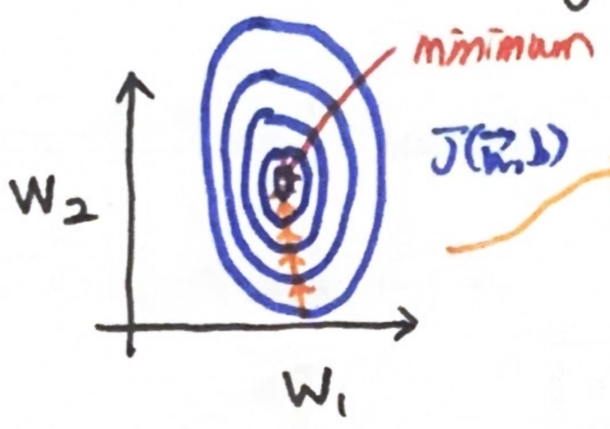
〈Advanced Optimization〉

* gradient descent

$$w_j := w_j - \underbrace{\alpha}_{\text{learning rate}} \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

⇒ When we use gradient descent as optimizer for learning algorithm,
we initialize learning rate and it never be changed until convergence (w, b no longer change)



with small value of learning rate
it takes too long time to get to the minimum

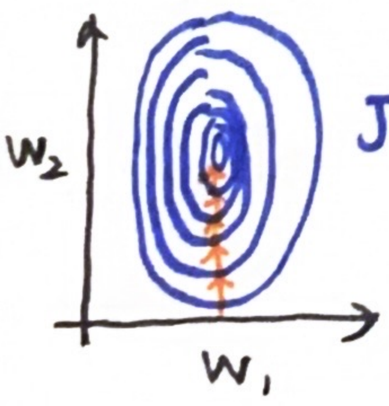
⇓ "want to get to the minimum faster"

* Adam algorithm

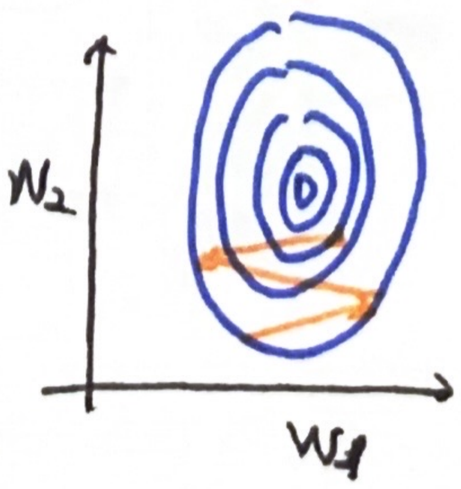
- if learning rate is too small and we are just taking tiny little steps in a similar direction
  ⇒ Let's make learning rate α bigger!
- adjust learning rate automatically (not just one α)
  ⇒ uses different learning rate for every single parameter update

ex)
$$w_1 := w_1 - \alpha_1 \frac{\partial}{\partial w_1} J(\vec{w}, b)$$
$$\vdots$$
$$w_{10} := w_{10} - \alpha_{10} \frac{\partial}{\partial w_{10}} J(\vec{w}, b)$$
$$b := b - \alpha_{11} \frac{\partial}{\partial b} J(\vec{w}, b)$$

* Intuition



(initialize tiny value of learning rate)
if w_j or b keeps moving in same direction
⇒ lets increase learning rate α_j
(lets go faster in that direction)



(initialize huge value of α)
if w_j or b keeps oscillating,
⇒ lets reduce α_j