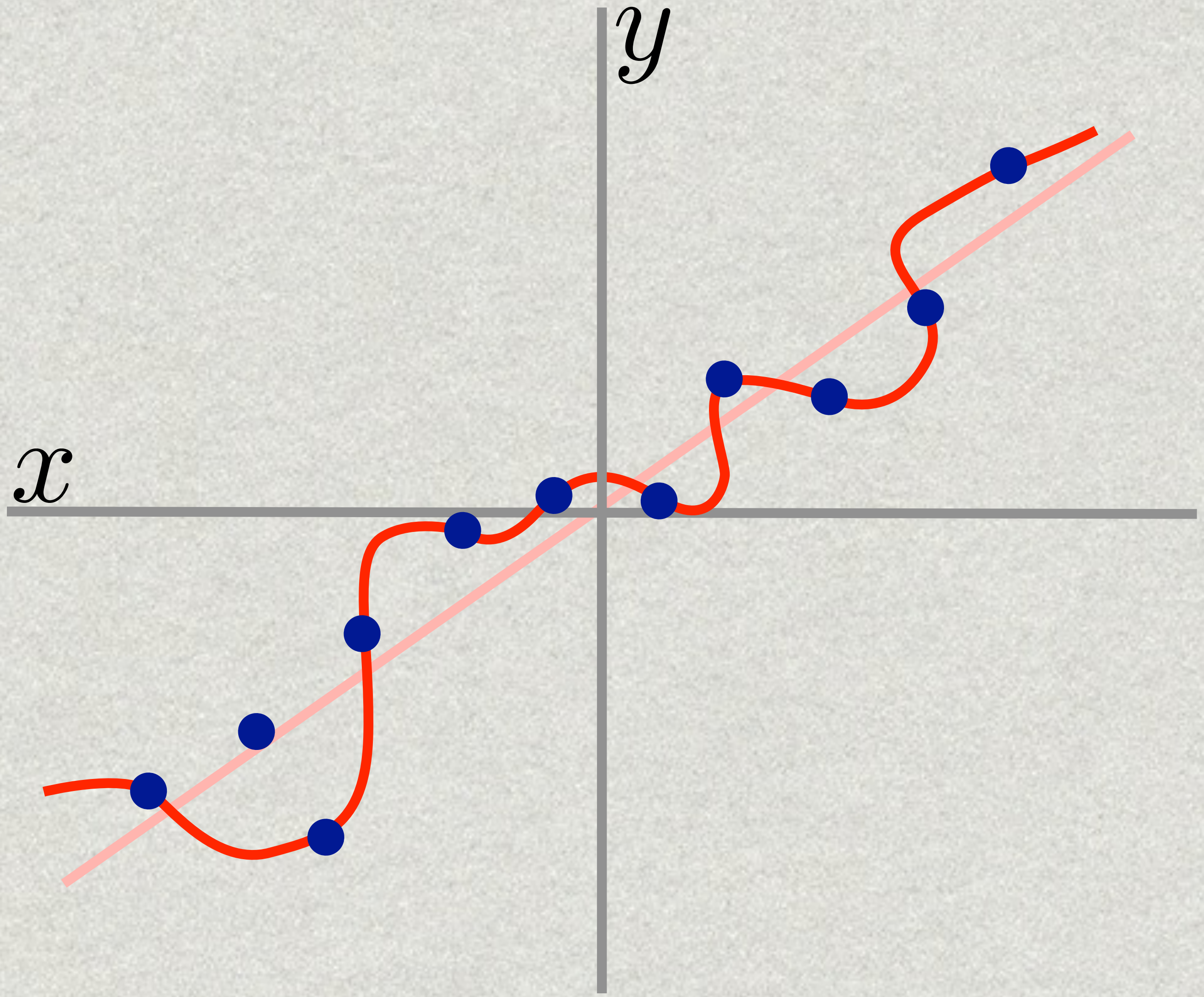


# Overfitting

Then the empirical loss ***L(w)*** would be 0



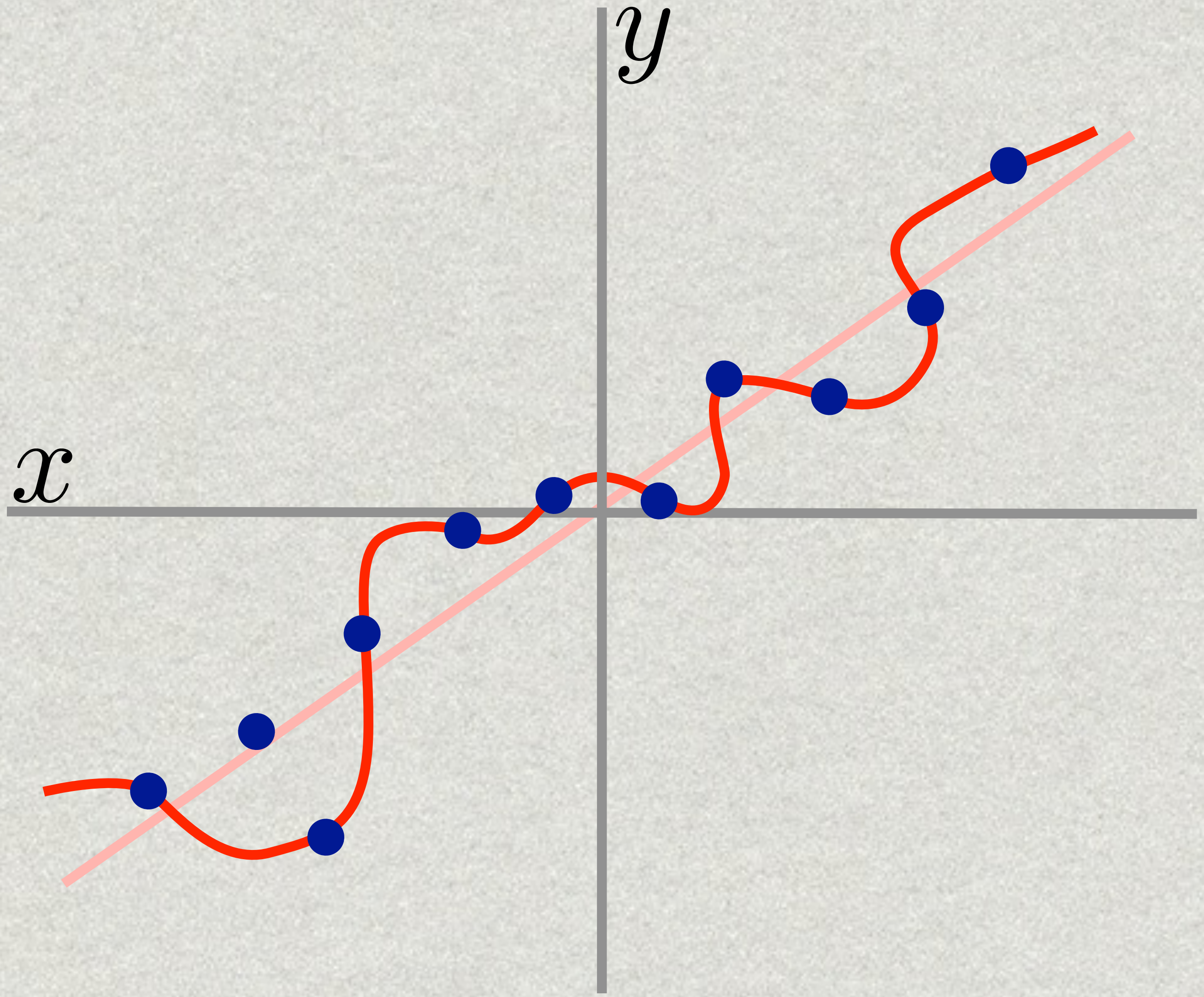
# Overfitting



However new data points are unlikely to reside on the piecewise linear curve

“*overfitting*”

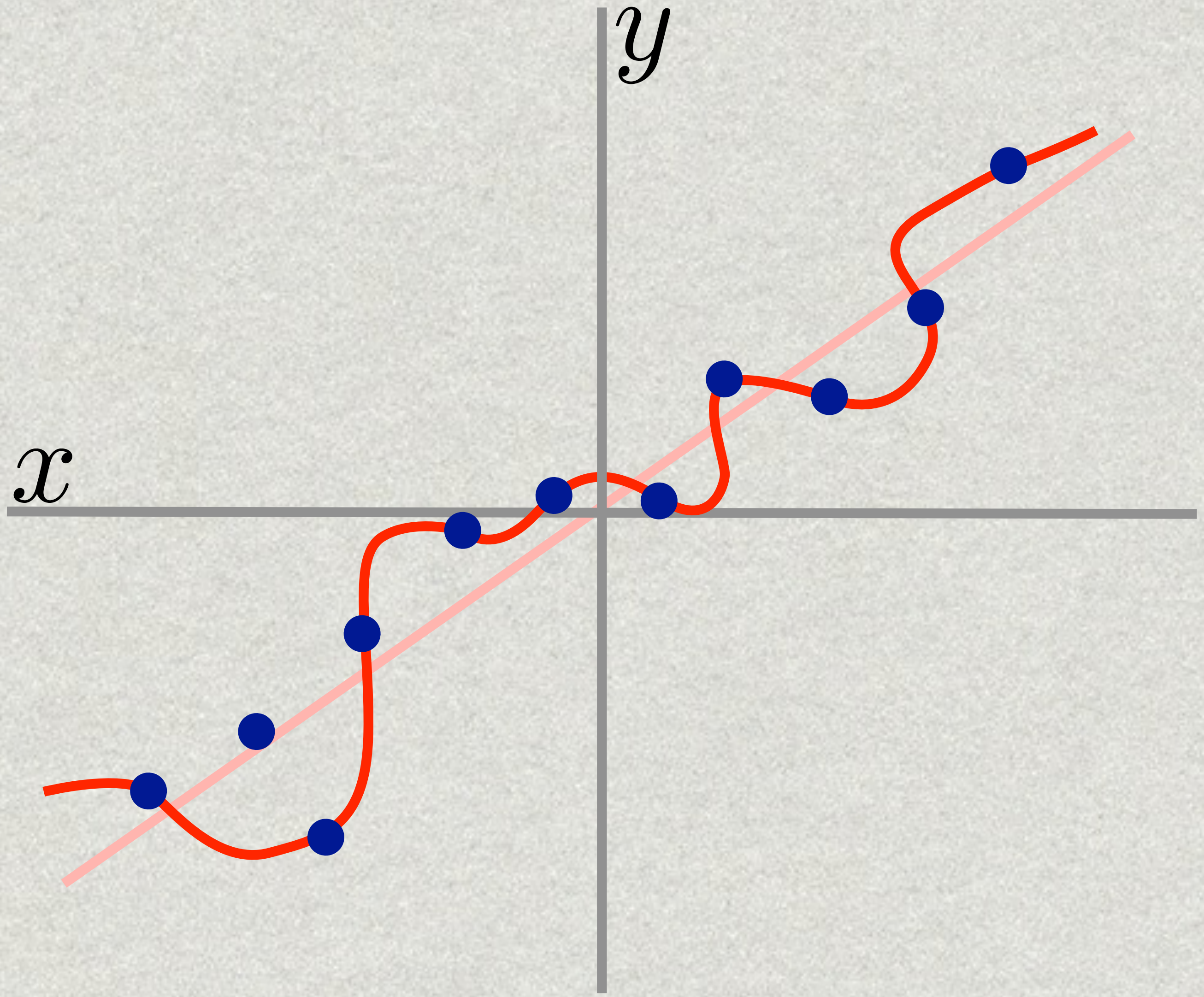
# Overfitting



But, is it possible to overfit with a linear model?

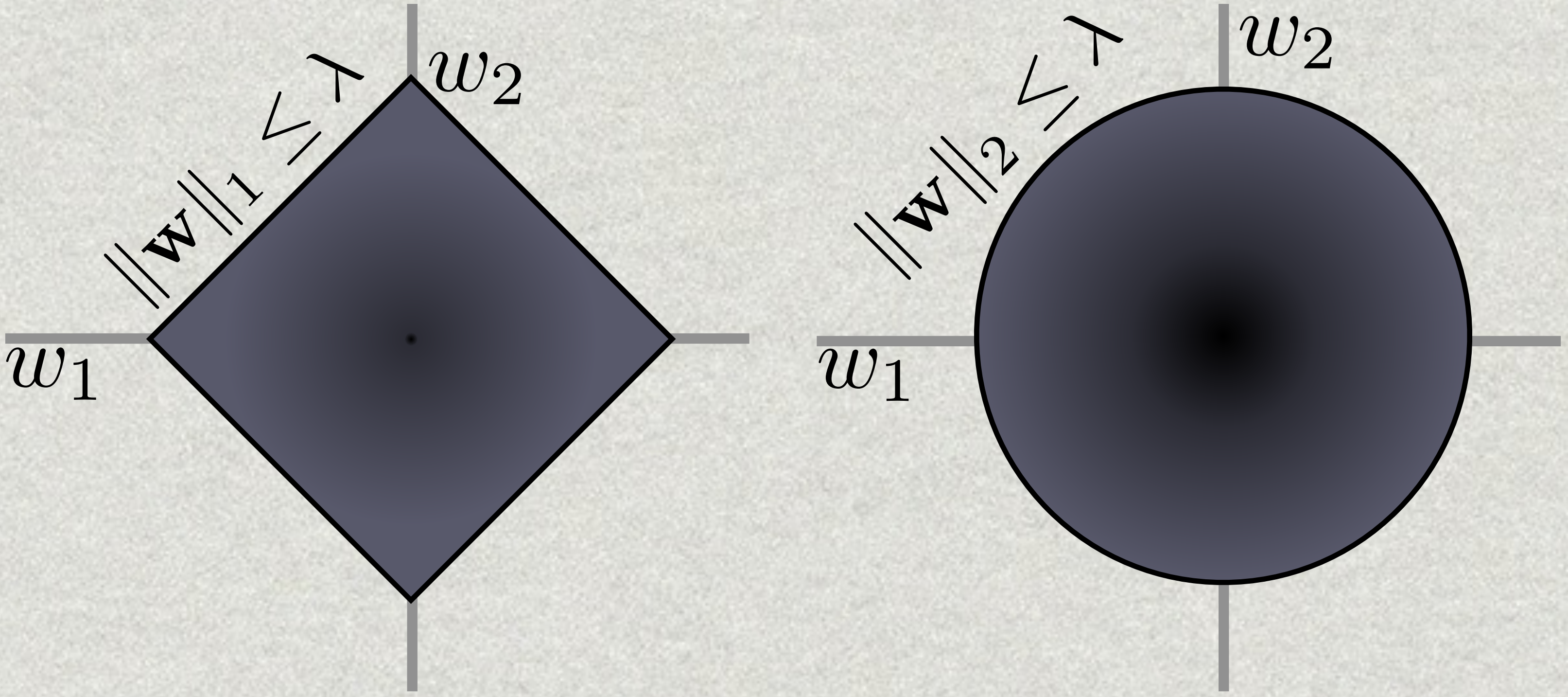
# Overfitting

Yes, when number of features is very large & many are irrelevant

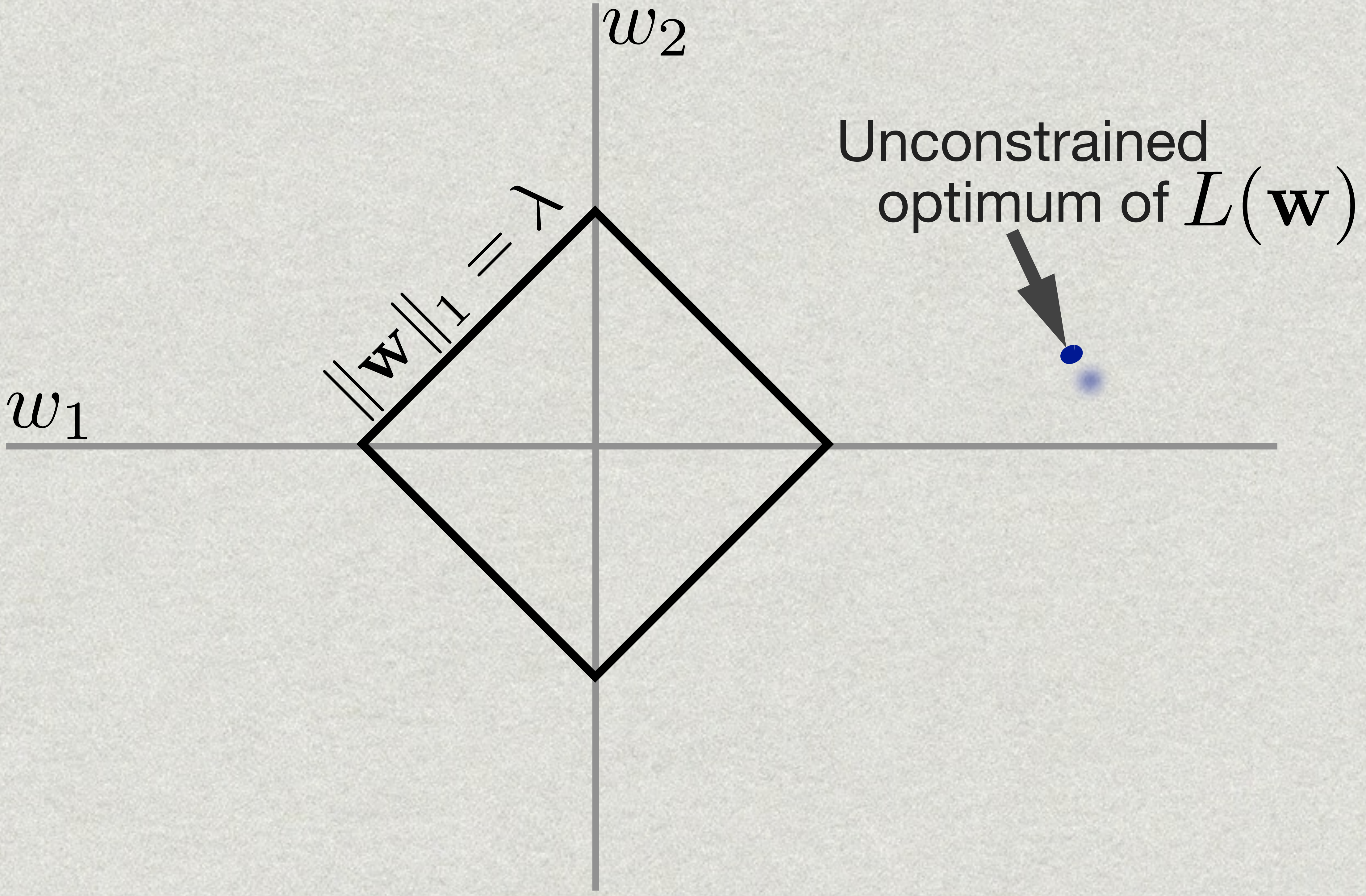


# Preventing Overfitting

- To prevent overfitting we need to constrain the volume of the space of the possible linear predictors
- A common approach is to limit the *p-norm* of **W**

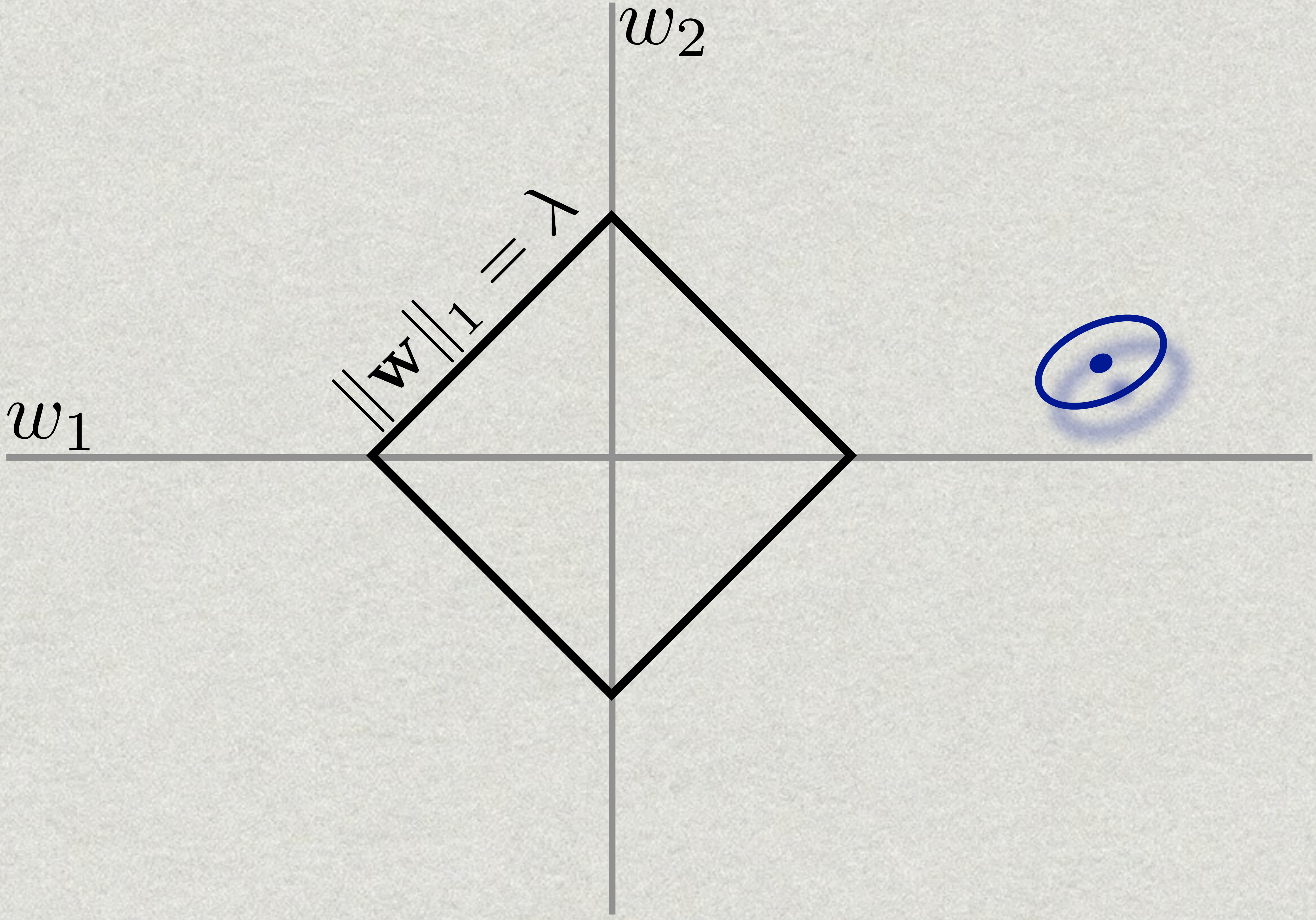

$\|\mathbf{w}\|_1 \leq \lambda$
$w_2$
$w_1$
$\|\mathbf{w}\|_2 \leq \lambda$
$w_2$
$w_1$

# Achieving Sparsity using 1-norm

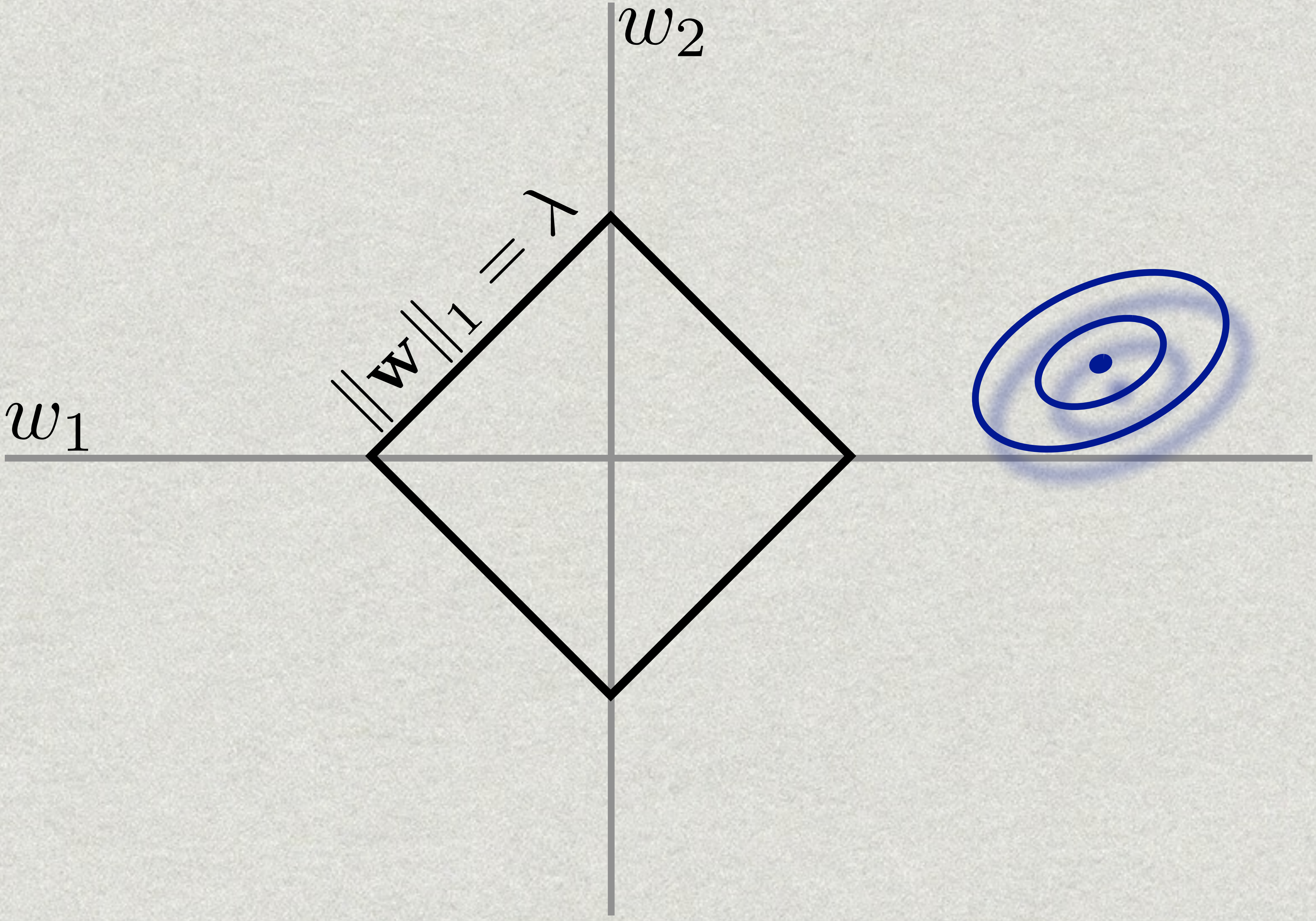


Unconstrained optimum of $L(\mathbf{w})$

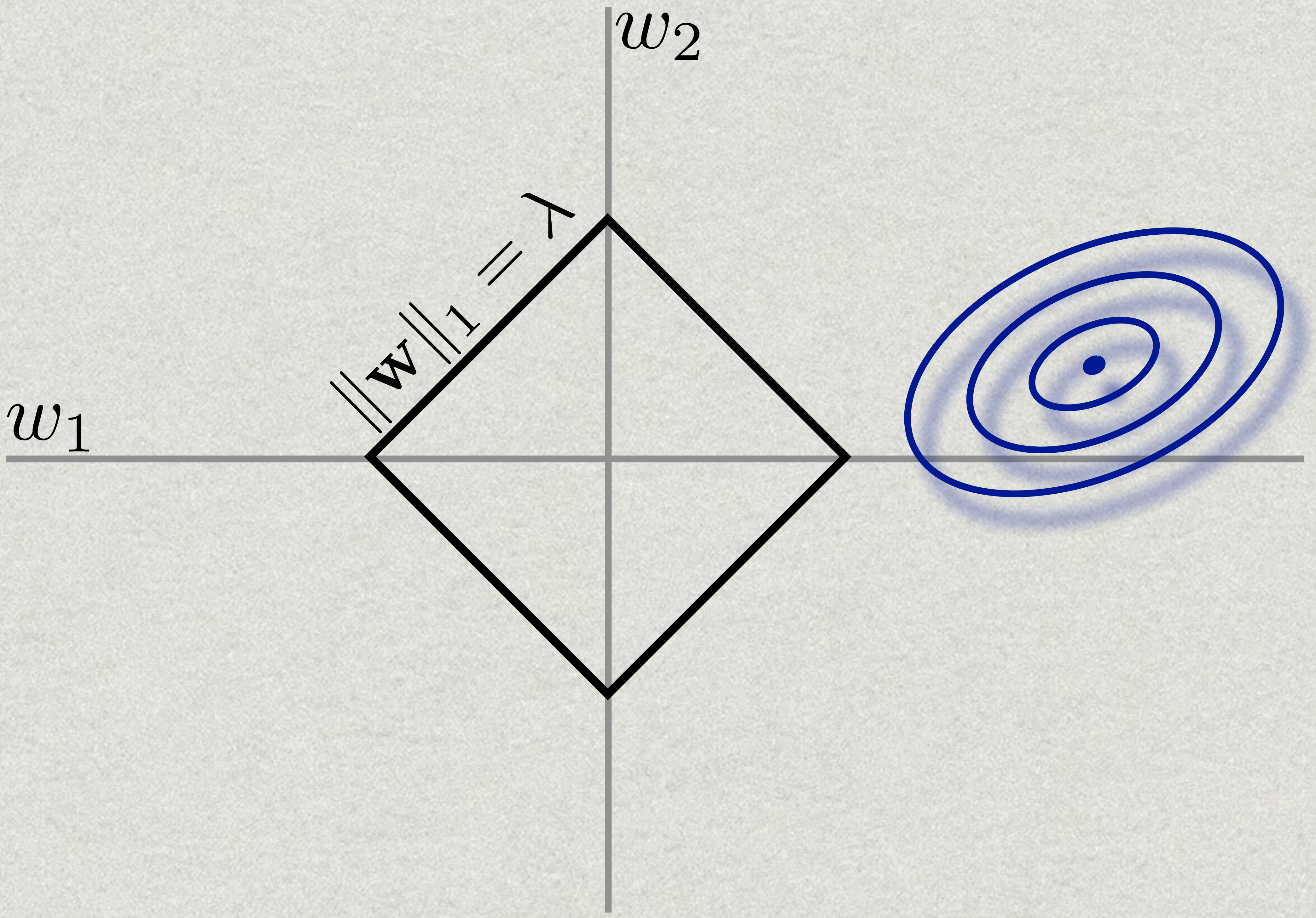
# Achieving Sparsity using 1-norm



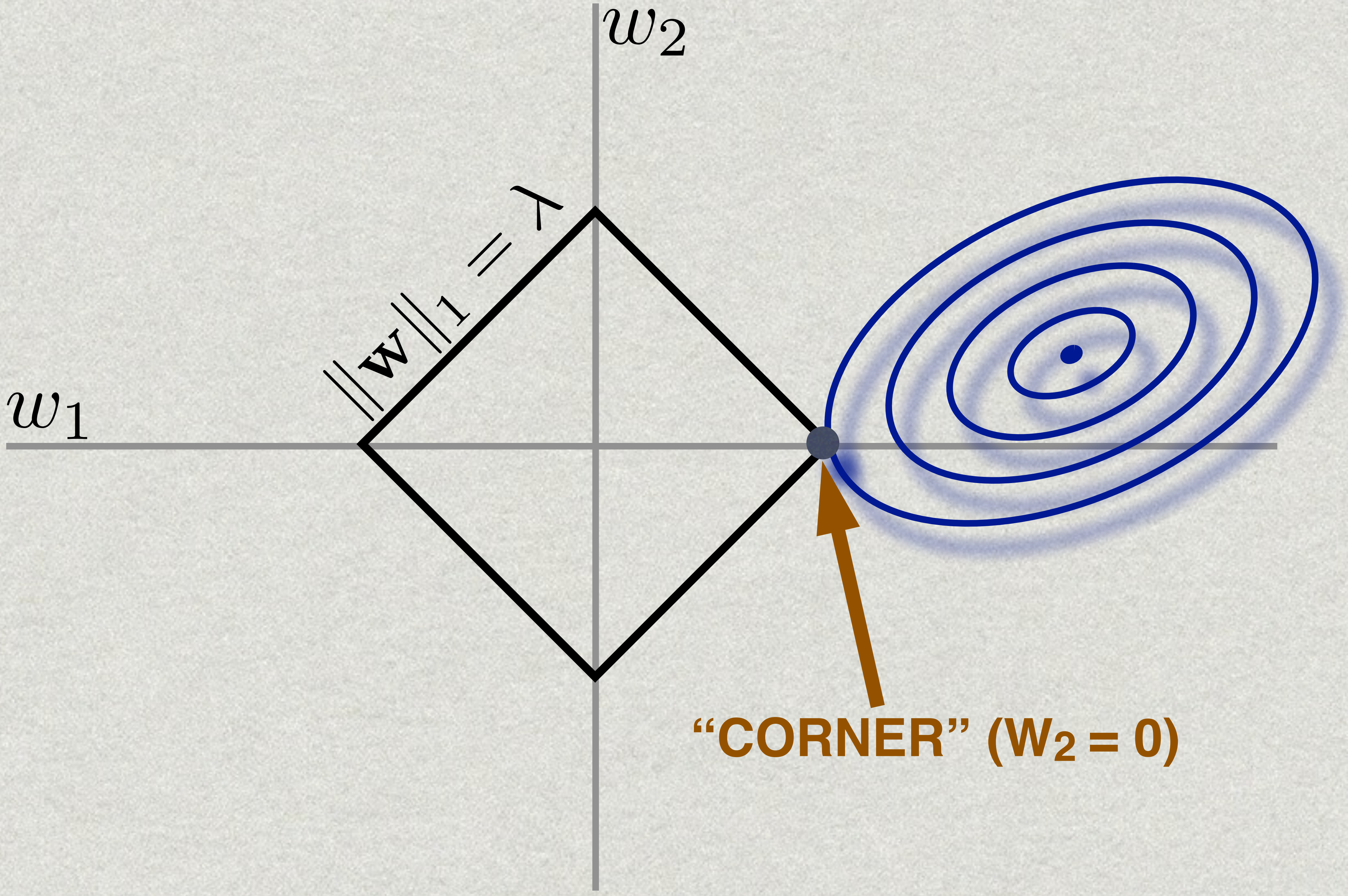
# Achieving Sparsity using 1-norm



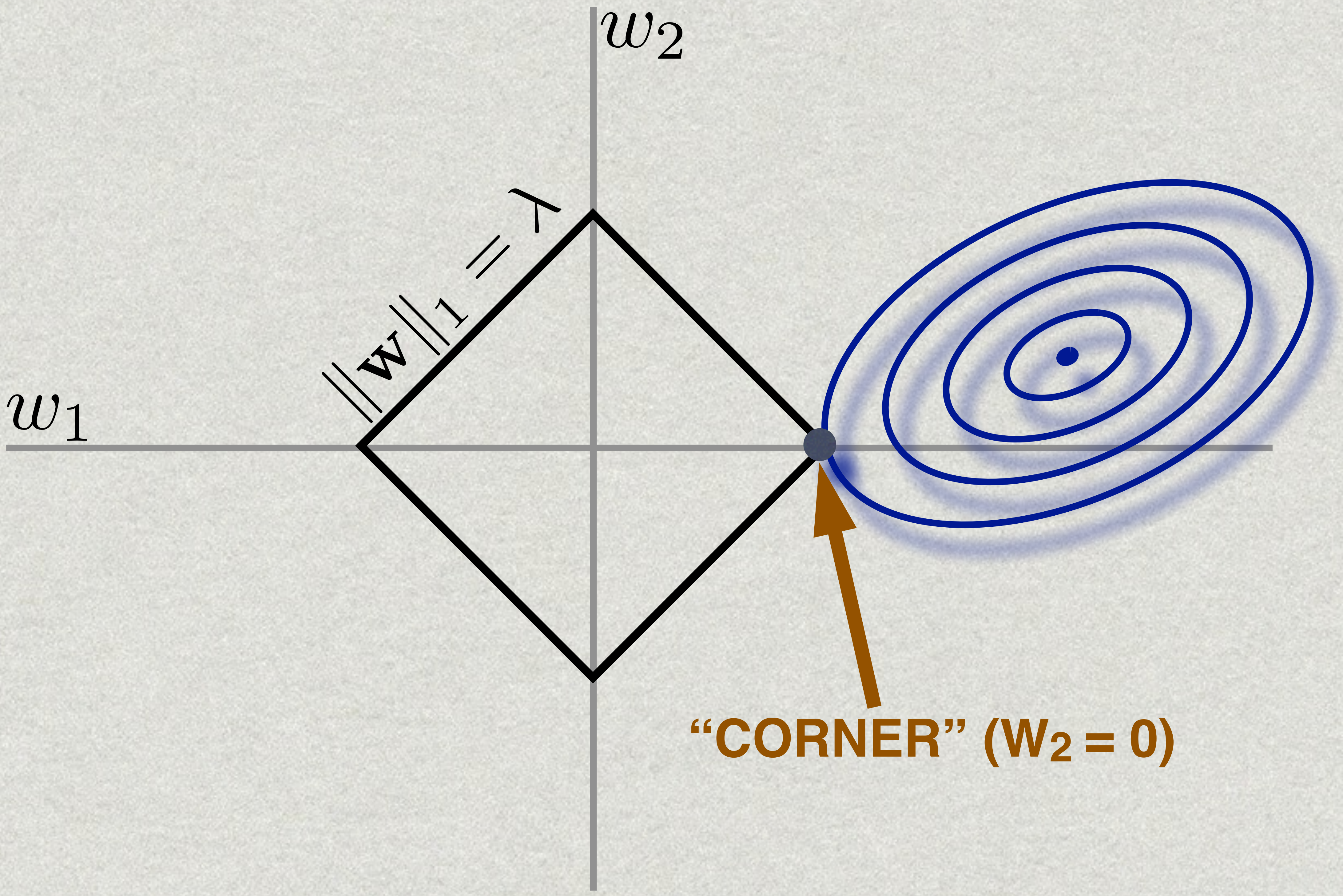
# Achieving Sparsity using 1-norm



# Achieving Sparsity using 1-norm



# Achieving Sparsity using 1-norm

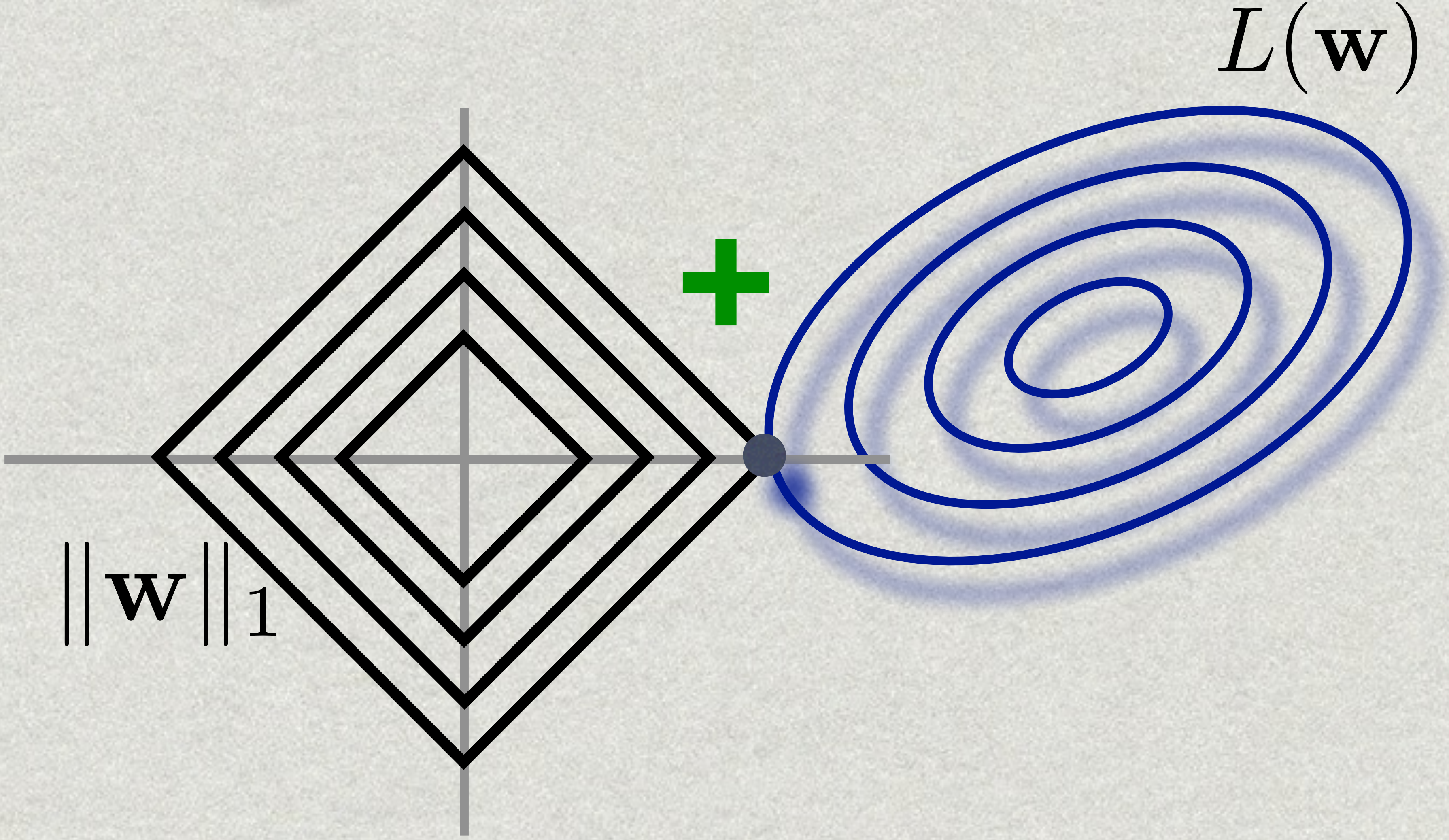


“CORNER” ($W_2 = 0$)

See e.g. Candes’06, Donoho’06

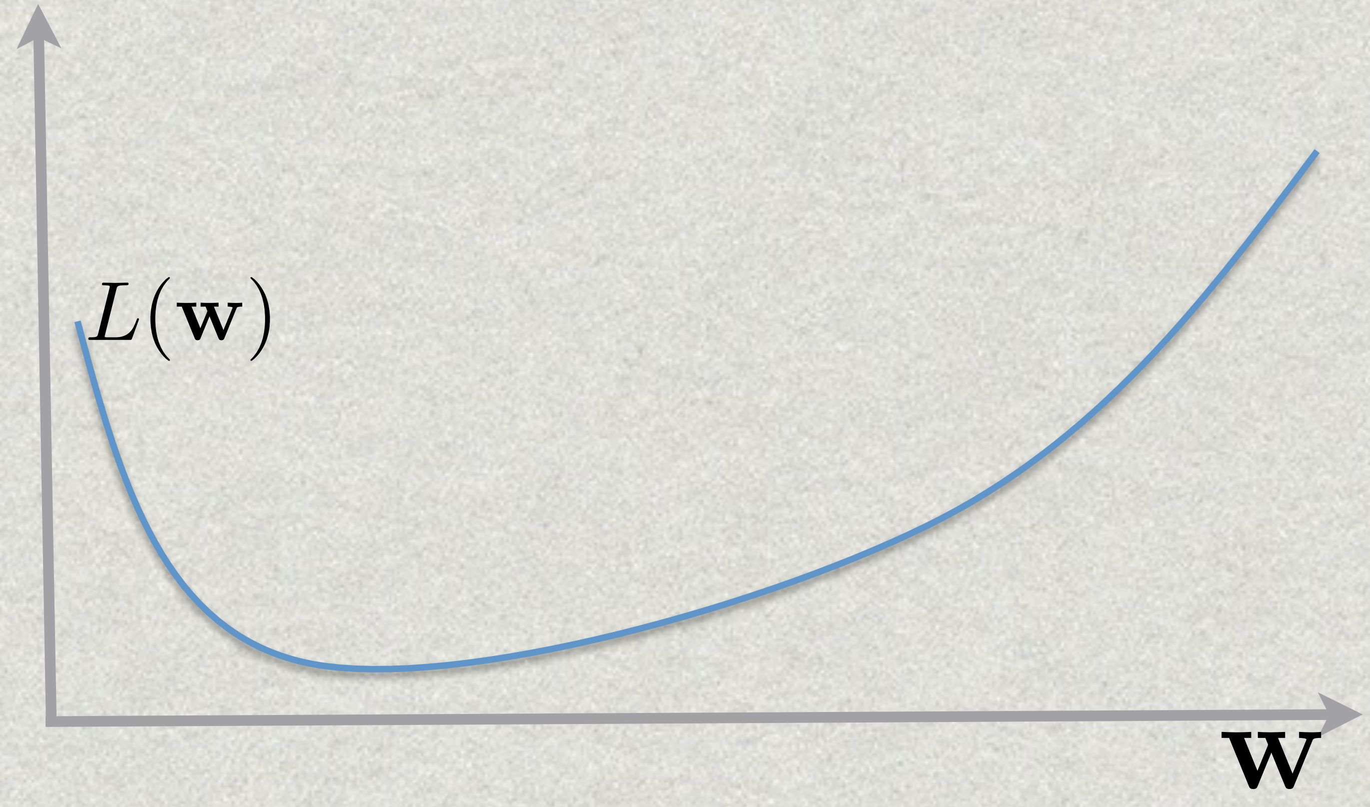
# Penalized Form

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

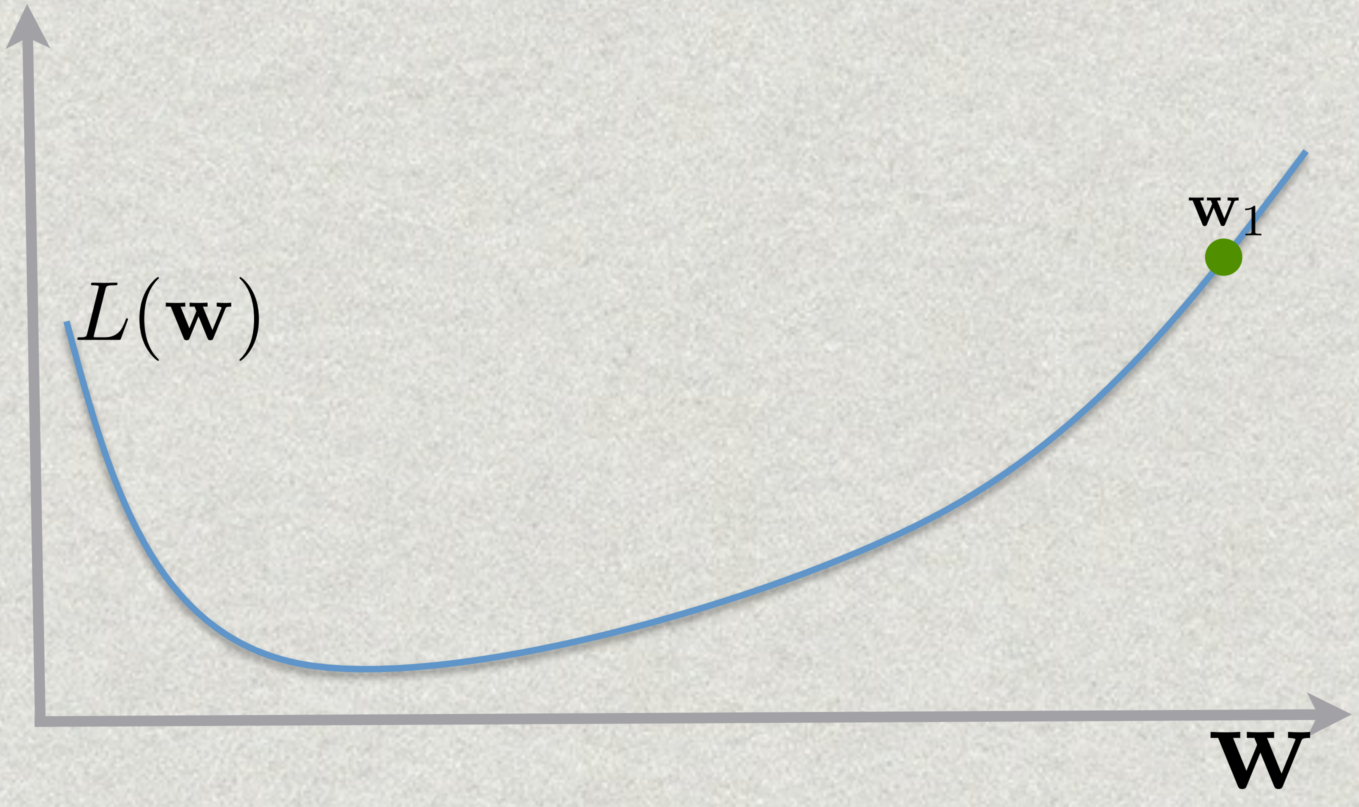


**Sparsity Properties Are Analogous to Constrained Form**

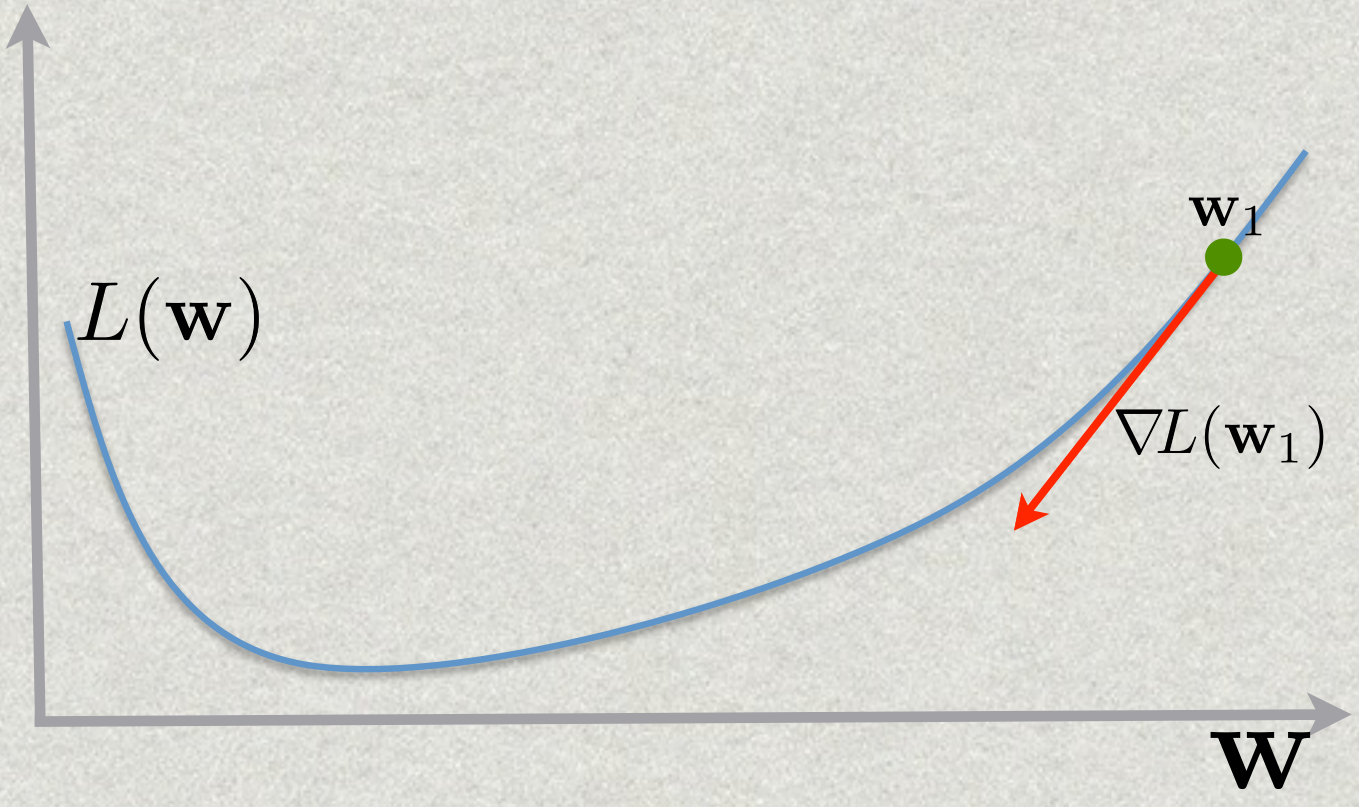
# Gradient Descent



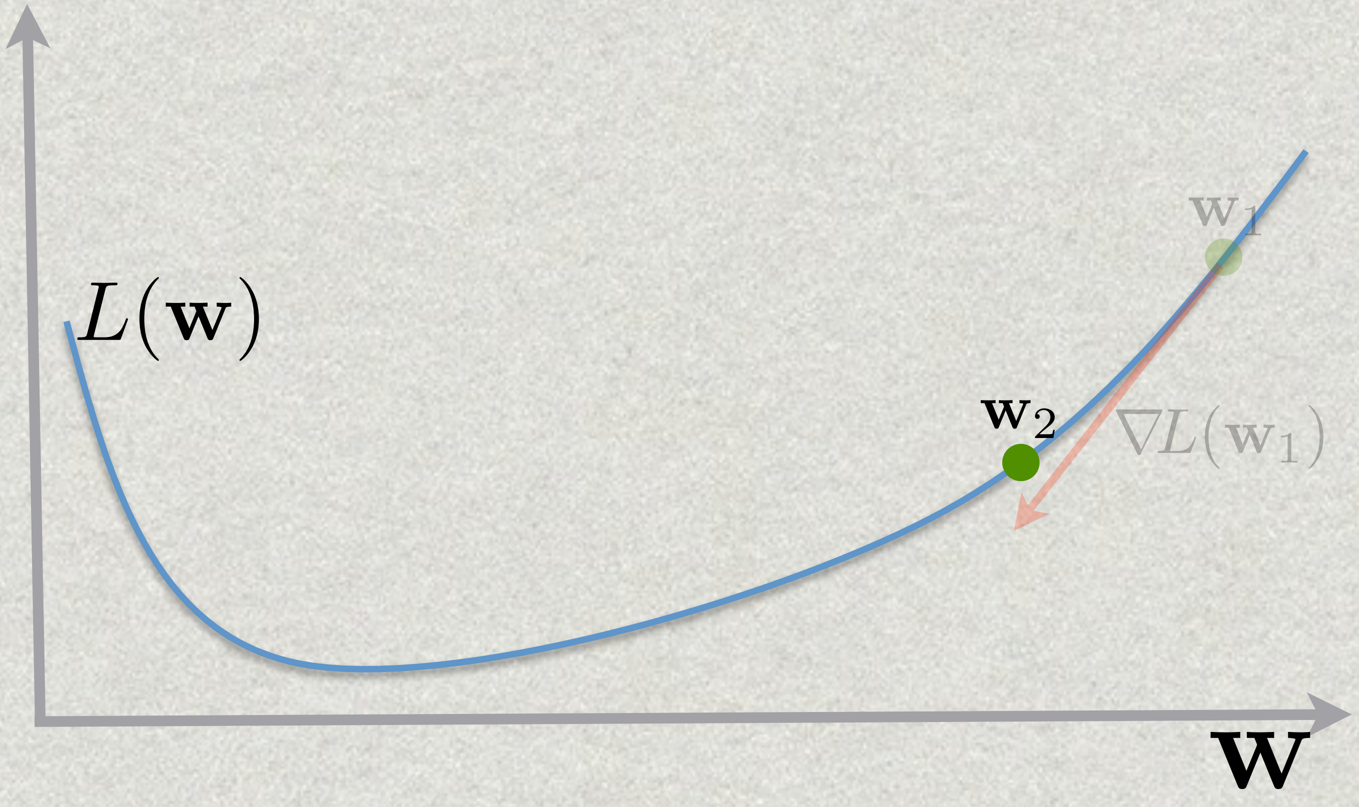
# Gradient Descent



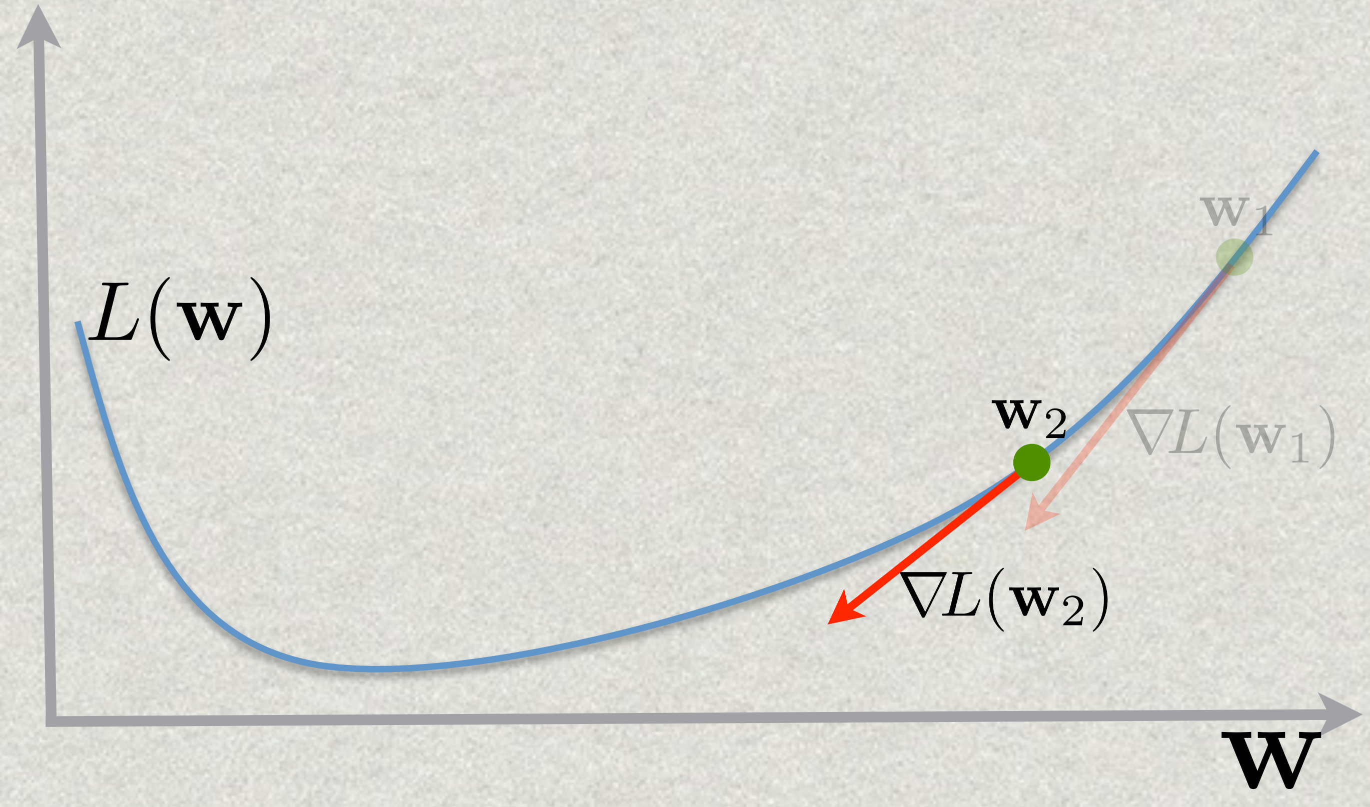
# Gradient Descent



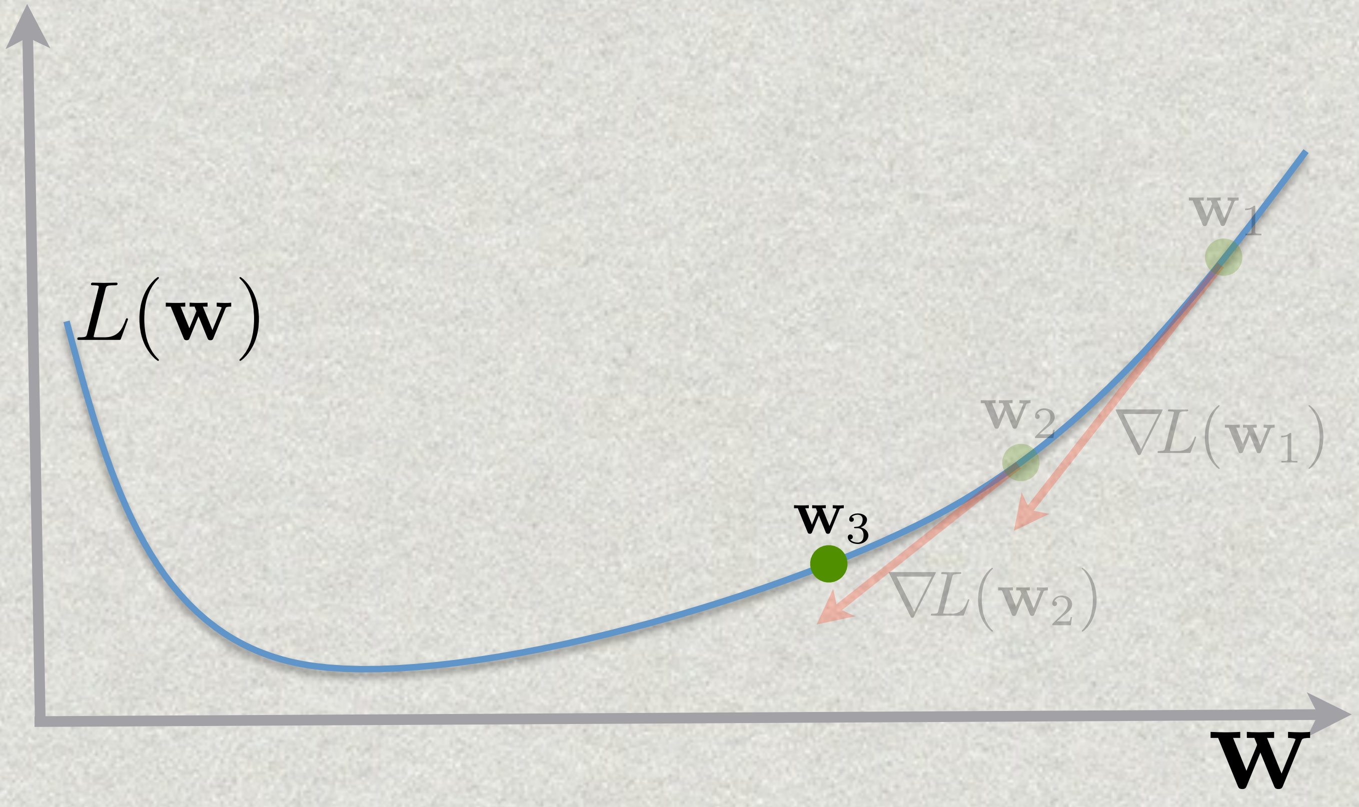
# Gradient Descent



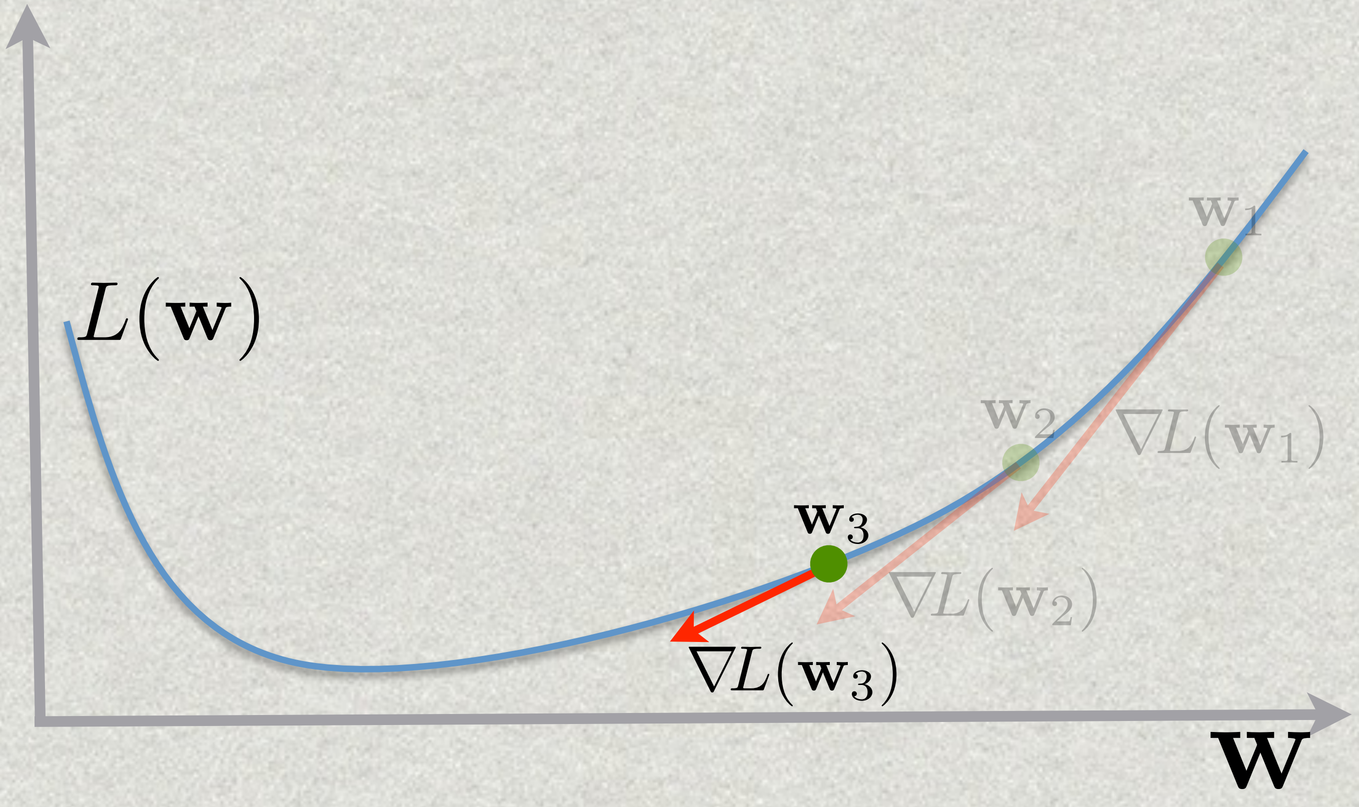
# Gradient Descent



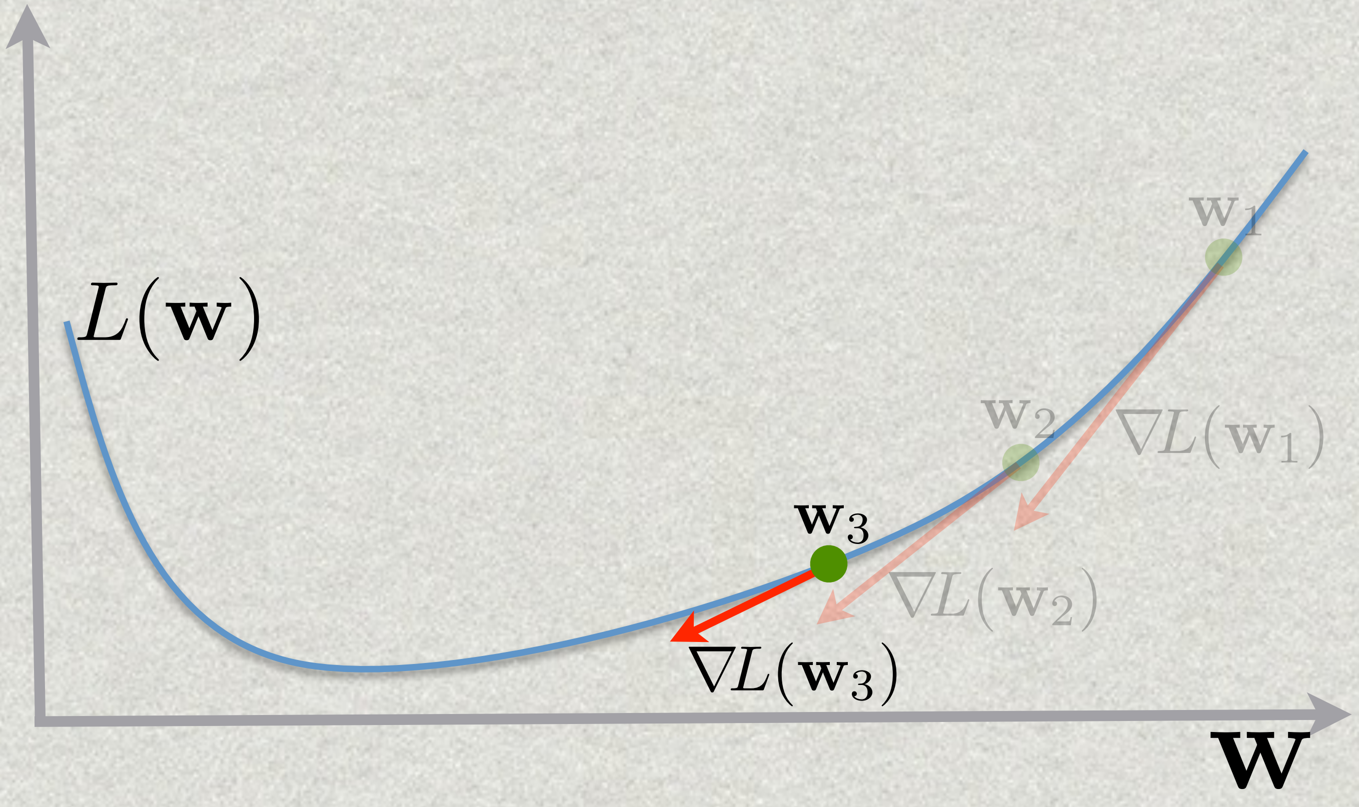
# Gradient Descent



# Gradient Descent



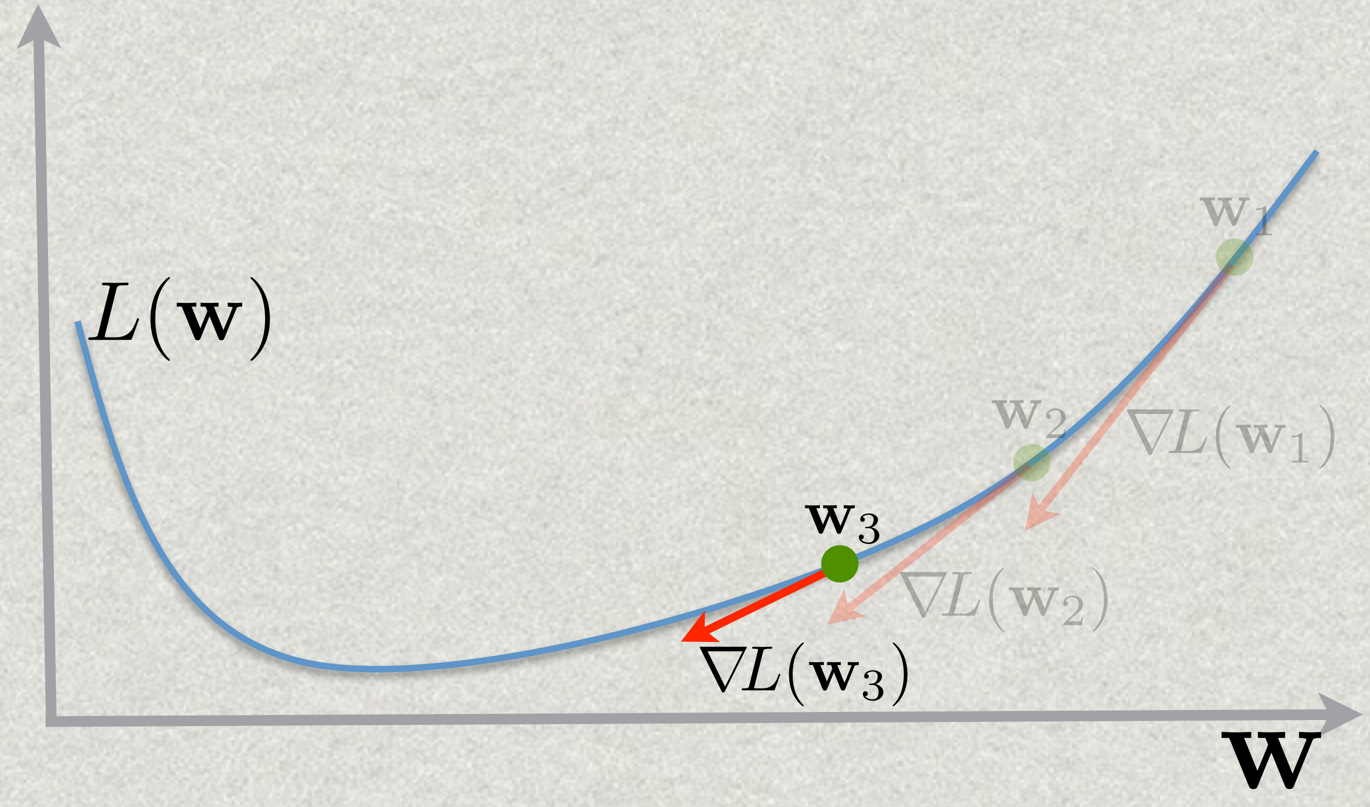
# Gradient Descent



- Gradient descent main loop:
  - Compute gradient $\nabla_t L = \frac{1}{|S|} \sum_{i \in S} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t; (\mathbf{x}_i, y_i))$
  - Update

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_t L$$

# Gradient Descent

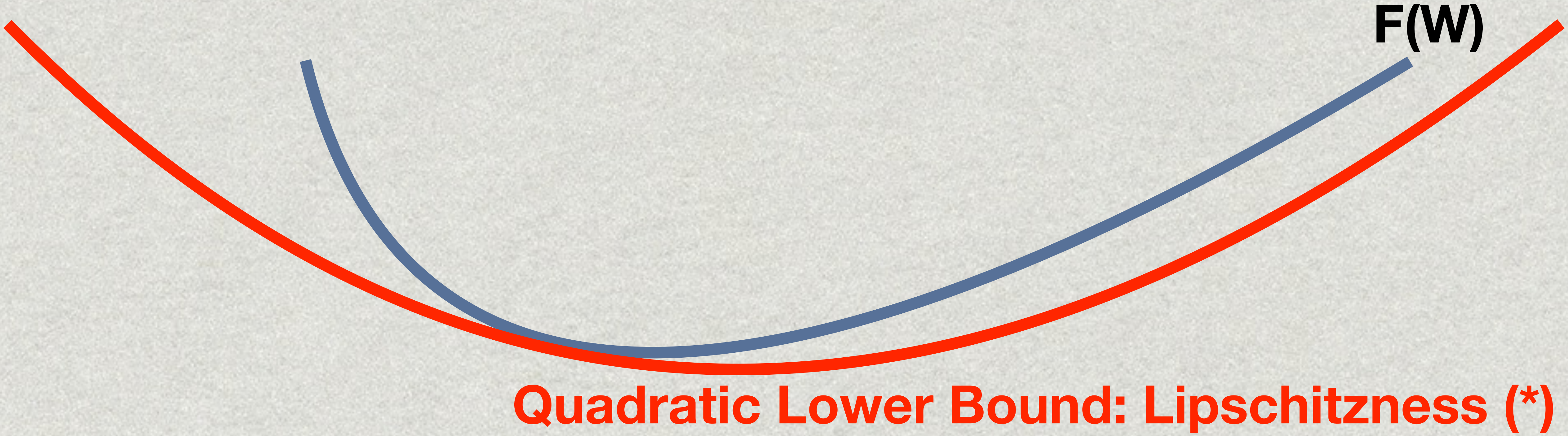


- Gradient descent main loop:
  - Compute gradient $$\nabla_t L = \frac{1}{|S|} \sum_{i \in S} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t; (\mathbf{x}_i, y_i))$$
  - Update

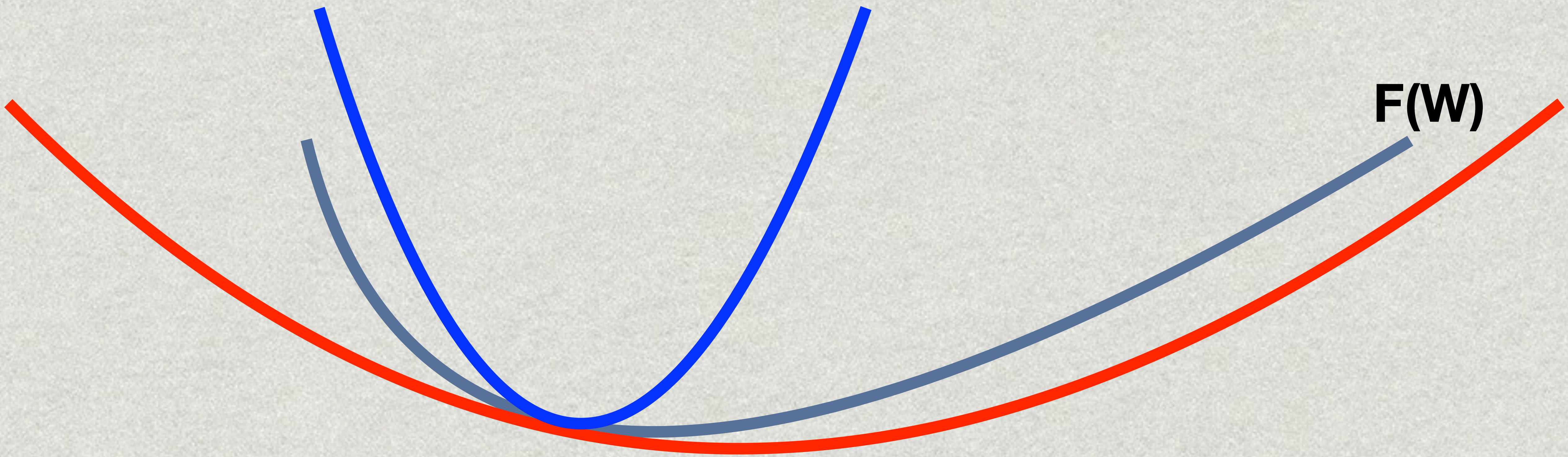
$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \overset{\text{STEP SIZE}}{\boxed{\eta_t}} \nabla_t L$$

# Lipschitz & Smooth Convex Losses

# Lipschitz & Smooth Convex Losses



# Lipschitz & Smooth Convex Losses



Quadratic Upper Bound: Smoothness

F(W)

Quadratic Lower Bound: Lipschitzness (*)

# Lipschitz Losses

- Domain $\Omega \subset \mathbb{R}^d$  Loss function: $\mathcal{L} : \mathbb{R}^d \rightarrow \mathbb{R}_+$

- Lipschitz losses change sufficiently "slow"

$$\beta - \text{Lipschitz} \;\Leftrightarrow\; |\mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{v})| \leq \beta \|\mathbf{w} - \mathbf{v}\|$$

- |x| is Lipschitz over the entire reals but $x^2$ is not!

- Homework Q.1: what is the Lipschitz constant for log(1+exp(x)) and what is the domain

- Homework Q.2: if L and Q are Lipschitz functions with constants $\beta_1$ & $\beta_2$, what is the Lipschitz constant for L(Q(w))  [Note that L is a scalar function while Q is a vector function]

# Smooth Losses

- A loss is β-smooth if its gradient is β-Lipschitz [Note that we extended Lipschitz to vector functions]

$$\|\nabla\mathcal{L}(\mathbf{w}) - \nabla\mathcal{L}(\mathbf{v})\| \leq \beta\|\mathbf{w} - \mathbf{v}\|$$

- Homework Q.3: show that if a loss is β-smooth then

$$\mathcal{L}(\mathbf{w}) \leq \mathcal{L}(\mathbf{v}) + \nabla\mathcal{L}(\mathbf{v}) \cdot (\mathbf{w} - \mathbf{v}) + \frac{\beta}{2}\|\mathbf{w} - \mathbf{v}\|^2$$

# Gradient Descent for Lipschitz Losses

- Assume that loss function is β-Lipschitz
- Perform the following updates:

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta_t \nabla \mathcal{L}(\mathbf{w}^{t}) \quad \text{where} \quad \eta_t = \tilde{O}\left(1/\sqrt{t}\right)$$

- Let w* be the minimizer of the loss over the domain {w s.t. ||w|| < r}
- Let u be the average of $w^t$ from t=1 through T
- Then, the gap between u and w* w.r.t loss is

$$\mathcal{L}(\mathbf{u}) - \mathcal{L}(\mathbf{w}^{\star}) = \mathcal{L}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{t}\right) - \mathcal{L}(\mathbf{w}^{\star}) \leq \frac{r\beta}{\sqrt{T}}$$

# Proof Outline

- Use convexity to upper bound the difference between the loss at u and the loss at w*
- Use the distance between $w^t$ and w* as potential
- Find a learning rate that minimizes at each iteration a bound on the potential
- Important comments on smoothness and stochastic optimization to follow the proof
- See also Section 14.1 in:
  Understanding Machine Learning: From Theory to Algorithms
  by Shai Shalev-Shwartz & Shai Ben-David

# Stochastic Optimization

Training set is large and the source is i.i.d then we can sub-sample S to obtain an estimate of the gradient

$$\hat{\nabla}_t L = \frac{1}{|S'|} \sum_{i \in S'} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t; (\mathbf{x}_i, y_i))$$

# Stochastic Optimization

Training set is large and the source is i.i.d then we can sub-sample S to obtain an estimate of the gradient

$$\hat{\nabla}_t L = \frac{1}{|S'|} \sum_{i \in S'} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t; (\mathbf{x}_i, y_i)) \quad S' \subset S$$

# Stochastic Optimization

Training set is large and the source is i.i.d then we can sub-sample S to obtain an estimate of the gradient

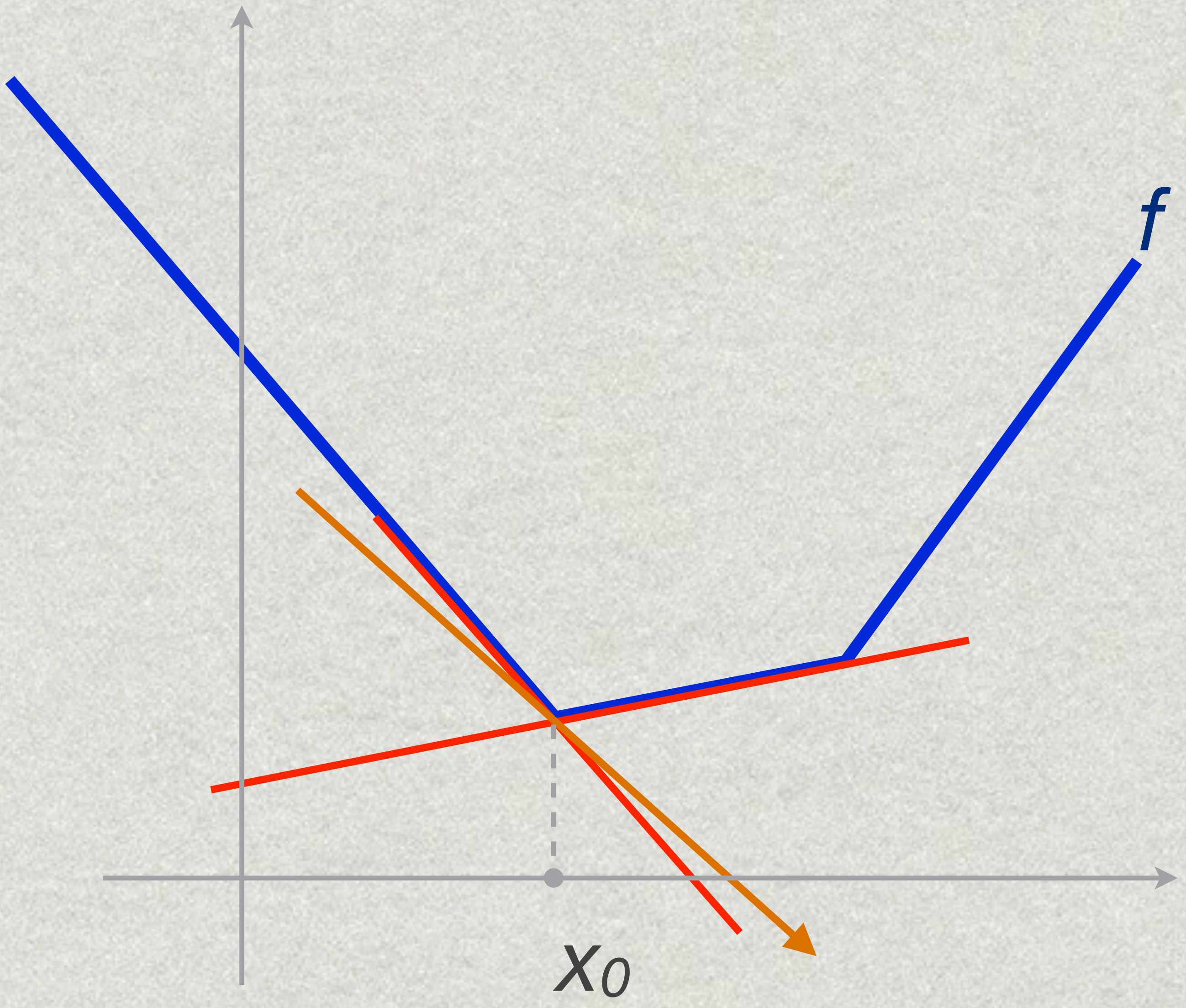
$$\hat{\nabla}_t L = \frac{1}{|S'|} \sum_{i \in S'} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{w}_t; (\mathbf{x}_i, y_i)) \quad S' \subset S$$

Convergence Rate still holds in expectation {over S'} !

# Subgradients

Subgradient set of a function *f* at x0

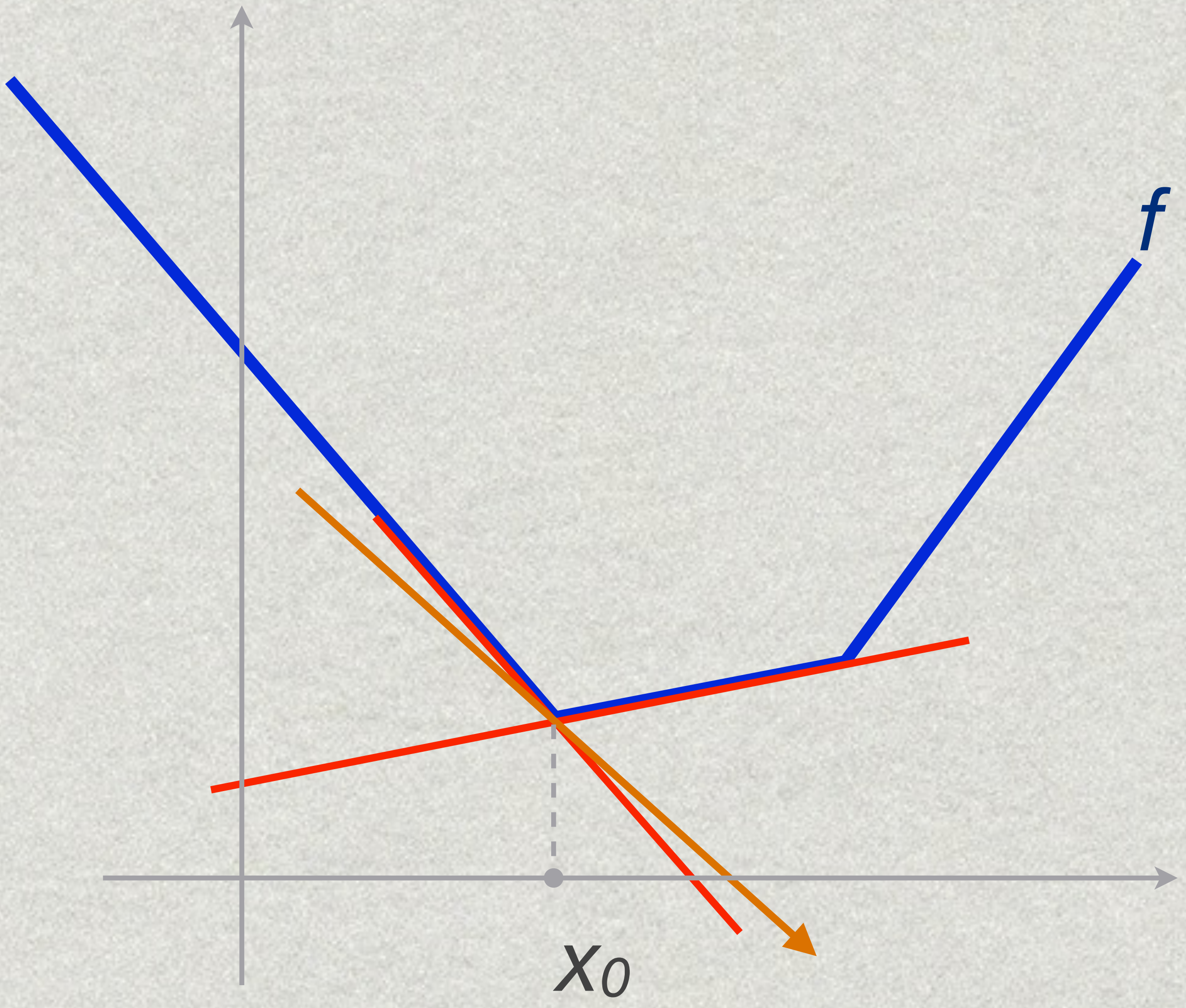
$$\partial f(\boldsymbol{x}_0) = \left\{\boldsymbol{g} : f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \boldsymbol{g}^\top(\boldsymbol{x} - \boldsymbol{x}_0)\right\}$$



# Subgradients

Subgradient set of a function *f* at x0

$$\partial f(\boldsymbol{x}_0) = \left\{ \boldsymbol{g} : f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \boldsymbol{g}^\top (\boldsymbol{x} - \boldsymbol{x}_0) \right\}$$



# Minimization using Subgradients

Minimize

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) + \lambda \left\|\boldsymbol{w}\right\|_1$$

- Unconstrained stochastic **sub**gradient descent

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t \qquad \boldsymbol{g}_t \in \hat{\nabla}_t L + \partial\|\boldsymbol{w}_t\|_1$$

# Minimization using Subgradients

Minimize

$$\min_{\boldsymbol{w}} L(\boldsymbol{w}) + \lambda \left\|\boldsymbol{w}\right\|_1$$

- Unconstrained stochastic **sub**gradient descent

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \boldsymbol{g}_t \qquad \boldsymbol{g}_t \in \hat{\nabla}_t L + \partial\|\boldsymbol{w}_t\|_1$$

$$\downarrow$$

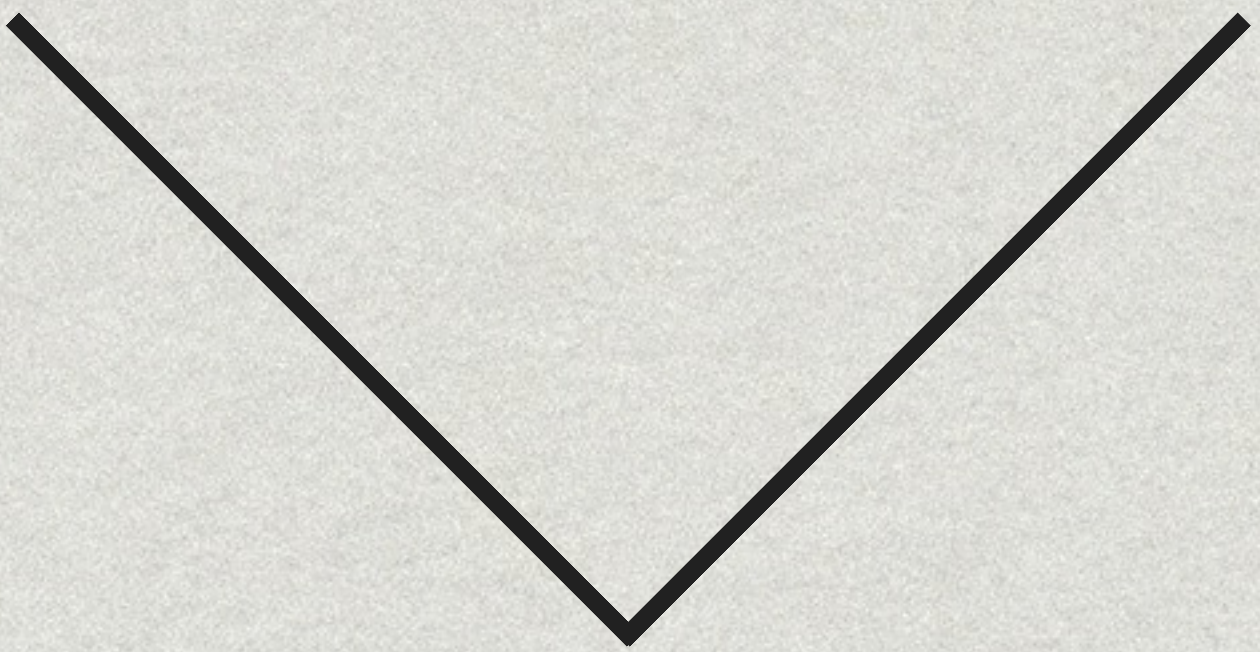
$$\partial|w_{t,j}| = \operatorname{sign}(w_{t,j})$$

# Subgradients: Caveat

Subgradients are “non-informative” at singularities

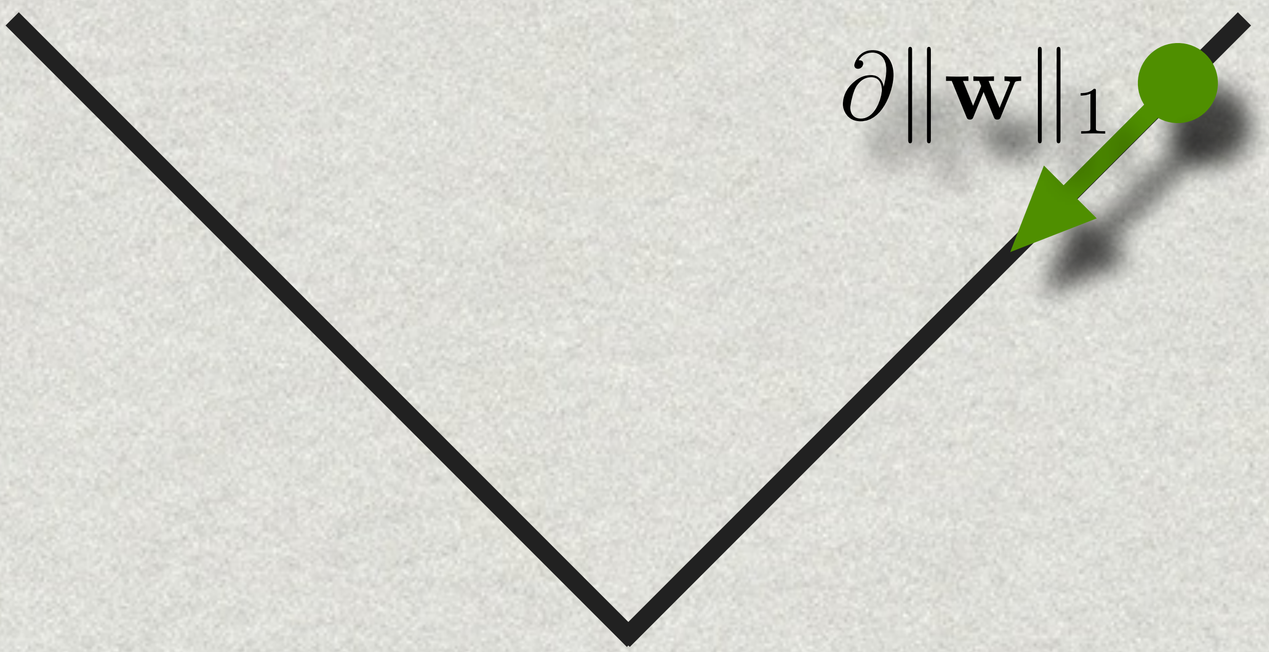
# Subgradients: Caveat

Subgradients are “non-informative” at singularities



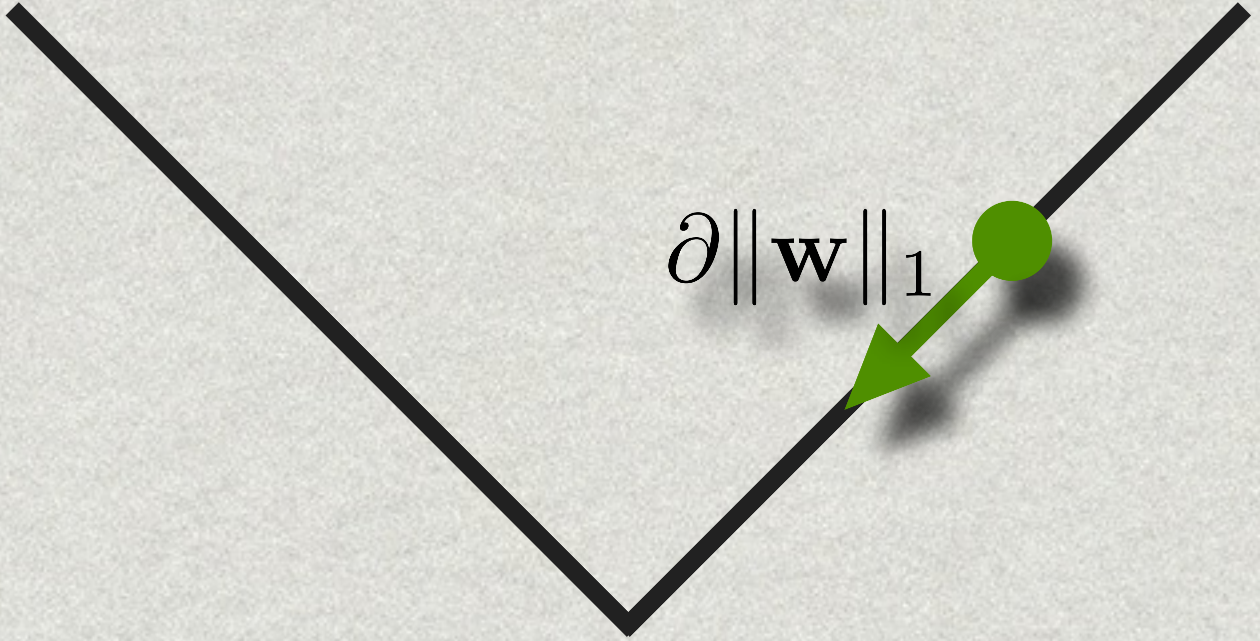
# Subgradients: Caveat

Subgradients are “non-informative” at singularities



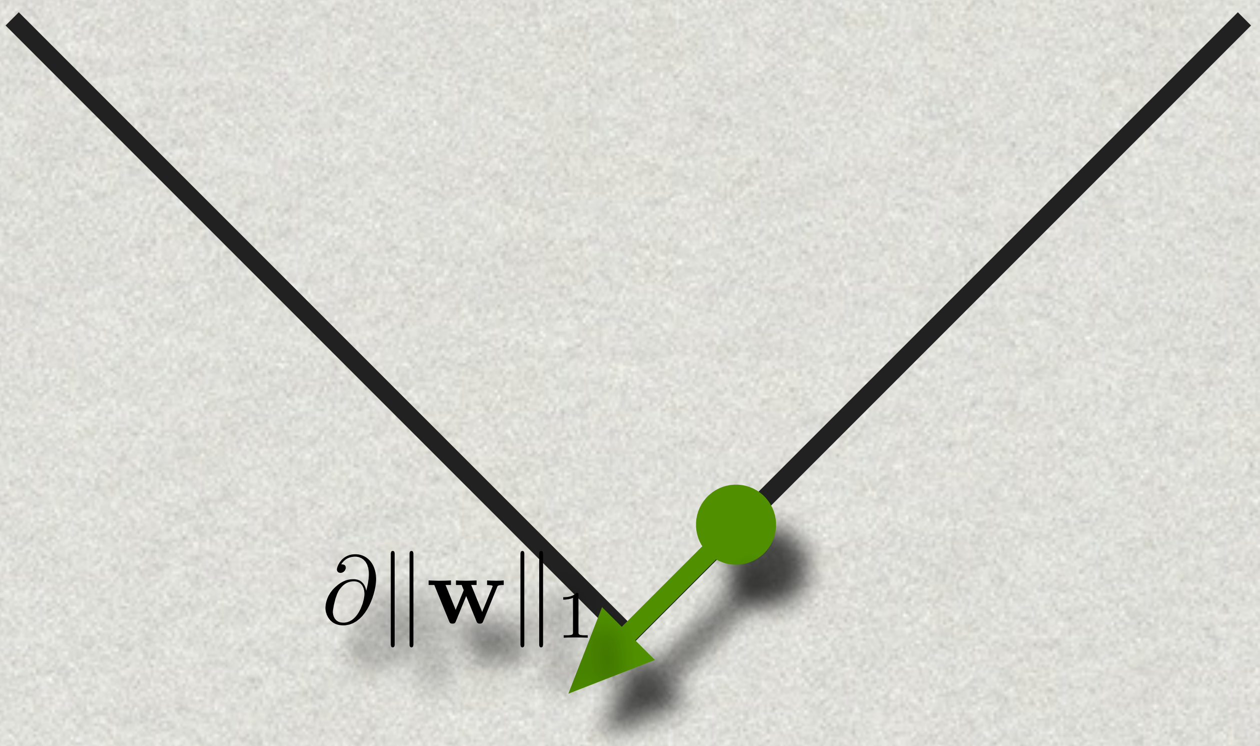
# Subgradients: Caveat

Subgradients are “non-informative” at singularities



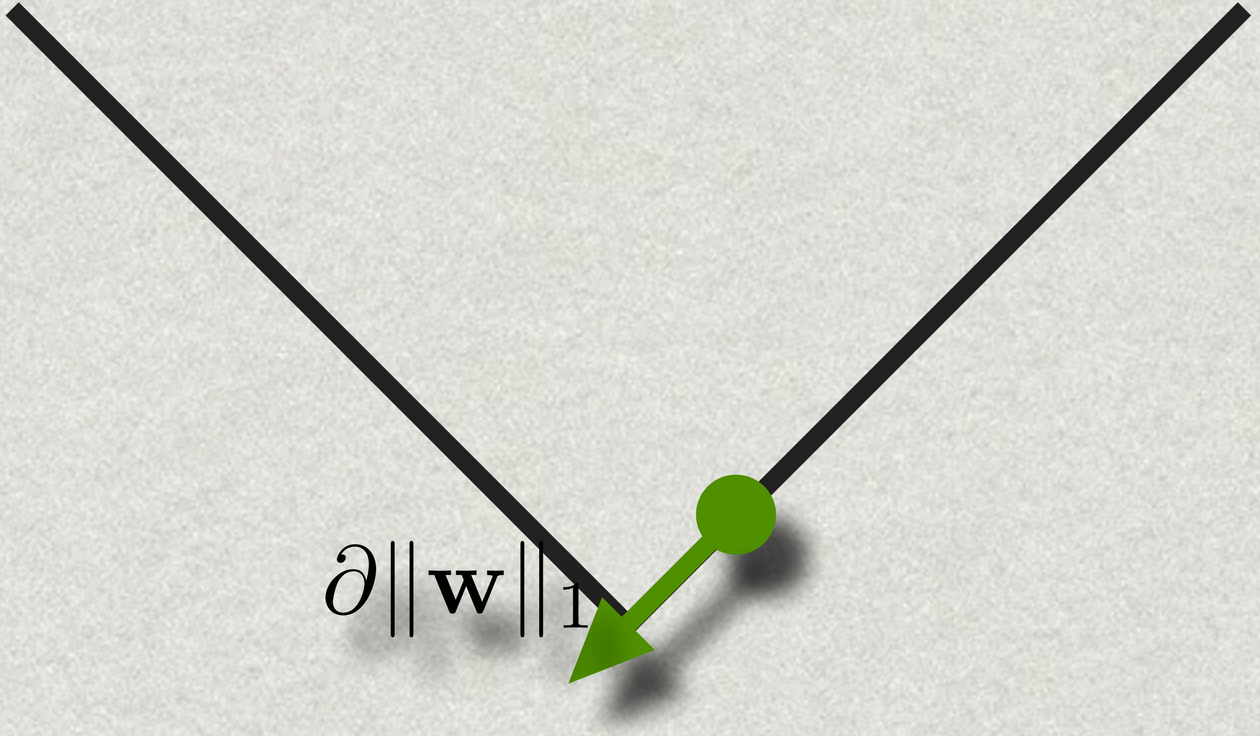
# Subgradients: Caveat

Subgradients are “non-informative” at singularities



# Subgradients: Caveat

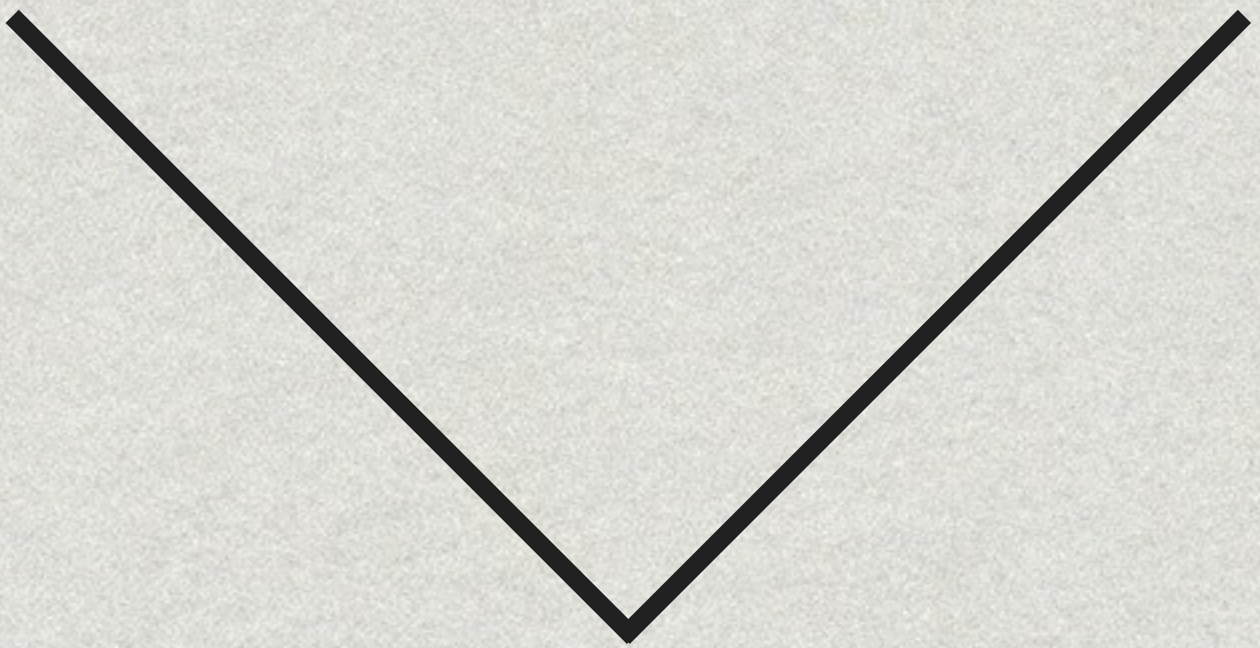
Subgradients are “non-informative” at singularities



- **Dense solution for W**

# Subgradients: Caveat

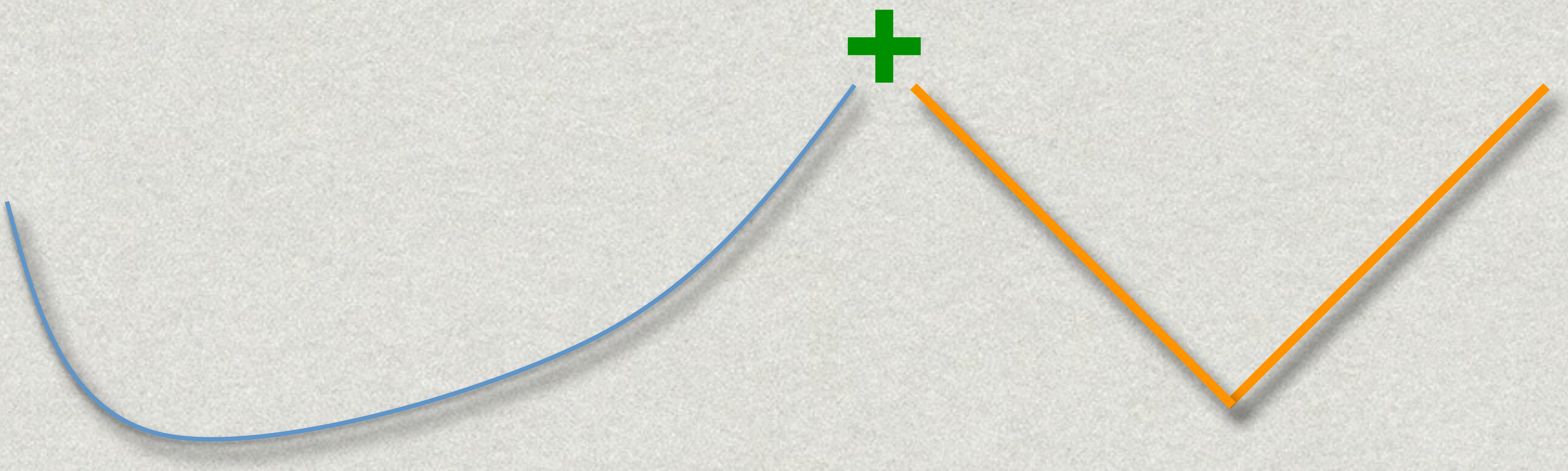
Subgradients are “non-informative” at singularities



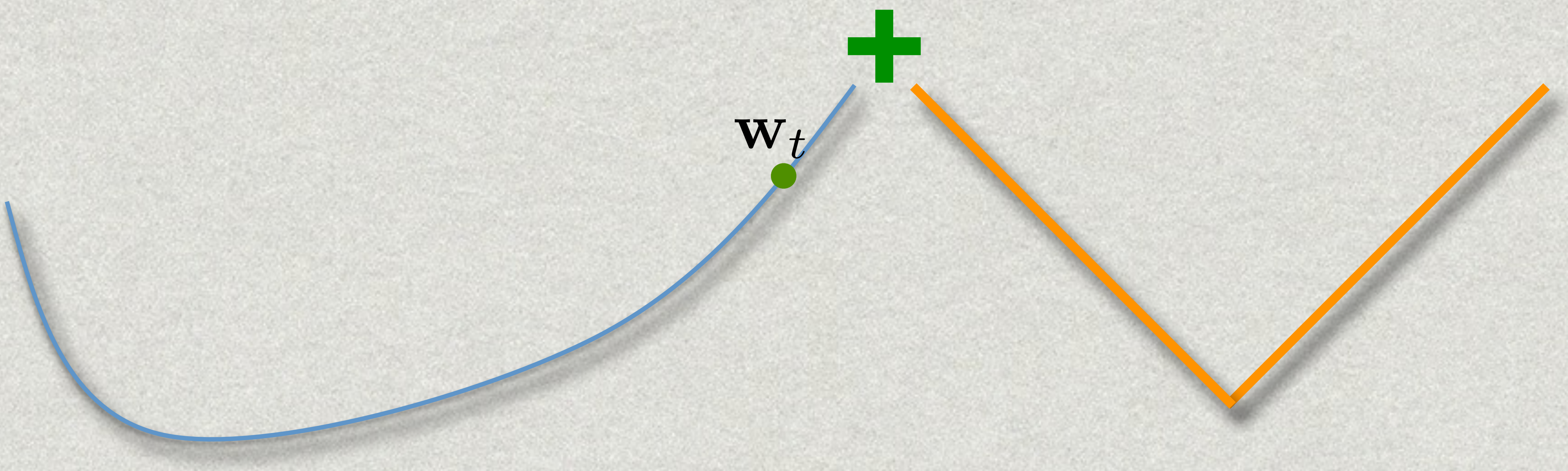
- **Dense solution for W**
- **Slow convergence**

# Fobos

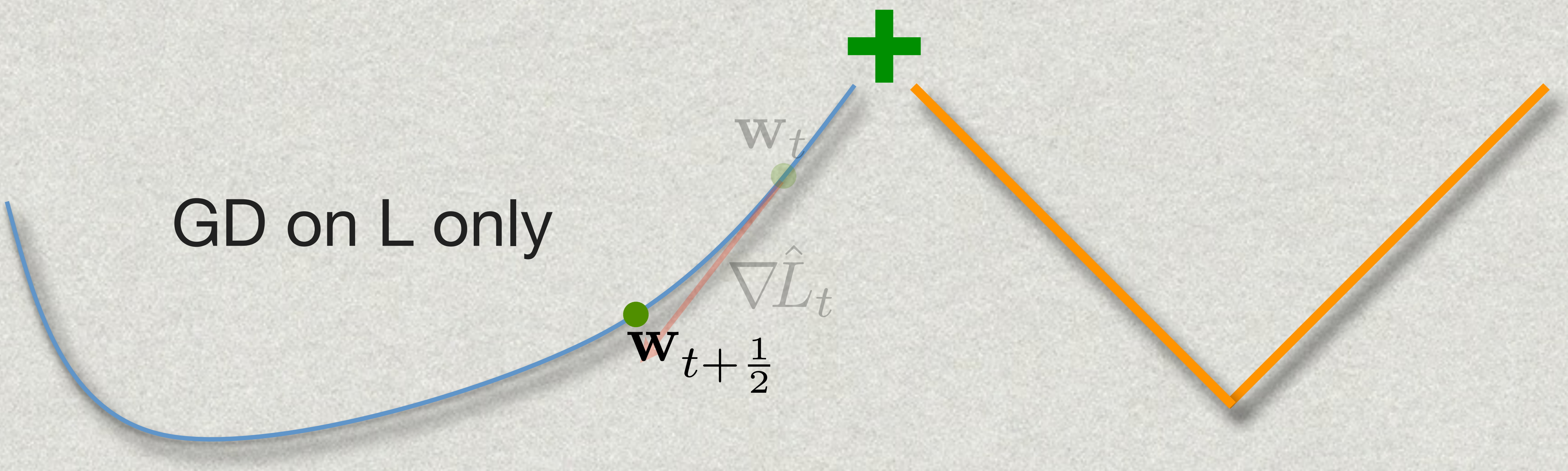
# Two Step Approach



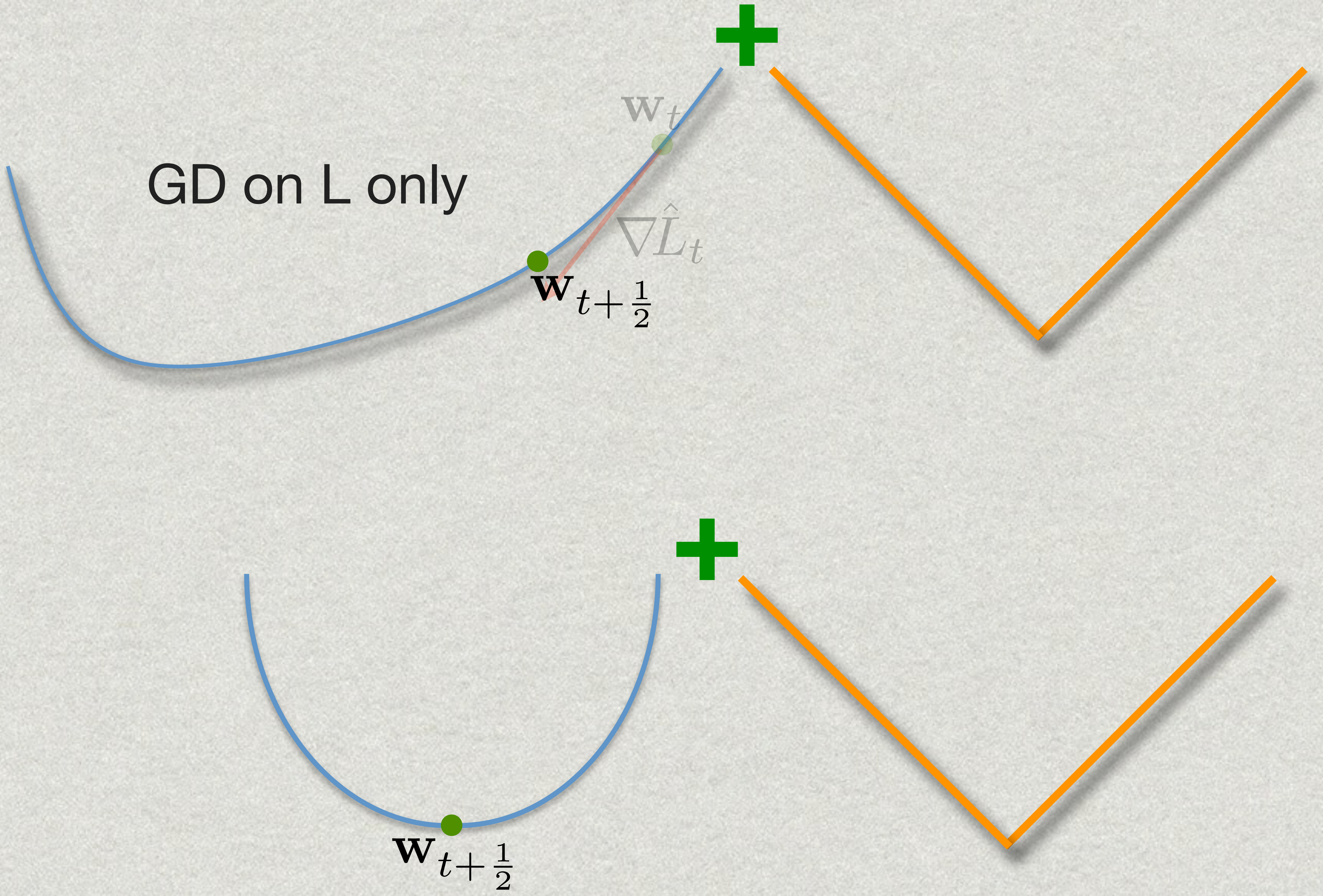
# Two Step Approach



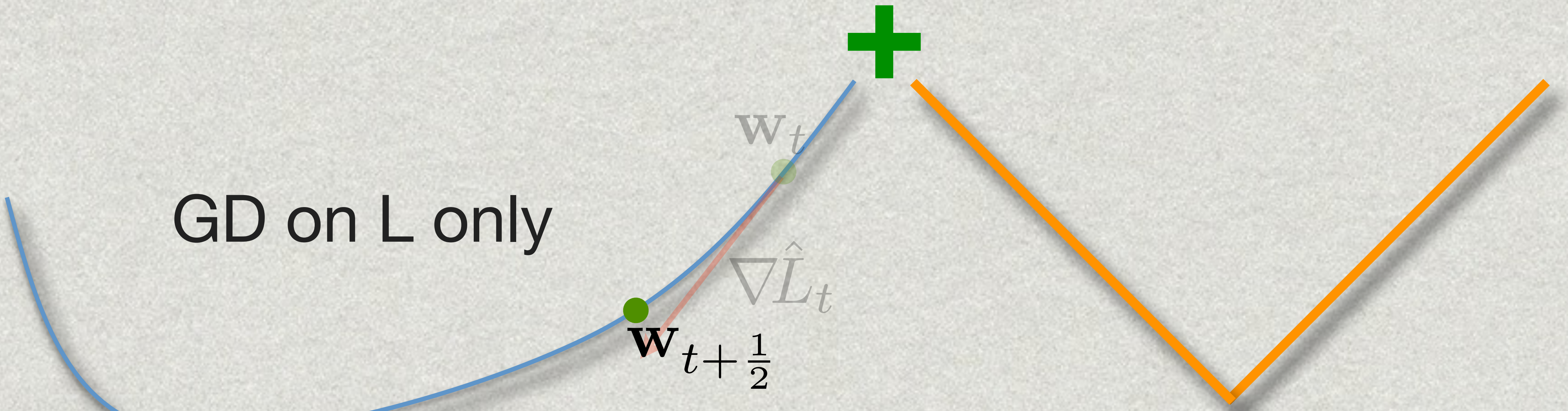
# Two Step Approach

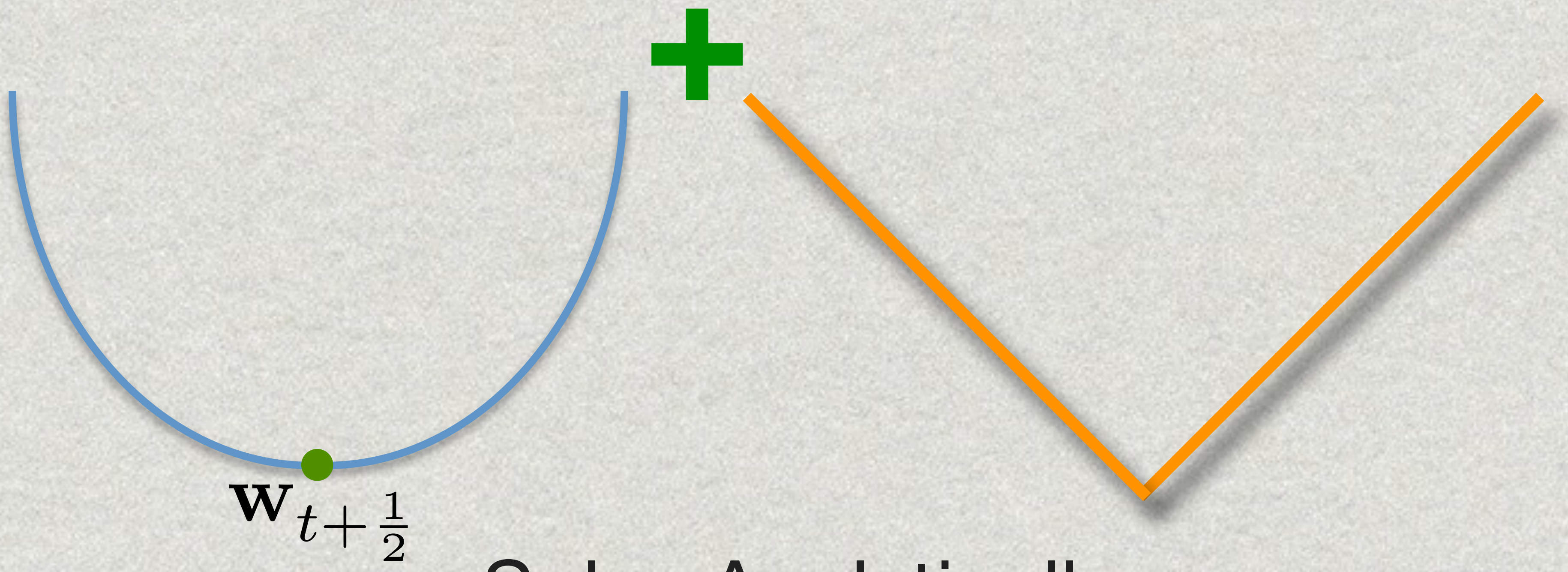


# Two Step Approach



# Two Step Approach


GD on L only



Solve Analytically

# Fobos: Two Step Approach

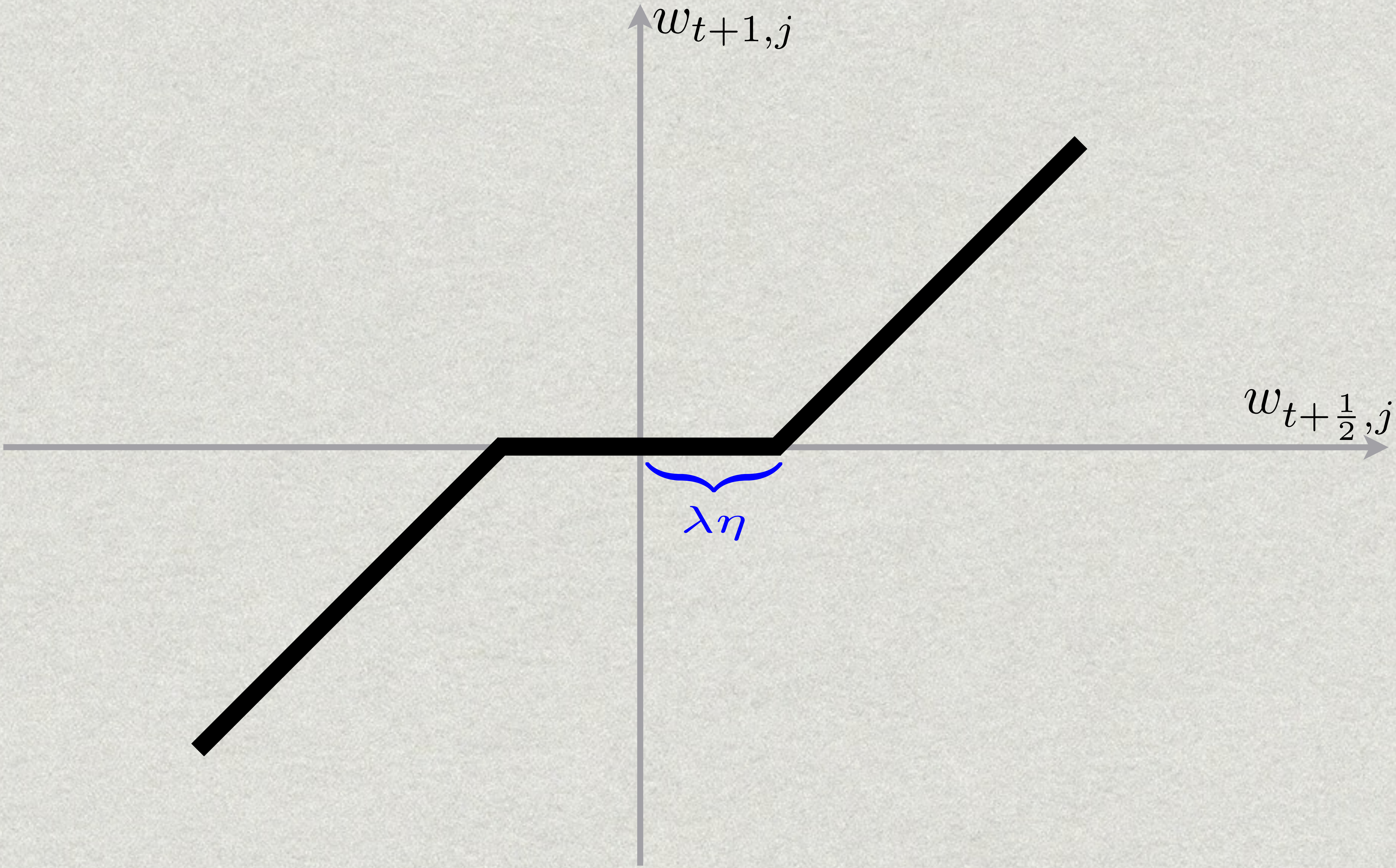
(1) Unconstrained stochastic gradient of loss

$$\boldsymbol{w}_{t+\frac{1}{2}} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$$

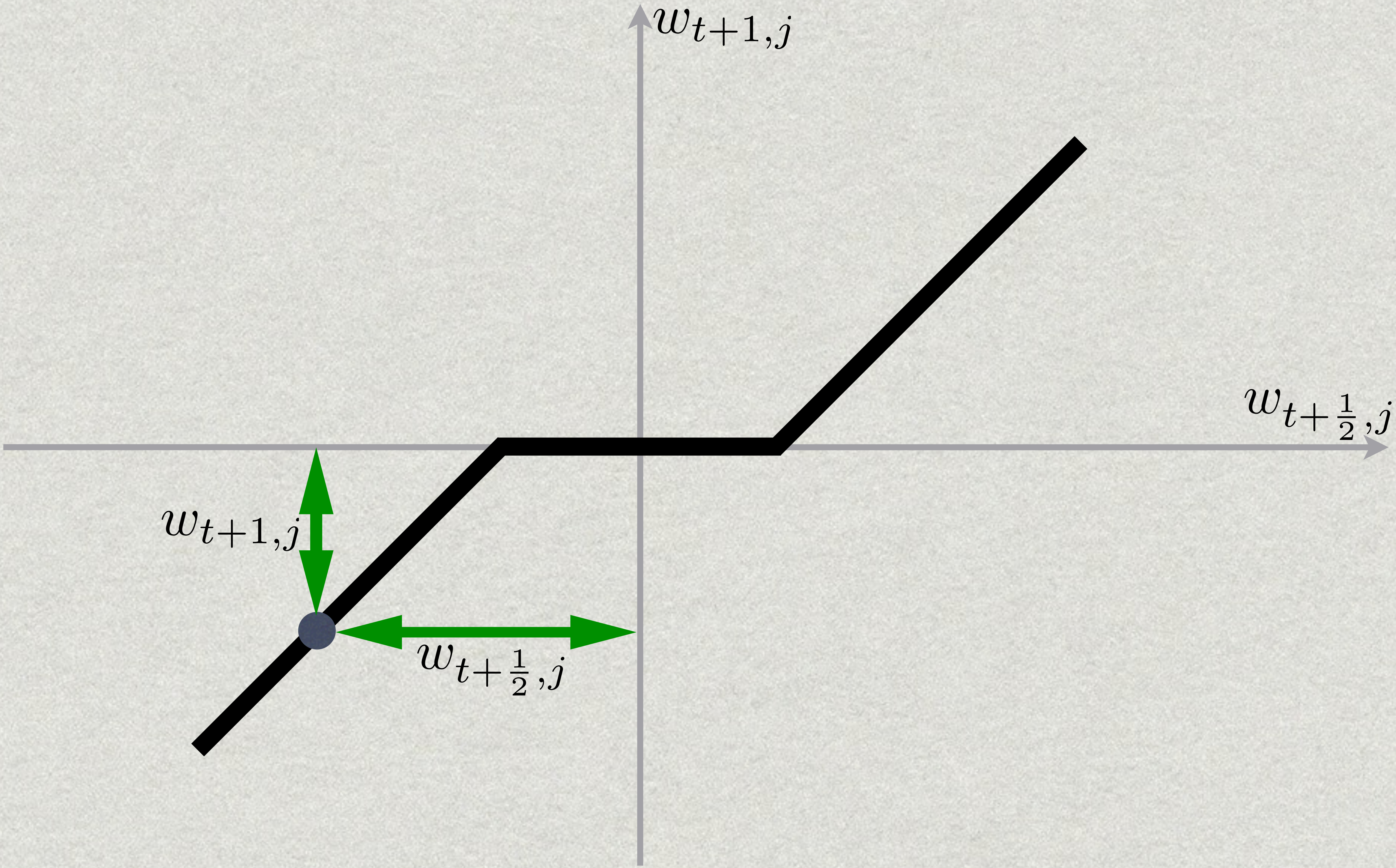
(2) Incorporate regularization and solve

$$\boldsymbol{w}_{t+1} \quad = \quad \underset{\boldsymbol{w}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \boldsymbol{w} - \boldsymbol{w}_{t+\frac{1}{2}} \right\|^2 + \eta \, \lambda \, R(\boldsymbol{w}) \right\}$$

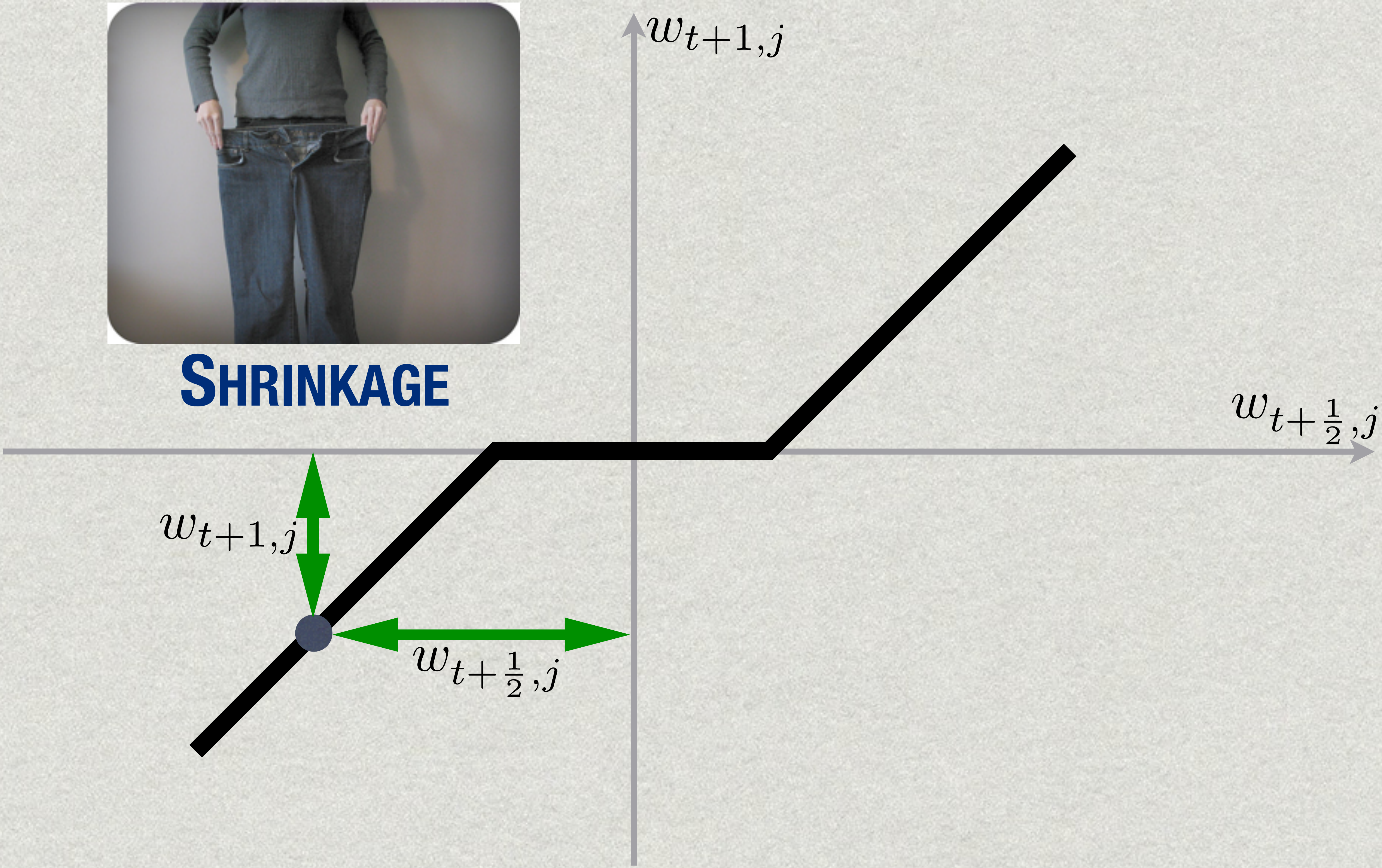
# Fobos for L$_1$ Regularization



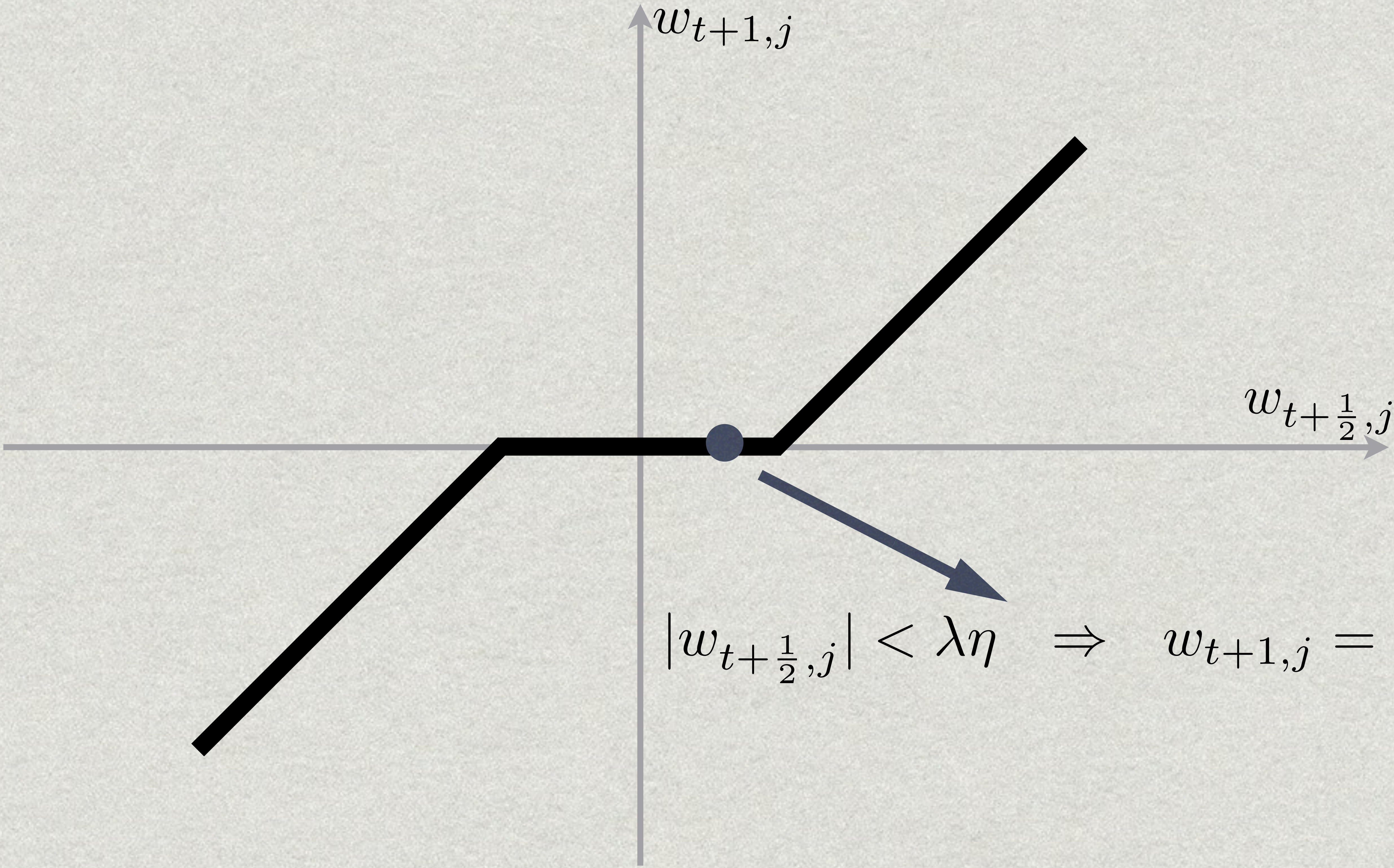
# Fobos for $L_1$ Regularization



# Fobos for $L_1$ Regularization

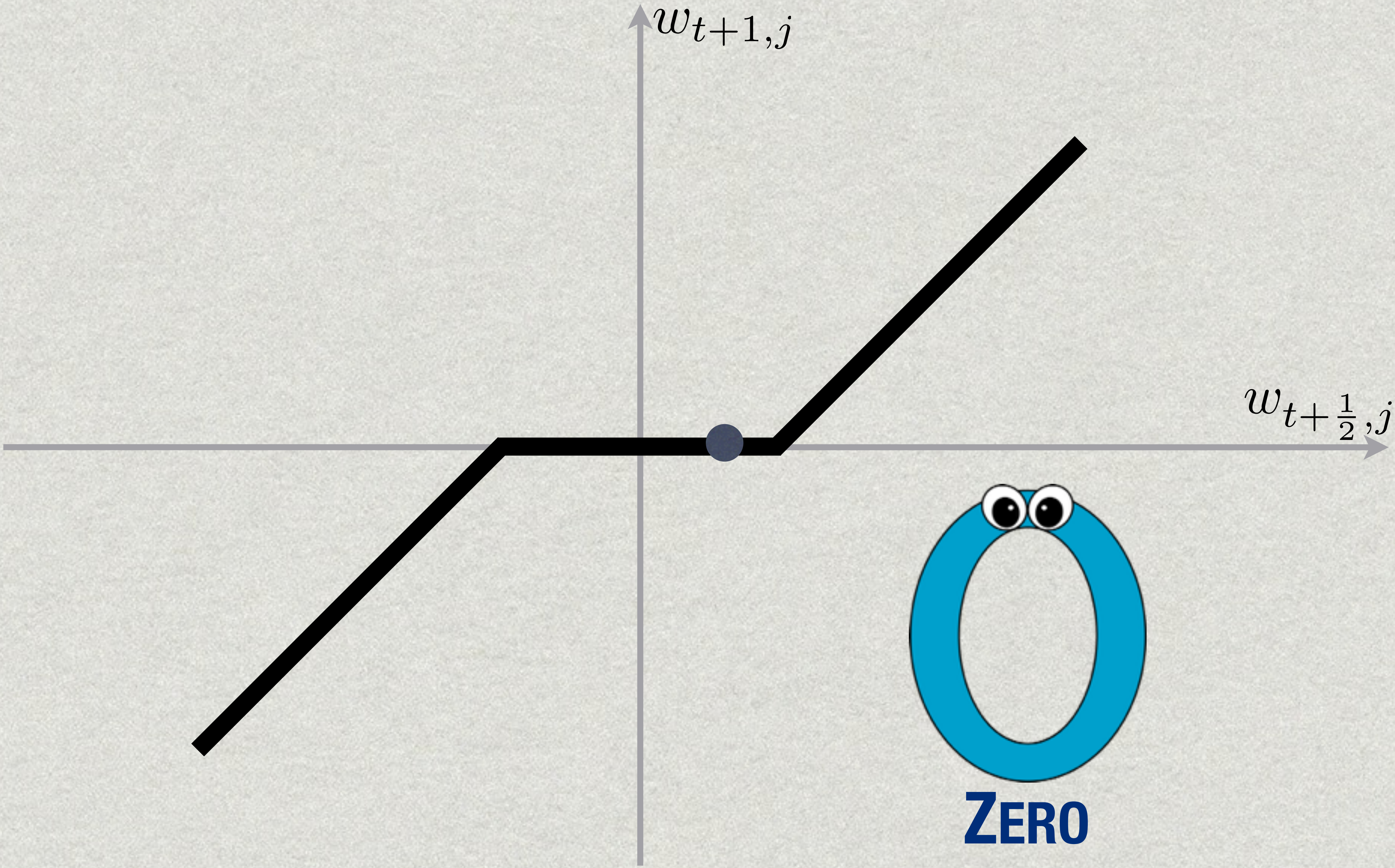


# Fobos for L$_1$ Regularization



$$|w_{t+\frac{1}{2},j}| < \lambda\eta \quad \Rightarrow \quad w_{t+1,j} = 0$$

# Fobos for $L_1$ Regularization



# Forward Looking Subgradient

- The optimum ($\mathbf{w}_{t+1}$) satisfies

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\,\boldsymbol{g}_t^L - \eta\,\lambda\,\boldsymbol{g}_{t+1}^R$$

# Forward Looking Subgradient

- The optimum ($\mathbf{w}_{t+1}$) satisfies

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\,\underline{\boldsymbol{g}_t^L} - \eta\,\lambda\,\boldsymbol{g}_{t+1}^R$$

**CURRENT GRADIENT OF EMPIRICAL LOSS**

# Forward Looking Subgradient

- The optimum ($\mathbf{w}_{t+1}$) satisfies

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\,\underline{\boldsymbol{g}_t^L} - \eta\,\lambda\,\underline{\boldsymbol{g}_{t+1}^R}$$

**CURRENT GRADIENT OF EMPIRICAL LOSS**

**FORWARD SUBGRADIENT OF REGULARIZATION**

# Forward Looking Subgradient

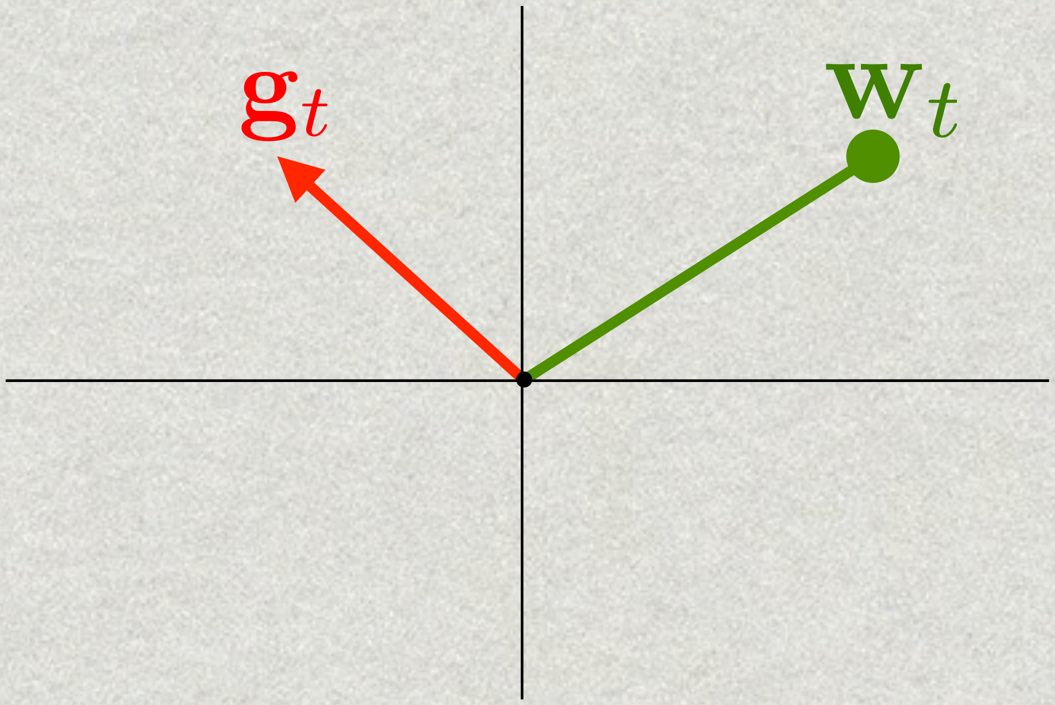
- The optimum ($\mathbf{w}_{t+1}$) satisfies

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \underbrace{\boldsymbol{g}_t^L}_{\text{CURRENT GRADIENT OF EMPIRICAL LOSS}} - \eta \lambda \underbrace{\boldsymbol{g}_{t+1}^R}_{\text{FORWARD SUBGRADIENT OF REGULARIZATION}}$$

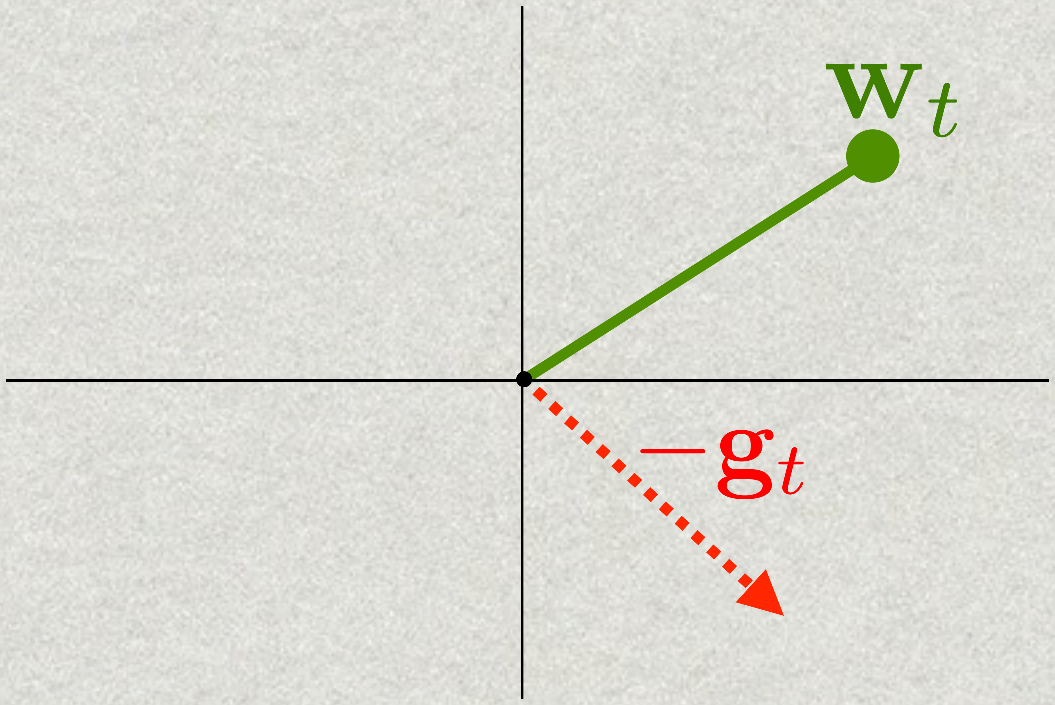
Yields very simple alternative analysis, in particular convergence to the optimum at a rate of

$$O\left(\frac{1}{\sqrt{T}}\right) \text{ or } O\left(\frac{\log(T)}{T}\right)$$

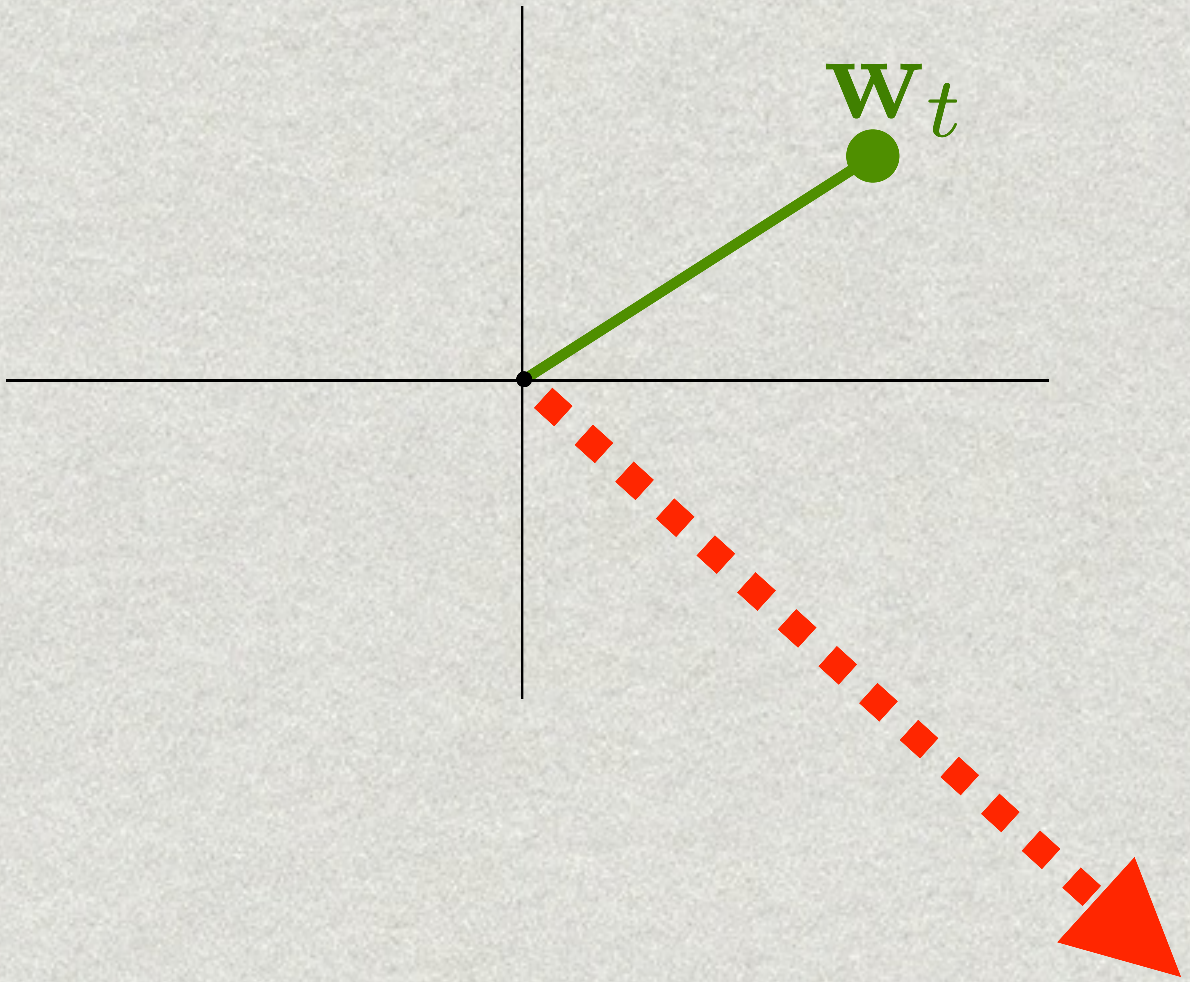
# Proximal Operators



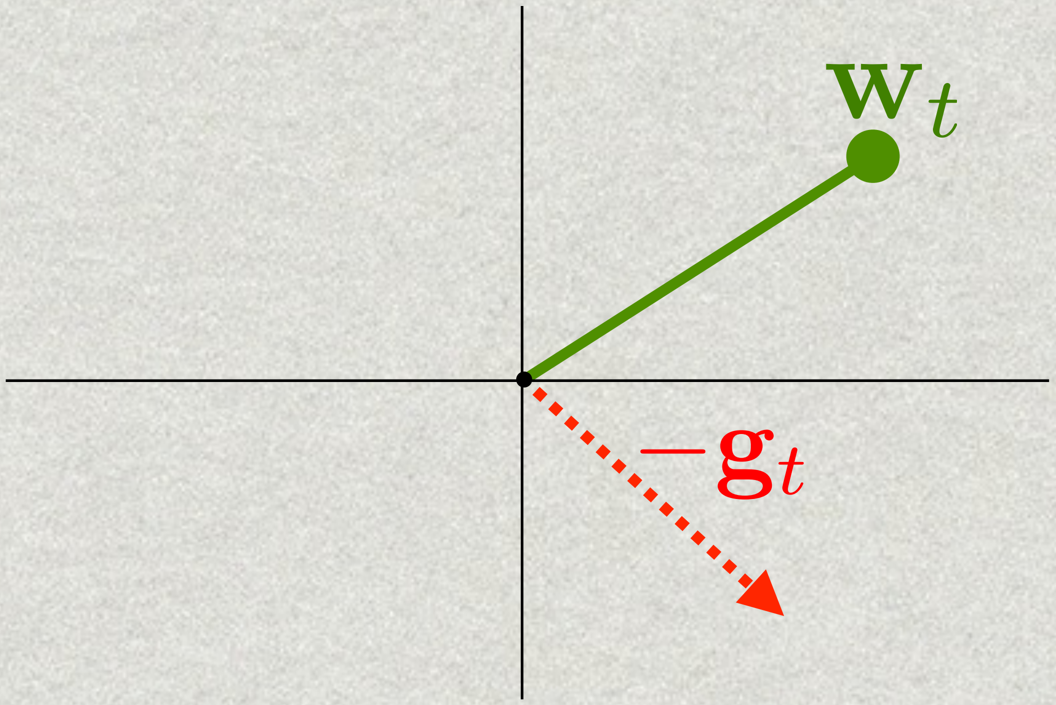
# Proximal Operators



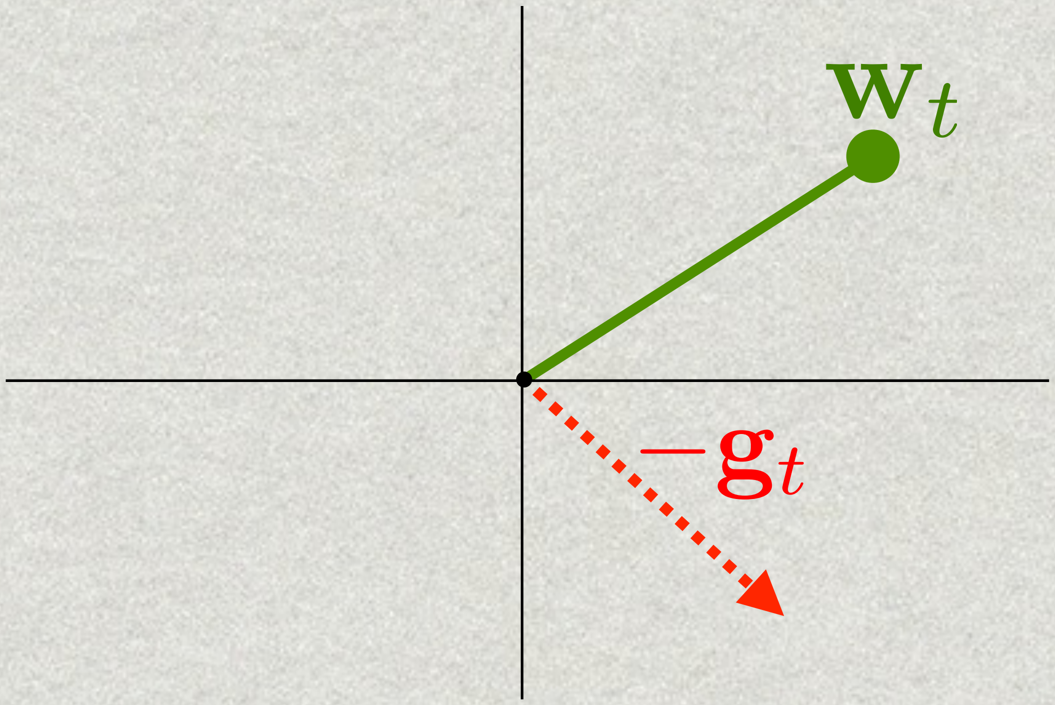
# Proximal Operators



# Proximal Operators



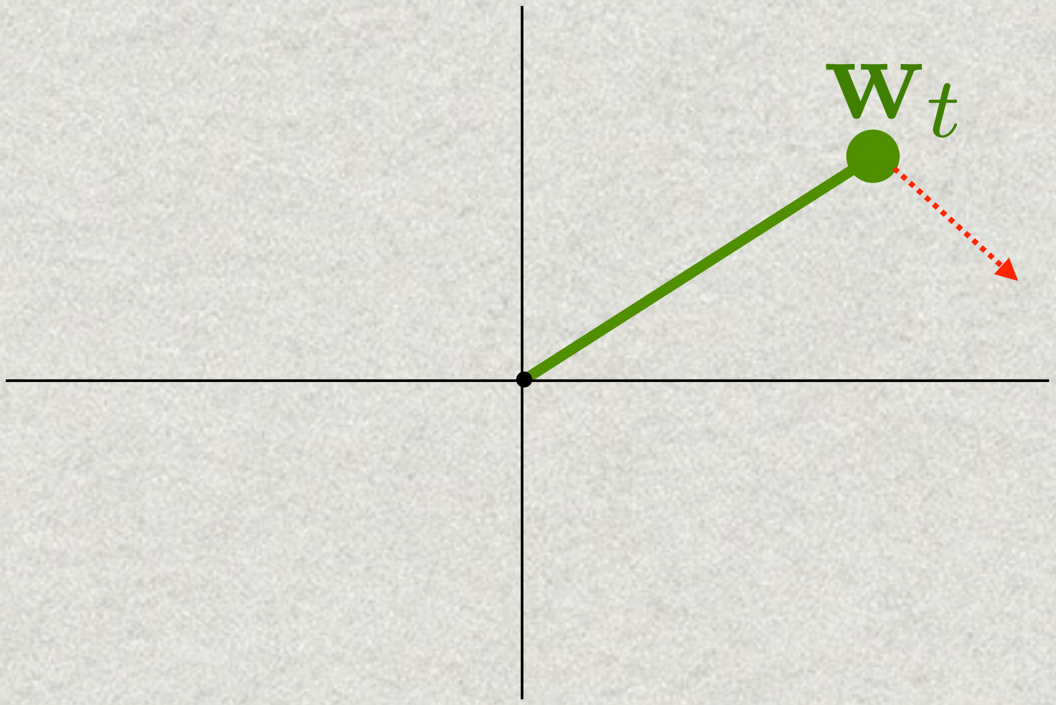
# Proximal Operators



Cast a tradeoff:

- Maintaining proximity to weight vector
- Following the steepest descent direction

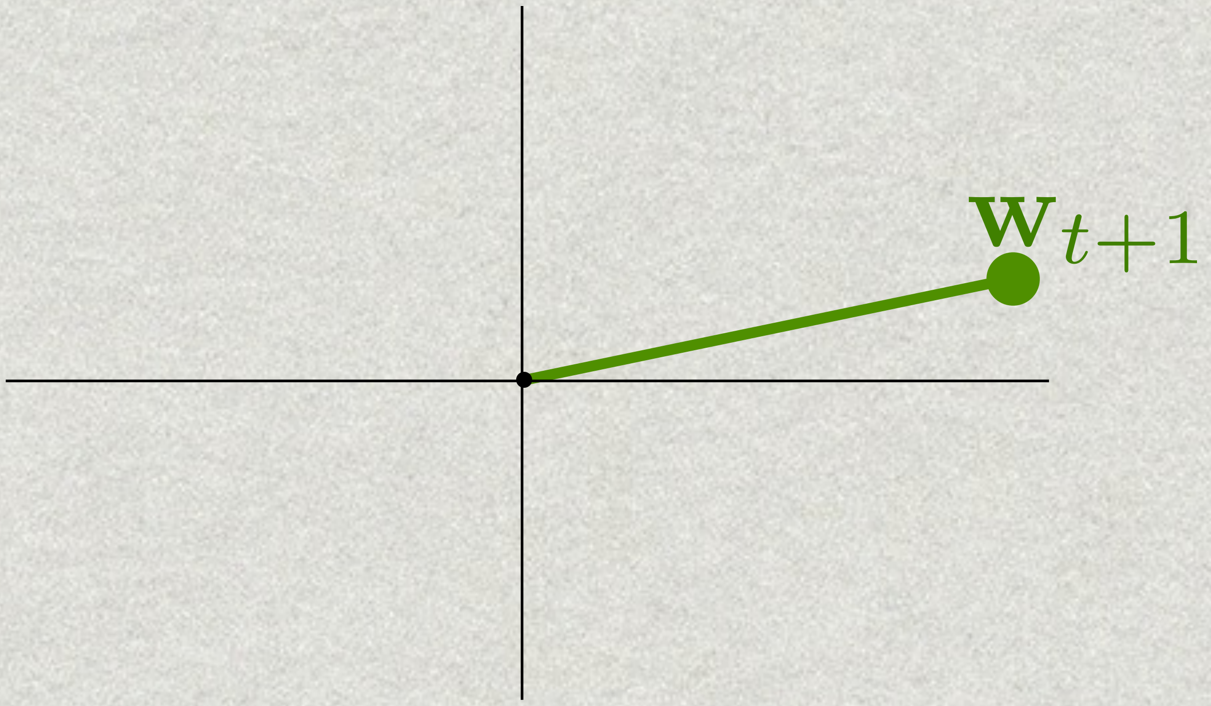
# Proximal Operators



Cast a tradeoff:

- Maintaining proximity to weight vector
- Following the steepest descent direction

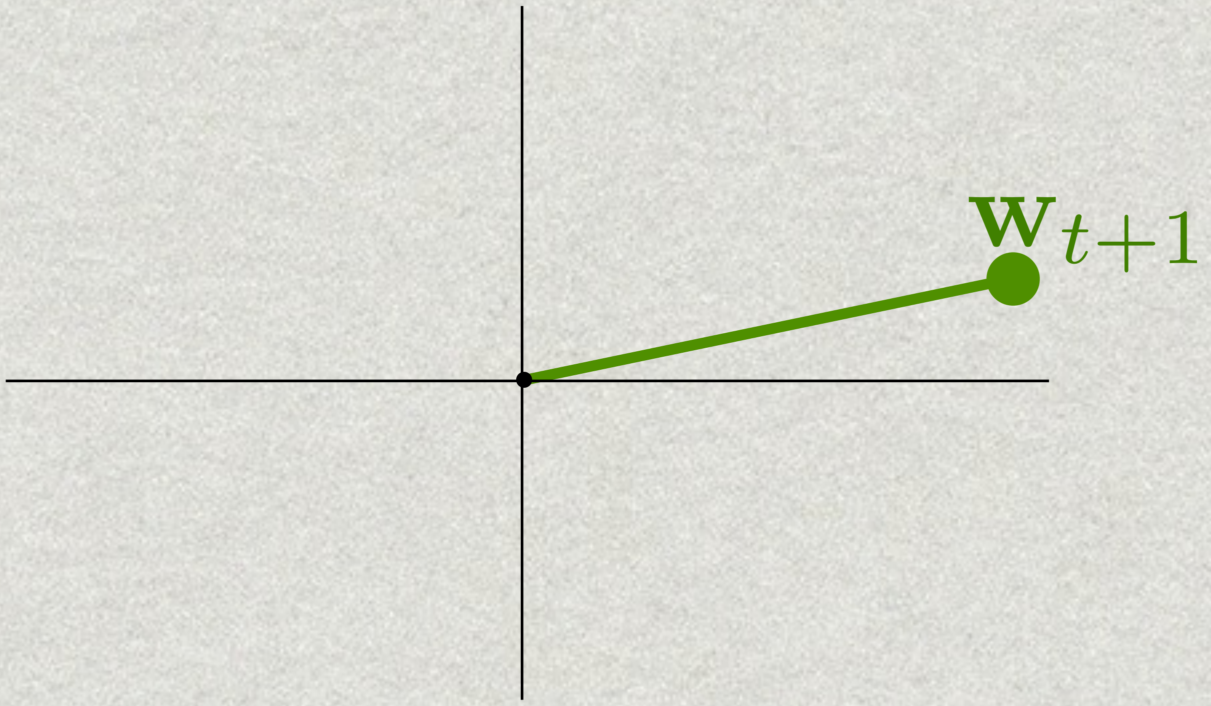
# Proximal Operators



Cast a tradeoff:

- Maintaining proximity to weight vector
- Following the steepest descent direction

# Proximal Operators

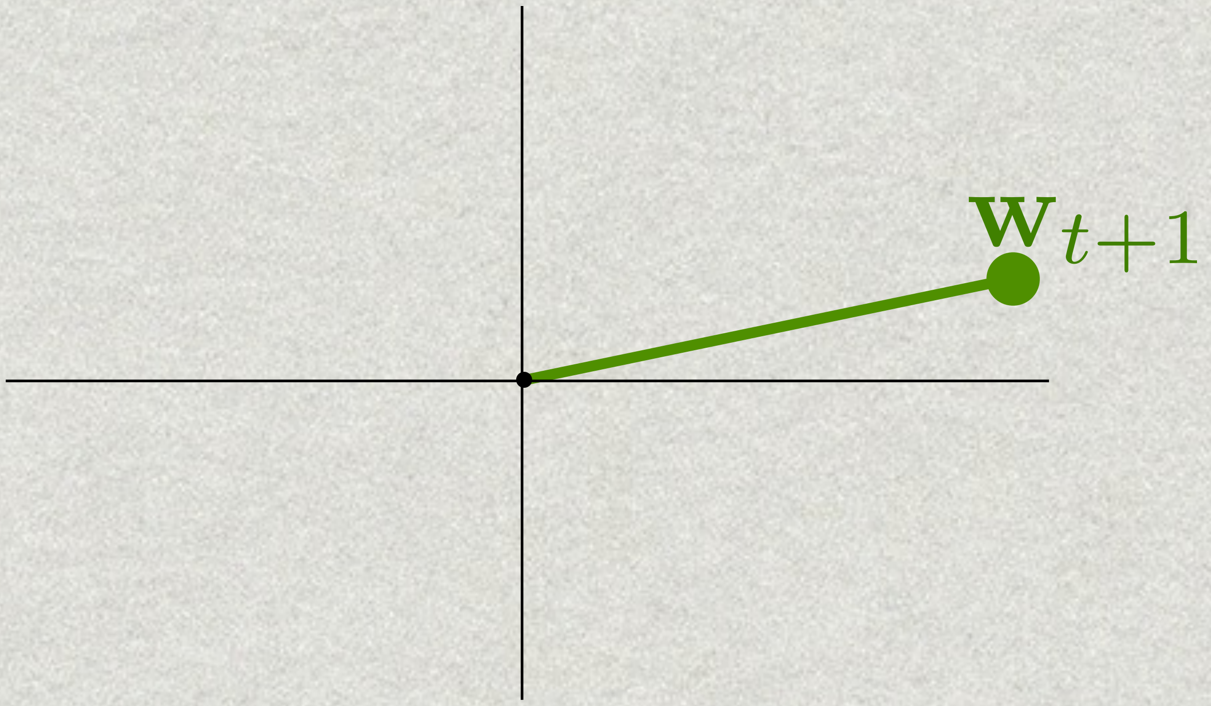


$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\Omega} \frac{1}{\eta} D(\mathbf{w}\|\mathbf{w}_{t+1}) + \langle \mathbf{w}, \mathbf{g}_t \rangle$$

proximity

gradient

# Proximal Operators



$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\Omega} \frac{1}{\eta} D(\mathbf{w}\|\mathbf{w}_{t\cancel{+1}}) + \langle \mathbf{w}, \mathbf{g}_t \rangle$$

MY BAD!

proximity

gradient

# Fobos & Proximal Operators

- Uses decomposition of the objective into an empirical risk minimization term and a regularization term
- Uses the squared Euclidian norm for proximity

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\Omega} \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_t\|^2 + \langle \mathbf{w}, \mathbf{g}_t \rangle + \lambda\|\mathbf{w}\|_1$$

# EG & Proximal Operators

- If we constrain **w** to the probability simplex, use relative entropy, we get Exponentiated Gradient (EG)

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}\in\Delta} \frac{1}{\eta} D_{\mathrm{KL}}(\mathbf{w}\|\mathbf{w}^t) + \langle \mathbf{w}, \mathbf{g}^t \rangle$$

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}\in\Delta} \sum_{j=1}^{d} w_j \left( \log\left(\frac{w_j}{w_j^t}\right) + \eta\, g_j^t \right)$$

$$w_j^{t+1} = \frac{1}{Z} w_j^t \exp(-\eta\, g_j^t)$$

$$\text{where} \quad Z = \sum_{l=1}^{d} w_l^t \exp(-\eta\, g_l^t)$$

# High Dim Data ⇨ Sparse Gradients

| | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... | | | | |

For an efficient implementation computation should:

- Scale with the number of non-zeros
- Not with the full dimension

The following lemma to the rescue:

$$\mathcal{P}.1: \quad \boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_{t-1}\|^2 + \lambda_t \|\boldsymbol{w}\|_q$$

$$\mathcal{P}.2: \quad \boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_0\|^2 + \left(\sum_{t=1}^{T} \lambda_t\right) \|\boldsymbol{w}\|_q$$

$$T \times \mathcal{P}.1 \;\equiv\; \mathcal{P}.2 \qquad q \in \{1, 2, \infty\}$$

$$\mathbf{w}_T\ (\mathcal{P}.1) \;=\; \mathbf{w}^\star\ (\mathcal{P}.2)$$

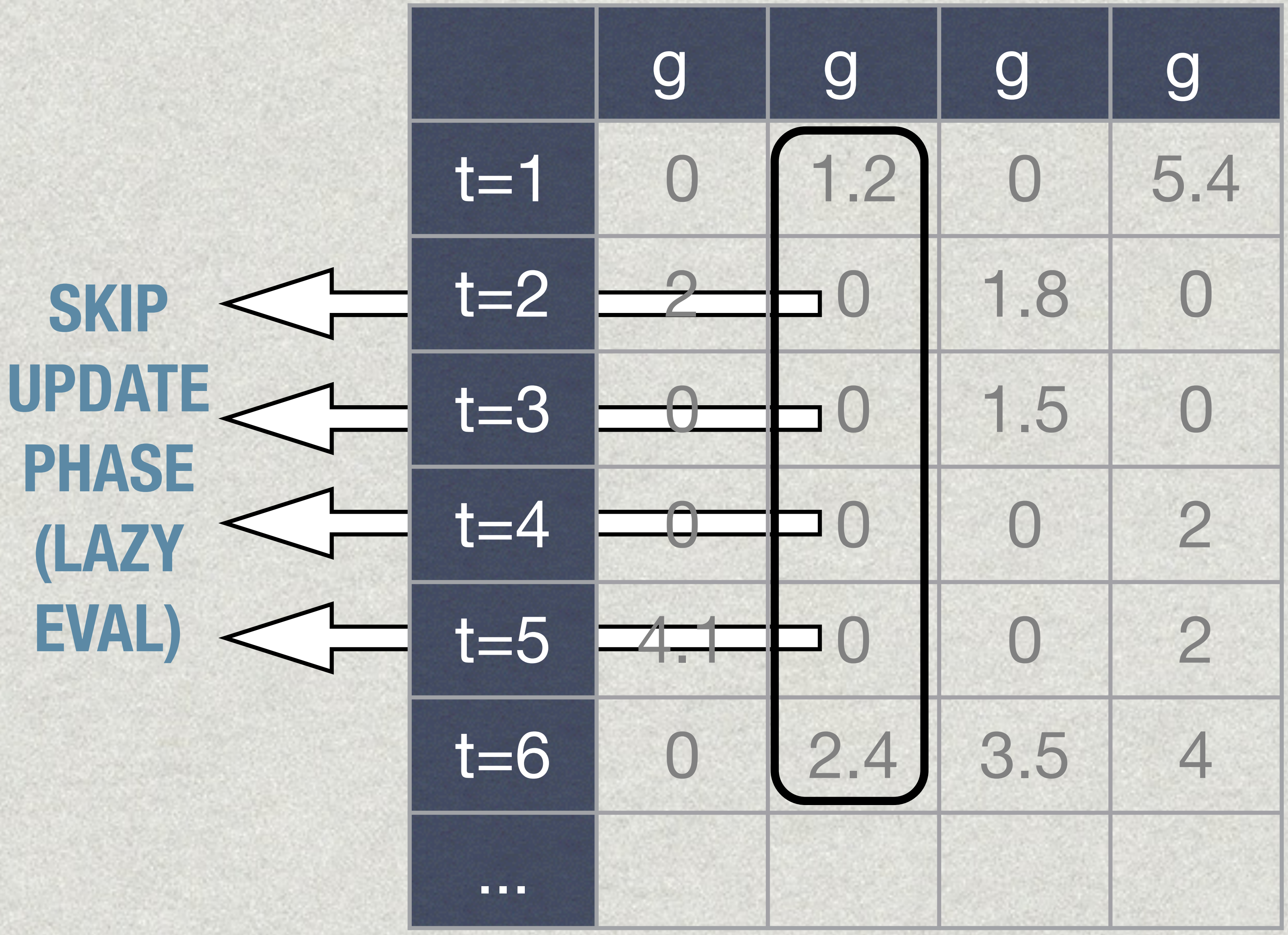
# Efficient High Dimensional Update

| | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... | | | | |

# Efficient High Dimensional Update

| | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... | | | | |

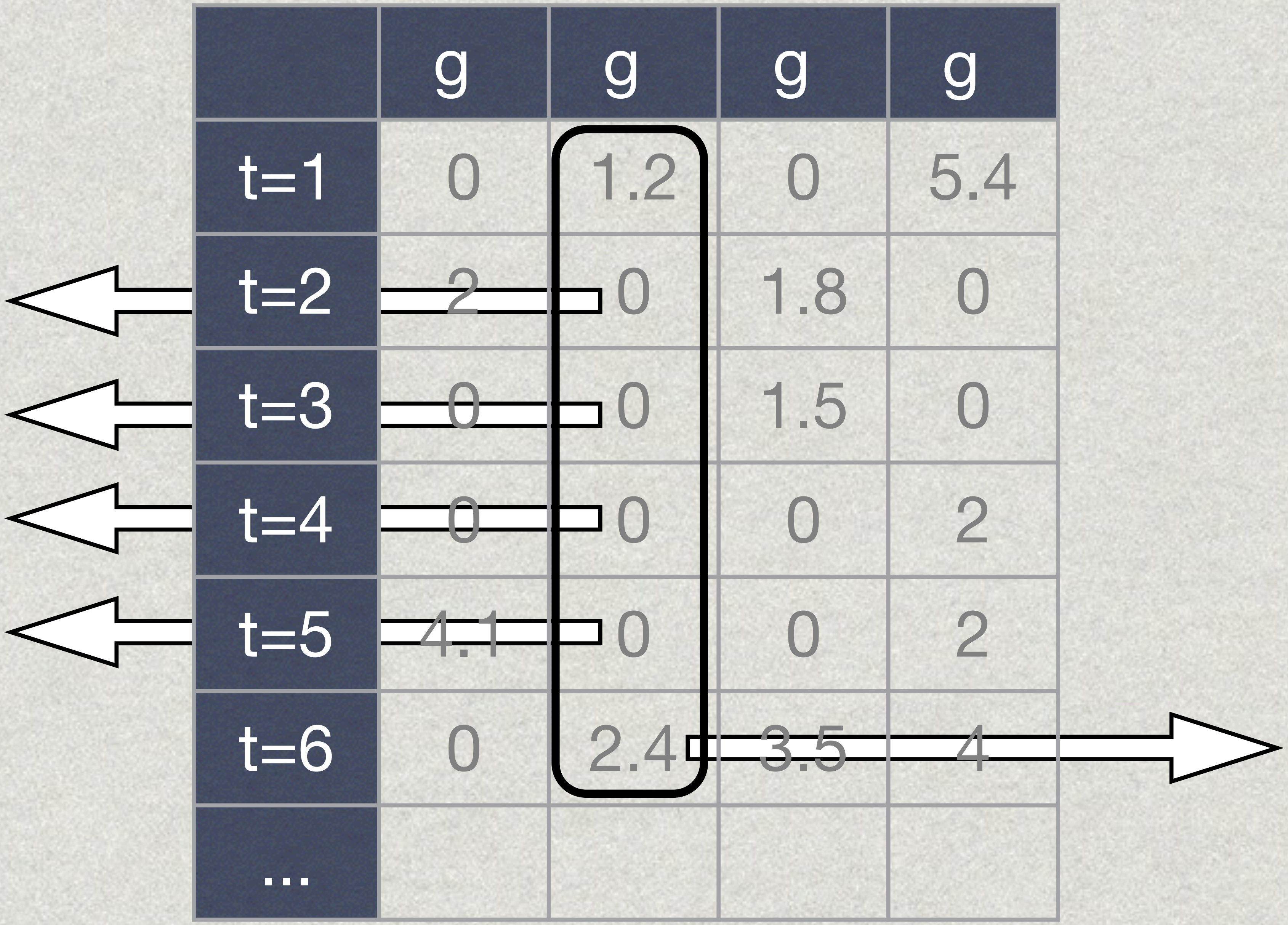
# Efficient High Dimensional Update



SKIP UPDATE PHASE (LAZY EVAL)

| | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... | | | | |

# Efficient High Dimensional Update



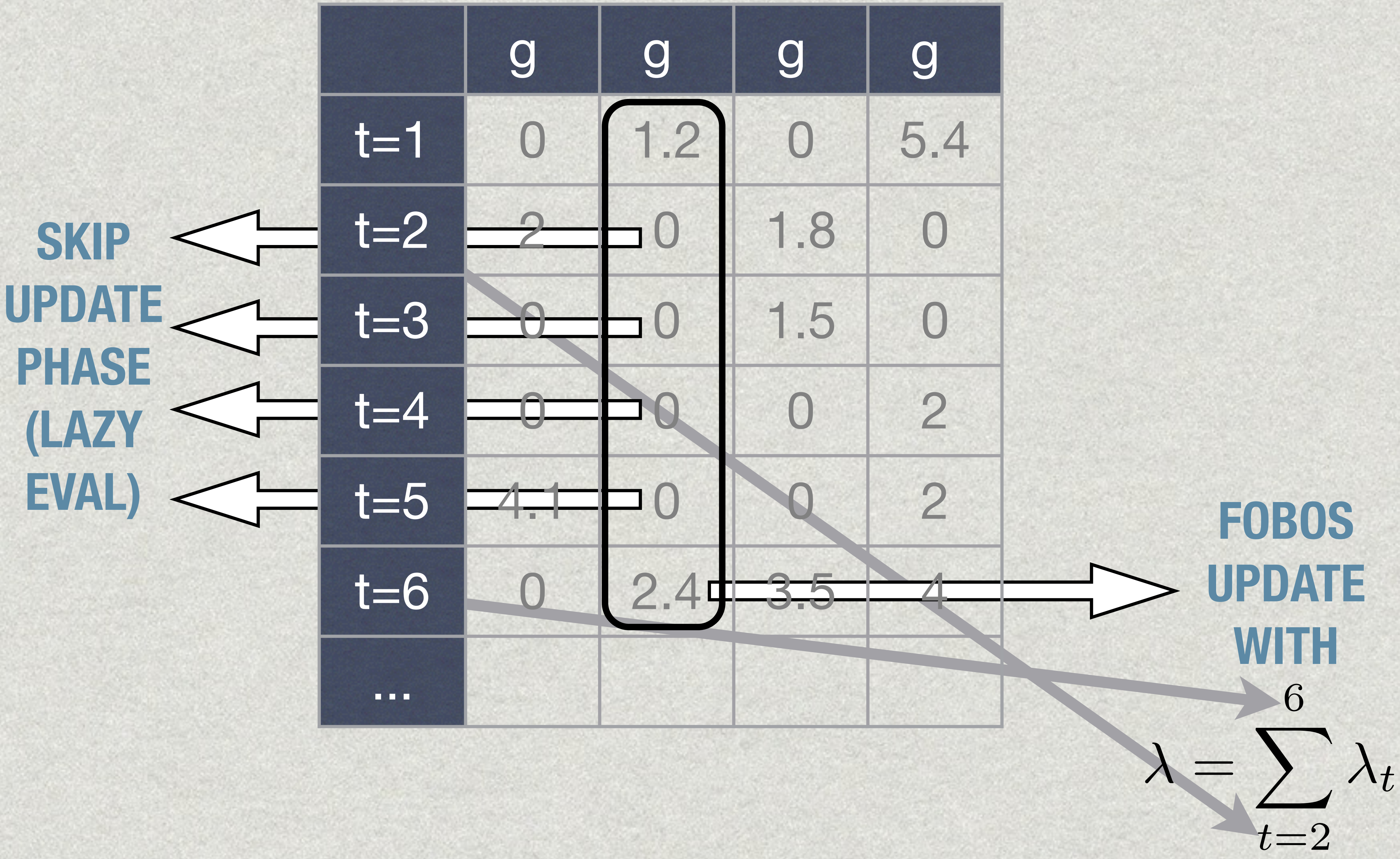
|  | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... |  |  |  |  |

**SKIP UPDATE PHASE (LAZY EVAL)**

**FOBOS UPDATE WITH**

$$\lambda = \sum_{t=2}^{6} \lambda_t$$

# Efficient High Dimensional Update



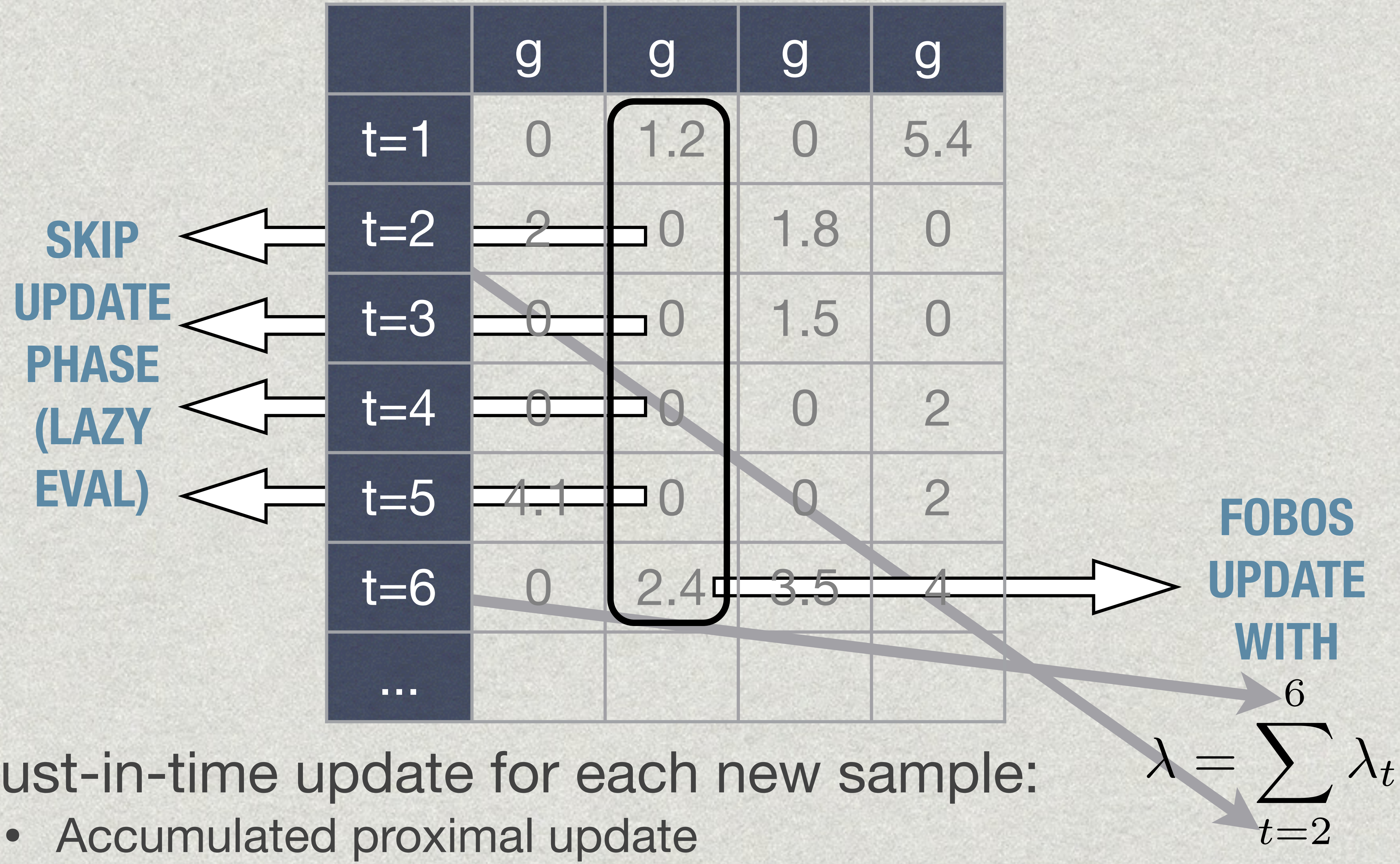
|  | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... |  |  |  |  |

SKIP UPDATE PHASE (LAZY EVAL)

FOBOS UPDATE WITH

$$\lambda = \sum_{t=2}^{6} \lambda_t$$

# Efficient High Dimensional Update



|  | g | g | g | g |
|---|---|---|---|---|
| t=1 | 0 | 1.2 | 0 | 5.4 |
| t=2 | 2 | 0 | 1.8 | 0 |
| t=3 | 0 | 0 | 1.5 | 0 |
| t=4 | 0 | 0 | 0 | 2 |
| t=5 | 4.1 | 0 | 0 | 2 |
| t=6 | 0 | 2.4 | 3.5 | 4 |
| ... |  |  |  |  |

$$\lambda = \sum_{t=2}^{6} \lambda_t$$

Just-in-time update for each new sample:

- Accumulated proximal update
- Stochastic gradient step (w/o further ops)