Iterative Gradient based ~~training~~ Descent Pseudo code:

1) - Standardize data X of size $N \times D$
   - Target value matrix Y of size $N \times 1$
2) add bias cd^n x so that $x = [1\ x]$ & has size $N \times [D+1]$
3) initialize params θ_j for $j = 1 \ldots D+1$ to some random ~~value~~
   θ = col^n vctr of size $(D+1) \times 1$
4) Until Convergence
   $\forall x \in X$
   - Compute gradient $\frac{dJ}{d\theta} = 2x^T(x\theta - y)$
   - update $\theta \rightarrow \theta = \theta - \eta \frac{dJ}{d\theta}$

Termination Criteria
We terminate when
- max iterations reached
- params change very little
- Δ training error = very little

all this was iterative gradient descent → considers 1 observation at a time
problem → gives local sol^n & takes long time

sol^n ? update params using avrg of gradients created by all observations
batch gradient descent

$$\frac{d}{d\theta} \frac{1}{N} \sum_{n \in N} x^T(x\theta - y)$$

$$\theta = \theta - \frac{\eta}{N} X^T(x\theta - Y)$$

let η absorb other scalars, 2

↑ now we need all tho!!

sol^n ? mini batch.

Overfitting ??
- add regularization term
- penalty proportional to magnitude of θ $|\theta| = \theta^T\theta$
- stop using validation set: compute validation error on each step

λ = blending term

$$\frac{1}{2}(Y - X\theta)^2 + \lambda\theta^T\theta$$