Machine Learning: Basic Principles

How to Specify A Machine Learning Problem?

Salo, September 2018

Guiding Questions

- How to formulate your business as a ML Problem ?
- How to determine which algorithm to use for your problem ?

Outline

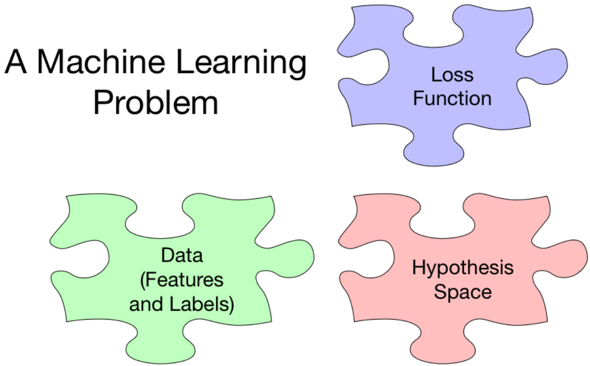
A ML Application

- hard disk full of images
- for a few images it is known if they show a cat or a dog
- develop a software tool ("app") to label all images

Design Choices to be Made

- in which format represent the images ? (what features?)
- which algorithms should we use ? (which hypothesis space?)
- how to evaluate our labelling tool? (how to validate?)
- how to tune for best performance? (how to train?)

Main Components of a ML Problem



Outline

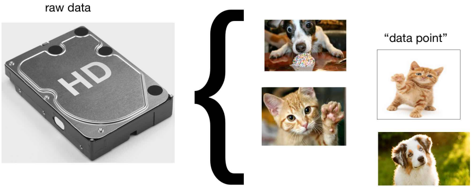
Raw Data

per se, the dataset is just a (huge) pile of bits

clever parsing of data might be most difficult part of ML problem!

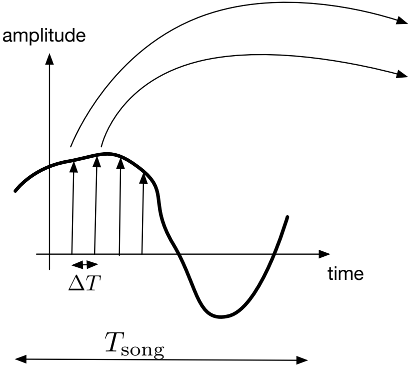
From Raw Data to Vectors (Data Points)

- need to parse raw data into more manageable form
- break raw data into atomic pieces (data points)



- ith data point encoded by vector $\mathbf{z}^{(i)} \in \mathbb{R}^d$
- dataset amounts to a bunch of vectors $\{\mathbf{z}^{(i)}\}_{i=1}^{N}$

From Audio to Vectors

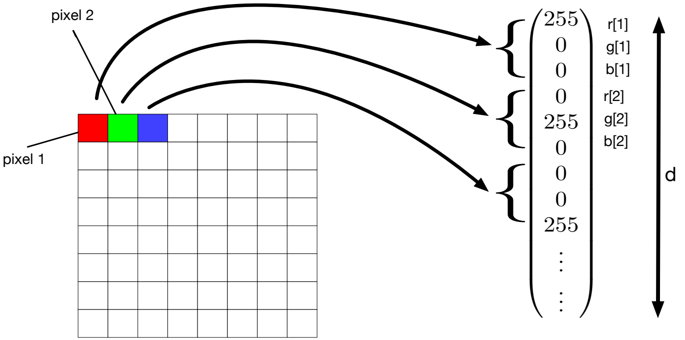


$$\begin{pmatrix} 0.85 \\ 0.9 \\ 0.87 \\ 0.7 \\ \vdots \\ \vdots \end{pmatrix} \in \mathbb{R}^d$$

$$d \approx T_{\text{song}} / \Delta T$$

what are typical values of ΔT and T_{song} for rock song ?

From RGB Images to Vectors



8 x 8 pixels (red green blue bitmap)

Labeled Data

- partition data point as $\mathbf{z} := (\mathbf{x}, y)$
- input "features" $\mathbf{x} \in \mathcal{X}$, "label" / "output" / "target" $y \in \mathcal{Y}$

data point $\mathbf{z} = (\mathbf{x}, \text{“dog”})$
image pixels $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$
label (ouput) $y \in \mathcal{Y} = \{\text{“dog”}, \text{“cat”}\}$

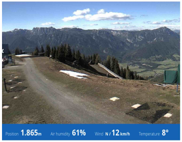
- applications with discrete $\mathcal{Y}$ called classification problems
- applications with continuous $\mathcal{Y}$ called regression problems

A Key Message

- in real-life applications its not obvious what part of data is label and what should be the features
- feature and label space $\mathcal{X}, \mathcal{Y}$ are design choice! (we have to find the most useful choices for our application at hand!)
- HOWEVER, there are methods to automatically choose good features ("Feature Learning")

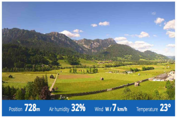
Labeled Data for Regression

data point $\mathbf{z}^{(i)}$ consists of pixels $\mathbf{x}^{(i)}$ (=features) and temperature $y^{(i)}$ (=label)



$$\mathbf{z}^{(1)} = \left(\mathbf{x}^{(1)}, y^{(1)} = 8\right)$$

$$\mathbf{z}^{(2)} = \left(\mathbf{x}^{(2)}, y^{(2)} = 8\right)$$



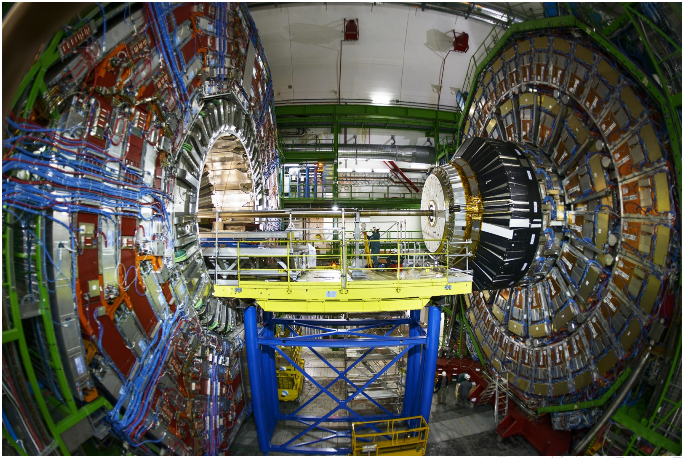
$$\mathbf{z}^{(3)} = \left(\mathbf{x}^{(3)}, y^{(3)} = 23\right)$$

Label Information is worth Gold!

- accurate label information is extremely precious
- ML most powerful with vast amounts of labeled data (=training data)
- HOWEVER, obtaining labels is typically costly
- "labelling" of data often requires human (expert) labour

Aquiring Labels in Marine Biology

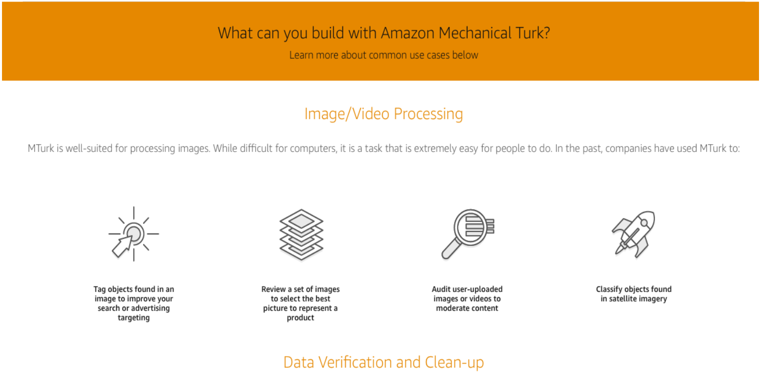
Acquiring Labels in Particle Physics



Acquiring Labels in Pharmacology

doing a good job as ML scientist/engineer might save lives !!!

The Amazon Mechanical Turk



you can hire human labelling workforce!

Labels are Costly!

Amazon Mechanical Turk Pricing

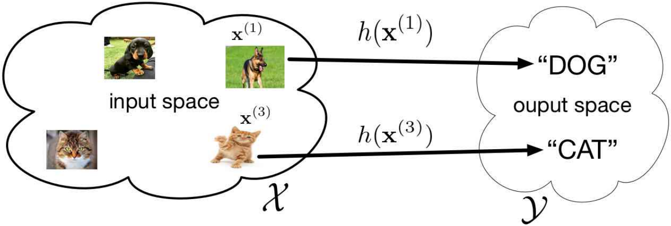
The price you (the Requester) pay for a Human Intelligence Task ("HIT") is comprised of two compone
pay Mechanical Turk. The fee you pay Mechanical Turk is based on the amount you pay Workers. Add

Worker Reward	You decide how much to pay Workers for each a
Mechanical Turk Fee	20% fee on the reward and bonus amount (if any Workers. HITs with 10 or more assignments will l additional 20% fee on the reward you pay Worke minimum fee is $0.01 per assignment or bonus p
Additional Fee for using the Masters Qualification (What are Masters?)	5% of the reward you pay Workers.
Additional Fee per assignment for using Premium Qualifications (How do I use Premium Qualifications?)	Blogger: $0.25 Born 1918 to 1960 (Age 55 or older): $0.50 Born 1961 to 1971 (Age 45-55): $0.50 Born 1972 to 1981 (Age 35-45): $0.50 Born 1982 to 1986 (Age 30-35): $0.50 Born 1987 to 1991 (Age 25-30): $0.50 Born 1992 to 1999 (Age 18-25): $0.50 Borrower - Auto Loans: $0.40

Outline

Hypothesis Maps Features to Labels

- want to predict label $y \in \mathcal{Y}$ from features $\mathbf{x} \in \mathcal{X}$ of data point
- consider hypothesis map $h(\cdot) : \mathcal{X} \to \mathcal{Y}$
- hypothesis for discrete $\mathcal{Y}$ (e.g., $\mathcal{Y} = \{0, 1\}$) called classifier
- hypothesis for continuous $\mathcal{Y}$ (e.g., $\mathcal{Y} = \mathbb{R}$) called predictor



How Good is A Predictor?

- ML is about finding good predictor $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$
- we predict label y from features $\mathbf{x}$ by $\hat{y} = h(\mathbf{x})$
- choose predictor $h(\cdot)$ such that $h(\mathbf{x}) \approx y$
- two issues here:
    - i1: set of maps $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ is typically LARGE (infinite)
    - i2: need a measure for quality of particular $h(\cdot)$

The Hypothesis Space

- GOAL of ML: find predictor $h(\cdot)$ such that $h(\mathbf{x}) \approx y$
- two issues here:
  - i1: set of maps $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ is typically LARGE
  - i2: how to measure approximation quality $h(\mathbf{x}) \approx y$
- solve i1 by restricting $h(\cdot)$ to subset $\mathcal{H}$ of maps $\mathcal{X} \rightarrow \mathcal{Y}$
- subset $\mathcal{H}$ referred to as hypothesis space

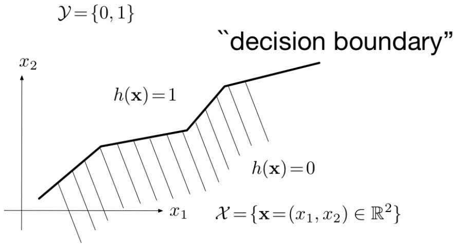
The Hypothesis Space Picture

Representing a Hypothesis/Predictor/Classifier

- ML revolves around finding a good predictor $h(\cdot) \in \mathcal{H}$
- need efficient (computer-friendly) representation of $\mathcal{H}$
- e.g., binary classification $\mathcal{Y} = \{0, 1\}$ with $|\mathcal{X}| = K$
- what would K be for 512×512 black/white bitmap?
- how many numbers specify an arbitrary map $\mathcal{X} \rightarrow \mathcal{Y}$?

Representing a Hypothesis via Decision Boundary

- binary classification with $\mathcal{Y} = \{0,1\}$ and $\mathcal{X} = \mathbb{R}^2$



- map $h(\cdot):\mathcal{X}\rightarrow\{0,1\}$ characterized by decision boundary (DB)
- hypothesis space defined by allowed shapes of DB
- e.g., $\mathcal{H}=\{$ classifiers with DB consisting of 4 line segments $\}$

A Regression Problem

- ith snapshot represented by feature vector $\mathbf{x}^{(i)} \in \mathcal{X} = \mathbb{R}^d$

- what is d for snapshots being 512×512 RGB bitmap?
- we label ith snapshot by local temperature $y^{(i)} \in \mathcal{Y} = \mathbb{R}$

Representing a Hypothesis for Regression ($\mathcal{Y} = \mathbb{R}$)

- 512×512 RGB webcam snapshot $\mathbf{x}^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^d$
- snapshot labeled with temperature $y^{(i)} \in \mathcal{Y} = \mathbb{R}$
- hypothesis space $\mathcal{H}$ of linear regression:
$$\mathcal{H} = \{h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}, \text{ with some } \mathbf{w} \in \mathbb{R}^d\}$$
- is $\mathcal{H}$ a proper subset of the set of all maps $\mathbb{R}^d \to \mathbb{R}$?
- choose $\mathbf{w}$ such that $y \approx h^{(\mathbf{w})}(\mathbf{x})$

Parametrizing the Hypothesis Space of Linear Regression

Representing a Hypothesis via Code

- $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x} \in \mathbb{R}$ predicts temperature for snapshot $\mathbf{x}$

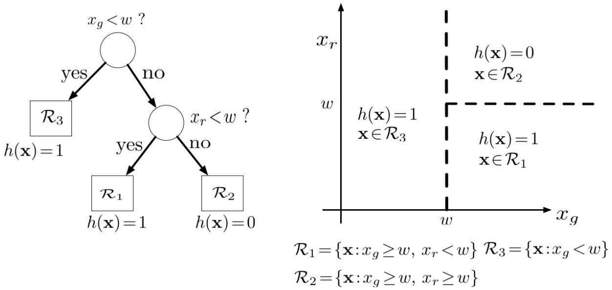
```
function temp=WhatIsTheTemperature (image,weight)

  temp = weight'*image ;
end
```

- think of hypothesis as a (Python/Matlab/...) subroutine
- hypothesis space could be, e.g.,

$$\mathcal{H} = \{ \text{ all python routines with runtime less than 10 sec. and having as input an image and a tuning parameter and output a temperature} \}$$

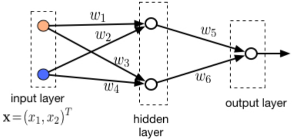
Representing a Hypothesis via DecisionTrees



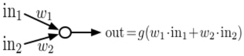
- fast evaluation of $h(\mathbf{x})$ by walking down the tree
- e.g., $\mathcal{H} = \{$ decision trees of depth less than six $\}$

Representing a Hypothesis via a “Neural Network”

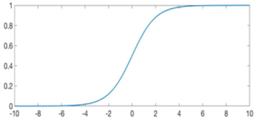
$\mathbf{w} = (w_1, w_2, \ldots, w_6)^T$



$$h(\mathbf{x};\mathbf{w}) = g\big(w_5 g(w_1 x_1 + w_2 x_2) + w_6 g(w_3 x_1 + w_4 x_2)\big)$$



“activation function” $g(\cdot)$



- network representation enables efficient computations !!!
- e.g.,
  $\mathcal{H} = \{$ NN with three hidden layers each having 10 units $\}$

Representing Hypothesis via Feature Maps

- consider original input vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$
- define feature map $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{F} \subseteq \mathbb{R}^n$ with $d \ll n$
- high-dimensional feature space $\mathcal{F}$
- construct non-linear classifiers $h^{(\mathbf{w})}(\mathbf{x}) := \mathcal{I}(\mathbf{w}^T \phi(\mathbf{x}) > \mathbf{0})$
- e.g., $d = 2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2 + x_2^2, 1)^T \in \mathbb{R}^4$ $(n = 4)$
- what is decision boundary of $h^{(\mathbf{w})}(\mathbf{x})$ for $\mathbf{w} = (0, 0, 1, -1)^T$?
- feature maps used in kernel methods (see course CS-E4830)

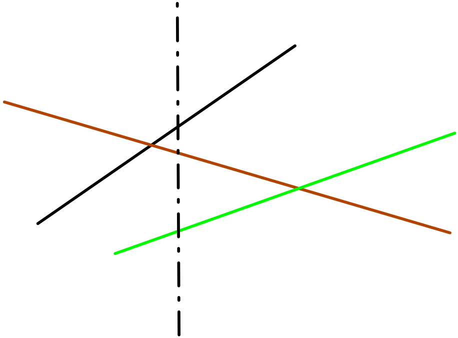
Linear Classifiers

- binary classification $\mathcal{Y} = \{0, 1\}$ with feature space $\mathcal{X} = \mathbb{R}^d$
- classifier $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ represented by decision boundary
- how many different decision boundaries are there?
- restrict $h(\cdot)$ to manageable subset $\mathcal{H}$ (hypothesis space)
- linear classifiers are particular hypothesis space

$$\mathcal{H} := \{h(\cdot) \text{ with decision boundary being hyperplane } \}$$

Linear Binary Classifiers for $\mathcal{X} = \mathbb{R}^2$

decision boundaries are straight lines



how many linear classifiers do exist for $\mathcal{X} = \mathbb{R}^2$?

Outline

The Quality of a Hypothesis

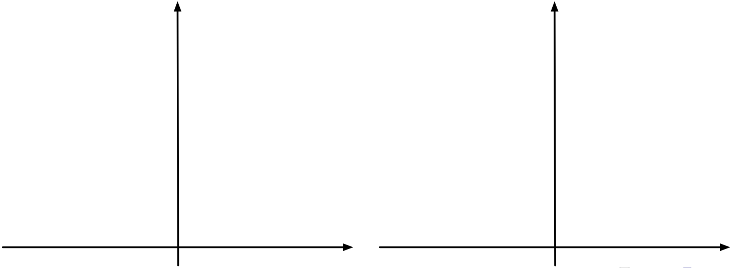
- GOAL of ML: choose hypothesis $h(\cdot)$ such that $h(\mathbf{x}) \approx y$
- two issues here:
  - i1: set of maps $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ is typically LARGE
  - i2: how to measure approximation quality $h(\mathbf{x}) \approx y$
- i2 requires measure for loss/error incurred by predictor $h(\mathbf{x})$
- define loss function $L(\mathbf{z}, h(\cdot))$ incurred by $h(\cdot)$ for data point $\mathbf{z}$
- most reasonable loss functions share structural similarities `http://web.mit.edu/lrosasco/www/publications/loss.pdf`

The Squared-Error Loss

- consider labeled data $\mathbb{X} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$
- continuous labels $y^{(i)} \in \mathbb{R}$ (regression problem)
- we predict label $y^{(i)}$ using predictor $h(\mathbf{x}^{(i)})$
- natural choice is squared error $L((\mathbf{x}, y), h(\cdot)) := (y - h(\mathbf{x}))^2$

The 0/1 Loss

- consider labeled data $\mathbb{X} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$
- binary labels $y^{(i)} \in \{-1, 1\}$ (classification problem)
- we predict label $y^{(i)}$ using predictor $h(\mathbf{x}^{(i)})$
- natural choice is 0/1-loss $L((\mathbf{x}, y), h(\cdot)) := \mathcal{I}(yh(\mathbf{x}) > 0)$



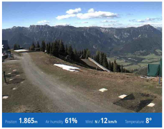
The Empirical Risk

- consider a particular loss function $L(\mathbf{z}, h(\cdot))$
- evaluate loss for data points in the dataset $\mathbb{X}$
- empirical/training loss/risk/error

$$\mathcal{E}(h(\cdot)|\mathbb{X}) := (1/N) \sum_{i=1}^{N} L((\mathbf{x}^{(i)}, y^{(i)}), h(\cdot))$$

- $\mathcal{E}(h(\cdot)|\mathbb{X})$ is mean squared error for squared error loss
- $\mathcal{E}(h(\cdot)|\mathbb{X})$ is misclassification rate for 0/1 loss

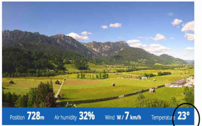
Mean Squared Error



$$\mathbf{z}^{(1)} = \left(\mathbf{x}^{(1)}, y^{(1)} = 8\right)$$
$$h(\mathbf{x}^{(1)}; \mathbf{w}) = 10$$

$$\mathbf{z}^{(2)} = \left(\mathbf{x}^{(2)}, y^{(2)} = 8\right)$$
$$h(\mathbf{x}^{(2)}; \mathbf{w}) = 7$$

$\mathbf{x}^{(3)}$



$y^{(3)}$

$$\mathbf{z}^{(3)} = \left(\mathbf{x}^{(3)}, y^{(3)} = 23\right)$$
$$h(\mathbf{x}^{(3)}; \mathbf{w}) = 20$$

$$\mathcal{E}(h|\mathbb{X}) = (1/N)\sum_{i=1}^{N}(h(\mathbf{x}^{(i)}) - y^{(i)})^2 = (1/3)(2^2 + 1^2 + 3^2) = 14/3$$

Multitask Learning

- might have to solve different tasks using same dataset
- consider, e.g., dataset of webcam snapshots
- Task 1: predict local temperature y using pixels $\mathbf{x}$
- Task 2: classify img into winter/summer using pixels $\mathbf{x}$
- individual loss function for each task: $\mathcal{E}_1(h(\cdot)|\mathbb{X})$, $\mathcal{E}_2(h(\cdot)|\mathbb{X})$
- choose $h(\cdot)$ to balance optimally between these two

Outline

Finding Optimal Hypothesis

- ML amounts to finding best predictor/classifier, i.e.,

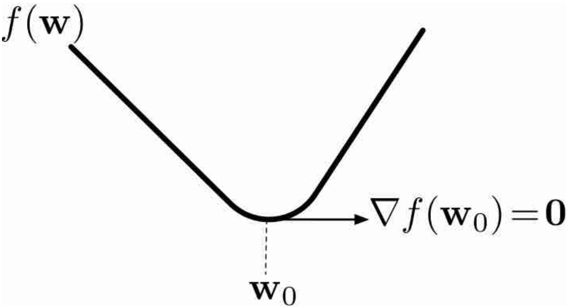
$$\hat{h} = \operatorname*{argmin}_{h(\cdot)\in\mathcal{H}} \mathcal{E}(h|\mathbb{X})$$

- solution of this empirical risk minimization yields two things:
  - "best" classifier/predictor $\hat{h}(\cdot)$ out of $\mathcal{H}$
  - minimum empirical error $\mathcal{E}(\hat{h}|\mathbb{X})$ achievable for $\mathcal{H}$
- if $\mathcal{E}(\hat{h}|\mathbb{X})$ is too high then we might enlarge $\mathcal{H}$
- $\mathcal{E}(\hat{h}|\mathbb{X})$ is indicator for accuracy of predicting y from $\mathbf{x}$ using $\hat{h}(\mathbf{x})$ for a "new" data point (with unknown label y)

Optimal Linear Regression

- regression problem $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$ with squared error loss
- predict label y from features $\mathbf{x}$ using predictors
$$\mathcal{H} = \{h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} \text{ with } \mathbf{w} \in \mathbb{R}^d\}$$
- optimal predictor $\hat{h} = \text{argmin}_{h(\cdot)\in\mathcal{H}} \mathcal{E}(h|\mathbb{X})$
- equivalent to find optimum weight vector $\mathbf{w}_0$, i.e.,
$$\mathbf{w}_0 = \underset{\mathbf{w}\in\mathbb{R}^d}{\text{argmin}}\, f(\mathbf{w}) := \mathcal{E}\left(h^{(\mathbf{w})}|\mathbb{X}\right)$$
- optimum $\mathbf{w}_0$ characterized by zero-gradient condition
$$\nabla f(\mathbf{w}_0) = \mathbf{0}$$

Zero Gradient is Necessary for Optimum



Gradient Descent

- best predictor $h^{(\mathbf{w}_0)}$ obtained for optimal weight

$$\mathbf{w}_0 = \operatorname*{argmin}_{\mathbf{w}} f(\mathbf{w}) \text{ with } f(\mathbf{w}) = \mathcal{E}(h^{(\mathbf{w})}|\mathbb{X})$$

- for convex $f(\mathbf{w})$, minimum $\mathbf{w}_0$ characterized by $\nabla f(\mathbf{w}) = \mathbf{0}$
- this is equivalent to fixed-point equation $\mathcal{T}\mathbf{w} = \mathbf{w}$
- here, $\mathcal{T}\mathbf{w} = \mathbf{w} - \alpha \nabla f(\mathbf{w})$ with step-size (learning rate) $\alpha > 0$
- find fixed-point $\mathbf{w}_0$ by fixed-point iterations

$$\mathbf{w}^{(k+1)} = \mathcal{T}\mathbf{w}^{(k)} = \mathbf{w}^{(k)} - \alpha \nabla f(\mathbf{w}^{(k)})$$

- known as gradient descent (GD)

Gradient Descent Picture

Outline

To Sum Up

- break raw data into atomic "data points"
- representing data points using features and labels
- various representations of predictor/classifier (decision boundaries, decision trees, neural networks, ...)
- concept of loss functions and empirical risk
- choosing optimal predictor by empirical risk minimization

The Nitty-Gritty Details

- how to efficiently solve ERM ?
- how to validate a predictor/classifier ?
- how to choose hypothesis space $\mathcal{H}$?
- what to do if we do not have any labels?

Design Choices

- loss function and hypothesis space are design choices
- squared error loss + linear hyp. space results in convex opt. problem
- choices guided by computational infrastructure
- if all you have is a spreadsheet app, then deep neural nets might not be the right choice for hypothesis space