Machine Learning 101

Supervised Learning
"right answers" given
task: estimate more "right answers"

Unsupervised Learning
no "right answer" → clusters
can find structure in data

[Regression Problem: continuous value answer
[Classification Problem: descrete valued output

Linear Regression
Notation
- Training set ~> dataset for given problem
- m ~> number of examples in training set
- x's ~> "input" variables
- y's ~> "output"/"target" variables
- $(x^{(i)}, y^{(i)})$ → i-th training example
- hypothesis ~> function given by learning algorithm
  ↳ maps from x's to y's

Cost Function
given: $h_\theta(x) = \theta_0 + \theta_1 x$ → input
↳ hypothesis; θ_0, θ_1 → parameters

↳ $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$ } minimize this:
get θ values that "cost less"
∴ fit data best

Gradient Descent
* Can reach different local min depending on starting parameters
  ↳ works best in cases in which there is only one min point
  (elliptic paraboloid)

Algorithm:
while !convergence {
$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ → univariate case
(α → learning rate)
} * Simultaneous ~~update~~ update: in literal algorithm, next state must be calculated and stored into temp values all at once, then updated all at once as well

→ Learning Rate (α)

[too small: gradient descent is too slow

[too big: "big steps", may lead to overshooting and failure to converge

→ Gradient Descent for Linear Regression

model $\begin{cases} h_\theta(x) = \theta_0 + \theta_1 x \\ J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \end{cases}$ → apply GD to get lowest cost parameters

$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$

→ $j=0$ → $\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$

→ $j=1$ → $\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)} - y^{(i)}) x^{(i)} \end{cases}$$

* "Batch" Gradient Descent: each iteration goes through the entire training set