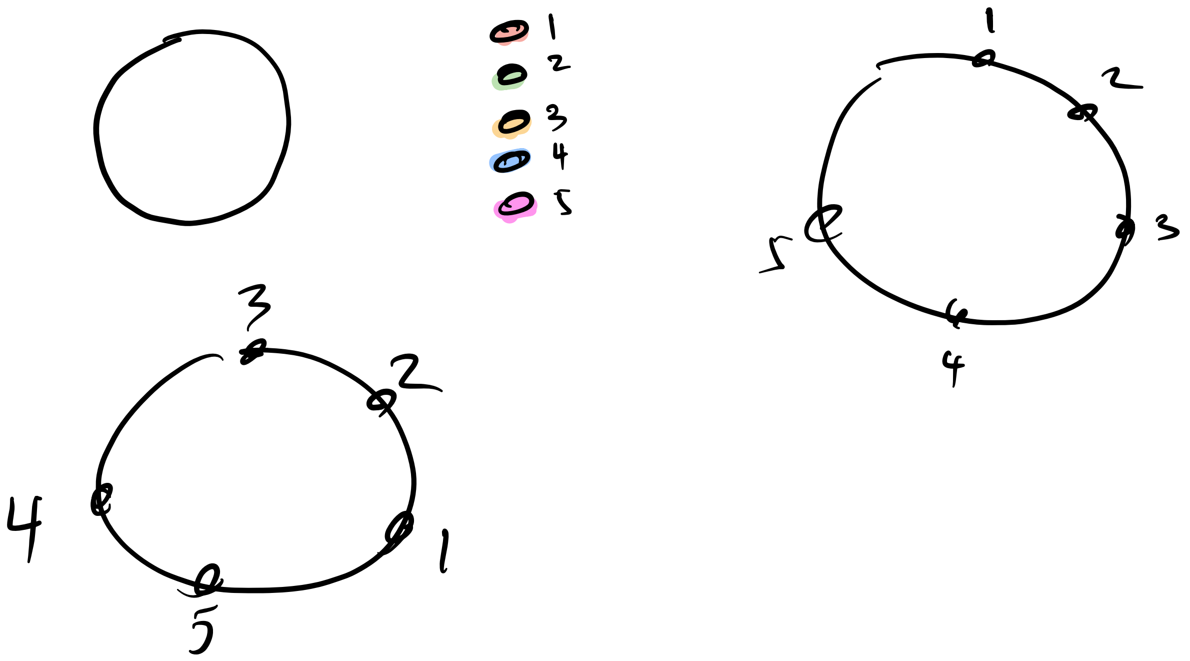


Overview:

- Large language models (LLM) built on problem of "next word prediction"
- Some perspective on this from brute force next letter prediction → language like output
- next word prediction w/ large texts unreasonably hard for brute force frequency analysis.

Problem: how to reasonably do next word prediction?

Answer = neural nets (lots of them)

Neural nets give a solution to the following ill posed problem

given inputs, we believe there should exist
$x_0, x_1, \ldots$ a function f

taking these to certain desired outputs
$y_0, y_1, \ldots$

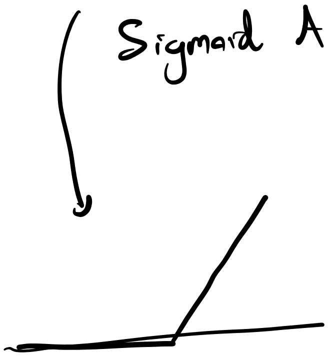
$f(x_0) = y_0$ $f(x_1) = y_1$ ... is one reasonable way.

and we believe if we can plug & check values for f,
we'll know if its right.

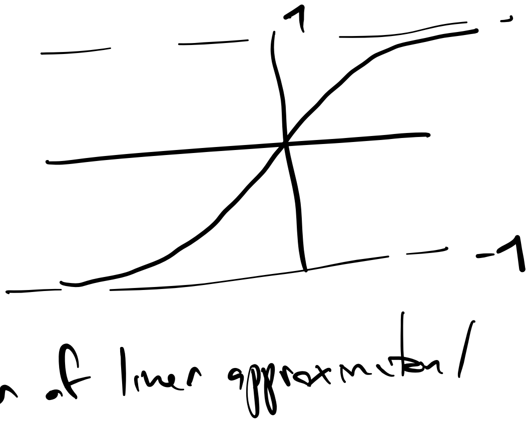
The design/architecture of nnet is highly variable,
depending on problem at hand.

ReLU Activation/similar for visual data (handwriting)

Sigmoid Activation for boolean/logic/classification



ReLu

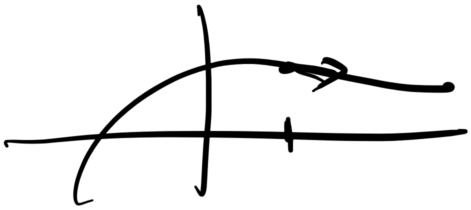


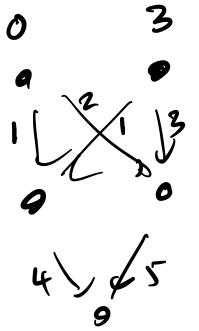
Neural nets have the flavor of linear approximation/
regression, but actually very nonlinear.

Gradient descent:

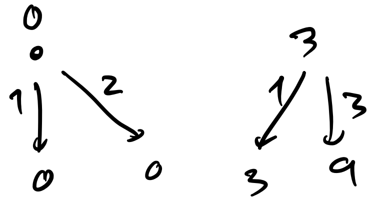
method of approximating solns to eqns

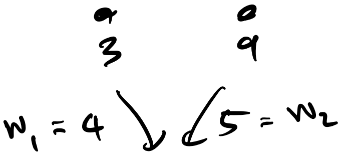
$f(x) = 0$





7 ?





$12 + 45 = 67$

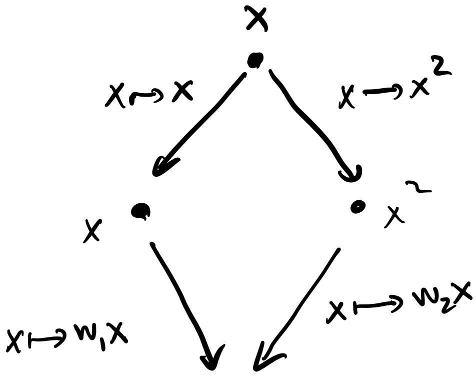
"

$3w_1 + 5w_2$

$w_1 \rightarrow w_1 - 3/10$

$w_2 \rightarrow w_2 - 5/10$

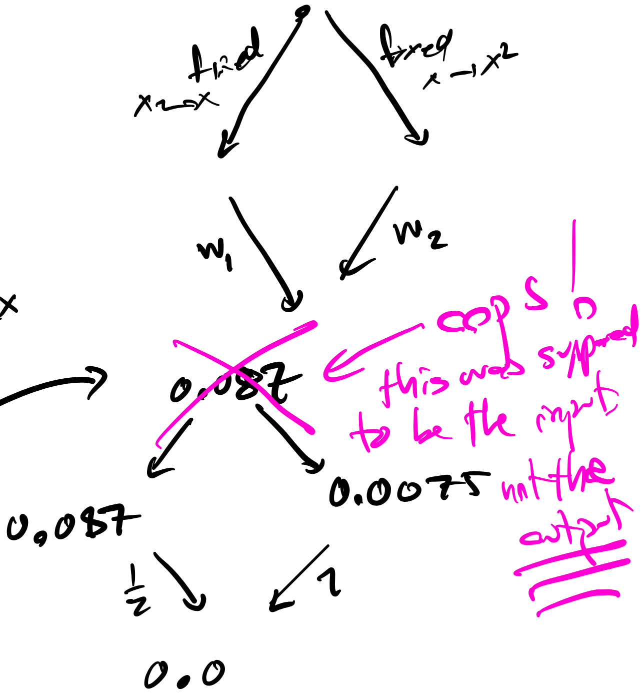
$f(x) = \sin x$ $\sin 30 = \frac{1}{2}$



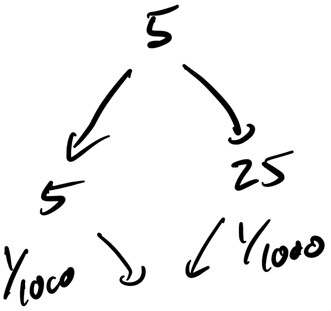
$\sin(5) = .087$

$\sin(40) = 0.64$

$\sin(20) = 0.34$



oops! this was supposed to be the input not the output



$0.005 + 0.025 = 0.03$