# Nov 1

we can assign weights to each value to get a linear combination
if it equals b when it is a diagonal pattern, else not
so find values of weights and b such that ↑
any pattern other than diag will have a lower value.
NOTE: equivalently can decide to move b to left of eq so diag pattern gives value of 0.
$\sigma(w_1 a_{00} + w_2 a_{01} + w_3 a_{10} + w_4 a_{11} + b) > 0.5$ when diag
↳ treats like a probability.
if $\sigma = \frac{1}{1+e^{-x}}$ this is essentially LogReg.

How to learn w and b

$\max \prod_{i=1}^{n} P(y_i = 1 \mid x_i)$

$\min -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\sigma(-w^T x_i + b)) + (1-y_i) \log(1 - \sigma(-w^T x_i + b))]$

$= \min \text{Cost}(w, b)$

## GRADIENT DESCENT

goal: estimate w (and b's)

consider a random weight $w_0$. What happens to $\text{Cost}(w_0)$ as you nudge $w_0$ slightly.
until we reach a point where


best nudge is in the direction of largest rate of change ($\nabla f(x) = f'(x)$)
w/ a func of mult variables $\nabla f(x,y,z) = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}$

e.x. $f(x) = 3x^2 - 2y$ $\quad f'(x) = 6x\hat{i} - 2\hat{j}$

$\nabla f$ @ $p = (0,0)$

$\nabla f_p = 6(0)\hat{i} - 2\hat{j} = -2\hat{j}$

$p_{new} = 1 \cdot \nabla f_p + p = (0, -2)$

1. Define step size $\alpha$ (tuning param)
2. initialize p to be random
3. $p_{new} = \alpha \nabla f_p + p$
4. $p = p_{new}$
5. Repeat 3/4 until $p \approx p_{new}$

To find local minima use $-\nabla f_p$

Compute $\nabla \text{Cost}(w, b)$

$\nabla \text{Cost}(w, b) = \left[\frac{\partial}{\partial w}\text{Cost}, \frac{\partial}{\partial b}\text{Cost}\right]$

$\frac{\partial}{\partial w}\text{Cost} = \frac{1}{n}\sum_{i=1}^{n} x_i (y_i - \sigma(-w^T x_i + b))$

## LIMITS

- expensive to run
- result we get depends on initial starting point

## STOCHASTIC GRADIENT DESCENT

goal: approximate the gradient of the cost using a sample of data (batch)

note: the magnitude of $\nabla f_p$ depends on p

# Nov 3rd

## Neural Networks

"learning features automatically"

- input layer
- hidden layer
- final layer

$x_1 + x_2 \rightarrow h_1$

$x_1 + x_2 \rightarrow h_2$

$h_1 + h_2 \rightarrow y$

- Forward Propagation

## Activation Functions