Logistic Regression & convex optimization

Announcements:

This week we will release P3 and HW3

Recap on Naive Bayes

NB is a **generative model** which models $P(x, y)$

$$P(y \mid x) \propto P(y)P(x \mid y) = P(y)\prod_{i=1}^{d} P(x[i] \mid y)$$

Conditional independent assumption given label

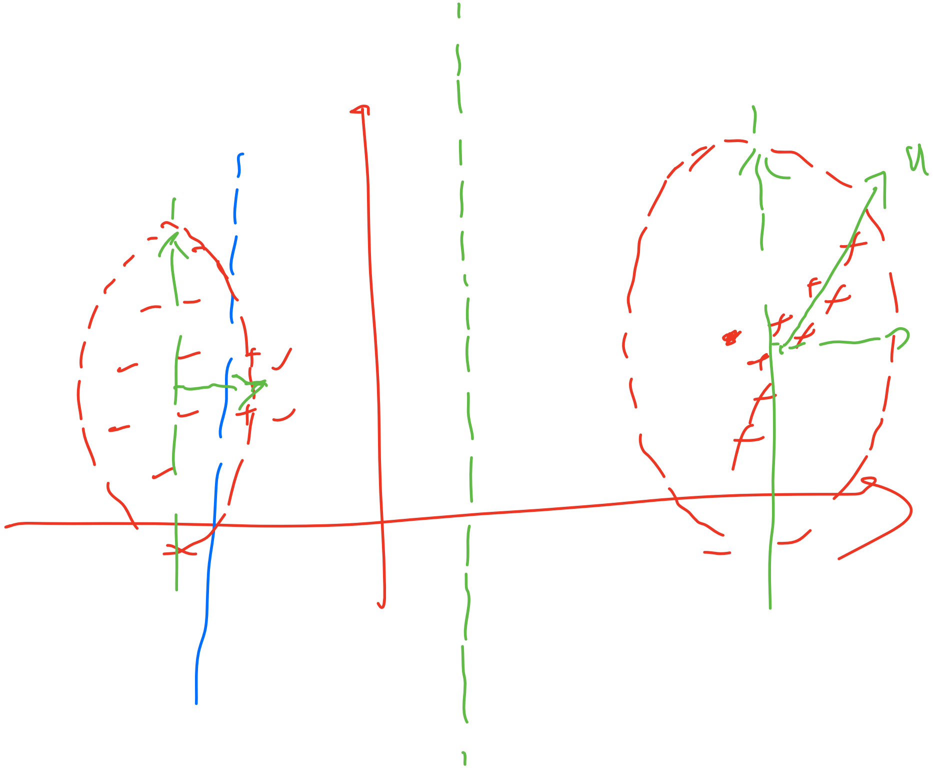
$$\arg\max_{y} P(y \mid x)$$

Perceptron VS Gaussian Naive Bayes

$\frac{1}{\sigma^2}$

$P(x \mid y=-1)$

$P(x \mid y=+1)$



$\arg\max_y P(y \mid x)$

$= \arg\max_y P(y)P(x \mid y)$

Today

Logistic regression — a ***discriminative learning*** approach that directly models $P(y \mid x)$ for classification

Outline for today

1. Logistic Regression

2. Convex optimization

3. Gradient Descent

Logistic Regression

Setting: binary classification $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$,
$x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$

iid

$x \sim P(x)$

$y \sim P(y|x)$

$p(x, y) = P(x)\,p(y|x)$

Logistic Regression

Setting: binary classification $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$, $x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$

(Note, we always assume x contains a constant 1)

Logistic Regression

Setting: binary classification $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$,
$x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$

(Note, we always assume x contains a constant 1)

Logistic regression **directly models** $P(y \mid x)$

$\arg\max_y P(y|x)$

Logistic Regression

Setting: binary classification $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$,
$x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$

(Note, we always assume x contains a constant 1)

Logistic regression **directly models** $P(y \mid x)$

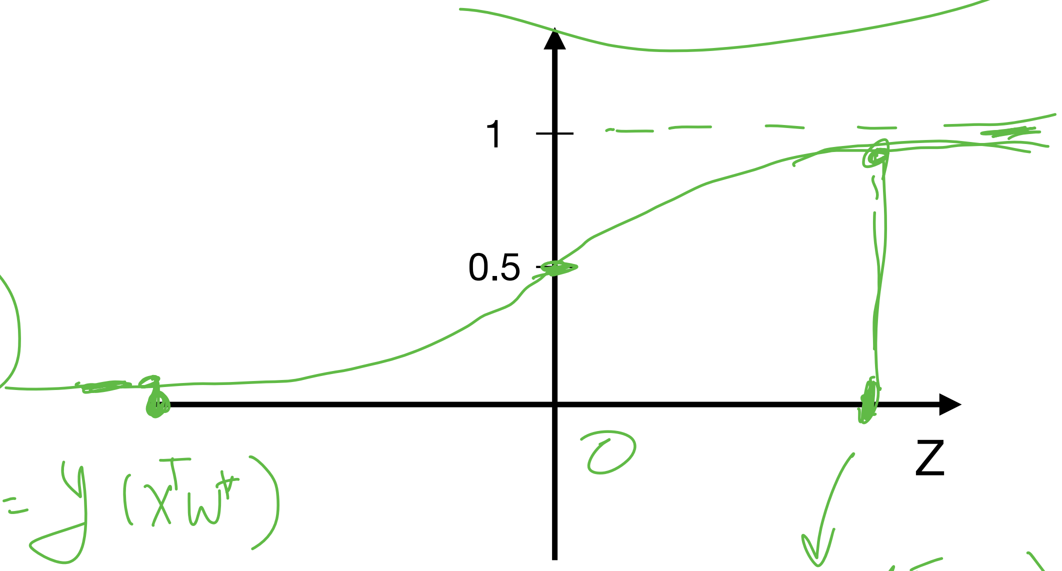
$$P(y \mid x) = \frac{1}{1 + \exp\left(-y(x^\top w^\star)\right)}$$

Logistic Regression

Logistic regression assumes:

$$P(y \mid x) = \frac{1}{1 + \exp\left(-y(x^\top w^\star)\right)}$$

Draw the Sigmoid function $1/(1 + \exp(-Z))$



Logistic Regression

Logistic regression assumes:

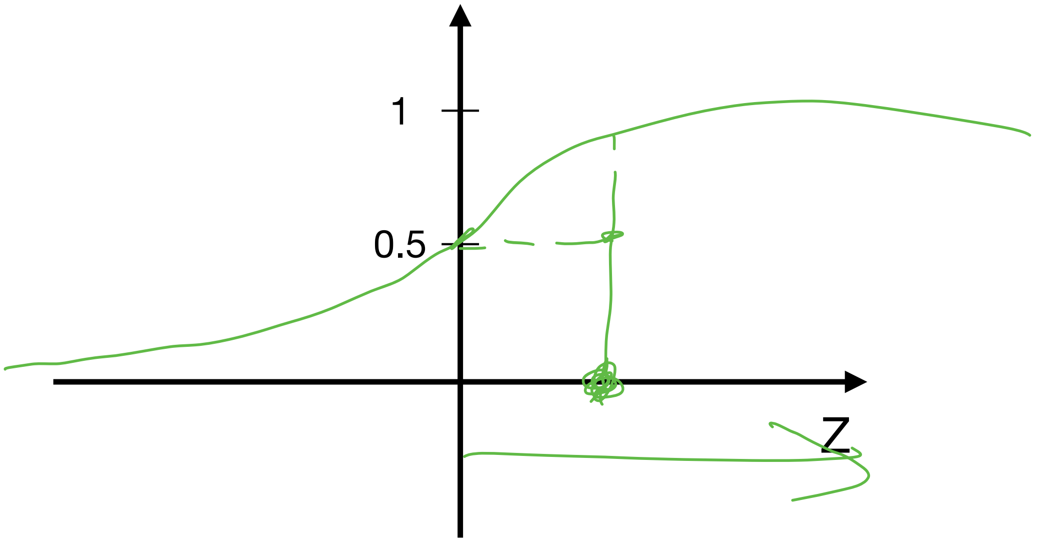
$$P(y \mid x) = \frac{1}{1 + \exp\left(-y(x^\top w^\star)\right)}$$

The model assigns higher prob to

$$y = \text{sign}(x^\top w^\star)$$

$\hookrightarrow y(x^\top w^\star) > 0$

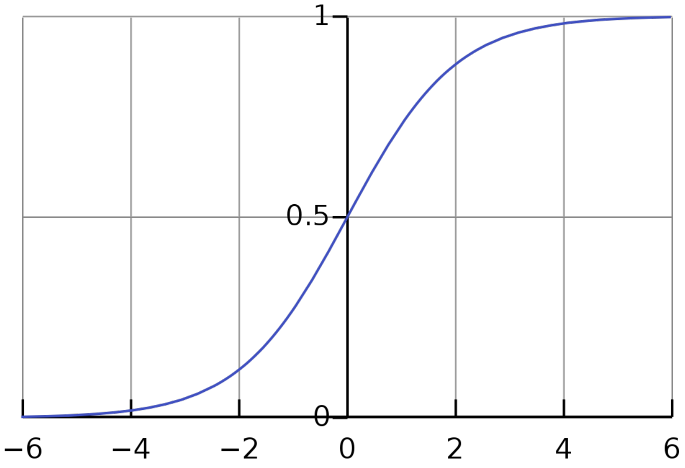
Draw the Sigmoid function $1/(1 + \exp(-Z))$



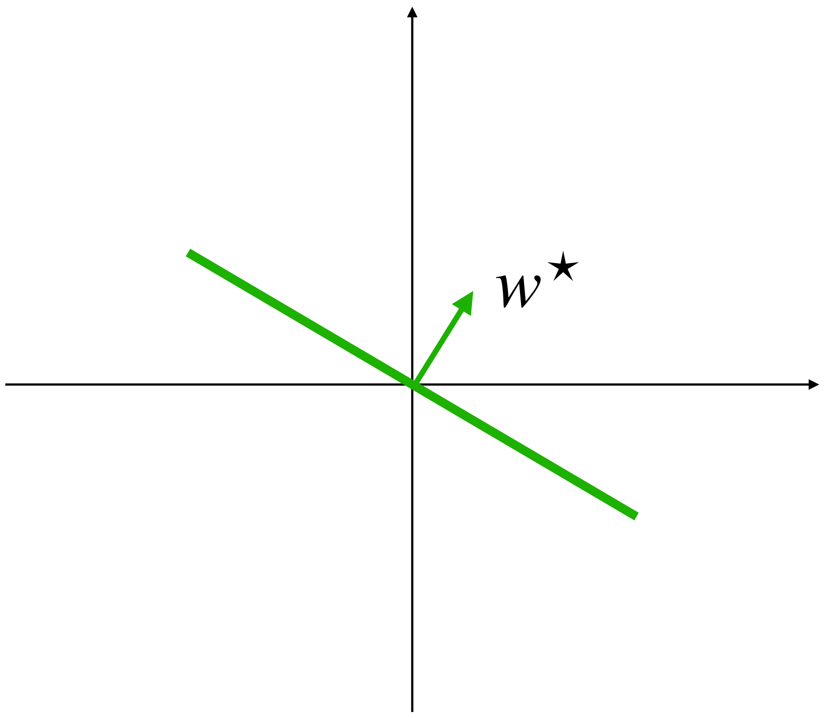
Logistic Regression

Logistic regression assumes:

$$P(y\,|\,x) = \frac{1}{1 + \exp\left(-y(x^\top w^\star)\right)}$$



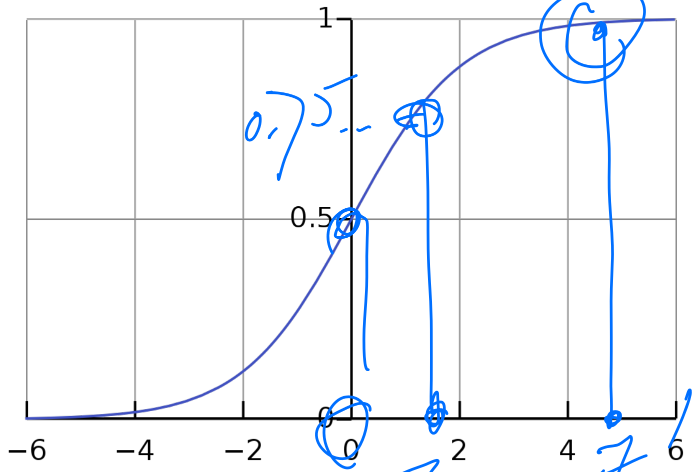
$$z := y(x^\top w^\star)$$



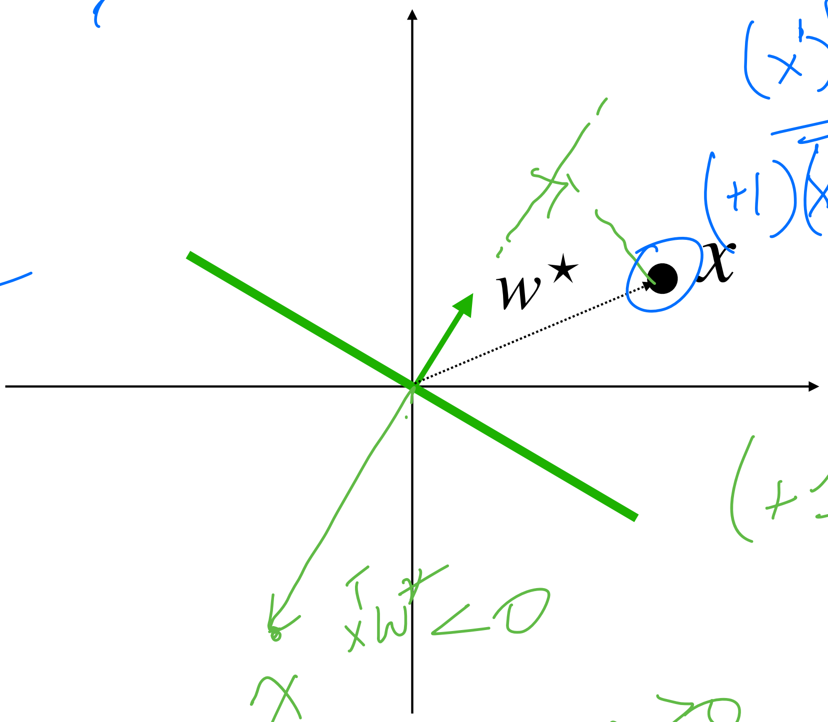
Logistic Regression

Logistic regression assumes:

$$P(y\,|\,x) = \frac{1}{1 + \exp\left(-y(x^\top w^\star)\right)}$$



$z := y(x^\top w^\star)$



Learn via MLE

$Y = \{y_1 \cdots y_n\}$

Recall we have data $\mathscr{D} = \{x_i, y_i\}_{i=1}^{n}$

$X = \{x_1 \cdots x_n\}$

$$\arg\max_{w} P(\mathscr{D} \mid w)$$

$$\hookrightarrow P(\mathscr{D} \mid w) = P(Y \mid X; w)\, P(X; w)$$

$$= P(X)$$

$$= P(Y \mid X; w)\, P(X)$$

Learn via MLE

Recall we have data $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$

$$\arg\max_w P(\mathscr{D} \mid w) = \arg\max_w P\left(\{y_i\}_{i=1}^n \mid \{x_i\}_{i=1}^n ; w\right)$$

Learn via MLE

Recall we have data $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$

$$\arg\max_w P(\mathscr{D} \mid w) = \arg\max_w P\left(\{y_i\}_{i=1}^n \mid \{x_i\}_{i=1}^n ; w\right)$$

i.i.d

$$= \arg\max_w \prod_{i=1}^n P\left(y_i \mid x_i ; w\right)$$

Learn via MLE

Recall we have data $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$

$$\arg\max_w P(\mathscr{D} \mid w) = \arg\max_w P\left(\{y_i\}_{i=1}^n \mid \{x_i\}_{i=1}^n ; w\right)$$

$$= \arg\max_w \prod_{i=1}^n P\left(y_i \mid x_i ; w\right)$$

$$P(y_i \mid x_i ; w) = \frac{1}{1+\exp\left(-y_i\left(x_i^\top w\right)\right)}$$

Plug in logistic assumption and add log:

$$\arg\max_w \sum_{i=1}^n -\ln\left[1 + \exp\left(-y_i(w^\top x_i)\right)\right]$$

Learn via MLE

$$\hat{w}_{mle} := \arg\max_{w} \sum_{i=1}^{n} \ln \left[\frac{1}{1 + \exp\left(-y_i(w^\top x_i)\right)} \right]$$

Intuitively, $\hat{w}_{mle}$ tries to explain the label:

Learn via MLE

$$\hat{w}_{mle} := \arg\max_{w} \sum_{i=1}^{n} \ln \left[\frac{1}{1 + \exp\left(-y_i(w^\top x_i)\right)} \right]$$

Intuitively, $\hat{w}_{mle}$ tries to explain the label:

Q: for $y_i = +1$, what we should expect from $\hat{w}_{mle}^\top x_i$?

$$\hat{w}_{MLE}^\top x_i >> 0$$

$$(y_i)(\hat{w}_{MLE}^\top x_i) >> 0$$

Learn via MLE

$$\hat{w}_{mle} := \arg\max_{w} \sum_{i=1}^{n} \ln \left[\frac{1}{1 + \exp\left(-y_i(w^\top x_i)\right)} \right]$$

Intuitively, $\hat{w}_{mle}$ tries to explain the label:

Q: for $y_i = +1$, what we should expect from $\hat{w}_{mle}^\top x_i$?

Q: for $y_i = -1$, what we should expect from $\hat{w}_{mle}^\top x_i$? < 0

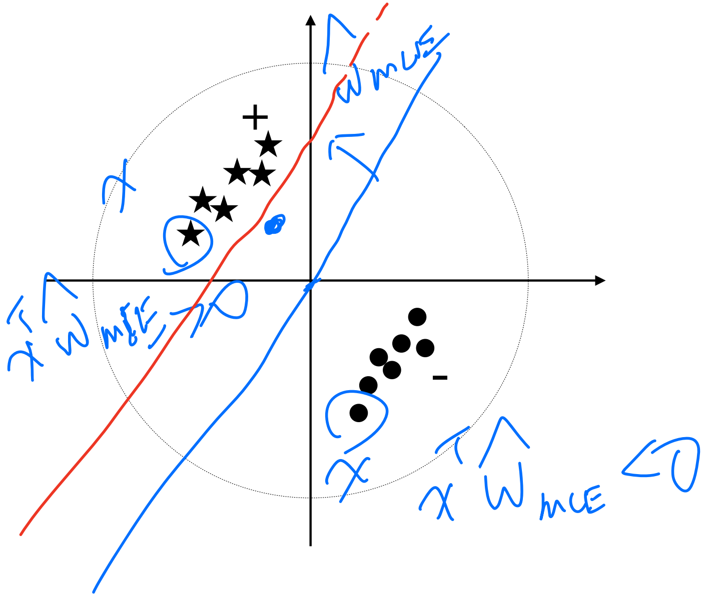
Learn via MLE

$$\hat{w}_{mle} := \arg\max_{w} \sum_{i=1}^{n} \ln \left[\frac{1}{1 + \exp\left(-y_i(w^\top x_i)\right)} \right]$$

Intuitively, $\hat{w}_{mle}$ tries to explain the label:

Q: for $y_i = +1$, what we should expect from $\hat{w}_{mle}^\top x_i$?

Q: for $y_i = -1$, what we should expect from $\hat{w}_{mle}^\top x_i$?



Learn via MAP

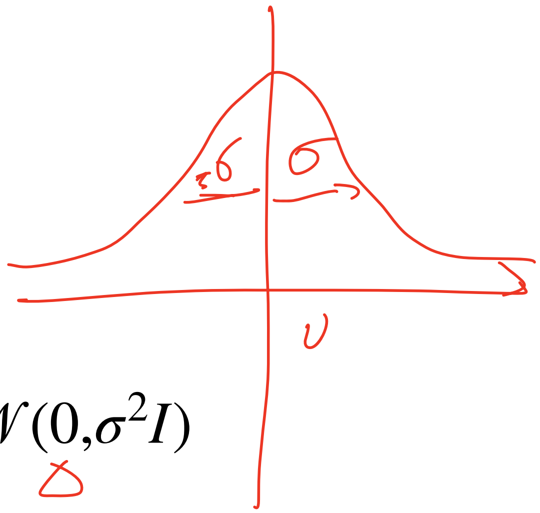
$$P(w \mid \mathscr{D}) \propto P(w)P(\mathscr{D} \mid w)$$

MLE

prior

Learn via MAP

$$P(w \mid \mathcal{D}) \propto P(w)P(\mathcal{D} \mid w)$$



We use Gaussian prior, i.e., $P(w) = \mathcal{N}(0, \sigma^2 I)$

Learn via MAP

$$P(w \mid \mathcal{D}) \propto P(w)P(\mathcal{D} \mid w)$$

We use Gaussian prior, i.e., $P(w) = \mathcal{N}(0, \sigma^2 I)$

$$\arg\max_{w} \ln \left(P(w) \prod_{i=1}^{n} P(y_i \mid x_i, w) \right) = \arg\max_{w} \ln P(w) + \sum_{i=1}^{n} \ln P(y_i \mid x_i, w)$$

Learn via MAP

$$P(w \mid \mathcal{D}) \propto P(w)P(\mathcal{D} \mid w)$$

We use Gaussian prior, i.e., $P(w) = \mathcal{N}(0, \sigma^2 I)$

$$\arg\max_w \ln\left(P(w)\prod_{i=1}^{n} P(y_i \mid x_i, w)\right) = \arg\max_w \ln P(w) + \sum_{i=1}^{n} \ln P(y_i \mid x_i, w)$$

Gaussian

$$= \arg\min_w \left(\sum_{i=1}^{n} \ln\left(1 + \exp(-y_i(w^\top x_i))\right) + \frac{\|w\|_2^2}{2\sigma^2}\right)$$

MLE

prior

Regularization

Comparison to Navie Bayes

1. Logistic regression does not model $P(x \mid y)$

Comparison to Navie Bayes

1. Logistic regression does not model $P(x \mid y)$

2. Gaussian NB leads a linear classifier in the form of
$$P(y \mid x) = 1/(1 + \exp(w^\top x))$$

Comparison to Navie Bayes

1. Logistic regression does not model $P(x|y)$

2. Gaussian NB leads a linear classifier in the form of

$$P(y|x) = 1/(1 + \exp(w^\top x))$$

Gaussian NB is a special case of logistic regression

Outline for today

1. Logistic Regression

2. Convex optimization

3. Gradient Descent

We needs to solve the optimization problem

MLE $\lambda = 0$

$$\hat{w} := \arg\min_w \underbrace{\sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right) + \lambda\|w\|_2^2\right]}_{:=\ell(w)}$$

$\nabla \ell(w) = 0$

Solve for w

We needs to solve the optimization problem

$$\hat{w} := \arg\min_{w} \underbrace{\sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right) + \lambda\|w\|_2^2\right]}_{:=\ell(w)}$$

There is no closed-form solution for the minimizer; luckily, $\ell(w)$ is convex

$y = 1, \quad x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

We needs to solve the optimization problem

$\ln\left[1 + \exp\left(-(w(1) + w(2))\right)\right]$

$$\hat{w} := \arg\min_{w} \underbrace{\sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right) + \lambda \|w\|_2^2\right]}_{:=\ell(w)}$$

There is no closed-form solution for the minimizer; luckily, $\ell(w)$ is convex

We will find an approximate minimizer via **gradient descent**

Setup for Optimization

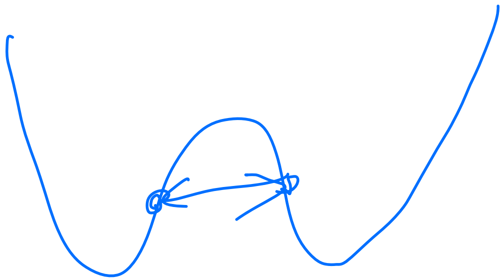
We consider minimizing a (convex) function $\arg\min_{w} \ell(w)$

Setup for Optimization

We consider minimizing a (convex) function $\arg\min_w \ell(w)$

Def of convexity:

$$\forall (x, x'), \alpha \in [0,1],\ \ell(\alpha x + (1-\alpha)x') \leq \alpha \ell(x) + (1-\alpha)\ell(x')$$

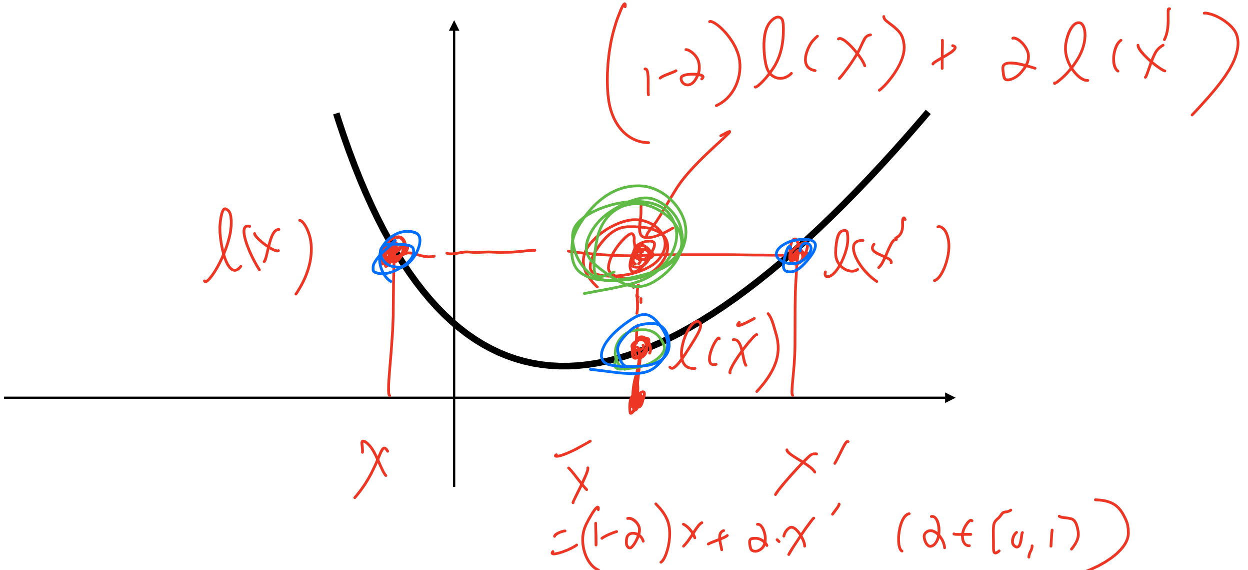


Setup for Optimization

We consider minimizing a (convex) function $\arg\min_{w} \ell(w)$

Def of convexity:

$$\forall (x, x'), \alpha \in [0,1],\ \ell(\alpha x + (1-\alpha)x') \leq \alpha \ell(x) + (1-\alpha)\ell(x')$$

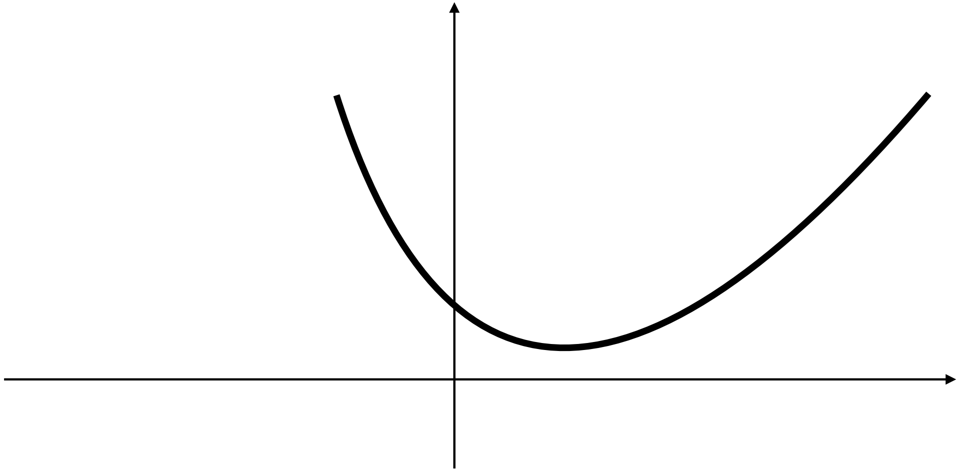


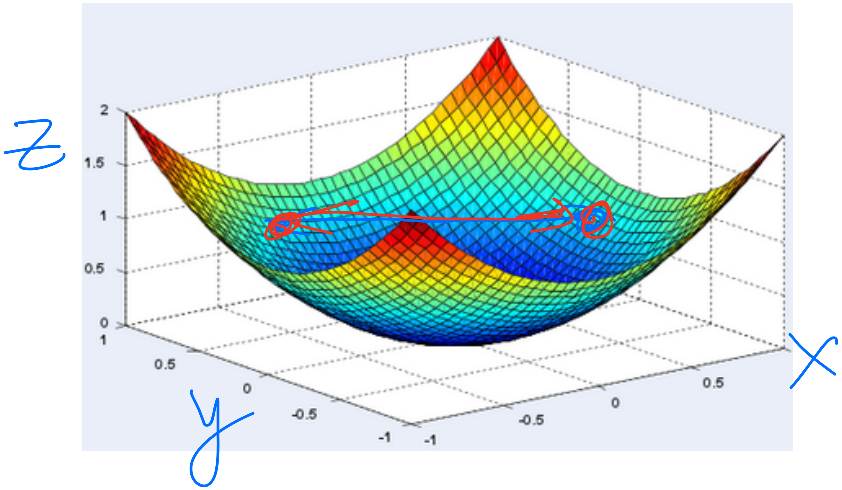
Setup for Optimization

We consider minimizing a (convex) function $\arg\min_{w} \ell(w)$

Def of convexity:

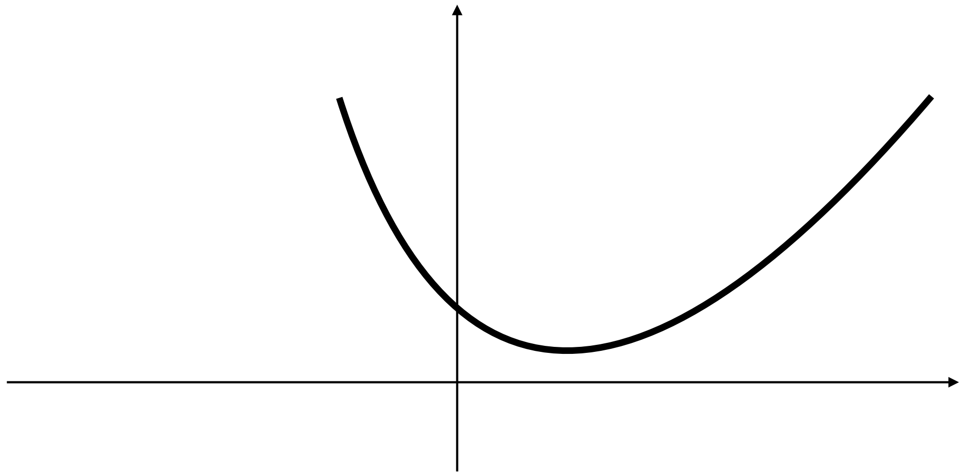
$$\forall (x, x'), \alpha \in [0,1],\ \ell(\alpha x + (1-\alpha)x') \leq \alpha \ell(x) + (1-\alpha)\ell(x')$$





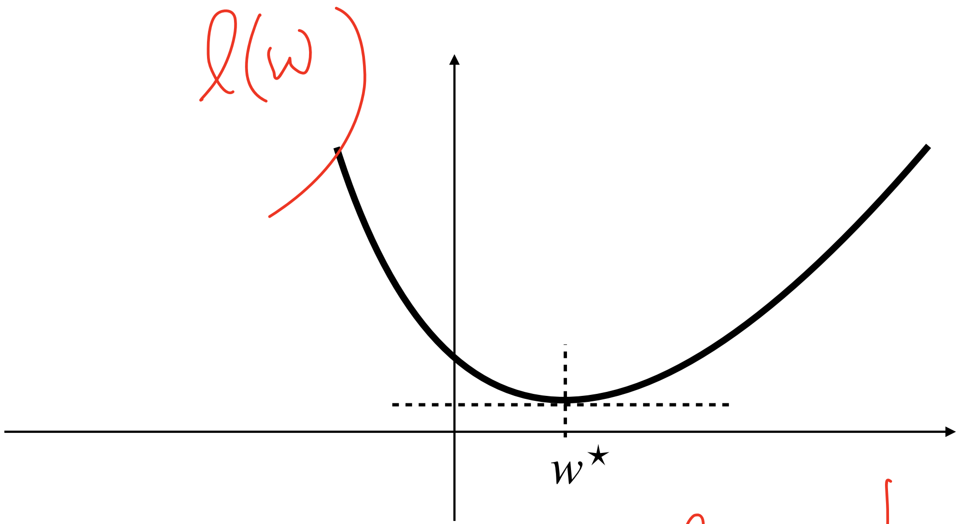
Global minimizer of a convex function

A convex function has global minimizer which has gradient equal to 0



Global minimizer of a convex function

A convex function has global minimizer which has gradient equal to 0

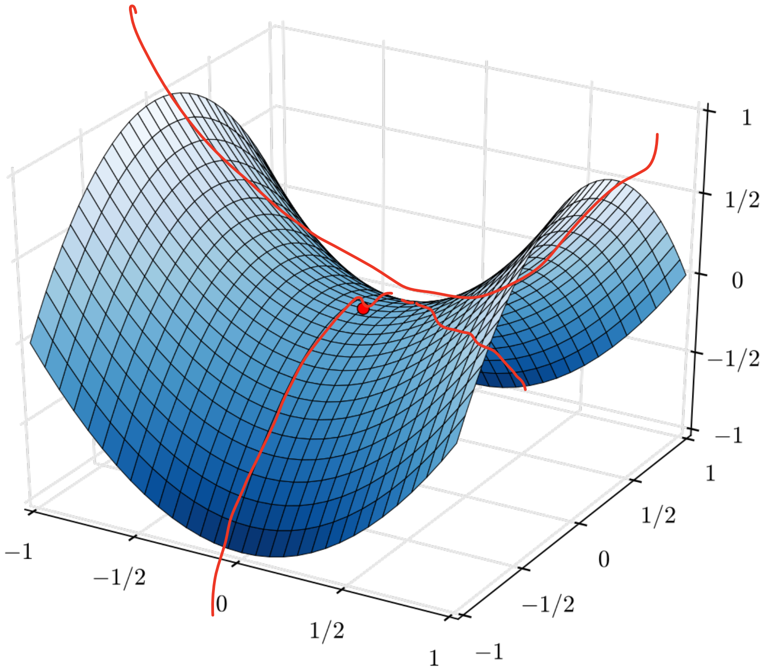


$$\nabla \ell(w) = \begin{bmatrix} \frac{\partial \ell}{\partial w(1)} \\ \vdots \\ \frac{\partial \ell}{\partial w[d]} \end{bmatrix}$$

$$\nabla \ell(w) \Big|_{w = w^\star} = 0$$

Examples of non-convex functions

Saddle point ($\ell(x, y) = x^2 - y^2$)



Outline for today

1. Logistic Regression

2. Convex optimization

3. Gradient Descent

The Gradient Descent algorithm

Goal: minimize $\ell(w)$

$\arg\min_{w} \ell(w)$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

The Gradient Descent algorithm

Goal: minimize $\ell(w)$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

1. Compute gradient $g^t = \nabla \ell(w) \,|_{w=w_t}$

The Gradient Descent algorithm

Goal: minimize $\ell(w)$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

1. Compute gradient $g^t = \nabla \ell(w)\,|_{w=w_t}$
2. Update (GD): $w^{t+1} = w^t - \eta g^t$

The Gradient Descent algorithm

Goal: minimize $\ell(w)$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

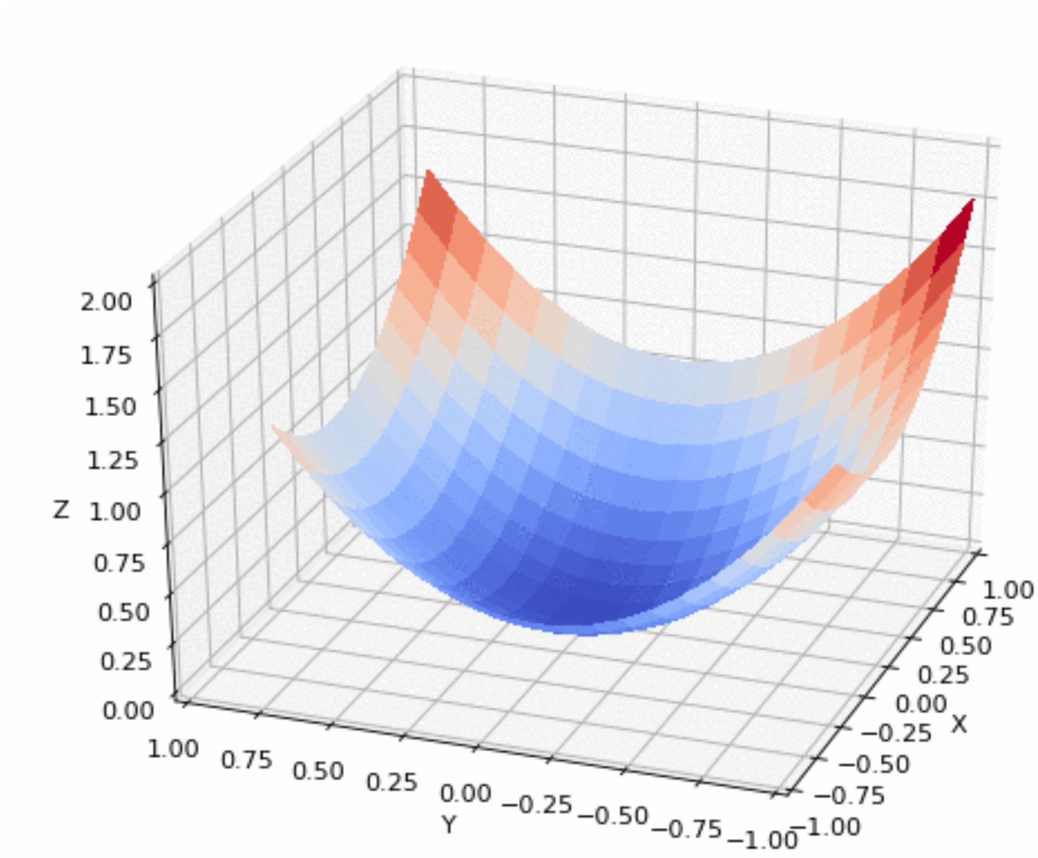
1. Compute gradient $g^t = \nabla \ell(w)|_{w=w_t}$
2. Update (GD): $w^{t+1} = w^t - \eta g^t$

η: learning rate

$\eta > 0$ ← small number

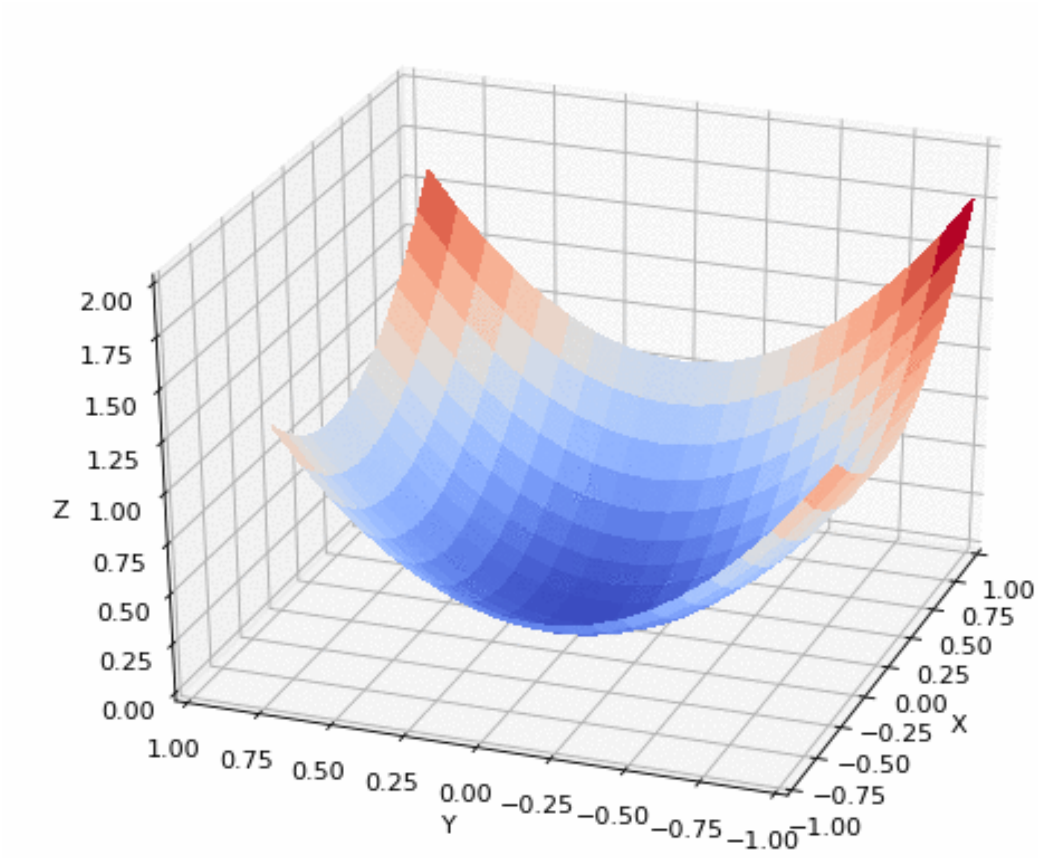
The Gradient Descent demo

$$\min_{x,y}(x^2 + y^2)$$



The Gradient Descent demo

$$\min_{x,y}(x^2 + y^2)$$



Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \rightarrow 0$), we have

Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \to 0$), we have

$$\ell(w - \delta) = \ell(w) - \nabla \ell(w)^\top \delta \quad + \delta^2 + \delta^3$$

$$\delta = \eta \nabla \ell(w)$$

Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \to 0$), we have

$$\ell(w - \delta) = \ell(w) - \nabla \ell(w)^\top \delta$$

Substitute $\delta = \eta \nabla \ell(w)$, with $\eta \to 0^+$

Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \to 0$), we have

$$\ell(w - \delta) = \ell(w) - \nabla \ell(w)^\top \delta$$

Substitute $\delta = \eta \nabla \ell(w)$, with $\eta \to 0^+$

set $\delta = \eta \nabla \ell(w)$

$$\ell(w - \eta \nabla \ell(w)) = \ell(w) - \eta \nabla \ell(w)^\top (\nabla \ell(w))$$

Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \to 0$), we have

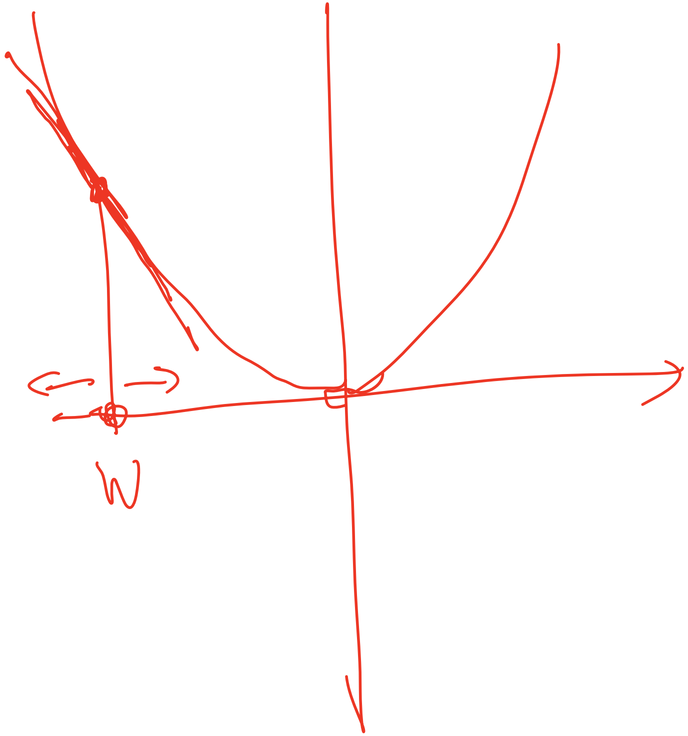
$$\ell(w - \delta) = \ell(w) - \nabla \ell(w)^\top \delta$$

Substitute $\delta = \eta \nabla \ell(w)$, with $\eta \to 0^+$

$$\ell(w - \eta \nabla \ell(w)) = \ell(w) - \eta \nabla \ell(w)^\top (\nabla \ell(w))$$

$$\|\nabla \ell(w)\|_2^2 > 0$$

$$\ell(w - \eta \nabla \ell(w)) < \ell(w), \text{ if } \nabla \ell(w) \neq 0$$



Informal proof for GD convergence

First-order Taylor expansion: for infinitesimally small δ (i.e., $\delta \to 0$), we have

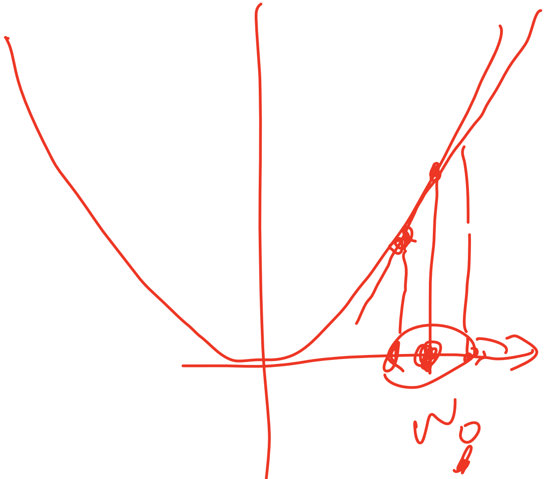
$$\ell(w - \delta) = \ell(w) - \nabla \ell(w)^\top \delta$$

Substitute $\delta = \eta \nabla \ell(w)$, with $\eta \to 0^+$

$$\ell(w - \eta \nabla \ell(w)) = \ell(w) - \eta \nabla \ell(w)^\top (\nabla \ell(w))$$

$$||\nabla \ell(w)||_2^2 > 0$$

i.e., w/ sufficiently small η, GD decrease obj value if $\nabla \ell(w) \neq 0$!



$$\min_w \ell(w_0) - \nabla \ell(w_0)^\top (w - w_0)$$

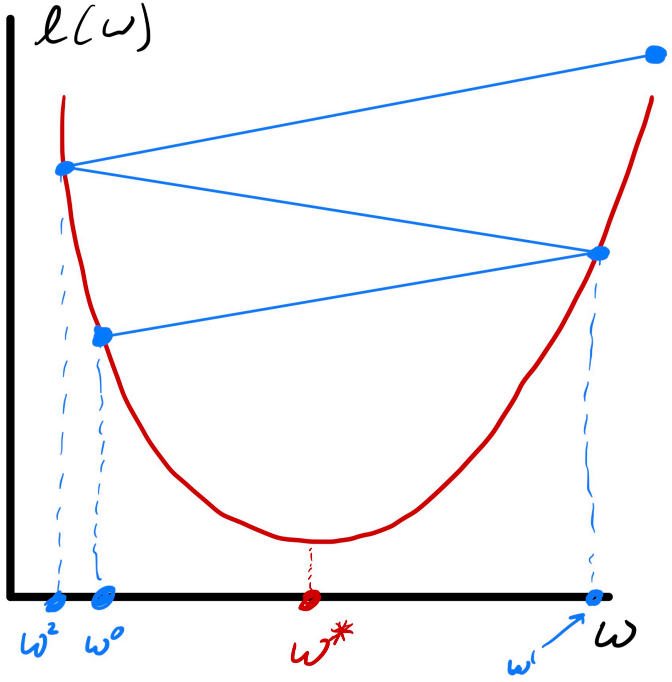
st $\|w - w_0\|_2 \le \delta$

How to set learning rate η in practice?

Large η typically is bad and
can lead to diverge

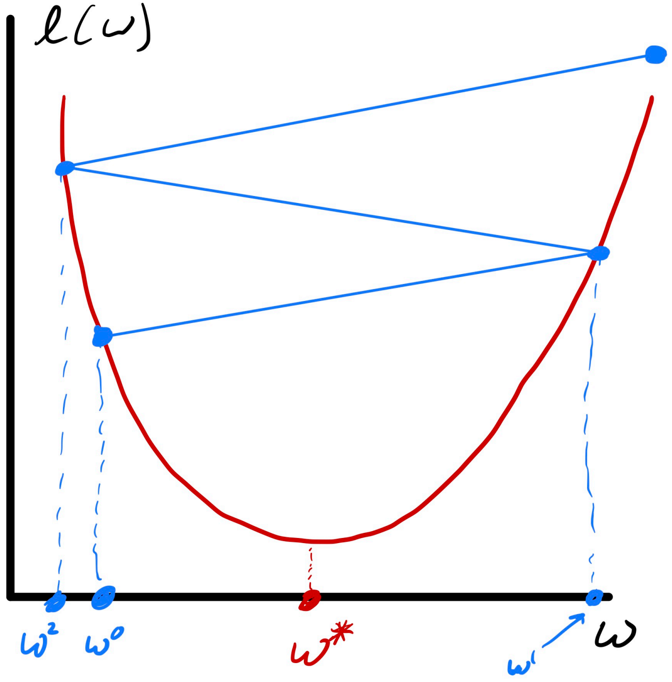
How to set learning rate η in practice?

Large η typically is bad and can lead to diverge



How to set learning rate η in practice?

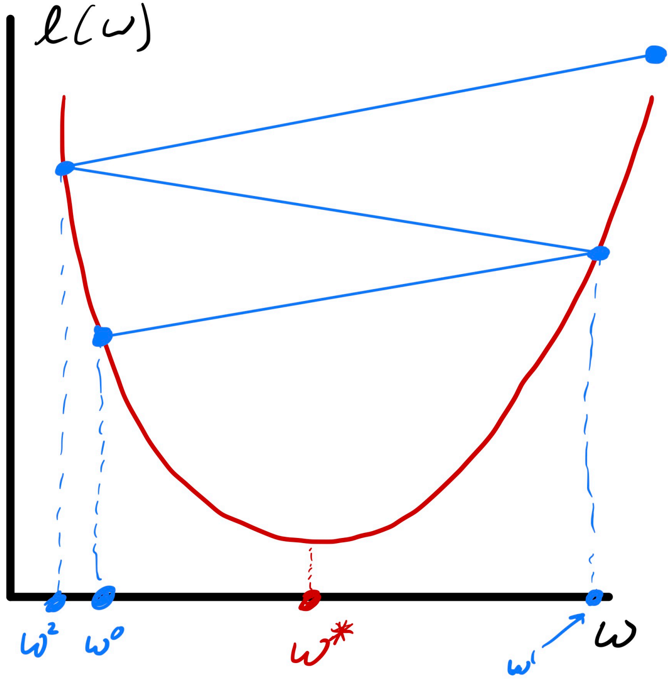
Large η typically is bad and can lead to diverge



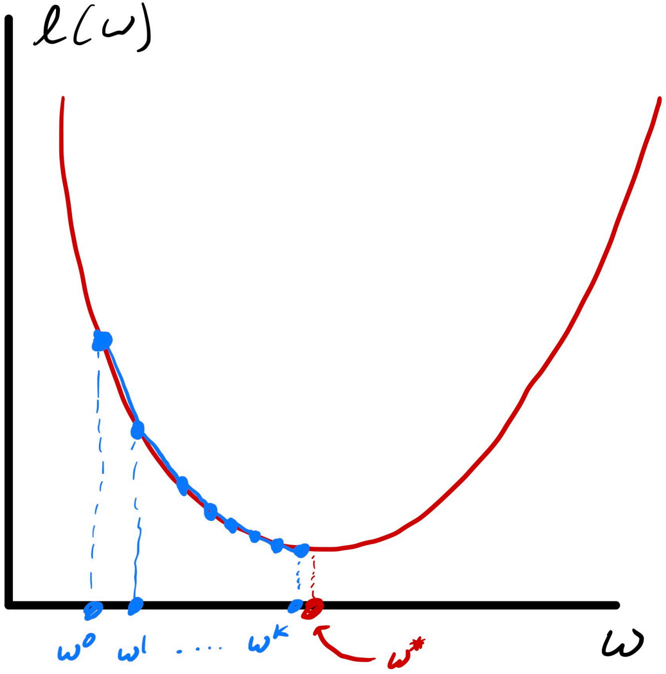
In theory, for convex loss, $\eta = c/\sqrt{k}$ guarantees convergence

How to set learning rate η in practice?

Large η typically is bad and can lead to diverge



In theory, for convex loss, $\eta = c/\sqrt{k}$ guarantees convergence



Let's summarize by applying GD to logistic regression

Recall the objective for LR:

$$\min_w \sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right)\right] + \lambda\|w\|_2^2$$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

Let's summarize by applying GD to logistic regression

Recall the objective for LR:

$$\min_w \sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right)\right] + \lambda \|w\|_2^2$$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

1. Compute gradient $g^t = \sum_i \frac{\exp(-y_i x_i^\top w^t)(-y_i x_i)}{1 + \exp(-y_i x_i^\top w^t)} + 2\lambda w^t$

Let's summarize by applying GD to logistic regression

Recall the objective for LR:

$$\min_w \sum_{i=1}^{n} \ln\left[1 + \exp\left(-y_i(w^\top x_i)\right)\right] + \lambda \|w\|_2^2$$

Initialize $w^0 \in \mathbb{R}^d$

Iterate until convergence:

1. Compute gradient $g^t = \sum_i \frac{\exp(-y_i x_i^\top w^t)(-y_i x_i)}{1 + \exp(-y_i x_i^\top w^t)} + 2\lambda w^t$
2. Update (GD): $w^{t+1} = w^t - \eta g^t$