Lecture 6/4

- [x] Recap NN / BackProp - Brilliant
- [x] Activation(s)
- [ ] AutoEncoder
- [ ] word vectors
- [ ] Demo/playground
- [x] Dropout
- [ ] HW 2B - implement 3ex NNets
  - compare yours-vs-pytorch

demo next Wed

Course Project

- Proposal by July 4th
- Similar to HW load/hours
- 3 options (standard)
  - replicate research paper results
  - use class method on new dataset
  - use new method on class-dataset
- non standard: something you are already working on.

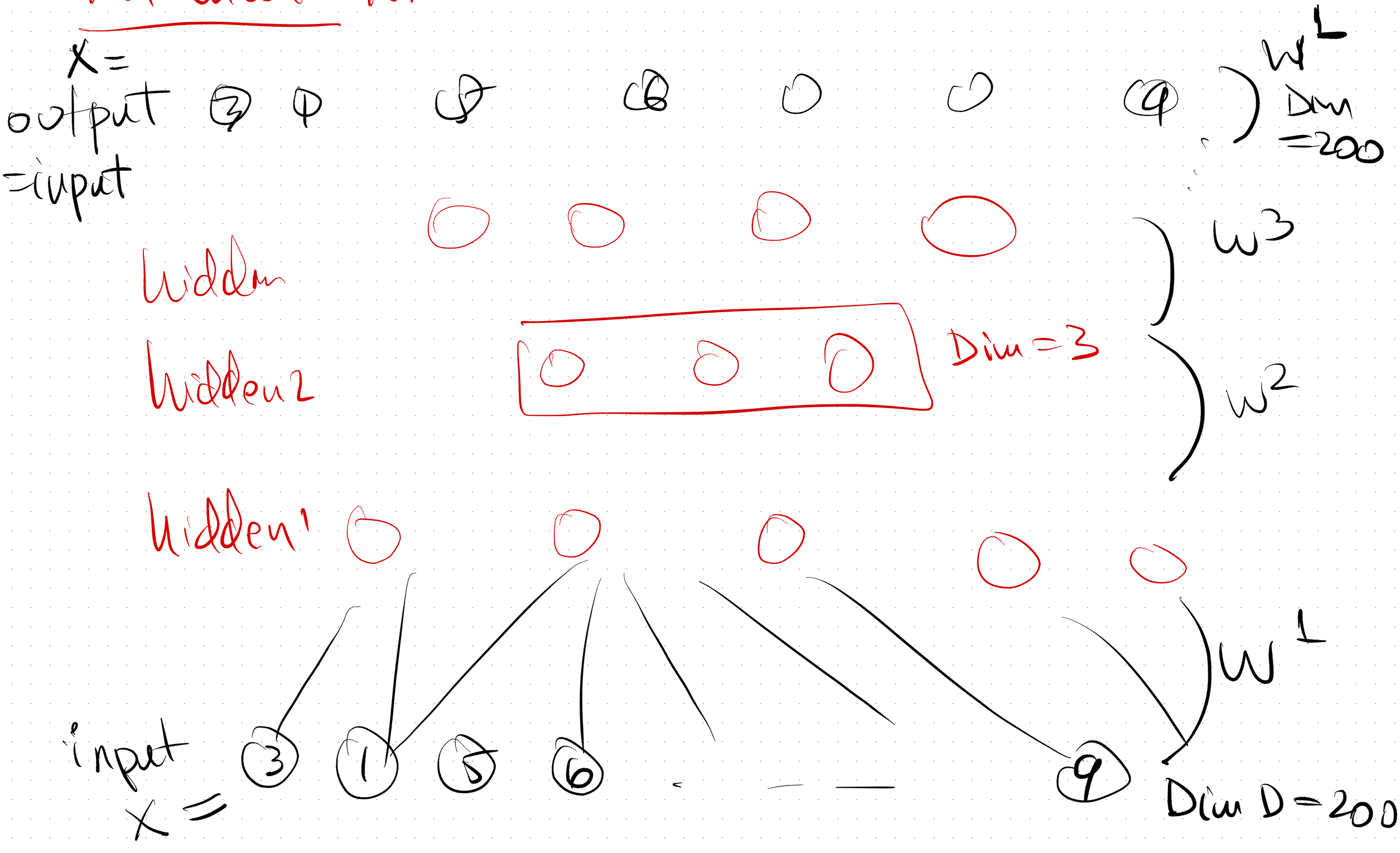
Dropout: Randomly (small prob) choose some neurons
to not update / consider for few iterations (iter = 1, 2)
⇒ ignore that neuron (usually for update)
⇒ regularization effect or not letting NN
concentrate weights on few neurons.
• use heuristic for sampling "idle" neurons.

Activation		
Relu	$\max(0,x) = \begin{cases} x & \text{if } x>0 \\ 0 & \text{if } x\le 0 \end{cases}$	not diff in 0 ?
Leaky Relu	$\begin{cases} x>0 \\ \alpha x \text{ if } x\le 0 \end{cases}$	reduce RELU risk for ignord neurons
Sigmoid	$\frac{1}{1+e^{-x}}$	Bin dassit
Tanh	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$	RNN
ELU	$\begin{cases} x \text{ if } x>0 \\ \alpha(e^x-1) \text{ if } x\le 0 \end{cases}$	
swish	. . .	slow, but diff
Softplus	. . .	

AutoEncoder NN

X=

output = input



Task find W^k $(k=1:L)$ matrices s.t.

Forward Comp

$$X \xrightarrow{W^1} L_1 \xrightarrow{W^2} L_2 \rightarrow \cdots \rightarrow \underbrace{L_L = X}_{\text{last}}$$

produces output $\simeq$ X input

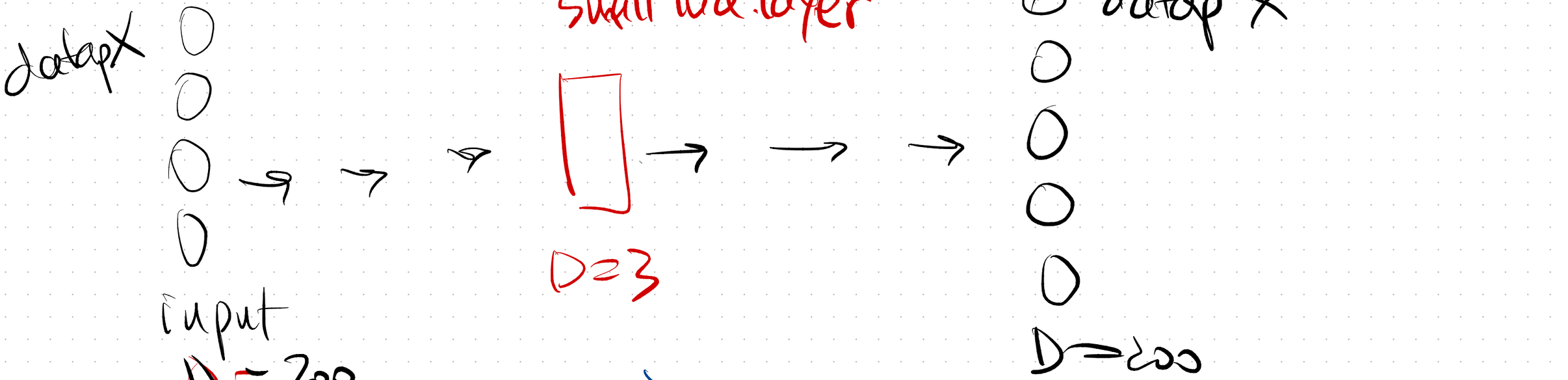
Success!

output $\simeq$ input

ALL datapoints

measure $\rightarrow$ Loss (obj)

restrict: hidden layers (at least one) is much smaller



compress = represent datax on 3-dim

decompress = recover orig datax from 3-dim repres

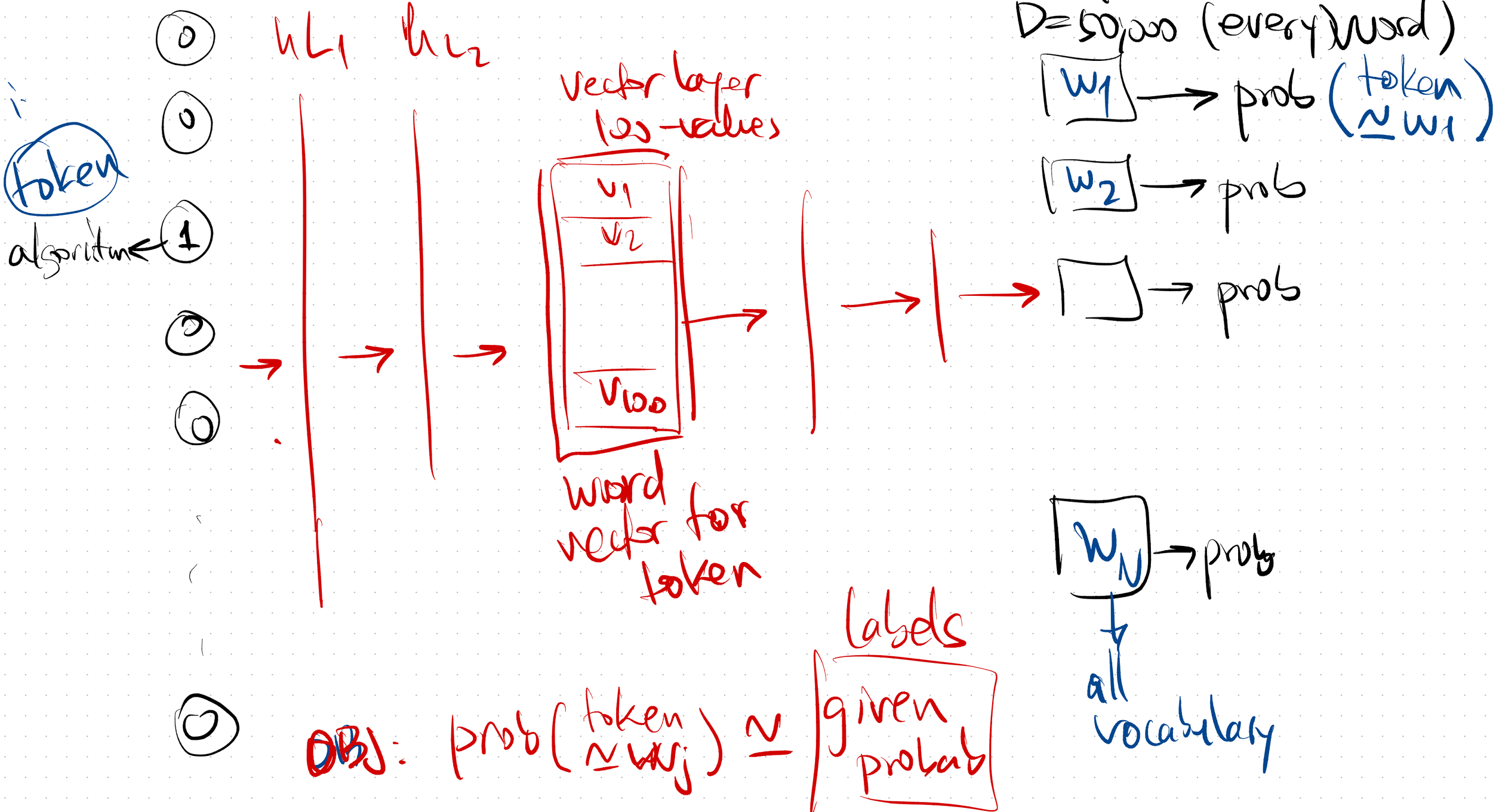
Word vectors word "Algorithm" → vector [1.2 | 13 | -5.71 | ...]

300-values

input 1-hot : D=w D = #words 50,000?

output 10,000-values

D=50,000 (every word)

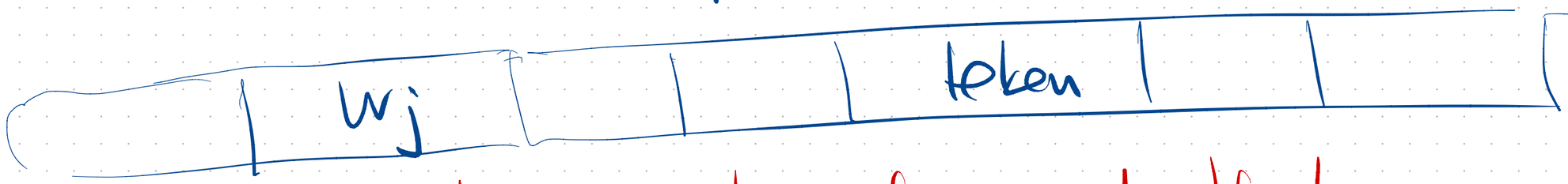


- label(token, w_j) = prob/likelihood of synonyms (same meaning)

  ⇒ word vector similar for words similar meaning

- label (token, w_j) = co-occurrence in text is span_window of 5 words

  prob



⇒ word vectors similar for words that co-occur