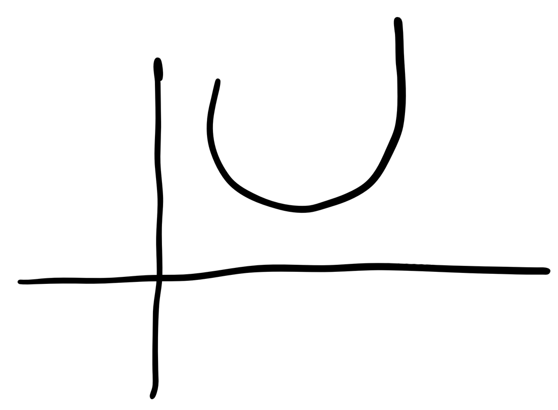
→ Gradient Descent

- Have some function $J(w,b)$
- Want $\min_{w,b} J(w,b)$



- Outline :
  - Start with some w, b
  - Keep changing w, b to reduce $J(w,b)$ until we settle at or near a minimum (may have > 1 min)

→ Implement Gradient Descent Algorithm

$$w = w - \alpha \frac{d}{dw} J(w,b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

Update steps

↳ Simultaneously update w and b

↳ Repeat until converge

$\alpha \equiv$ Learning Rate

↳ Controls the step size

- Learning Rate
  - → If α is to small ⇒ Gradient descent may be slow
  - → If α is to large ⇒ Gradient descent may:
    - overshoot, never reach minimum
    - fail to converge

- Near a local minimum
  - → Derivatives become smaller
  - → Update steps become smaller

  Can reach minimum without decreasing learning rate α

→ Gradient descent for linear regression

- Linear Regression Model

$$f_{w,b}(x) = wx + b$$

- Cost Function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left[f_{w,b}(x^{(i)}) - y^{(i)} \right]^2$$

- Gradient Descent Algorithm

$$w = w - \alpha \frac{\partial}{\partial w} J(w,b) \iff w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} \left[f_{w,b}(x^{(i)}) - y^{(i)} \right]^2$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b) \iff b = b - \alpha \frac{1}{m} \sum^{m} \left[f_{w,b}(x^{(i)}) - y^{(i)} \right]$$