Machine Learning from Data

Lecture 9: Spring 2021

Today's Lecture

- Logistic Regression
- Gradient Descent

3 types

Probability

Previous Lecture

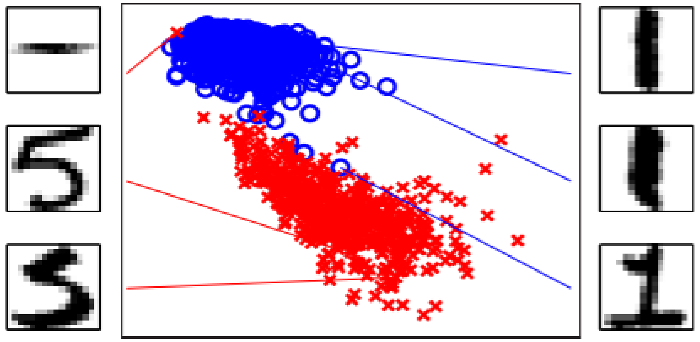
$E_{in} \neq 0$

The linear signal:

$$\boxed{s = \mathbf{w}^{\mathrm{T}}\mathbf{x}}$$

feature construction

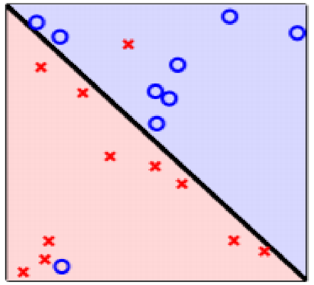
Good Features are Important



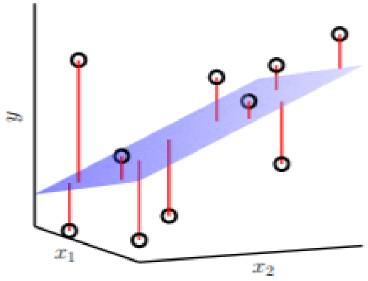
d_{vc}

Before looking at the data, we can **reason** that symmetry and intensity should be good features **based on our knowledge of the problem**.

Algorithms



Linear Classification.
Pocket algorithm can tolerate errors
Simple and efficient



Linear Regression.

$y \in \mathbb{R}$

Single step learning:

$$\mathbf{w} = \mathrm{X}^{\dagger}\mathbf{y} = (\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}\mathbf{y}$$

Very efficient $O(Nd^2)$ *exact* algorithm.

Predicting a Probability (Logistic Regression)

linear signal

Will someone have a heart attack over the next year?

age	62 years
gender	male
blood sugar	120 mg/dL40,000
HDL	50
LDL	120
Mass	190 lbs
Height	5′ 10″
…	…

x

transform
sigmoid

Probability

± 1

$-1, +1$

0.5
0.3
0.9

Classification: $\boxed{\text{Yes/No}}$

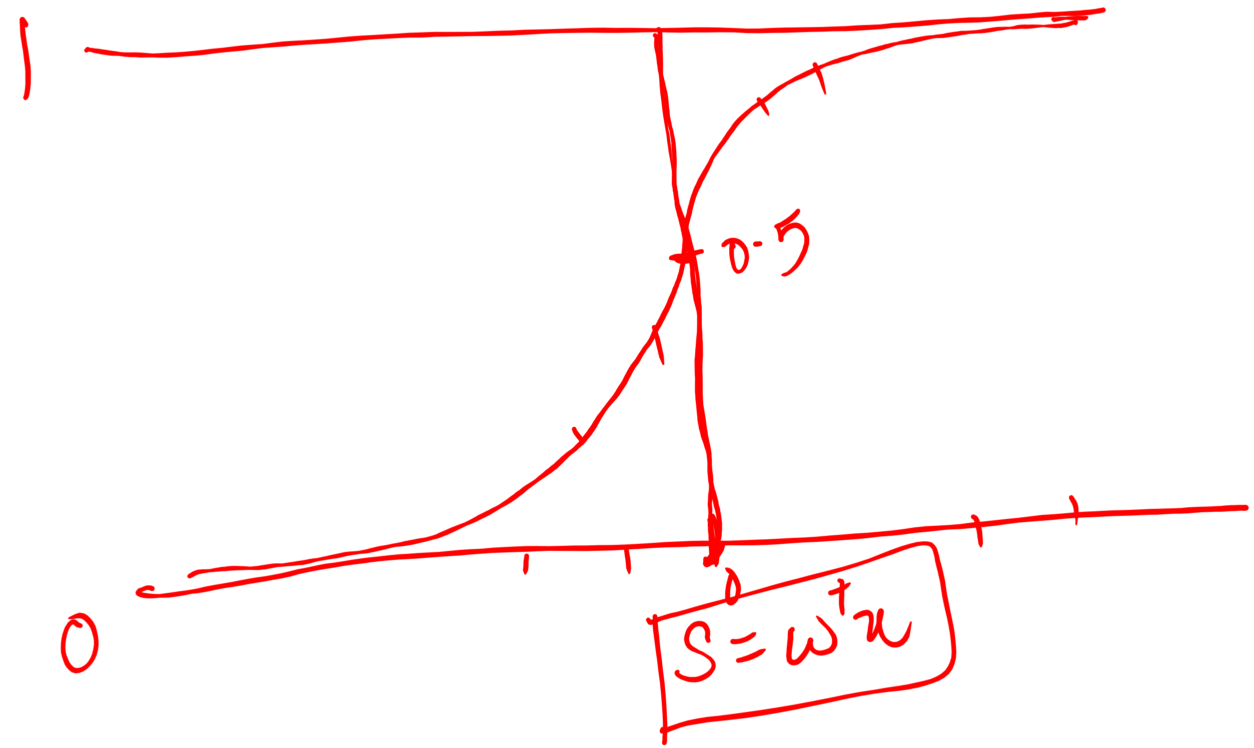
Logistic Regression: Likelihood of heart attack

$$\boxed{\text{logistic regression} \equiv y \in [0, 1]}$$

Logistic Regression $\longrightarrow$ Probability

$s = w^T x \longrightarrow \theta(w^T x)$

$y \in [0, 1]$



$$\theta(s) = \frac{e^s}{1 + e^s}$$

(Mathematically convenient)

Properties of the Sigmoid

- $\theta(s) = \dfrac{e^{s}}{1+e^{s}} = \dfrac{1}{1+e^{-s}}$

$$\frac{1}{\theta(s)} = 1 + e^{-s}$$

$$\theta(-s) = \frac{e^{-s}}{1+e^{-s}} = \frac{1}{1+e^{s}} = 1-\theta(s)$$

How to get the Probability?

- Patients : $x_1 \quad x_2 \cdots\cdots \quad x_n$

$\downarrow \quad \downarrow$

Observe : $y_1 \quad y_2 \cdots\cdots\cdots \quad y_n \; (+1/-1)$

(Yes/No) $\qquad y_i$

Don't have the exact data

$\longrightarrow$ Probability Values

Target function

- Probabilistic target function (Ground Truth)

$$P[+1 \mid x] = f(x)$$

$$P[-1 \mid x] = 1 - f(x)$$

$$h(x) = \theta(w^T x) \simeq f(x)$$

$\underline{g(x)}$ $\underline{E_{in}(w)}$

$$P[+1 \mid x] \simeq h(x)$$

$$P[-1 \mid x] \simeq 1 - h(x)$$

Observe $x_n \longrightarrow y_n = +1$ then $h(x_n) \simeq 1$

$x_n \longrightarrow y_n = -1$ then $h(x_n) \simeq 0$

$$\therefore h(x_n) \simeq \frac{1}{2}(1 + y_n)$$

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}\left(h(x_n) - \frac{1}{2}(1+y_n)\right)^2$$

We cannot use this because:

i) Inconvenient to minimize.

ii) Where is the probabity approx.?

What is a better error function?

CROSS ENTROPY ERROR

$$E_{in}(w) = \frac{1}{N} \sum_{i=1}^{N} \ln\left(1 + e^{-y_n(w^T x_n)}\right)$$

Better behaved + Probabilistic interpretation.

Data : $(x_1, y_1)\ (x_2 y_2) \cdots (x_N\ y_N) \cdots\ y_n \in \{-1, +1\}$

$\downarrow$

IID $\longrightarrow$ One of the pillars of theory

Observing $y_1 y_2 \cdots y_N \longrightarrow x_1, x_2 \cdots x_N$

$$P[y_1 y_2 \ldots y_N \mid \underline{x_1, x_2 \ldots x_N}] \longrightarrow \text{LIKELIHOOD}$$

$$= P(y_1|x_1) \cdot P(y_2|x_2) \ldots P(y_N|x_N)$$

$$= \underbrace{\prod_{n=1}^{N} P(y_n|x_n)}$$

$$\longrightarrow \begin{cases} h(x_n) & \text{if } y_n = +1 \\ 1 - h(x_n) & \text{if } y_n = -1 \end{cases}$$

$$h(x_n) = \underline{\theta(w^T x_n)}$$

$$1 - h(x_n) = 1 - \theta(w^T x_n)$$

$$= \theta(-w^T x_n)$$

$$\boxed{\theta(-s) = 1 - \theta(s)}$$

$$\left[\begin{array}{l} \text{If } y_n = +1 \text{ then } h(x_n) \longrightarrow \theta(w^T x_n) \\ \text{If } y_n = -1 \quad \text{''} \quad h(x_n) \longrightarrow \theta(-w^T x_n) \end{array}\right\}$$

$$\longrightarrow \theta(y_n(w^T x_n))$$

$$\prod_{n=1}^{N} P(y_n|x_n) \longrightarrow \theta(y_n(w^T x_n))$$

$$\boxed{\mathcal{L} = \prod_{n=1}^{N} \theta(y_n(w^T x_n))}$$ Likelihood

Means

Goal: Choose w's $\rightarrow$ maximize

Maximum Likelihood (MLE)

$L(w) \longrightarrow$ maximize

$$\underset{w}{\operatorname{argmax}} L(w) \Longleftrightarrow \underset{w}{\operatorname{argmax}} \ln L(w)$$

$$\underset{w}{\operatorname{argmax}} \frac{1}{N} \ln L(w) \Longleftrightarrow \underbrace{\underset{w}{\operatorname{argmin}} -\frac{1}{N} \ln L(w)}$$

Choose weights w to minimize $-\frac{1}{N} \ln L(w)$

$$\therefore \min -\frac{1}{N} \sum_{i=1}^{N} \ln \theta(y_n w^T x_n) = \frac{1}{N} \sum_{i=1}^{N} \ln \frac{1}{\theta(y_n w^T x_n)}$$

Cross entropy error!

$$E_{in}(w) = \frac{1}{N} \sum_{i=1}^{N} \ln\left(1 + e^{-y_n w^T x_n}\right)$$

How to minimize E_{in}?

-

$$\underline{E_{\text{in}}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \ln(1 + e^{-y_n \cdot \mathbf{w}^{\mathrm{T}}\mathbf{x}_n})$$

$y_n = -1$, $w^T x_n << 0$

$y_n = +1$
$w^T x_n >> 0$

Classification: Pocket Algo.
Regression: Pseudo Inverse $\quad \underline{\nabla E_{in}(w) = 0}$

Analytically not possible
Iteratively possible.

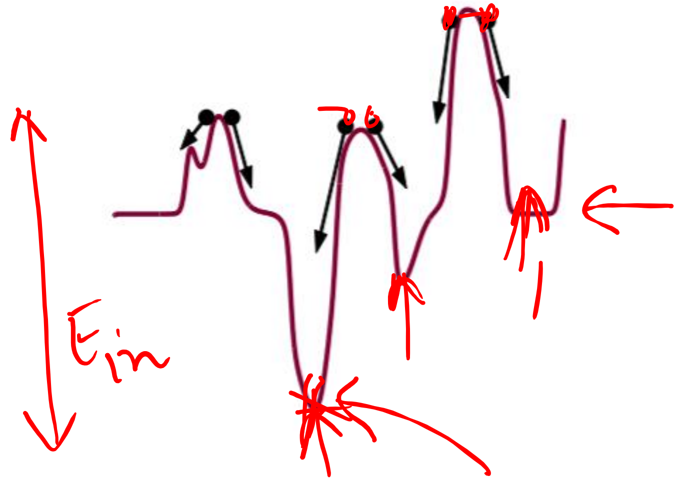
Ball Phenomena

Ball on a complicated hilly terrain

— rolls down to a *local valley*

↑

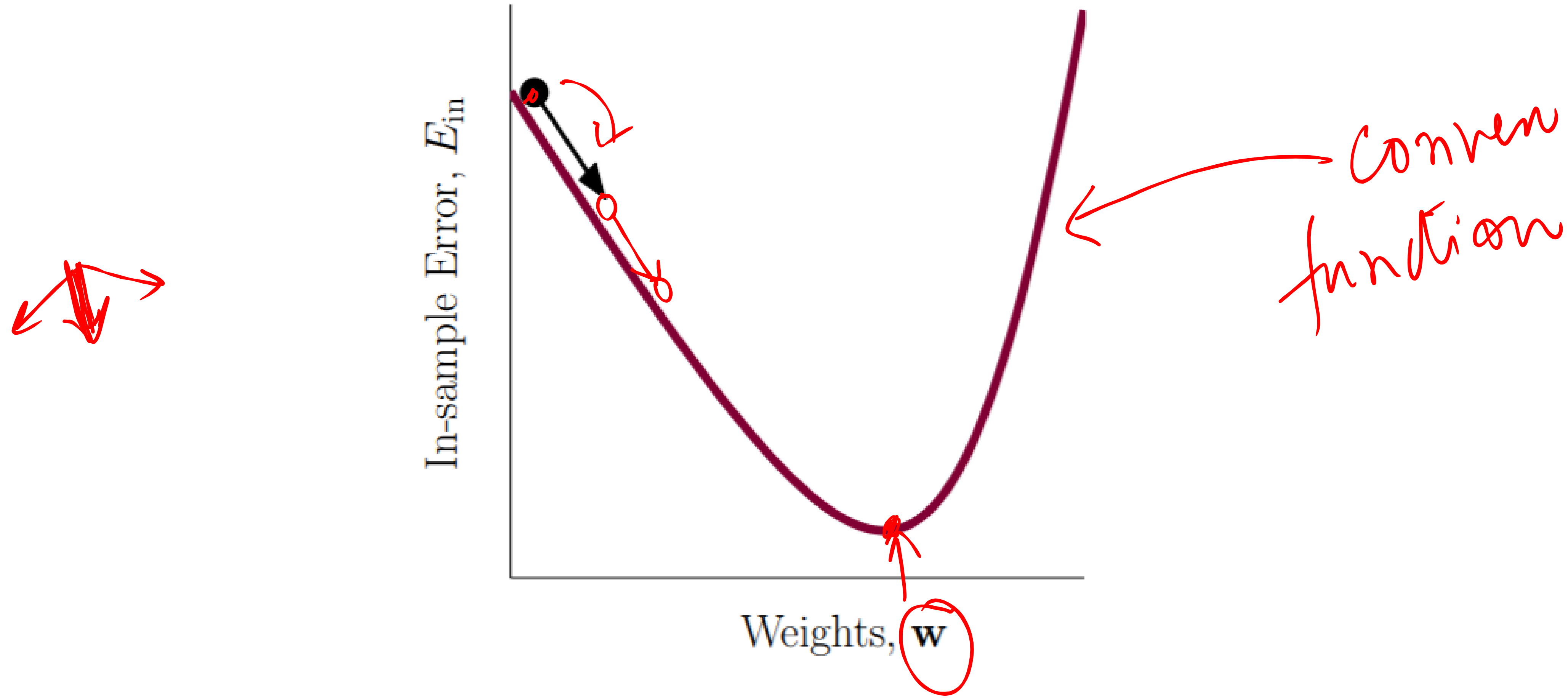
this is called a *local minimum*



Questions:

How to get to the bottom of the deepest valey?

How to do this when we don't have gravity?


In-sample Error, E_{in}
Weights, **w**
convex
function

How do we roll down?

Iteration (t)

$w(t)$: Pick a direction i.e. a unit vector $\hat{v}$. Take a small step in that direction. η

ΔE_{in}

$\downarrow$

$\hat{v}$

Pick $\hat{v}$ to make $E_{in}(w(t+1))$ as small as possible

ΔE_{in}

$w(t+1) = w(t) + \eta\hat{v}$

$E_{in}(w(t))$

$\Delta = E_{in}(w(t+1)) - E_{in}(w(t))$

$\Delta E_{in} = \nabla E_{in}(w(t))(\eta\hat{v})$

By Cauchy Shwatz inequality

$\geq -\eta\|\nabla E_{in}(w(t))\|$

$$\hat{v} = - \frac{\nabla E_{in}(w(t))}{\|\nabla E_{in}(w(t))\|}$$ -ve of the gradient

Summarize

i) Initialize $t=0$, $w(0)$

ii) t : $w(t)$

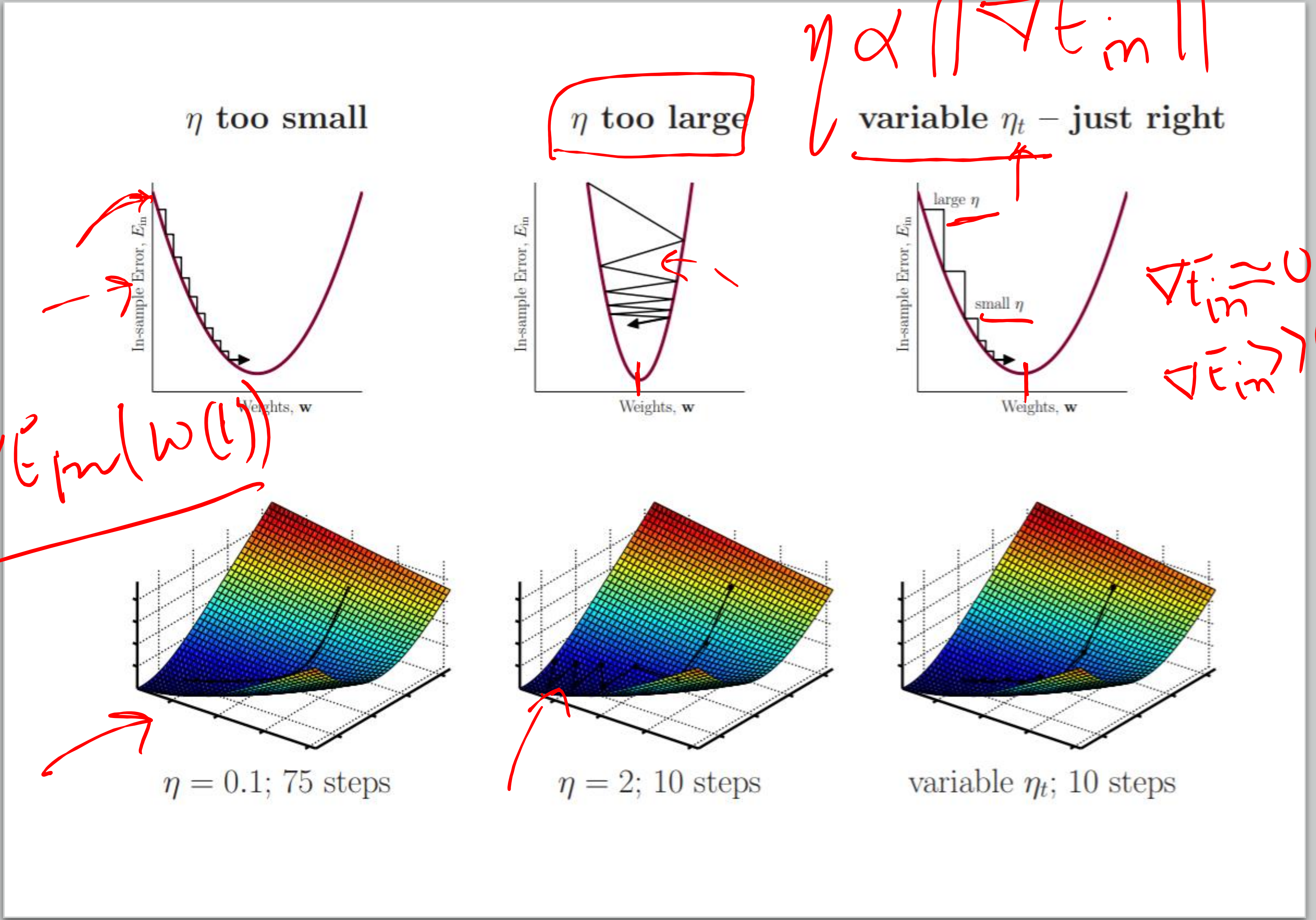
Compute $\nabla E_{in}(w(t))$

$$w(t+1) = w(t) - \eta \frac{\nabla E_{in}(w(t))}{\cancel{\|\nabla E_{in}\, w(t)\|}}$$

$t+1$

Normalized gradient descent.

Step Size



$\eta \propto \|\nabla E_{in}\|$

$\nabla E_{in} \approx 0$

$\nabla E_{in} > 0$

$w(t+1) = w(t) - \eta_t \nabla E_{in}(w(t))$

Batch gradient descent!

i) $E_{in}(w) \longrightarrow$ differentiable

ii) Inefficient. $\nabla E_{in} \longrightarrow O(Nd)$

Updating weights $\rightarrow O(d+1)$

How to roll down efficiently?

Stochastic gradient descent

$$E_{in}(w) = \frac{1}{N} \sum_{i=1}^{N} e(w^T x, y_n)$$

Randomly

$x_n \rightarrow e(w^T x_n, y_n) \longrightarrow$ minimize

$$w(t+1) \Rightarrow w(t) - \eta \nabla e(w^T x_n, y_n)$$

∇E_{in}

$$e(w^T x_n, y_n) = \ln\left(1 + e^{-y_n(w^T x_n)}\right)$$

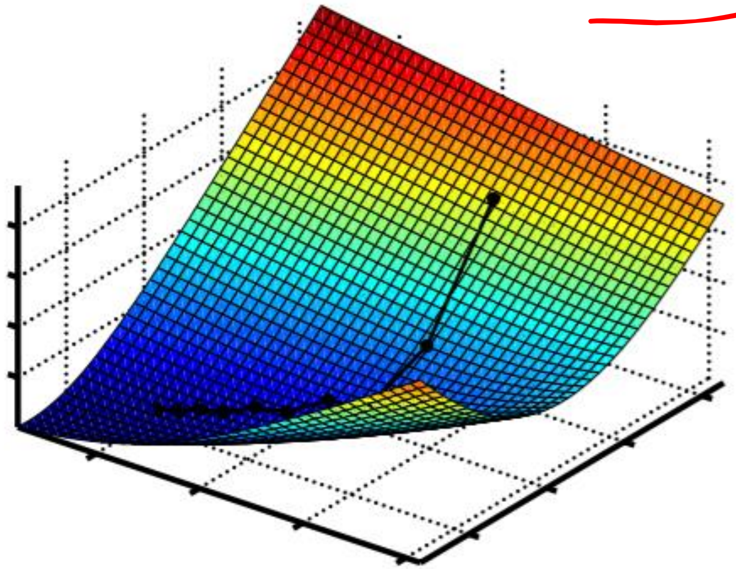
$$\nabla e(w^T x_n, y_n) = \frac{1}{\left(1 + e^{-y_n w^T x_n}\right)} \cdot -y_n x_n \cdot e^{-y_n(w^T x_n)}$$

$$= \frac{-y_n x_n}{1 + e^{y_n w^T x_n}}$$

$$w(t+1) = w(t) + \frac{y_n x_n \eta}{1 + e^{y_n w^T x_n}}$$

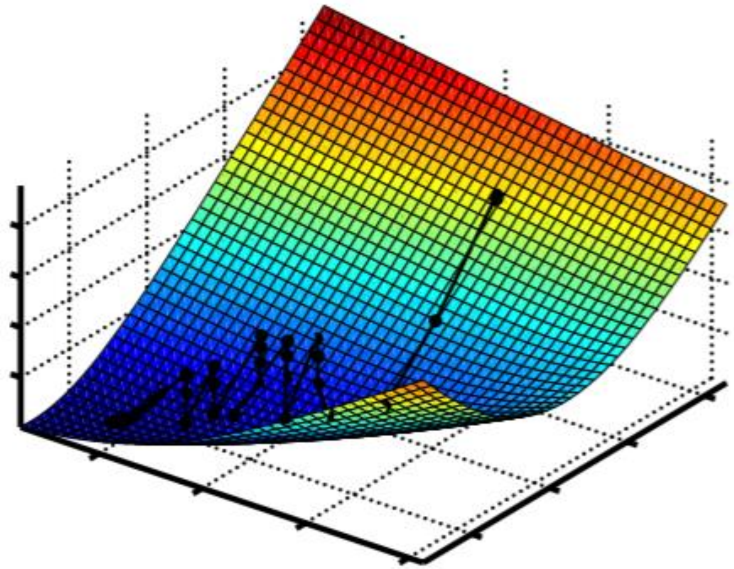
N

GD Batch



$\eta = 6$
10 steps
$N = 10$

SGD



$\eta = 2$
30 steps

Thanks!