In this math $\vec{\beta} \in \mathbb{R}^d$

$$c, \vec{\beta}_{lasso} = \arg\min_{\vec{\beta}, c} \sum_{i=1}^{n} \left(\vec{x}_i^T \vec{\beta} + c - y_i \right)^2 + \lambda \sum_{i=1}^{d} |\beta_i|$$

$$c = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \underbrace{\sum_{j=1}^{d} x_{ij} \beta_j}_{\text{no intercept}} \right)$$

Math:
- 1-based
- 0-based code

$$b_i = 2 \sum_{j=1}^{n} x_{ji} \left(y_j - c - \sum_{\substack{k=1 \\ k \neq i}}^{d} x_{jk} \beta_k \right)$$

residual w/o using i^{th} predictor

$$\begin{bmatrix} c \\ \vec{\beta} \end{bmatrix}$$

Today: neural networks

Tomorrow:
in-class HW help

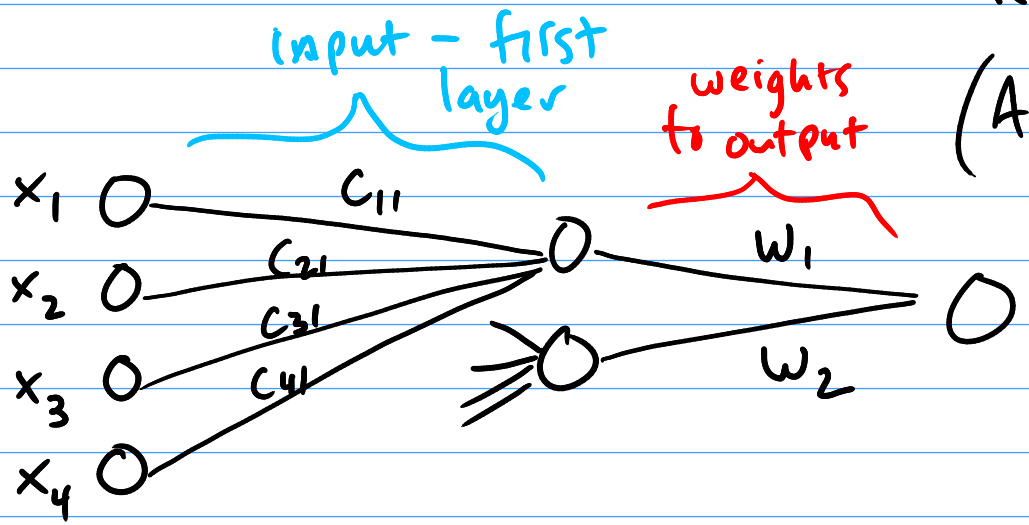
Goals: functional form (2-layer)
where neuro connection comes from
RBF networks

RBF networks

$$f(\vec{x}) = \sum_{i=1}^{m} w_i \, g(\|\vec{x} - \vec{c}_i\|)$$

i^{th} center

RBF function



$\vec{x} \in \mathbb{R}^4$

$m = 2$ centers RBFs

linear

(Artificial) Neural networks (NNs) relax the RBF constraint

$$f(\vec{x}) = \sum_{i=1}^{m} w_i \, g(\vec{x}, \vec{c}_i)$$

weights

arbitrary nonlinear function of $\vec{x}$ and parameters $\vec{c}_i$

weights

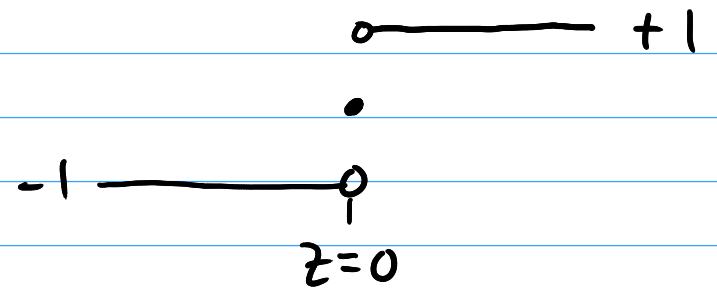
Often $g(\vec{x}, \vec{c}_i) = g(\underbrace{\vec{c}_i^T \vec{x}}_{\text{dot product}}) = g(z)$ (z: scalar)

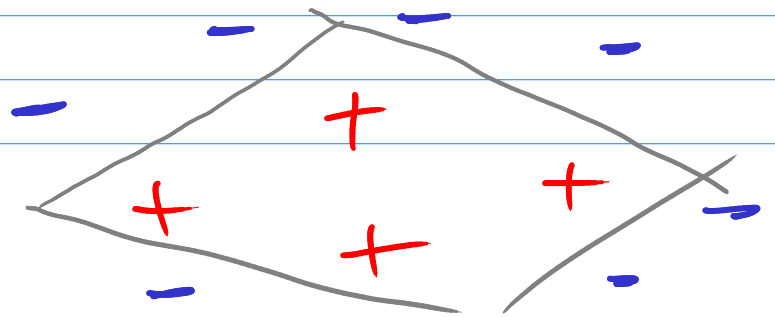
Usually g = thresholded nonlinearity

$g(z) = \text{sgn}(z)$

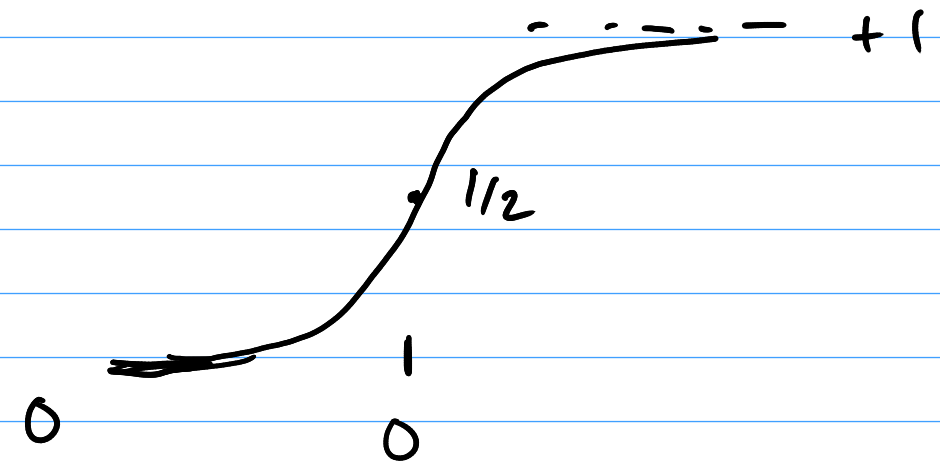
$= \text{sgn}(\vec{c}^T \vec{x})$

predictions of linear classifier





More realistic: Saturating nonlinearity

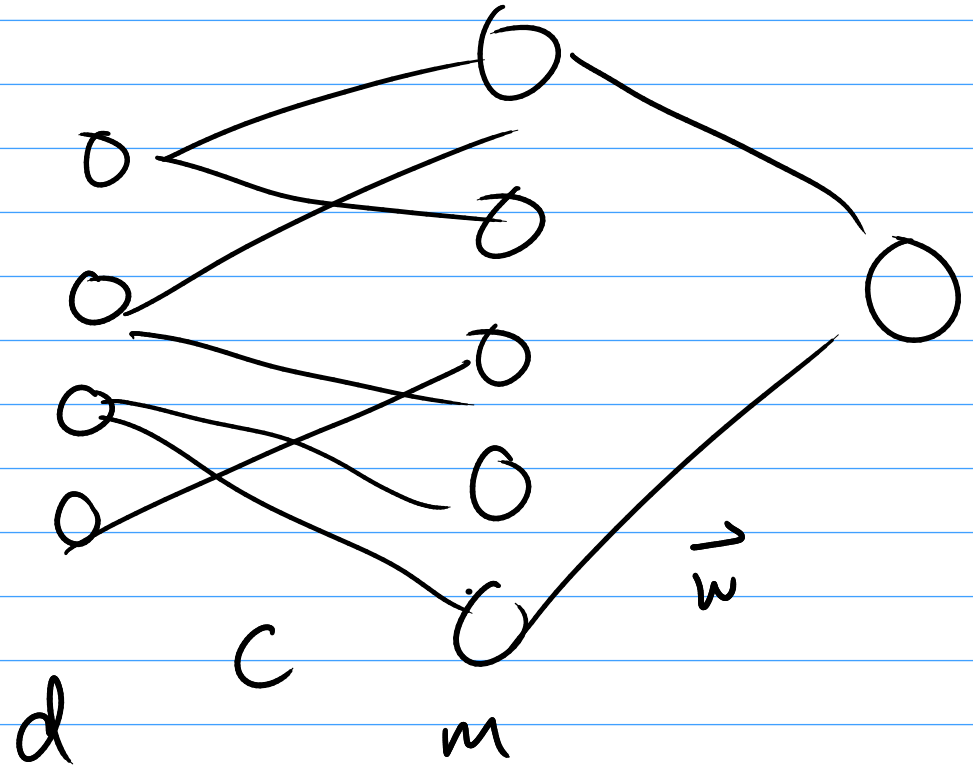


logistic regression

modeled $Pr[Y=+1]$

$$g(z) = \frac{1}{1+e^{-z}}$$

Design m units, nonlinearity

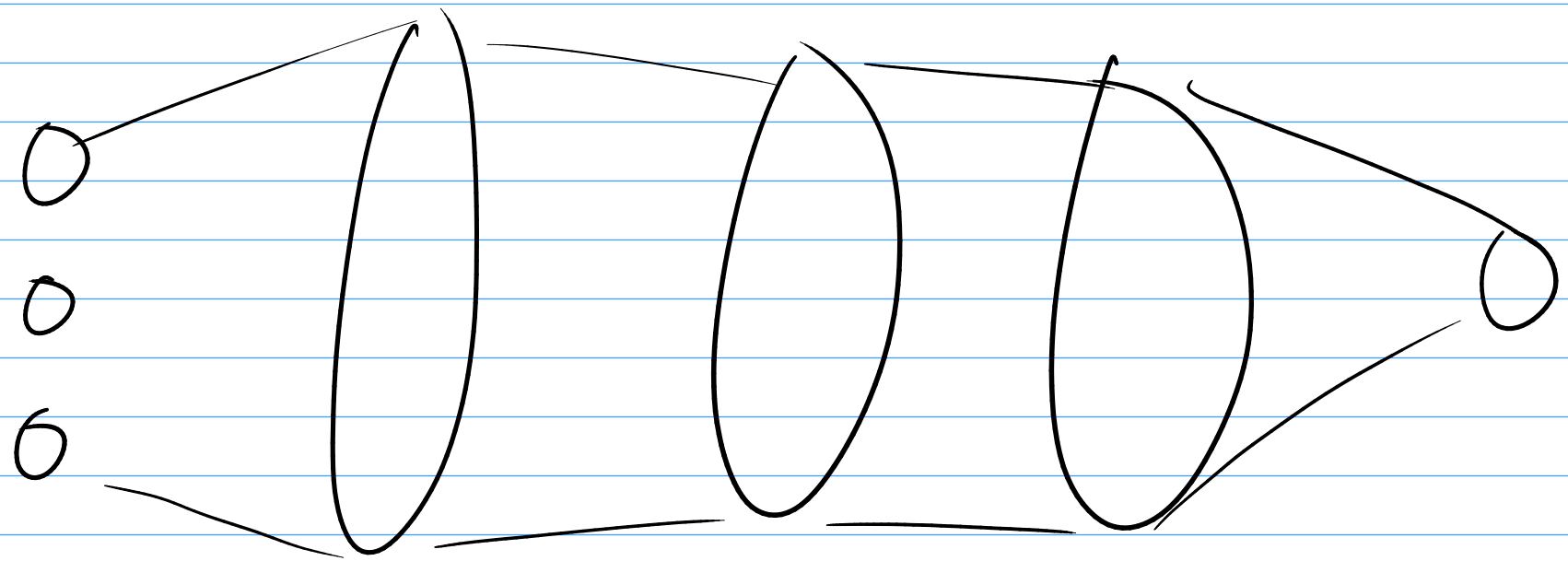


$$f(\vec{x}) = \sum_{i=1}^{m} w_i g(\vec{c}_i^T \vec{x})$$

similar to RBFs:
If C is fixed, training $\vec{w}$ is just like training a linear model

- area of research
- study structures in C & functions you can learn

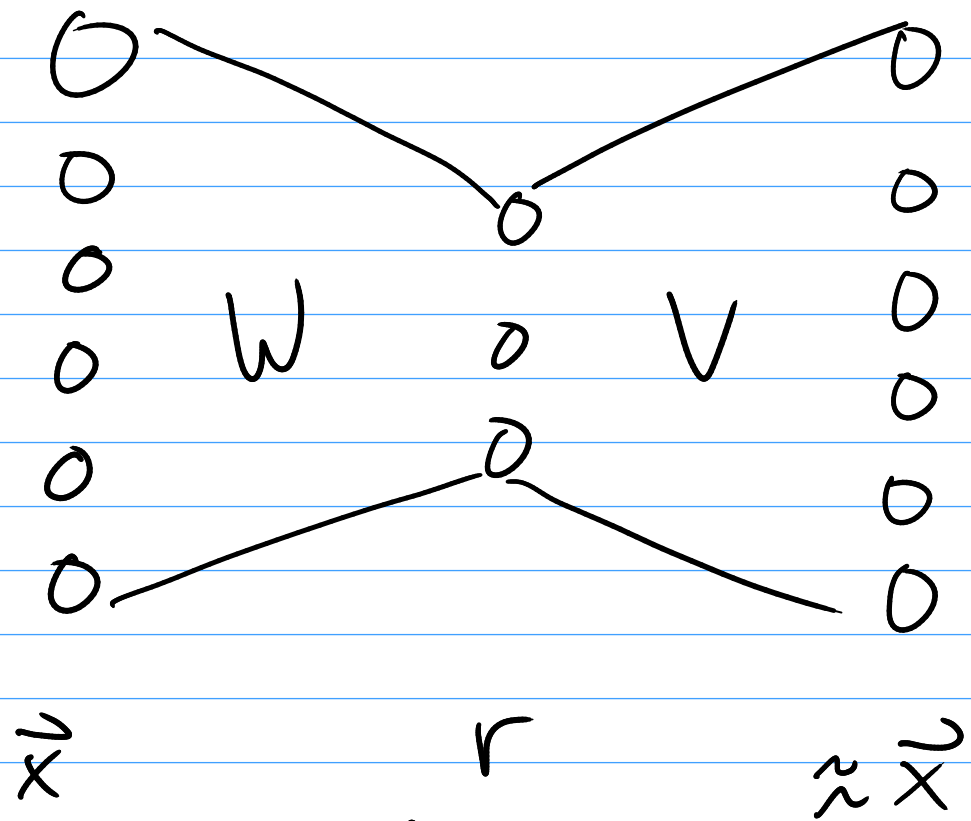
deep learning



In practice:

- train C's SGD
- try lots of layers (1, 2, ...) diff architectures
- still working on theory

autoencoder

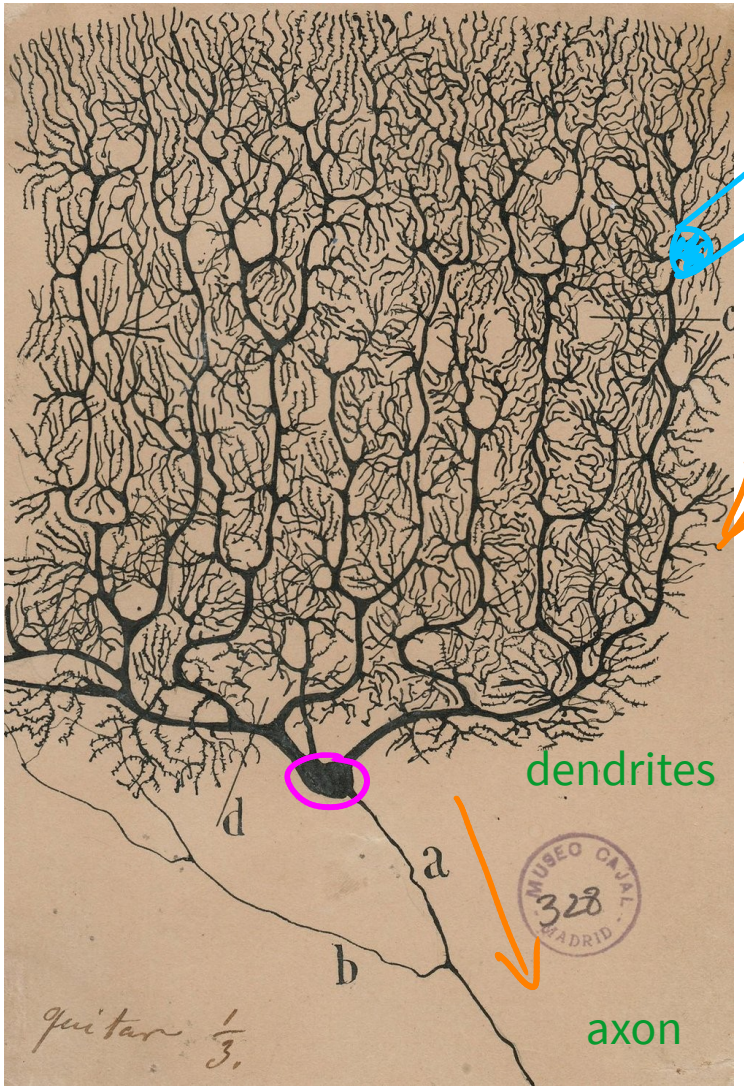


linear

$g(z) = z$

Supporting slides, networks intro

CSCI 471/571 Fall 2020



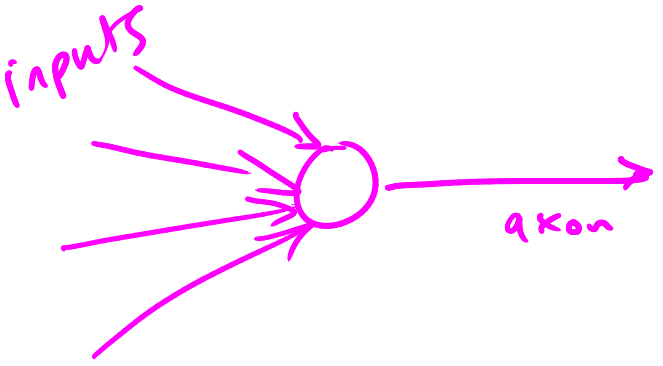
Electrode

inputs

Drawing of a Purkinje neuron
by Santiago Ramón y Cajal

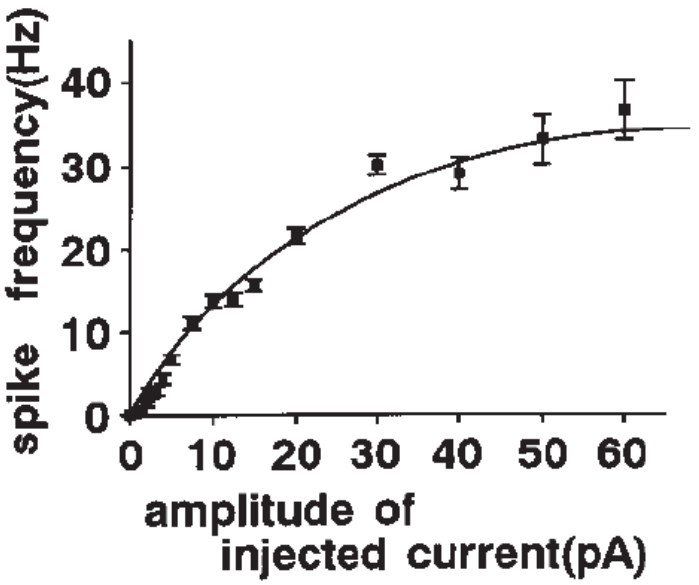
Golgi (silver) staining technique

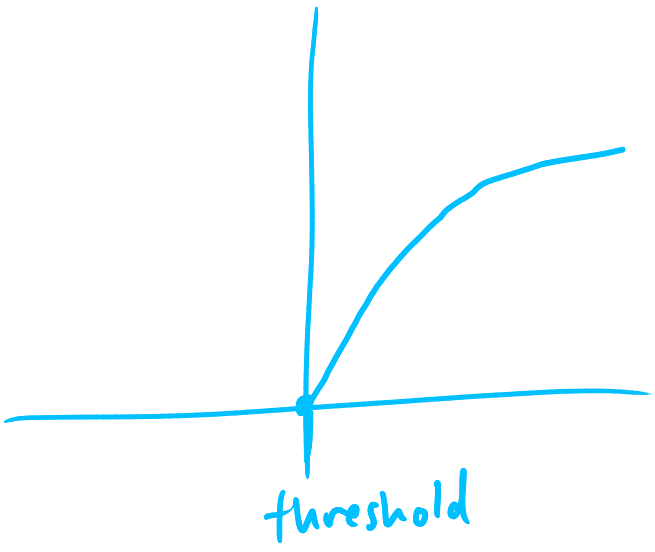
only labels a few neurons



Nonlinearities in real neurons

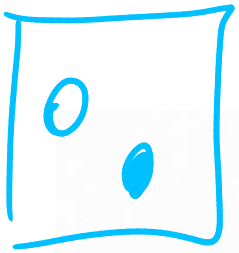
(frog ones, in fact)

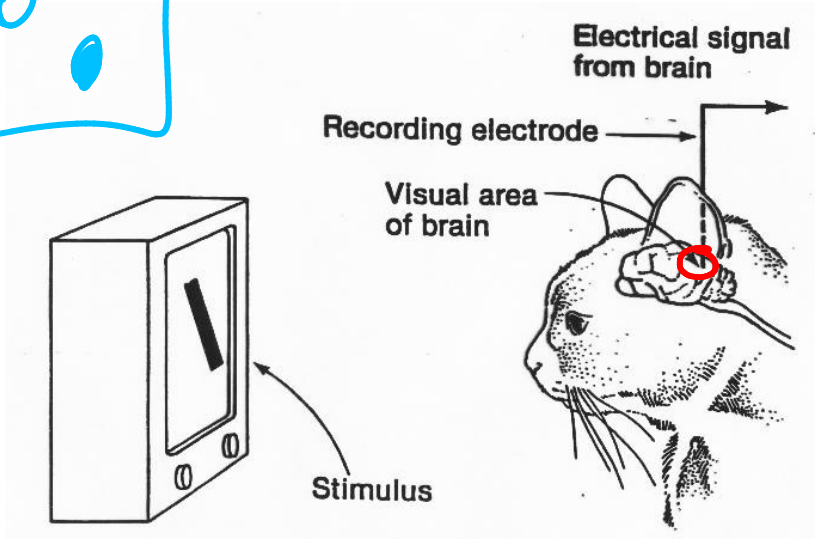




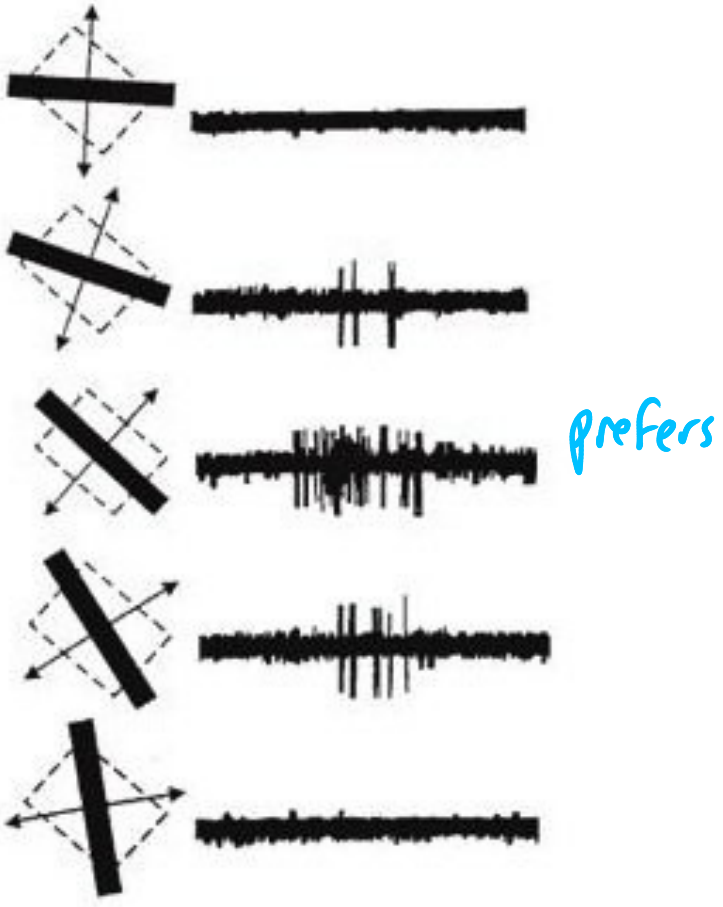
Iida & Kashiwayanagi, J Gen Physiol (1999)

Real neurons are selective to inputs





Hubel & Wiesel
late ‘50s onward



First artificial neural network

Perceptron

Rosenblatt
1950's

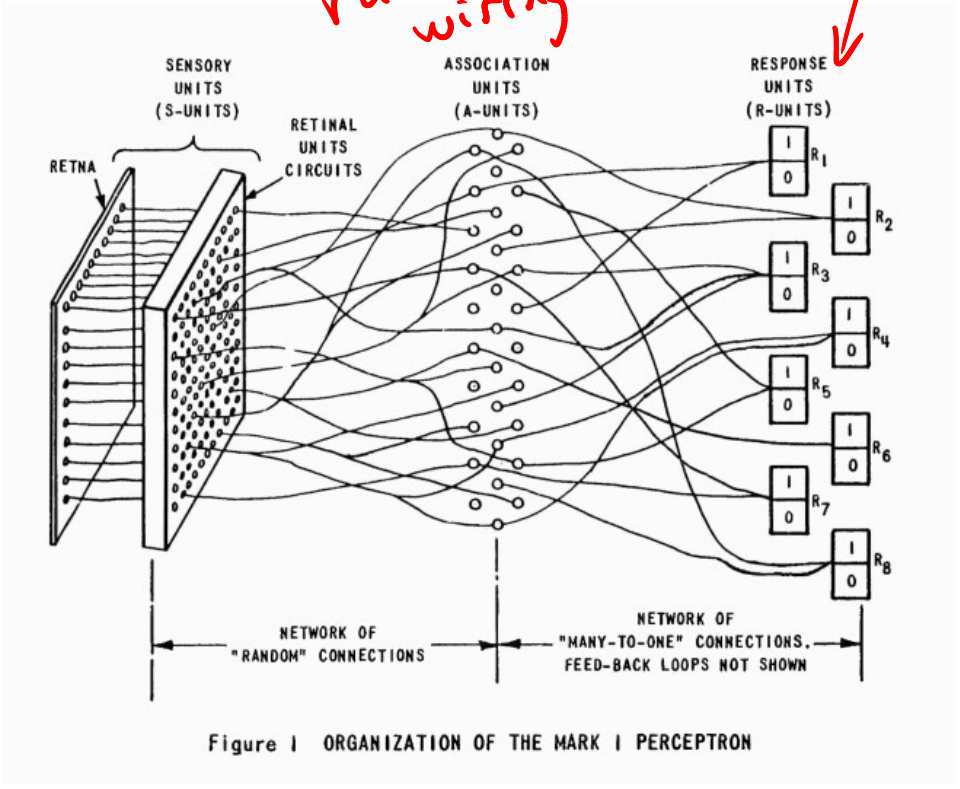
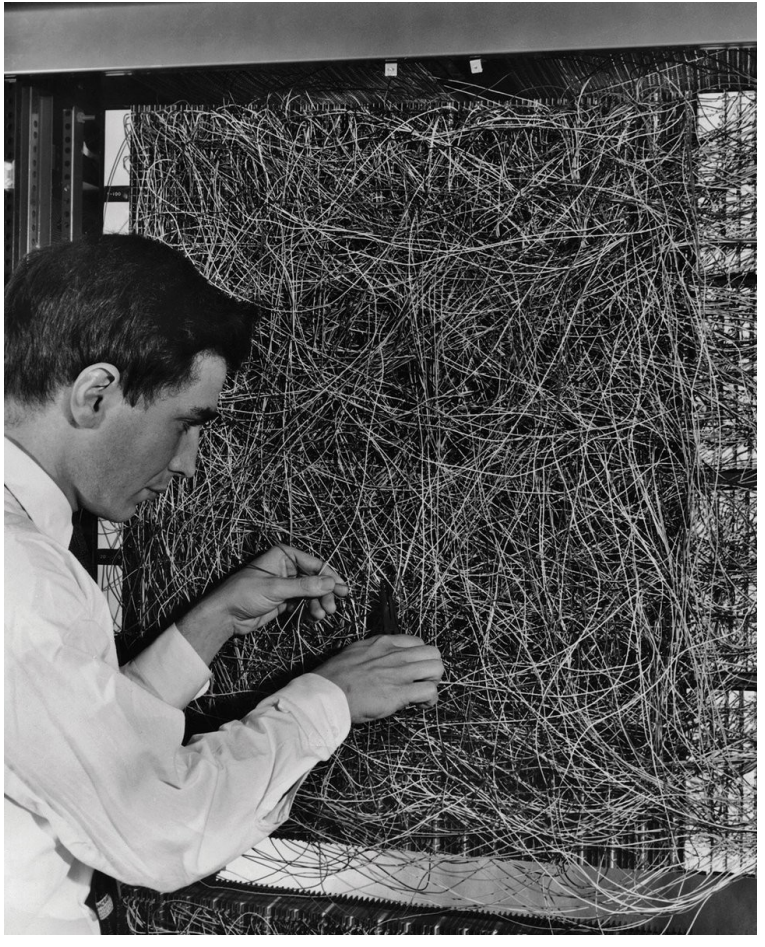


Figure 1 ORGANIZATION OF THE MARK I PERCEPTRON



A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

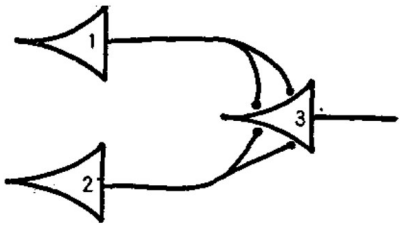
invented symbols →

a

0, 1

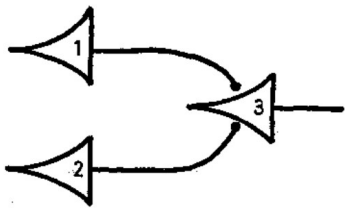
copy

b



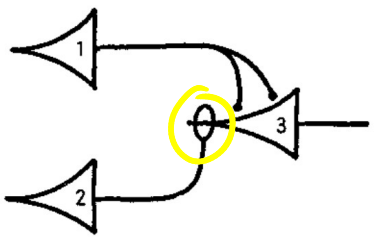
1 and 2

c



1 or 2

d



1 and not 2

All boolean functions are realizable by some network
(Later: all functions are realizable) 70's / 80's