logistic Regression Cost function:

→ new stuff from ml course

→ to train the parameters W & b of logistic regression model, you need to define a cost function.

$z^{(i)} = w^T x^{(i)} + b$

$$\hat{y} = \sigma(w^T x + b) \quad \text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$

given $\underline{\{(x^{(1)}, y^{(1)}) \dots (x^{m}, y^{(m)})\}}$ we want $\hat{y}^{(i)} \approx y^{(i)}$

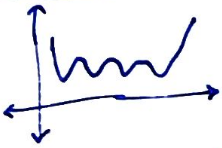
Loss (error function):

1) You could do $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ → half square error

→ this basically measures how well our algo is doing

→ If you do this however, the optimization problem (later) becomes non-convex → end up with optimization problem with lots of local optima

↓ gradient descent might not find local optima



2 another way $$\boxed{L(\hat{y}, y) = -\left(\{y \log \hat{y}\} + \{(1-y) \log(1-\hat{y})\}\right)}$$

rationale: if $y=1$ $L(\hat{y}, y) = -\log \hat{y}$ (← want as small as possible)

$\hat{y}$ = large → sigmoid can't be > 1

if $y=0$ $L(\hat{y}, y) = -\log(1-y)$ want $\log 1-\hat{y}$ large want $\hat{y}$ small

Cost function

→ loss f^n was defined wrt single training example

→ Cost f^n determines how well you're doing on entire training set

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)})\right]$$

basically gives average of $L(\hat{y}, y)$

→ In training log. reg. model, we're going to find params w & b that minimize overall cost f^n J.