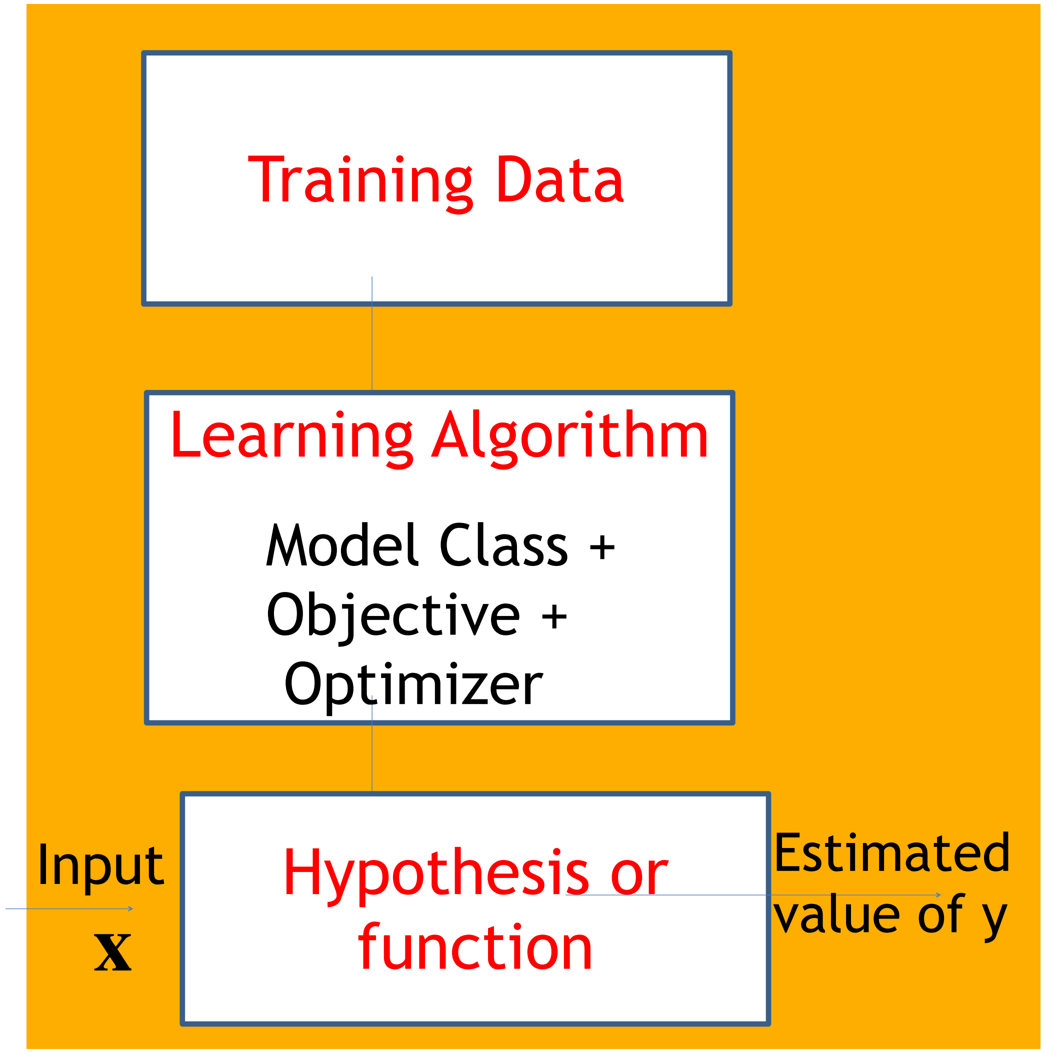
Notation and Math



Example: predicting the mpg of a car based on the *horsepower* of the car ($d = 1$):

$$(\mathbf{x}^{(1)} = [307],\ y^{(1)} = 18)$$
$$(\mathbf{x}^{(2)} = [350],\ y^{(2)} = 15)$$
$$(\mathbf{x}^{(3)} = [318],\ y^{(3)} = 18)$$
$$(\mathbf{x}^{(4)} = [304],\ y^{(4)} = 17)$$

$$X = \begin{bmatrix} 307 \\ 350 \\ 318 \\ 304 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 18 \\ 15 \\ 18 \\ 17 \end{bmatrix}$$

If our model class (hypothesis class) is a linear function

$f(\mathbf{x}) = w_0 + w_1 x_1$

we need to find the "best" w_0, w_1

e.g. if $w_0 = 39.94, w_1 = -0.16$

then $\hat{y} = h(\mathbf{x}) = 39.94 - 0.16x_1$

Notation: We will use the notation (mostly…) from Stanford and the Deep Learning Book

- **Input (features):** $\mathbf{x} \in \mathbb{R}^d$ ($\mathbf{x}^{(i)}$ for the ith example)

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix} \text{ ith example, } X = \begin{bmatrix} \mathbf{x}^{(1)T} \\ \mathbf{x}^{(2)T} \\ \vdots \\ \mathbf{x}^{(N)T} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots x_d^{(2)} \\ \vdots & \vdots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots x_d^{(N)} \end{bmatrix} \text{ design matrix}$$

Aka data matrix

- **Output (target/label):** $y \in \mathbb{R}$
  ($y^{(i)}$ for the ith example)

- **Training data:** $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})$

- The number of training examples: N

Example predicting the mpg of a car based on the horsepower of the car ($d = 1$):

$(\mathbf{x}^{(1)} = [307],\ y^{(1)} = 18)$

$(\mathbf{x}^{(2)} = [350],\ y^{(2)} = 15)$

$(\mathbf{x}^{(3)} = [318],\ y^{(3)} = 18)$

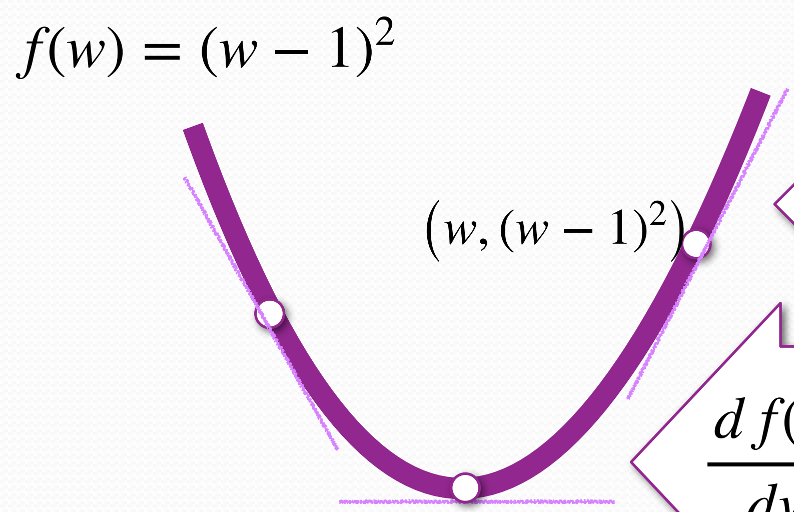
$(\mathbf{x}^{(4)} = [304],\ y^{(4)} = 17)$

$$X = \begin{bmatrix} 307 \\ 350 \\ 318 \\ 304 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 18 \\ 15 \\ 18 \\ 17 \end{bmatrix}$$

If $w_0 = 39.94, w_1 = -0.16$

$\hat{y} = h(\mathbf{x}) = 39.94 - 0.16x_1$

Calculus Review



$\dfrac{d\,f(w)}{dw}$ = $2(w-1)$ is the rate of change of f at the point w

$\dfrac{d\,f(1)}{dw}$ = $2(1-1) = 0$, the function is at a minimal value

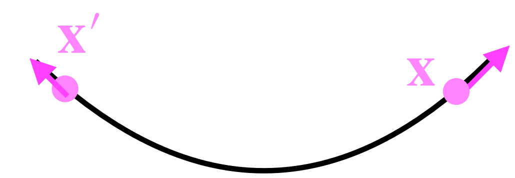
Global optimization: Find the minimum value of the function.

$\dfrac{d\,f(w)}{dw}$ = $2(w-1)$ = 0

Gradient - generalization of derivative

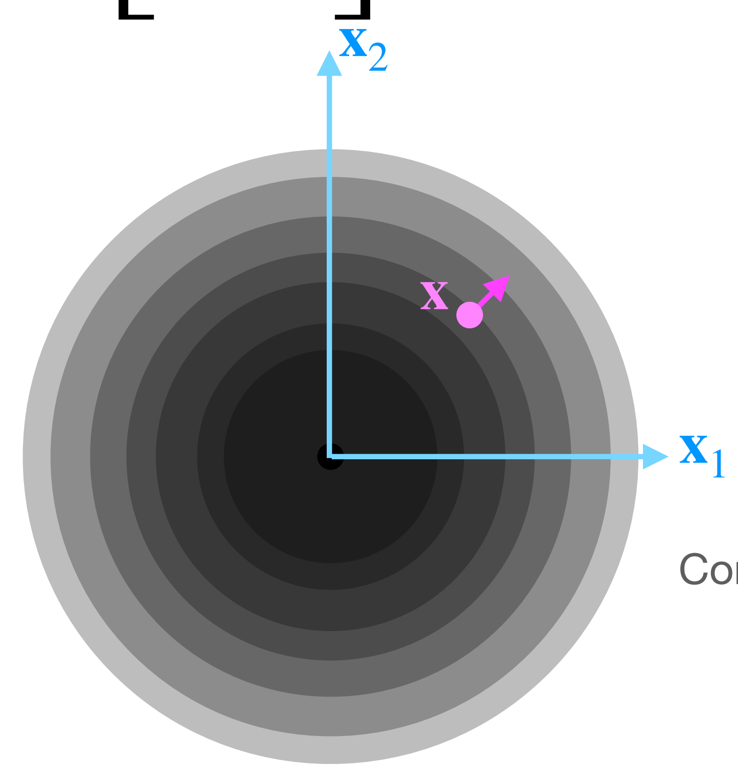
Derivative and Gradient

- $h(\mathbf{x})$ a differentiable function ($h : \mathbb{R}^d \to \mathbb{R}$)



- If $d = 1$, the derivative $\dfrac{dh(\mathbf{x})}{d\mathbf{x}}$ gives the direction of the fastest increase

- For $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$, the gradient $\nabla h(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial h(\mathbf{x})}{\partial x_1} \\ \vdots \\ \dfrac{\partial h(\mathbf{x})}{\partial x_d} \end{bmatrix}$ gives direction fastest increase



Contour plot of $h(\mathbf{x}) = x_1^2 + x_2^2$

Direction of steepest increase $\nabla h(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$

Warning!
$h'(\mathbf{x})$ will most often mean derivative
$\mathbf{x}'$ will often mean a variable

Gradient Examples

Slide only shown in 2:00 pm lecture

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$f(\mathbf{w}) = (w_0 + 4w_1 - 3)^2$$

$$\frac{\partial f(\mathbf{w})}{dw_0} = 2 \cdot (w_0 + 4w_1 - 3), \quad \frac{\partial f(\mathbf{w})}{dw_1} = 2 \cdot (w_0 + 4w_1 - 3) \cdot 4$$

$$\nabla f(\mathbf{w}) = \begin{bmatrix} 2 \cdot (w_0 + 4w_1 - 3) \\ 2 \cdot (w_0 + 4w_1 - 3) \cdot 4 \end{bmatrix}$$

$$f(\mathbf{w}) = (w_0 + w_1 - 3)^2 + (w_0 + 4w_1 - 4)^2$$

$$\frac{df(\mathbf{w})}{dw_0} = 2 \cdot (w_0 + w_1 - 3) + 2 \cdot (w_0 + 4w_1 - 4)$$

$$\frac{df(\mathbf{w})}{dw_1} = 2 \cdot (w_0 + w_1 - 3) + 2 \cdot (w_0 + 4w_1 - 4) \cdot 4$$

$$\nabla f(\mathbf{w}) = \begin{bmatrix} 2 \cdot (w_0 + w_1 - 3) + 2 \cdot (w_0 + 4w_1 - 4) \\ 2 \cdot (w_0 + w_1 - 3) + 2 \cdot (w_0 + 4w_1 - 4) \cdot 4 \end{bmatrix}$$

Slides not covered in class

Notation

Average *Training* Error is called “in sample error”

$$E_{\text{in}}(w_0, w_1) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}}_{\substack{\text{Average error on}\\ \text{the } N \text{ training}\\ \text{examples}}} \overbrace{\text{error}}^{\substack{\text{Cost (loss) for}\\ \text{prediction not}\\ \text{being the same}\\ \text{as true label}}}(y^{(i)}, \underbrace{g(\mathbf{x}^{(i)})}_{\substack{\text{Prediction on}\\ \text{input } \mathbf{x}^{(i)}}})$$

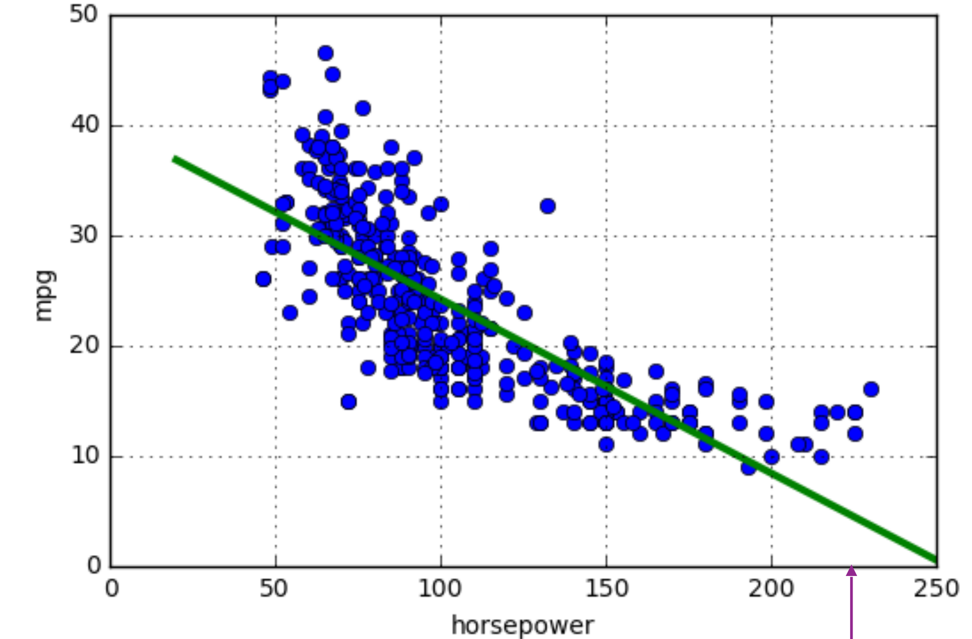
- If our objective function (cost function) is RSS, then

$$E_{\text{in}}(w_0, w_1) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \underbrace{(w_0 + w_1\mathbf{x}^{(i)})}_{\substack{\text{Prediction on}\\ \text{input } \mathbf{x}^{(i)}}}\right)^2$$

- If instead, we had chosen our objective function to be the absolute error (another very reasonable choice) then

$$E_{\text{in}}(w_0, w_1) = \; = \frac{1}{N}\sum_{i=1}^{N}\left|y^{(i)} - \underbrace{(w_0 + w_1\mathbf{x}^{(i)})}_{\substack{\text{Prediction on}\\ \text{input } \mathbf{x}^{(i)}}}\right|$$

Recap



mpg = $\mathbf{w}_0$ + $\mathbf{w}_1$ horsepow

- **Model** relationship between horsepower and mpg as a line

$$\hat{y} = h(\mathbf{w}) = w_0 + w_1 \mathbf{x}$$

- We chose to minimize:

$$\text{RSS}(w_0, w_1) = \sum_{i=1}^{N} \left(y^{(i)} - (w_0 + w_1 \mathbf{x}^{(i)})\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Residual Sum of Squares (RSS)

Also called the sum of squared residuals (SSR) and sum of squared errors (SSE)