data science
for (physical) scientists XI

Neural networks

dr.federica bianco | fbb.space | fedhere | fedhere

this slide deck:

http://bit.ly/dspsXI

- **Machine Learning basic concepts**
  - interpretability
  - parameters vs hyperparameters
  - supervised/unsupervised

- ~~**CART methods**~~
- **Clustering methods**
- **Neural Networks**

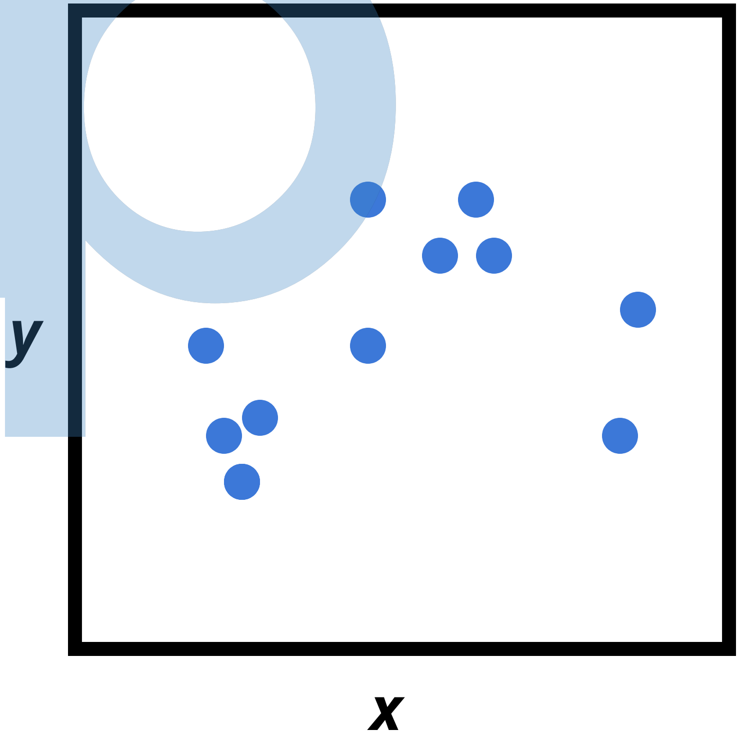
- **Neural Networks**
  - the brain connection
  - perceptron
  - learning
  - activation functions
  - shallow nets
  - deep nets architecture
  - back-propagation
  - preprocessing and whitening (minibatch)

recap

machine
learning

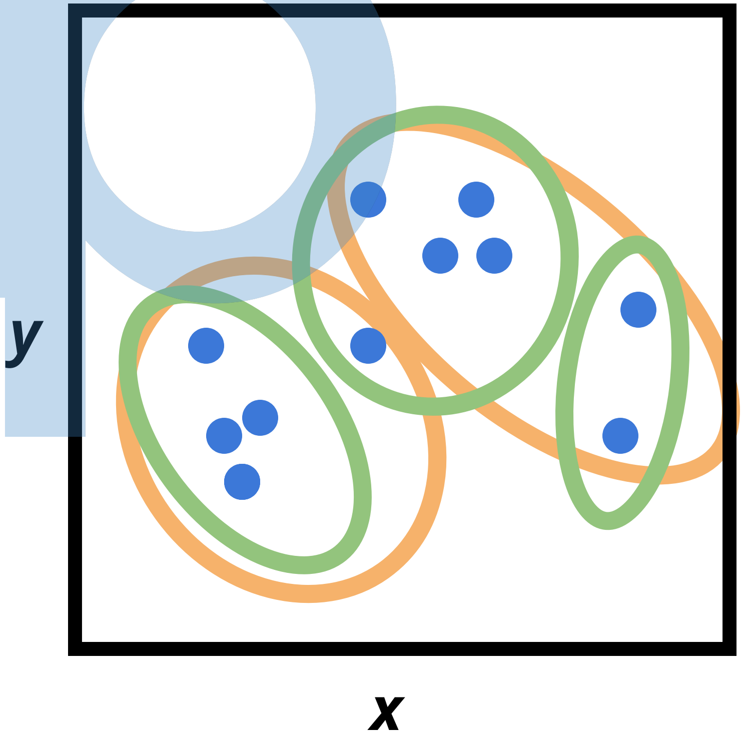
clustering is an unsupervised learning method

GOAL: partitioning data in *maximally homogeneous, maximally distinguished* subsets.



clustering is an unsupervised learning method

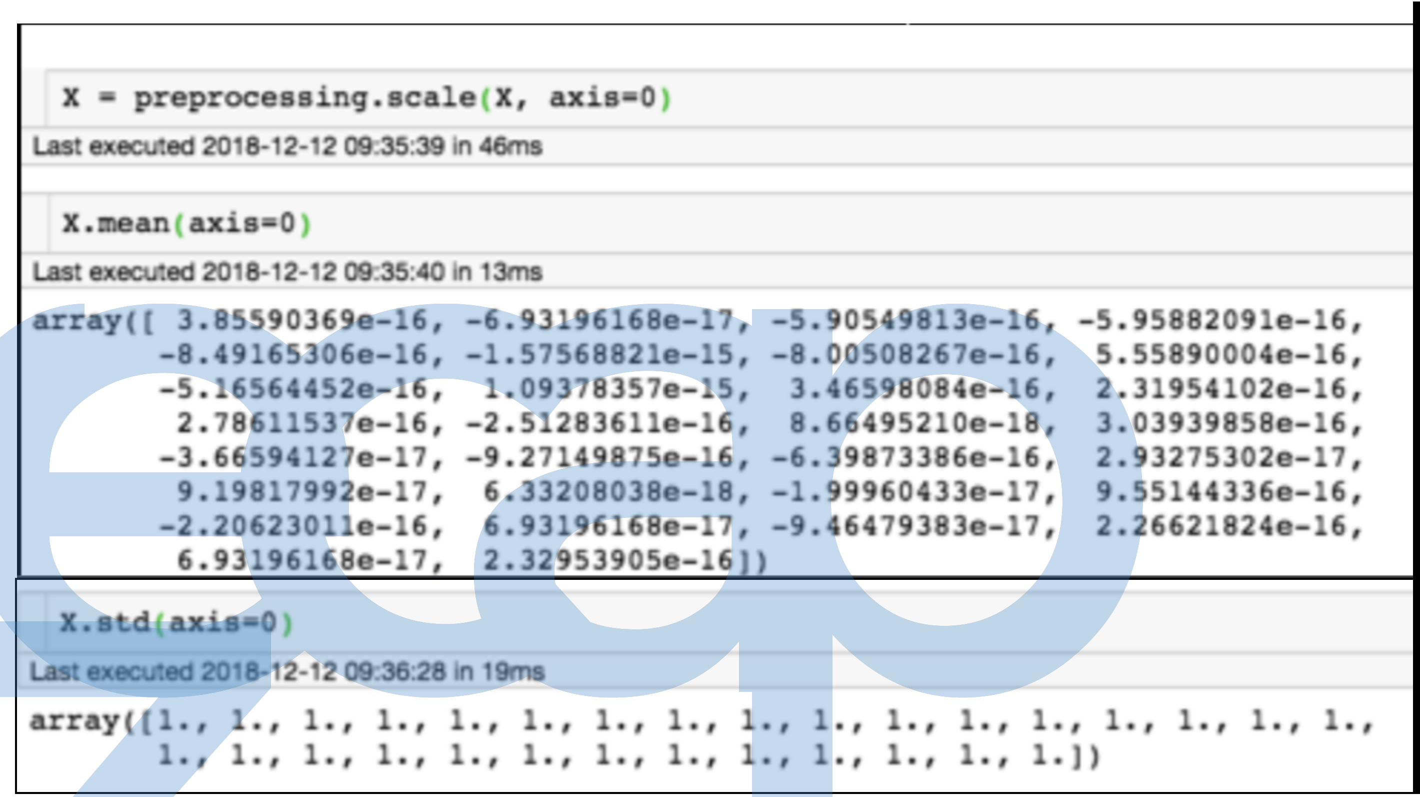
GOAL: partitioning data in *maximally homogeneous, maximally distinguished* subsets.



what optimal clustering is cannot be said outside of context: e.g. purpose, domain knowledge

Generic preprocessing

for each feature: divide by standard deviation and subtract mean



```
X = preprocessing.scale(X, axis=0)
```

Last executed 2018-12-12 09:35:39 in 46ms

```
X.mean(axis=0)
```

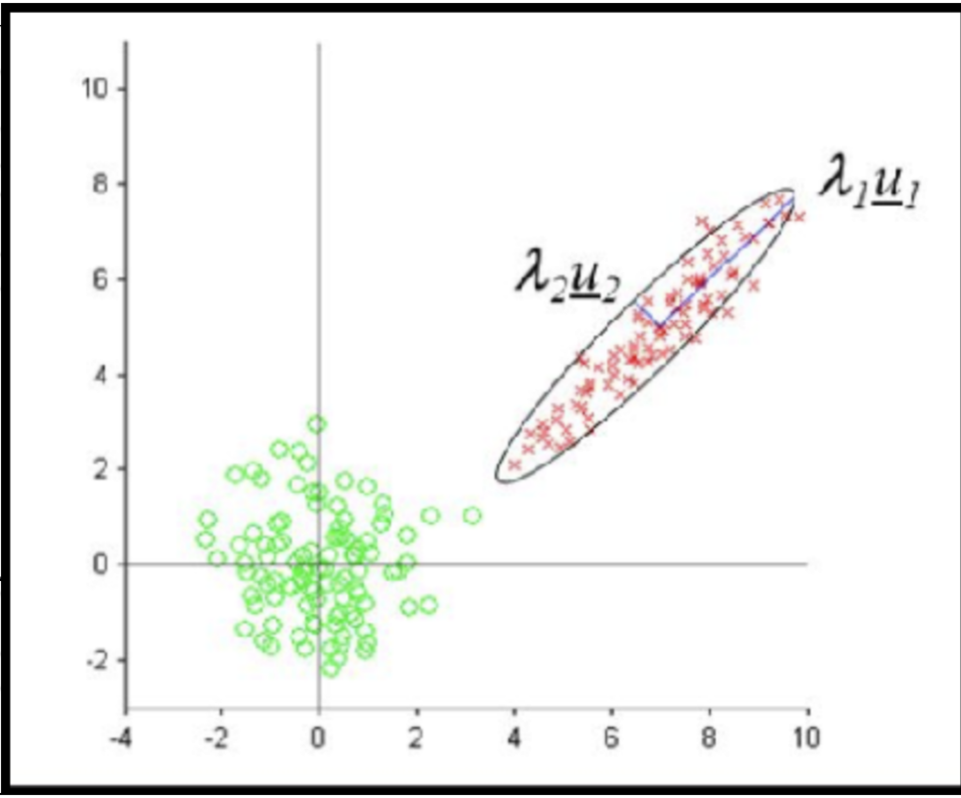
Last executed 2018-12-12 09:35:40 in 13ms

```
array([ 3.85590369e-16, -6.93196168e-17, -5.90549813e-16, -5.95882091e-16,
       -8.49165306e-16, -1.57568821e-15, -8.00508267e-16,  5.55890004e-16,
       -5.16564452e-16,  1.09378357e-15,  3.46598084e-16,  2.31954102e-16,
        2.78611537e-16, -2.51283611e-16,  8.66495210e-18,  3.03939858e-16,
       -3.66594127e-17, -9.27149875e-16, -6.39873386e-16,  2.93275302e-17,
        9.19817992e-17,  6.33208038e-18, -1.99960433e-17,  9.55144336e-16,
       -2.20623011e-16,  6.93196168e-17, -9.46479383e-17,  2.26621824e-16,
        6.93196168e-17,  2.32953905e-16])
```

```
X.std(axis=0)
```

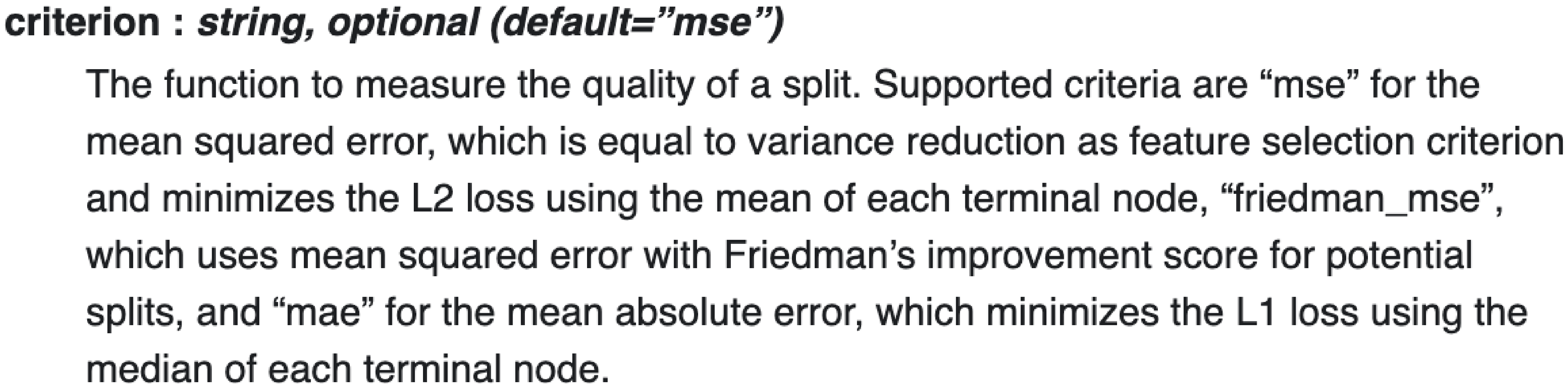
Last executed 2018-12-12 09:36:28 in 19ms

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```



mean of each feature should be 0, standard deviation of each feature should be 1

Hyperparameters



criterion : *string, optional (default=”mse”)*

The function to measure the quality of a split. Supported criteria are “mse” for the mean squared error, which is equal to variance reduction as feature selection criterion and minimizes the L2 loss using the mean of each terminal node, “friedman_mse”, which uses mean squared error with Friedman’s improvement score for potential splits, and “mae” for the mean absolute error, which minimizes the L1 loss using the median of each terminal node.

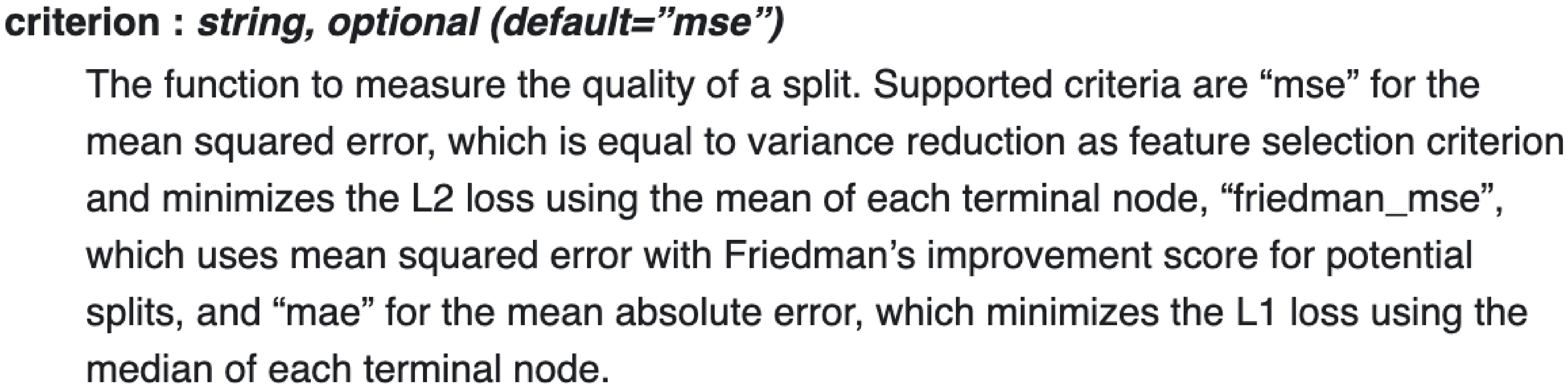
mean square error

$$L_2 = \Sigma\left(y_{true} - y_{predicted}\right)^2$$

mean absolute error

$$L_1 = \Sigma\left|y_{true} - y_{predicted}\right|$$

A single tree: hyperparameters



criterion : *string, optional (default="mse")*

The function to measure the quality of a split. Supported criteria are "mse" for the mean squared error, which is equal to variance reduction as feature selection criterion and minimizes the L2 loss using the mean of each terminal node, "friedman_mse", which uses mean squared error with Friedman's improvement score for potential splits, and "mae" for the mean absolute error, which minimizes the L1 loss using the median of each terminal node.

mean square error

$$L_2 = \Sigma\left(y_{true} - y_{predicted}\right)^2$$

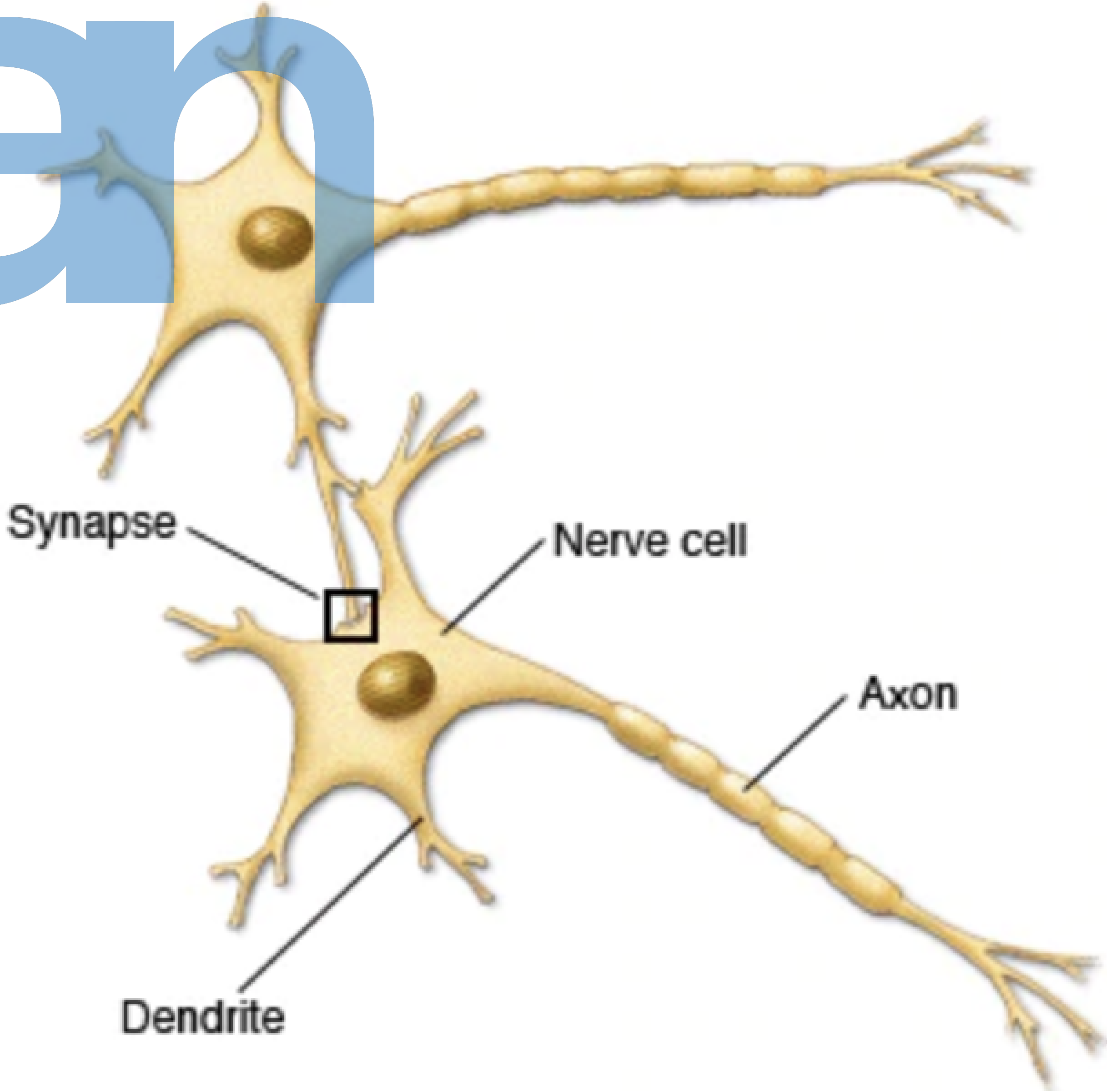
mean absolute error

$$L_1 = \Sigma\left|y_{true} - y_{predicted}\right|$$

neural
networks

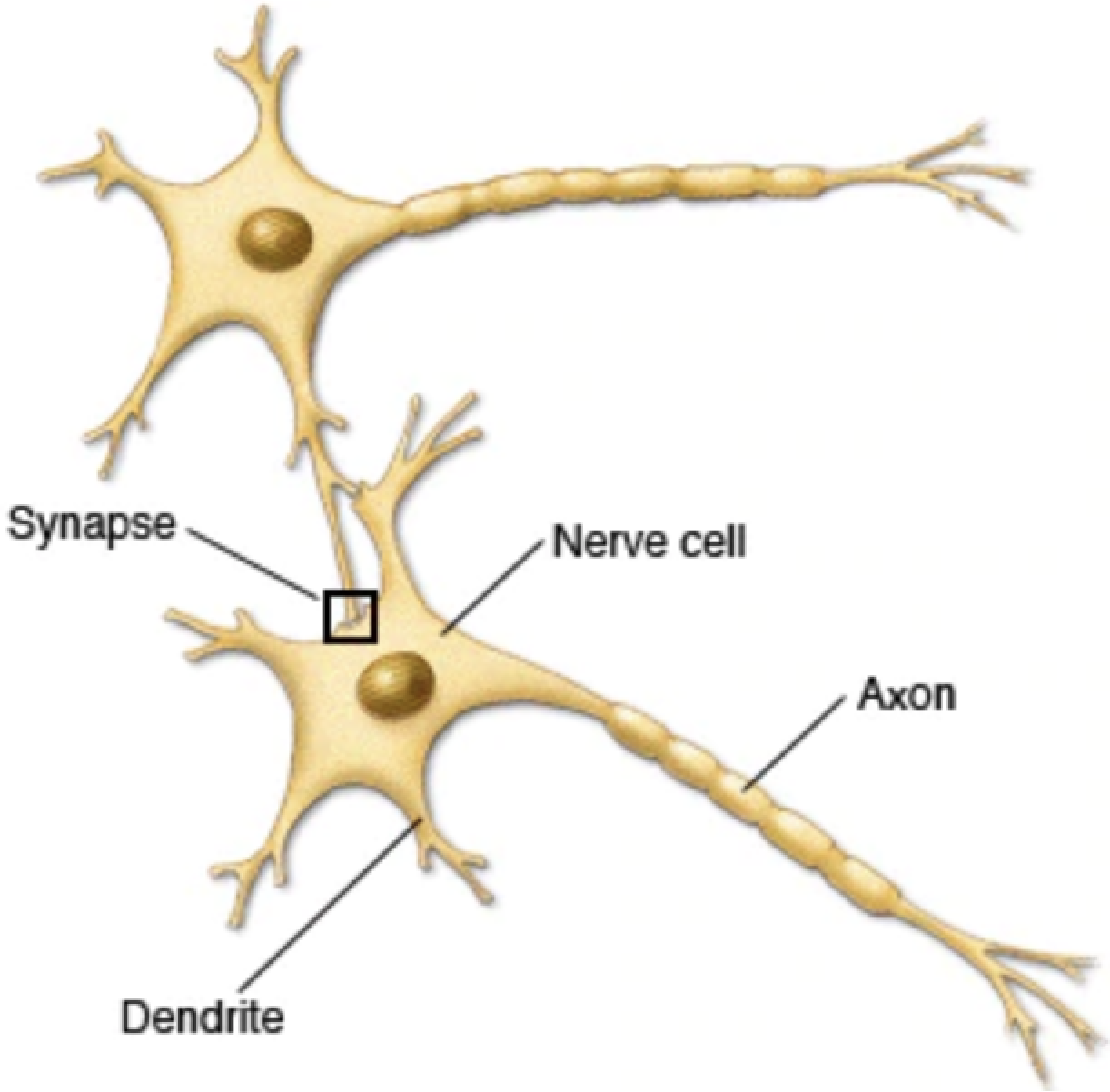
the brain

1



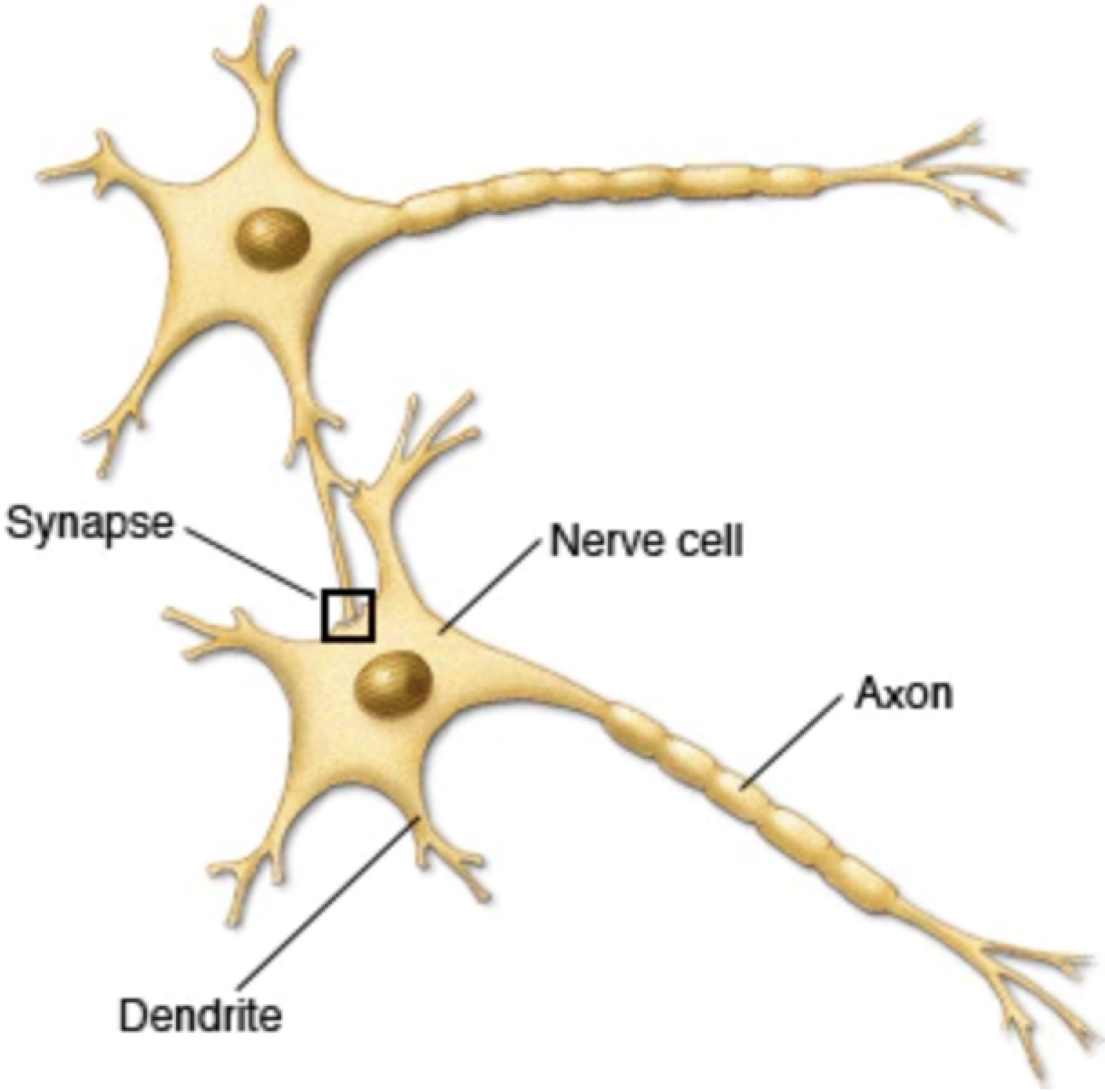
Neurons (nerve cells) are connected into a network: dendrites receive incoming messages from other nerve cells; axons carry outgoing signals,

How brains works



Neurons communicates with other cells through electrical impulses releasing chemicals that pass through the synapse, the gap between two nerve cells, and attach to receptors on the receiving cell.

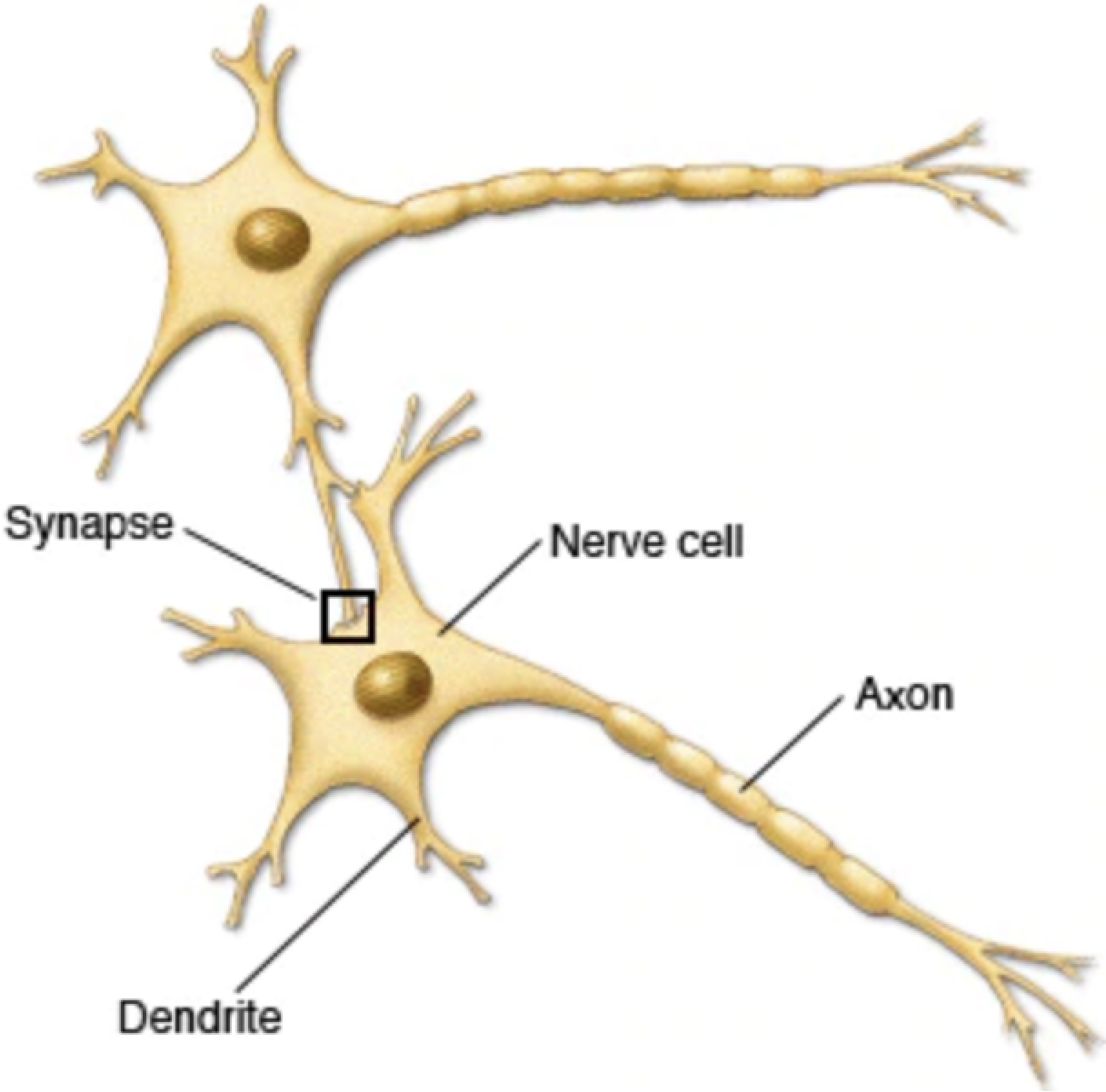
How brains works



How brains works

In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using electrical circuits.

https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html



perceptions

2

The perceptron algorithm : 1958, Frank Rosenblatt

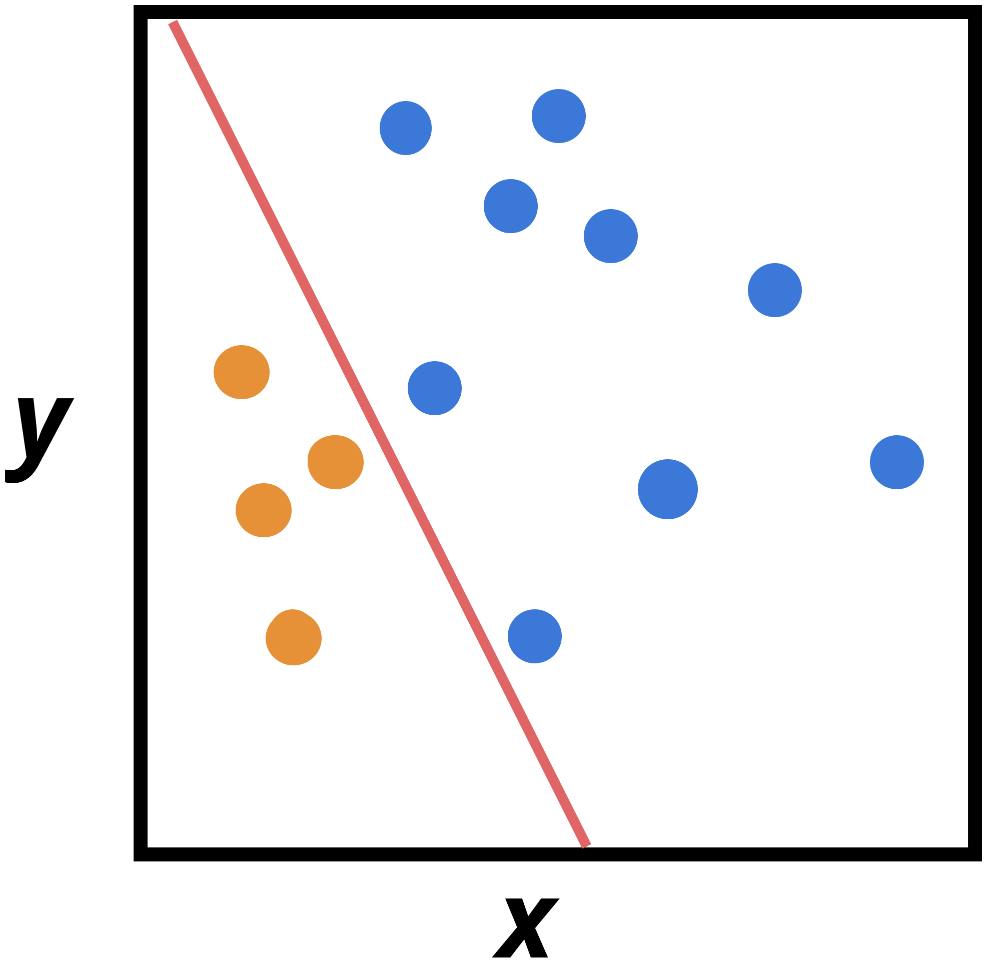
Perceptrons are ***linear classifiers:*** makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.

$$y = wx + b$$

"the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."

F. Rosenblatt, 1958



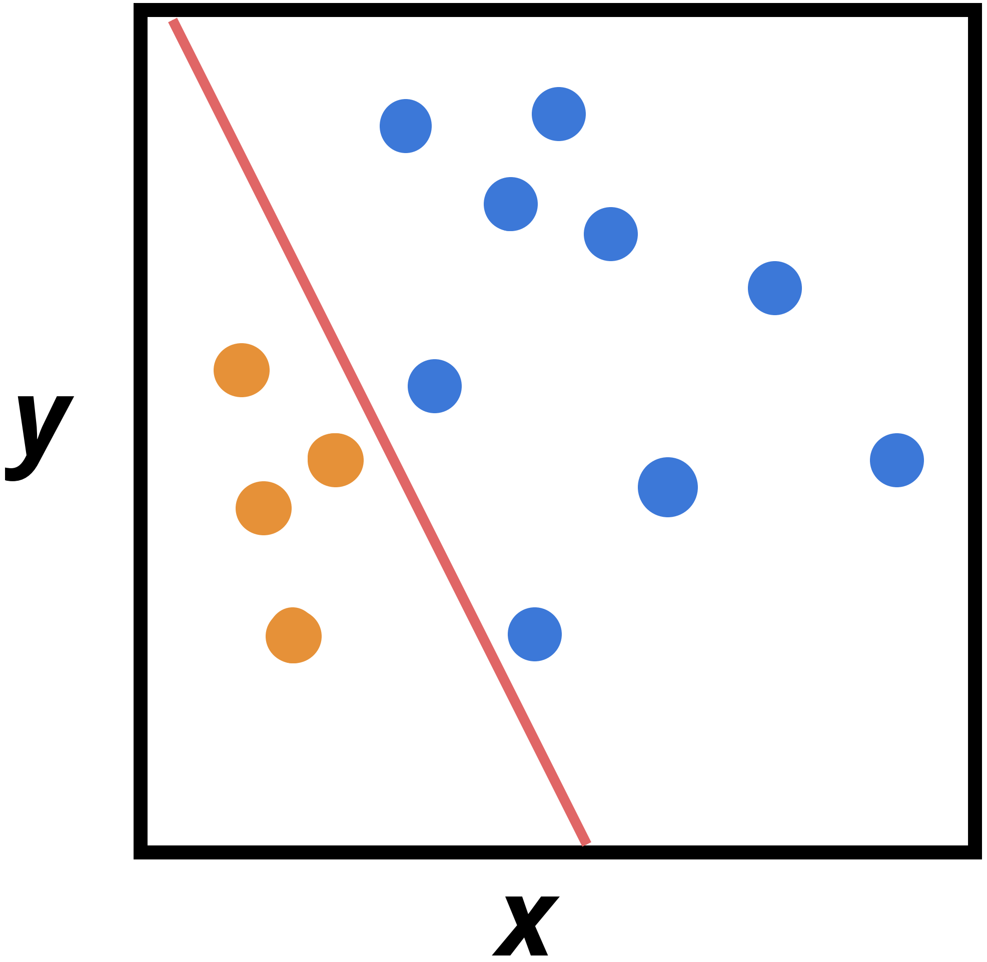
The perceptron algorithm : 1958, Frank Rosenblatt

Perceptrons are ***linear classifiers:*** makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.

$$y = wx + b$$ in 1D

$$y = \sum_i w_i x_i + b$$ in N-D



The perceptron algorithm : 1958, Frank Rosenblatt

Perceptrons are ***linear classifiers:*** makes its predictions based on a linear predictor function

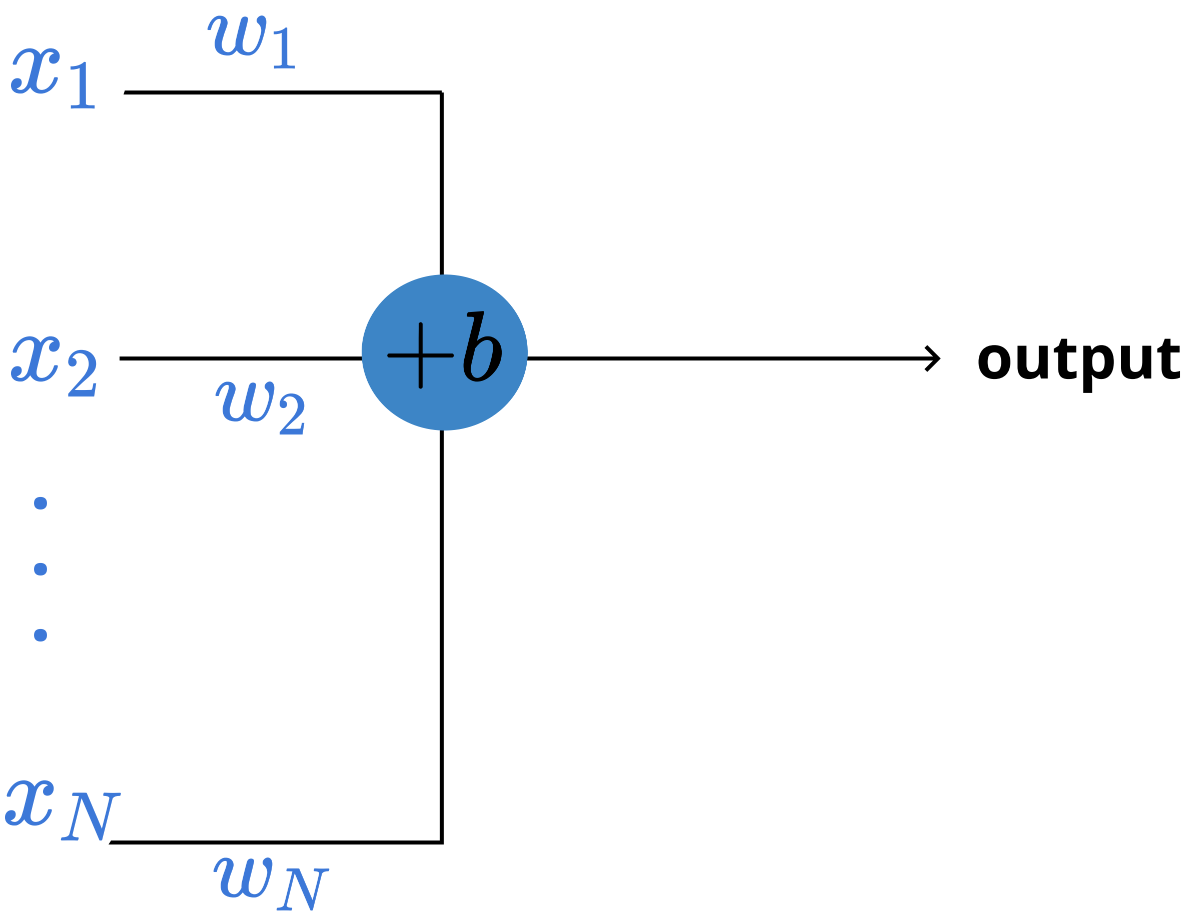
combining a set of weights (=parameters) with the feature vector.

$$y = wx + b$$

$$y = \sum_i w_i x_i + b$$

linear regression: w_i weights

b bias



ADELINE and MADELINE 1962 - B. Widrow & M. Hoff

SELF-ORGANIZING SYSTEMS 1962

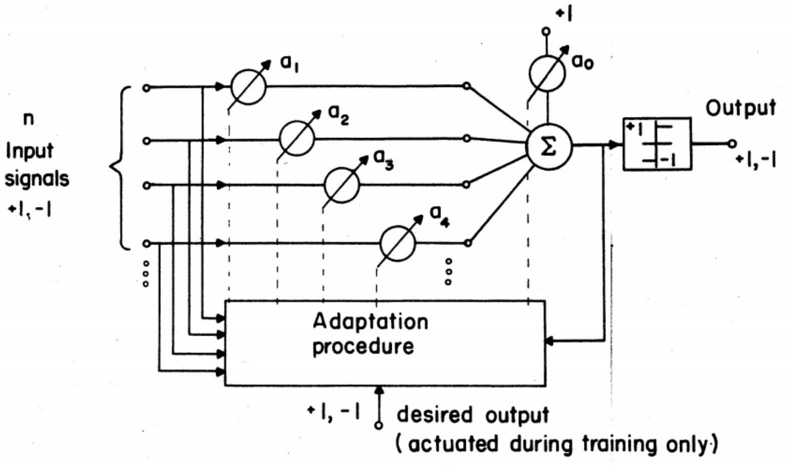


Figure 1. An Automatically-Adapted Threshold Element.

Edited By:

MARSHALL C. YOVITS, Office of Naval Research

GEORGE T. JACOBI, Armour Research Foundation

GORDON D. GOLDSTEIN, Office of Naval Research

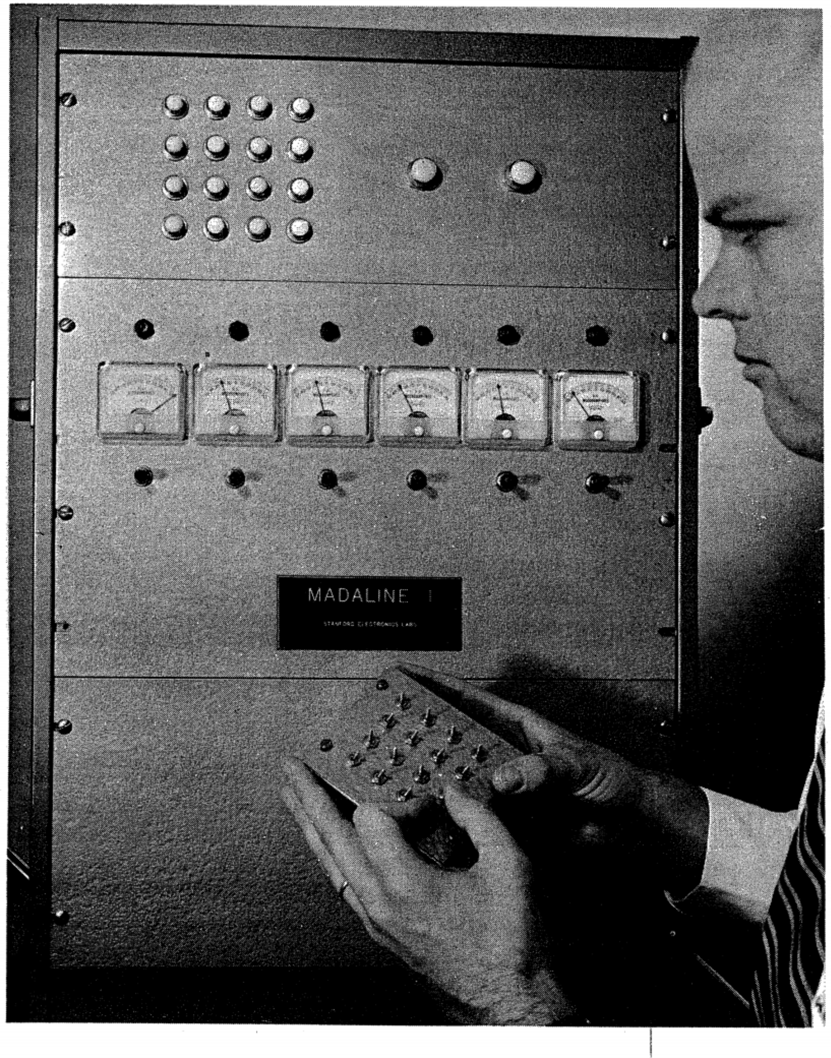


Figure 14. MADALINE I and W. C. RIDGWAY, III.

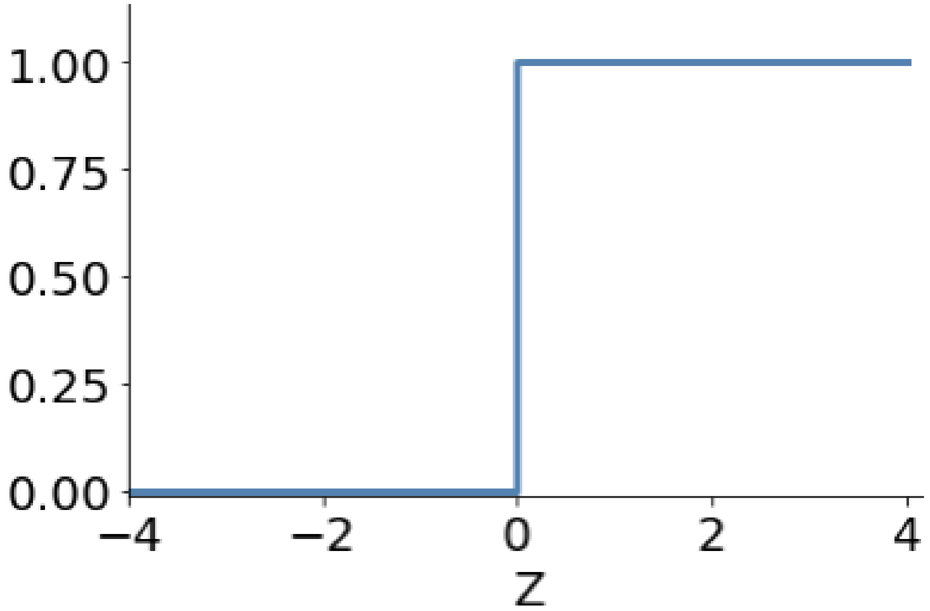
http://www-isl.stanford.edu/~widrow/papers/c1961generalizationand.pdf

ADELINE and MADELINE 1962 - B. Widrow & M. Hoff

Perceptrons are ***linear classifiers***: makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.

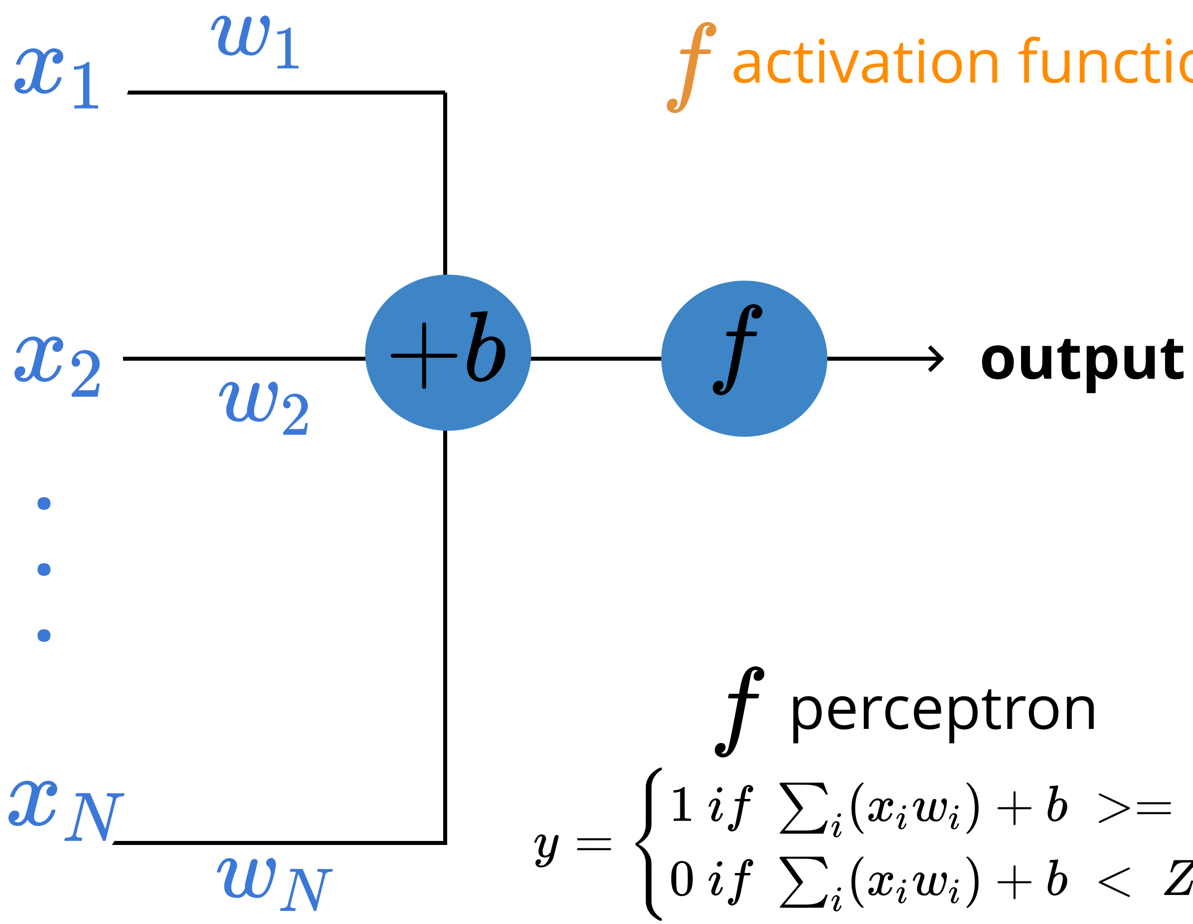
$$y = f(\textstyle\sum_i w_i x_i + b)$$



w_i weights

b bias

f activation function



f perceptron

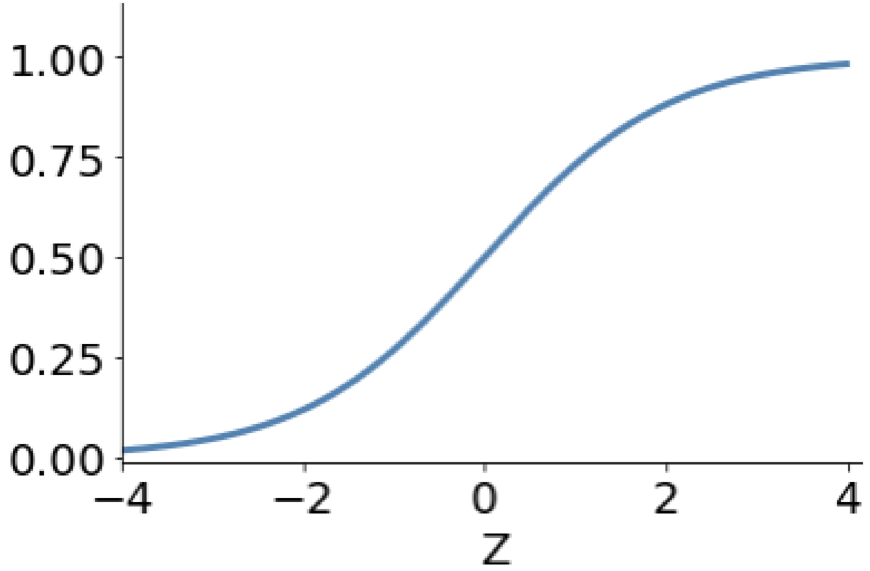
$$y = \begin{cases} 1\ if\ \sum_i (x_i w_i) + b\ >=\ Z \\ 0\ if\ \sum_i (x_i w_i) + b\ <\ Z \end{cases}$$

ADELINE and MADELINE 1962 - B. Widrow & M. Hoff

Perceptrons are ***linear classifiers*:** makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.

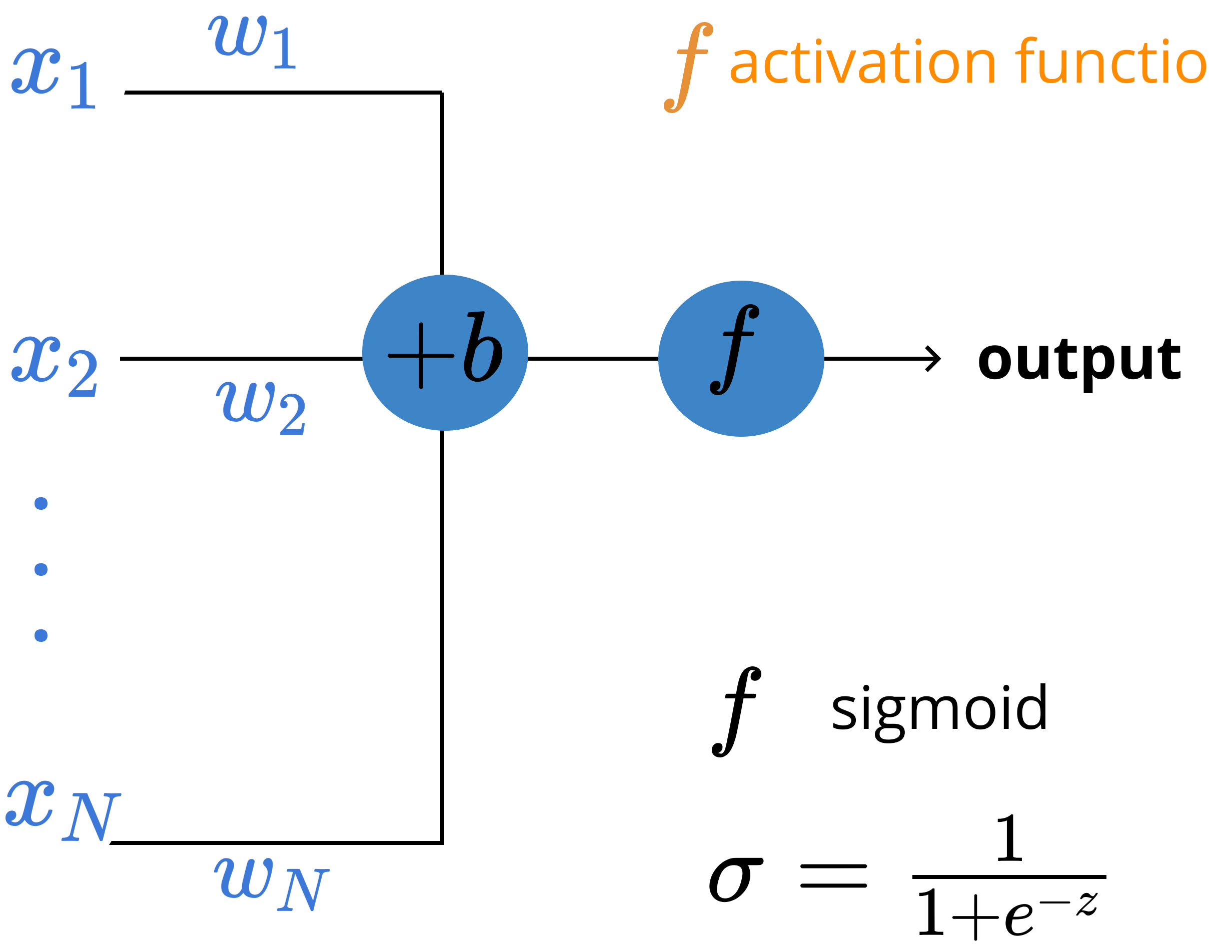
$$y \ = \ f(\textstyle\sum_i w_i x_i \ + \ b)$$



w_i weights

b bias

f activation function



f sigmoid

$$\sigma = \frac{1}{1+e^{-z}}$$

ADELINE and MADELINE 1962 - B. Widrow & M. Hoff

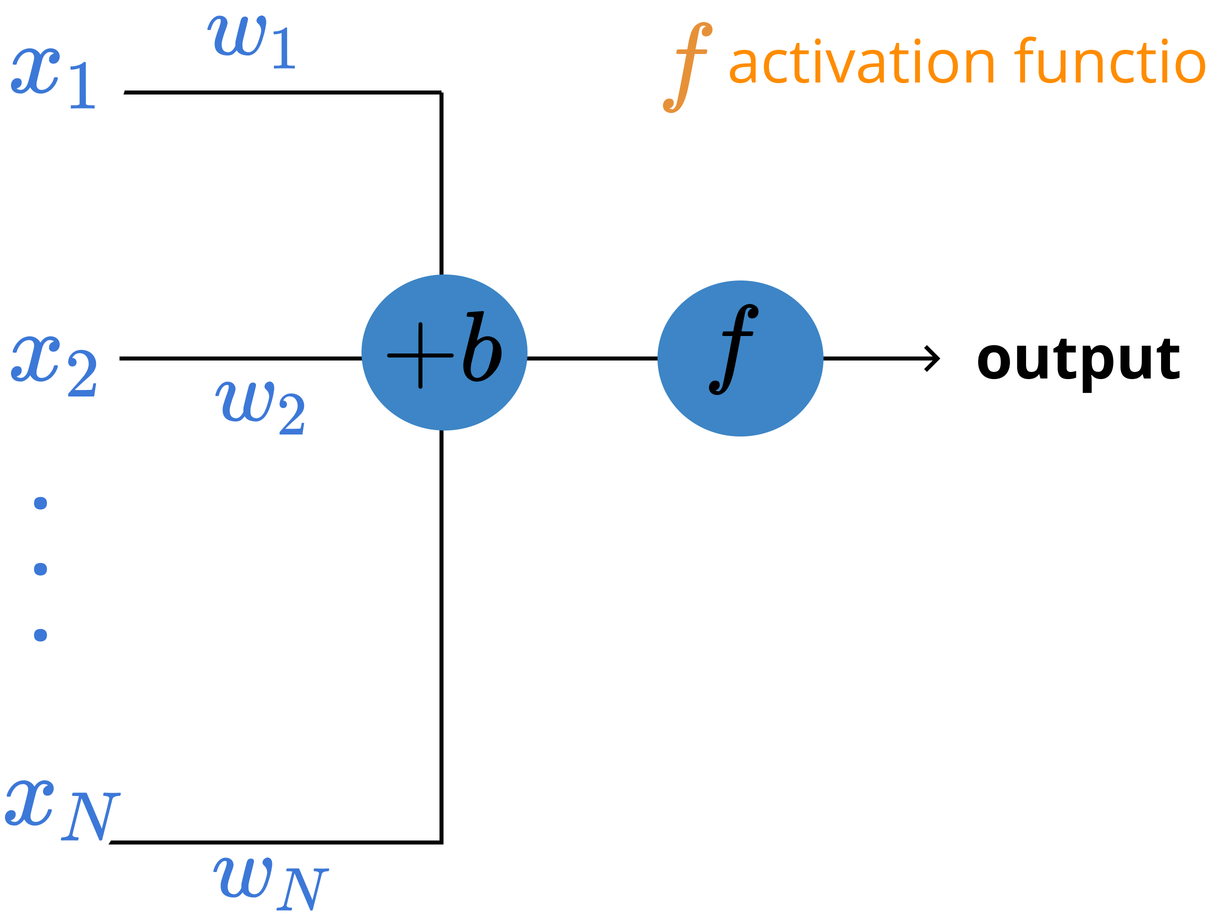
Perceptrons are ***linear classifiers*:**
makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.

$$y = wx + b$$

$$y = \sum_i w_i x_i + b$$

$$y = f(\sum_i w_i x_i + b)$$



w_i weights

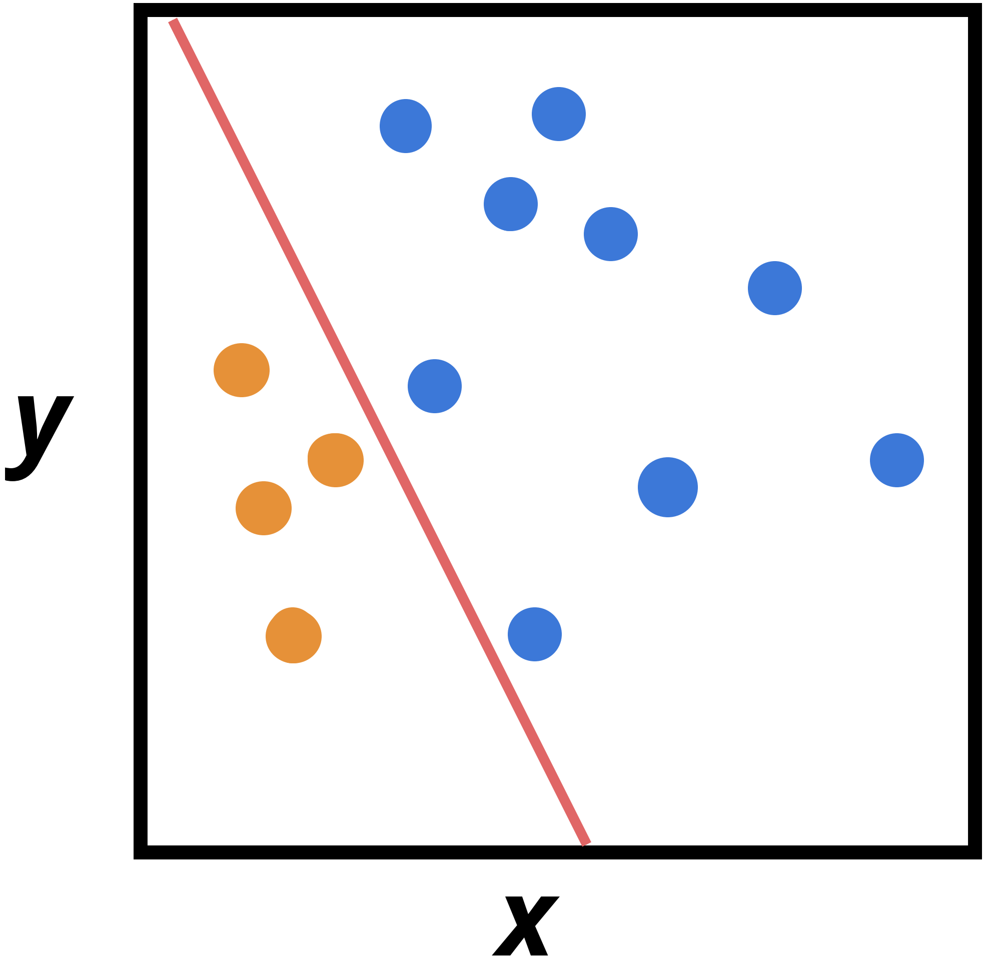
b bias

f activation function

ADELINE and MADELINE 1962 - B. Widrow & M. Hoff

Perceptrons are ***linear classifiers:*** makes its predictions based on a linear predictor function

combining a set of weights (=parameters) with the feature vector.



Problem:

can only learn linearly separable patterns

... time went by... 2+ DECADES

learning

3

widrow-hoff rule

Weight Change = (Pre-Weight line value)(Error / (Number of Inputs)).

http://www-isl.stanford.edu/~widrow/papers/c1988madalinerule.pdf

widrow-hoff rule

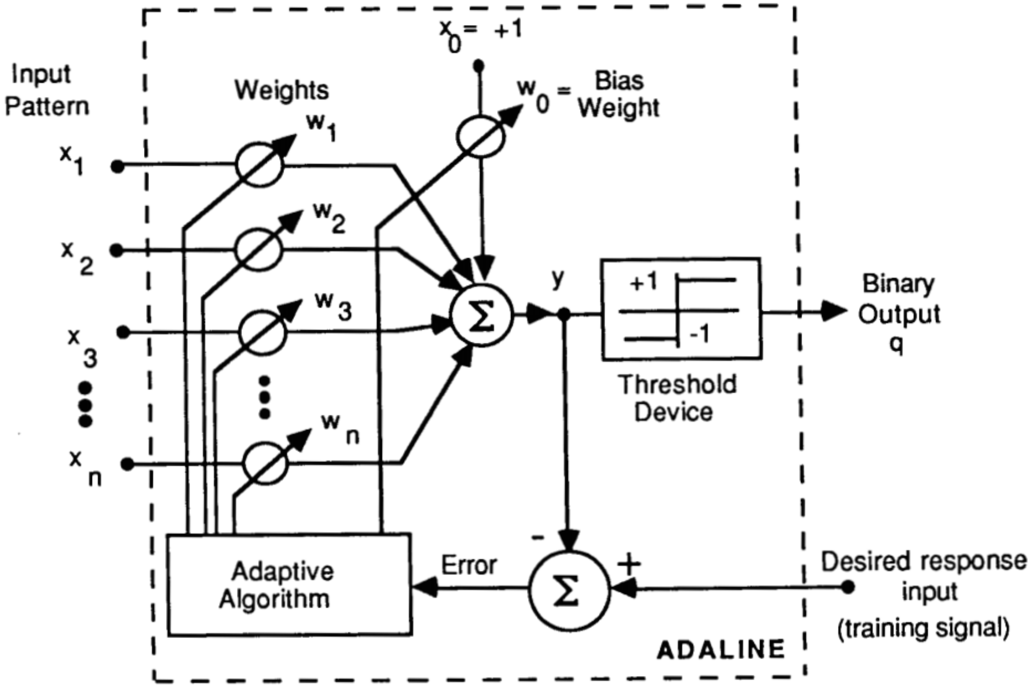


Figure 2: Adaptive linear neuron (ADALINE)

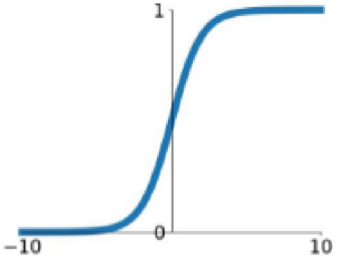
http://www-isl.stanford.edu/~widrow/papers/c1988madalinerule.pdf how do you choose the parameters?

activation functions

4

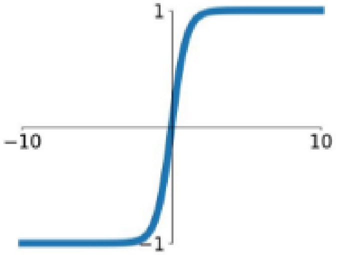
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



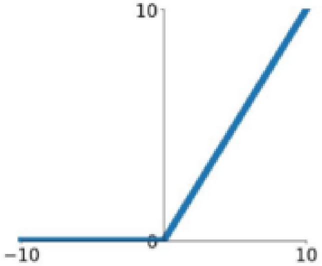
tanh

$\tanh(x)$



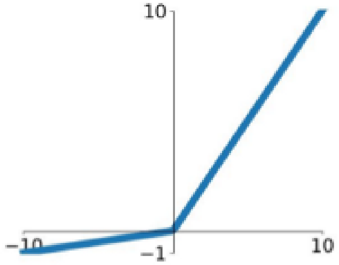
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

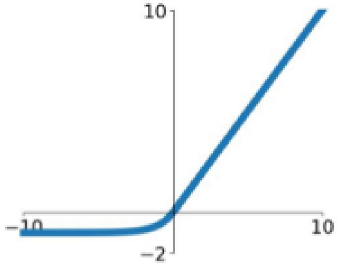


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

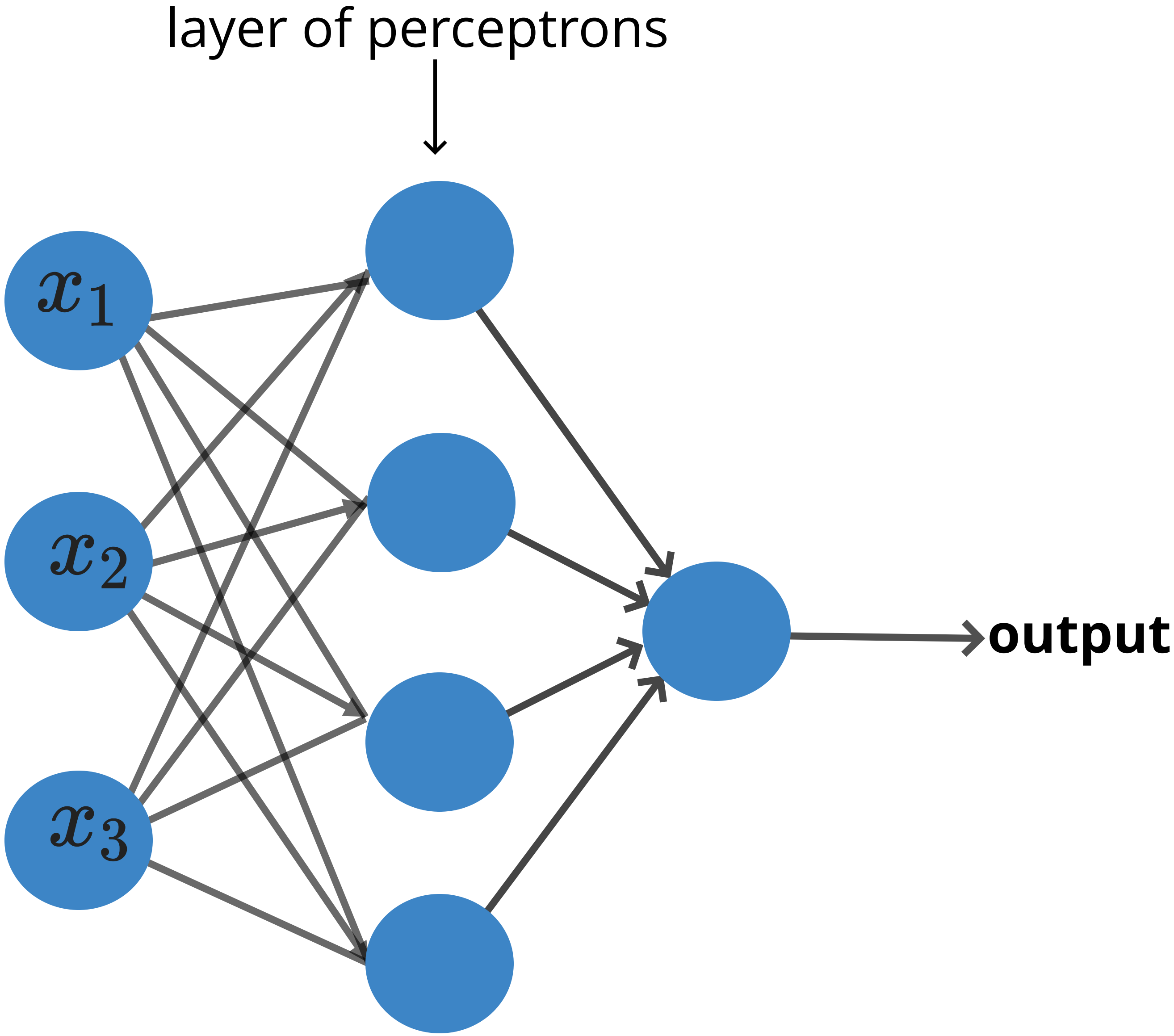


shallow networks

5

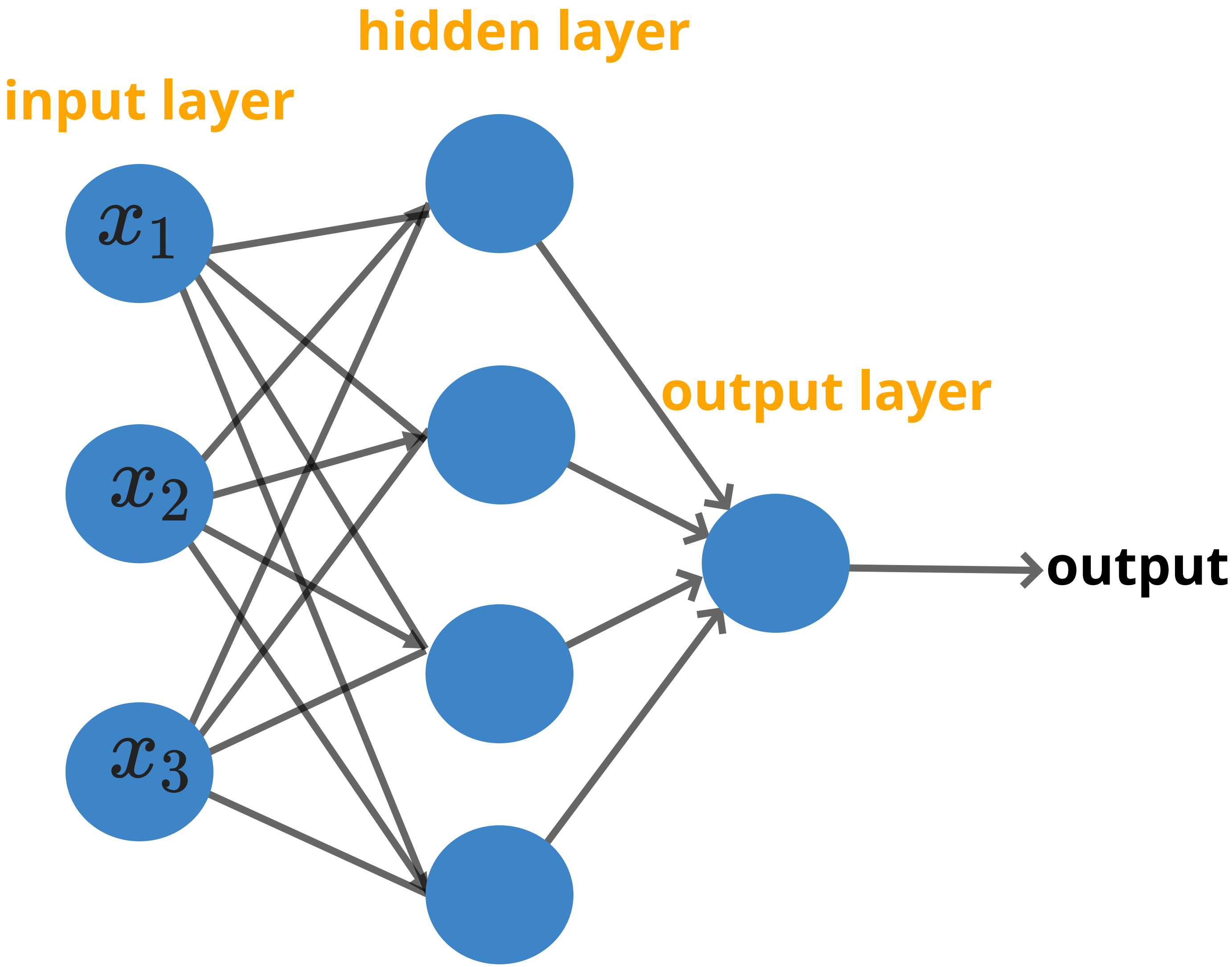
multilayer perceptron

Fully connected: all nodes go to all nodes of the next layer.



multilayer perceptron

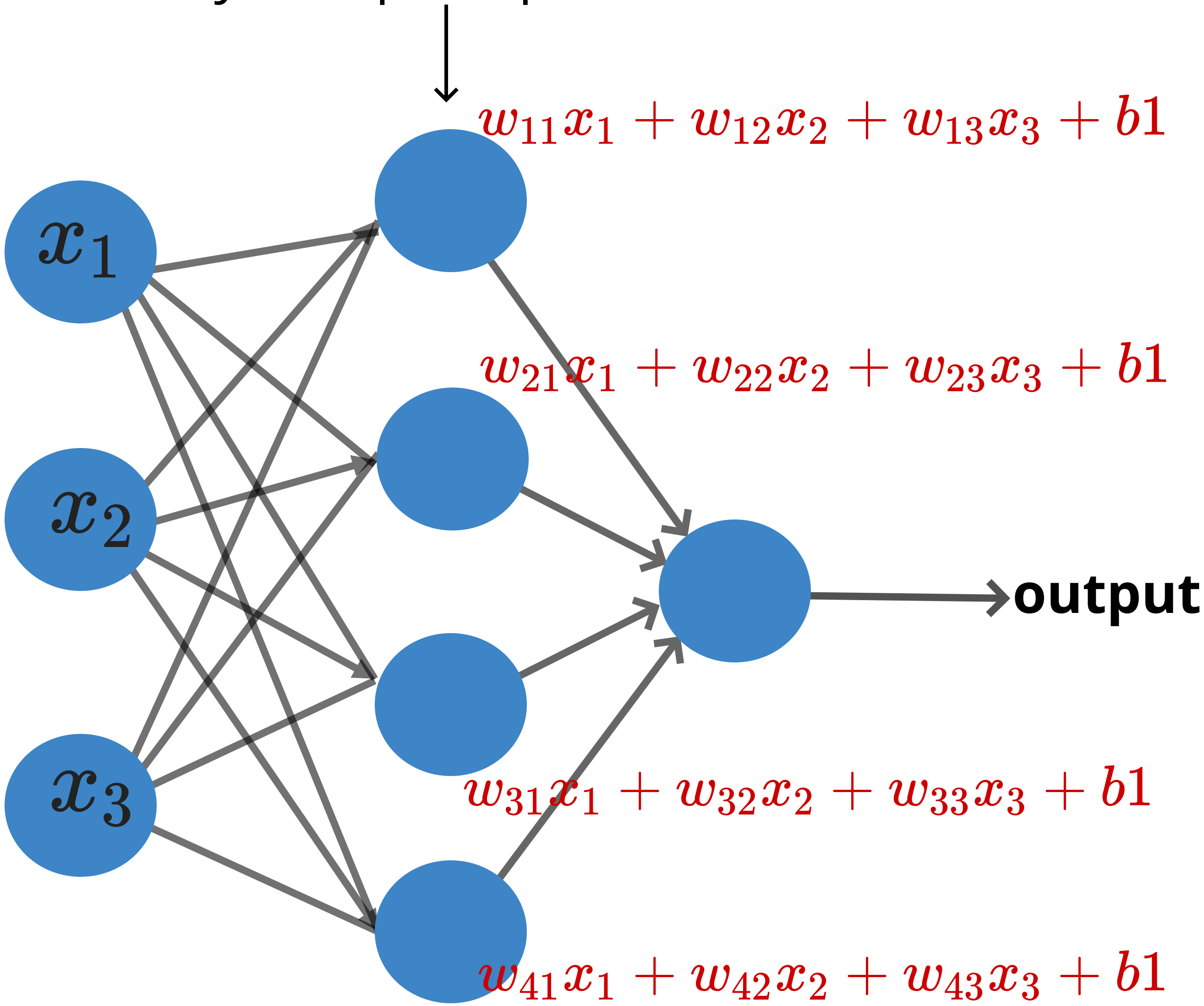
1970: multilayer perceptron architecture



Fully connected: all nodes go to all nodes of the next layer.

multilayer perceptron

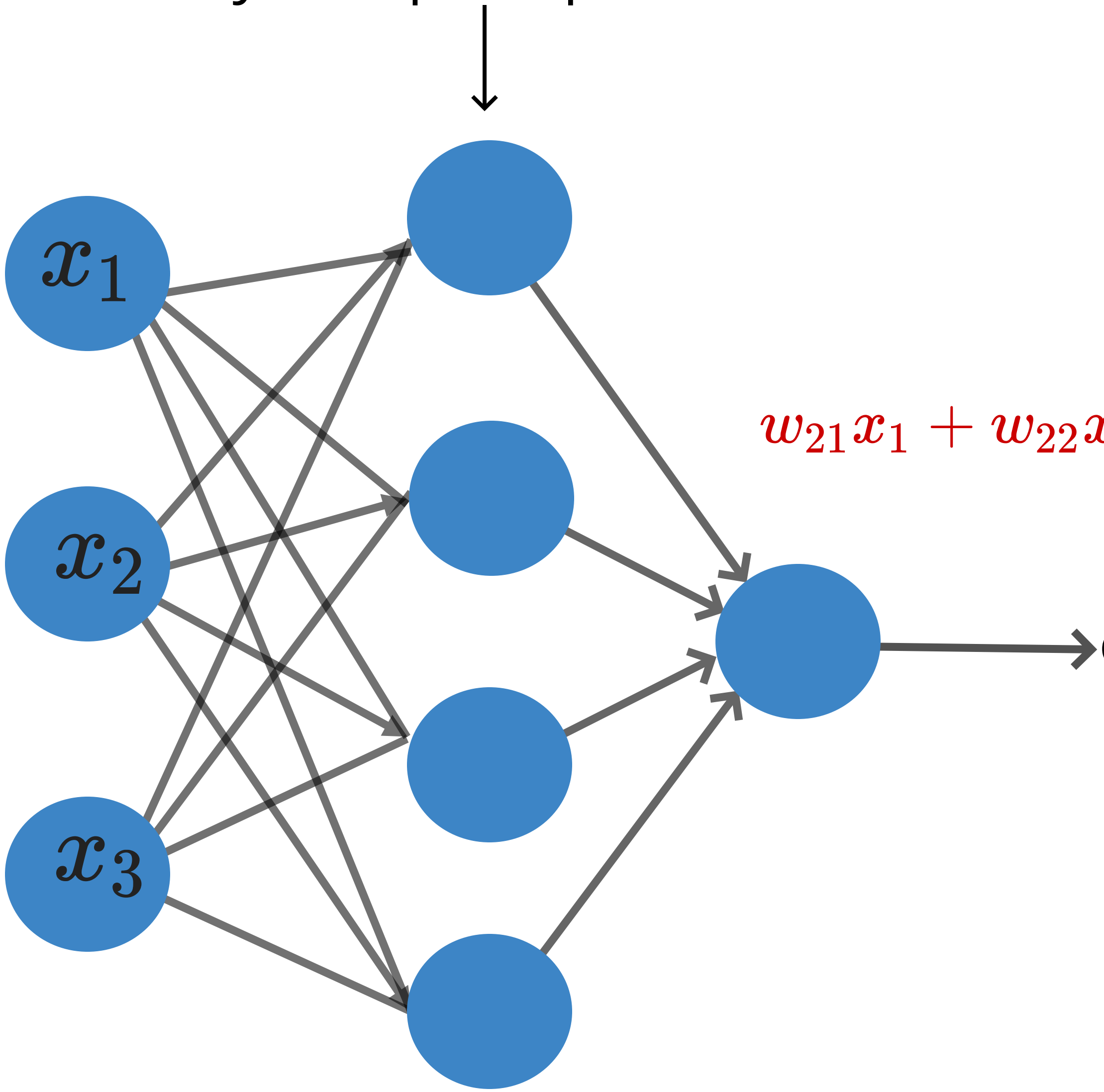
Fully connected: all nodes go to all nodes of the next layer.


layer of perceptrons

x_1
x_2
x_3
$w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b1$
$w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b1$
$w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + b1$
$w_{41}x_1 + w_{42}x_2 + w_{43}x_3 + b1$
output

multilayer perceptron

Fully connected: all nodes go to all nodes of the next layer.



layer of perceptrons

$w_{21}x_1 + w_{22}x_2 + w_{23}$

output

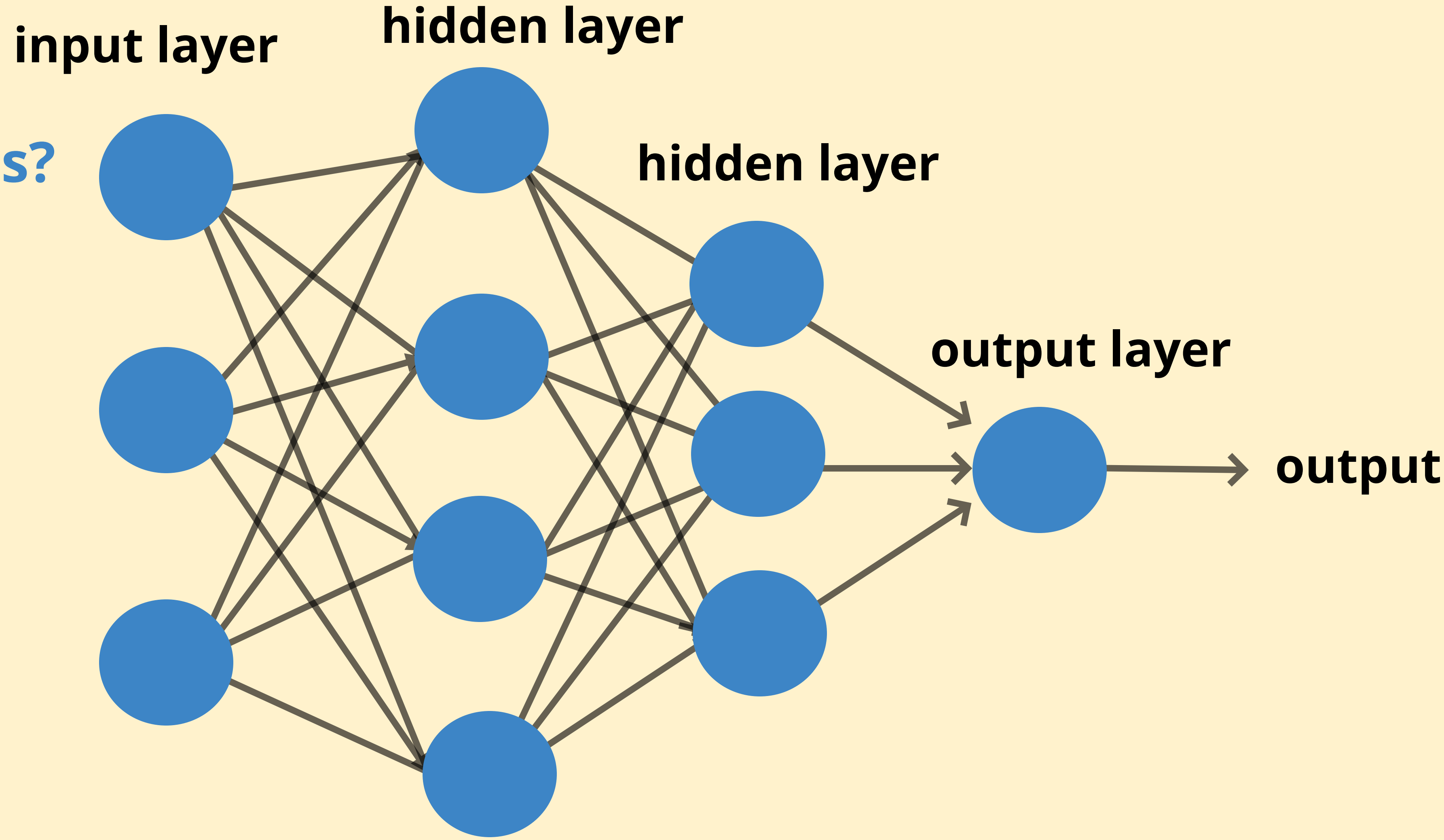
multilayer perceptron

what we are doing is exactly a series of matrix multiplictions.

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ b_1 & b_2 & b_3 & \cdots & b_n \\ c_1 & c_2 & c_3 & \cdots & c_n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_1 & m_2 & m_3 & \cdots & m_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} (a_1x_1) + (a_2x_2) + (a_3x_3) + \ldots + (a_nx_n) \\ (b_1x_1) + (b_2x_2) + (b_3x_3) + \ldots + (b_nx_n) \\ (c_1x_1) + (c_2x_2) + (c_3x_3) + \ldots + (c_nx_n) \\ \cdots \\ (m_1x_1) + (m_2x_2) + (m_3x_3) + \ldots + (m_nx_n) \end{bmatrix}$$

EXERCISE

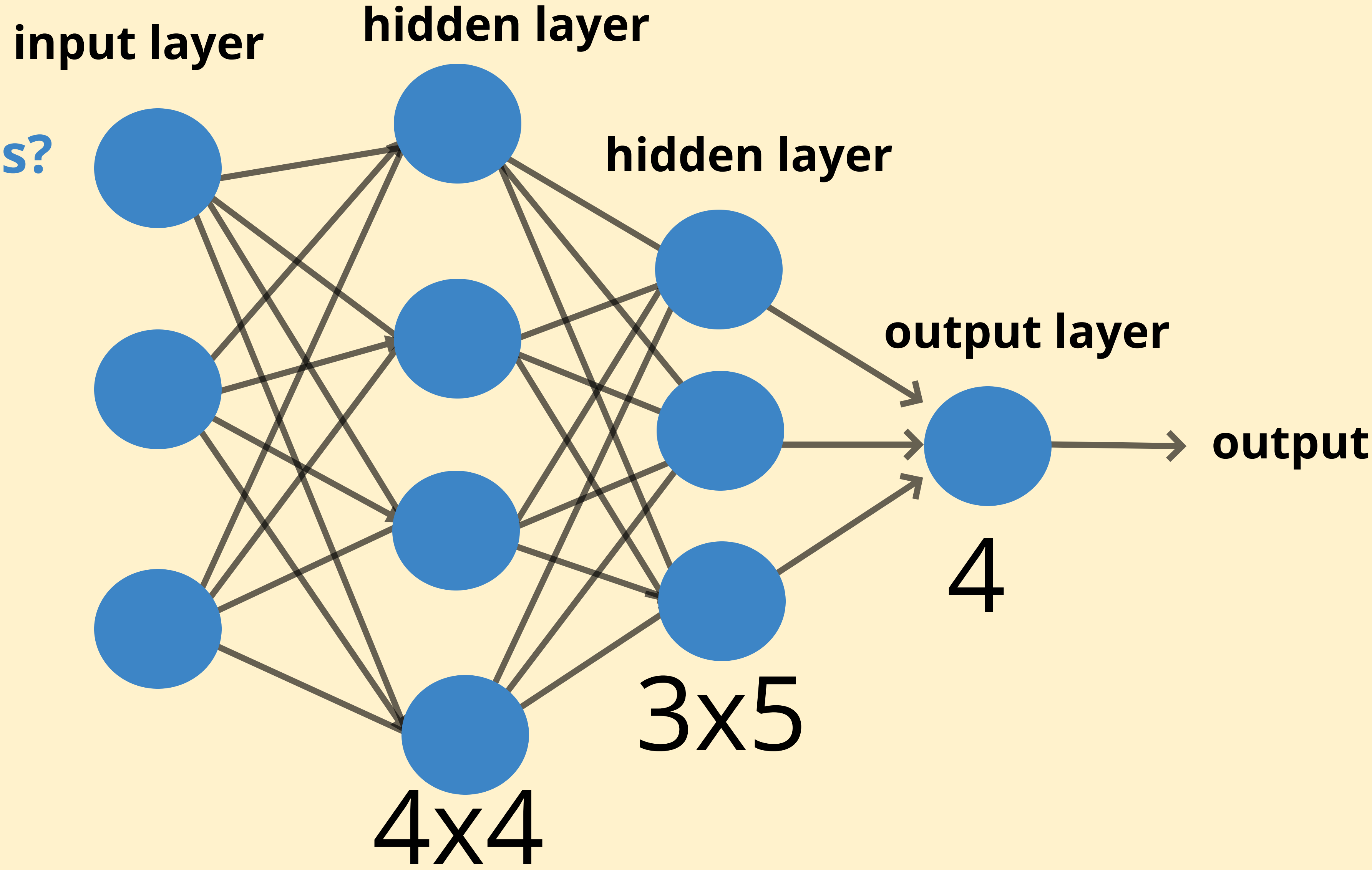
how many parameters?



EXERCISE

how many parameters?

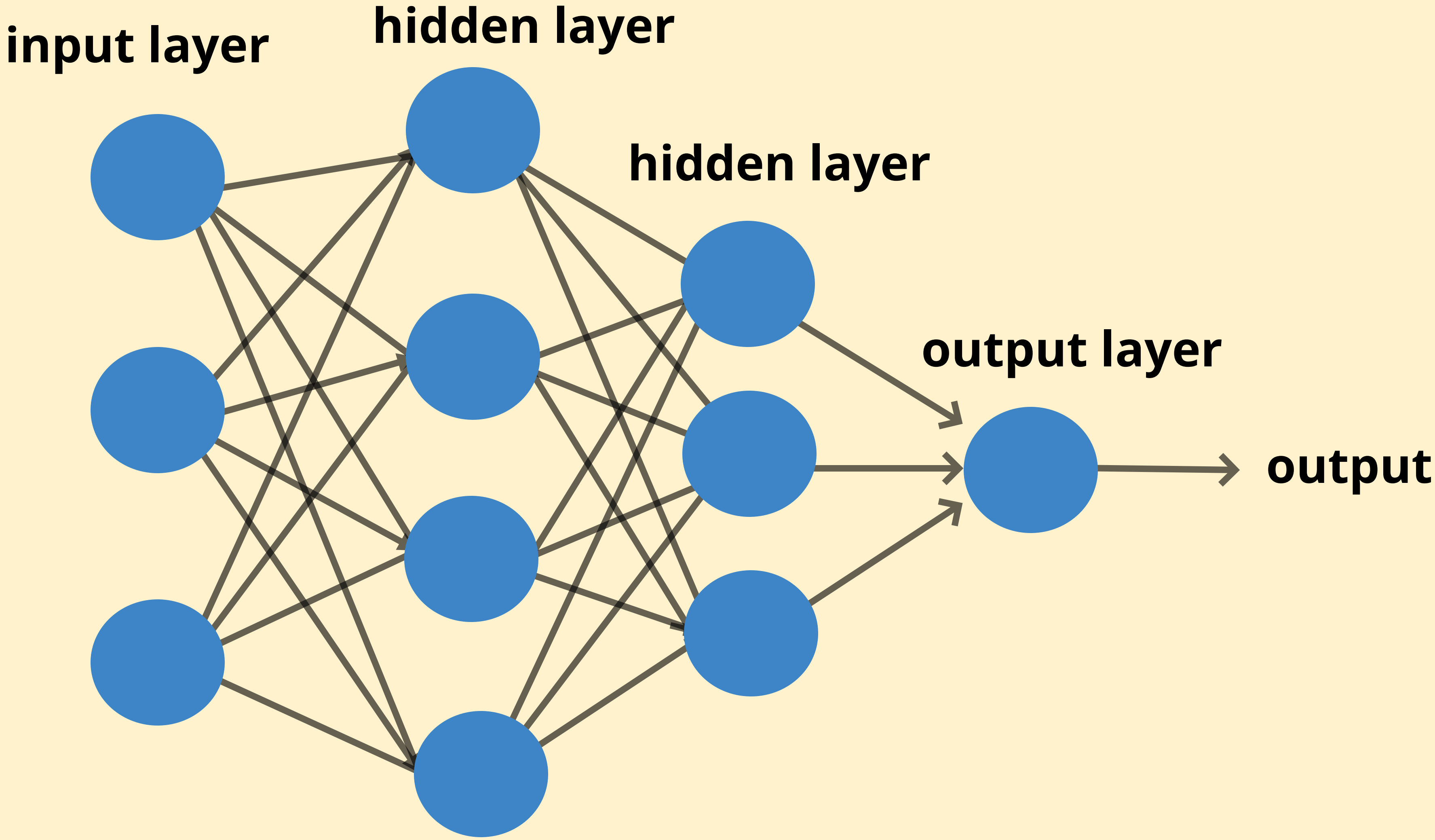
35



EXERCISE

how many hyperparameters?

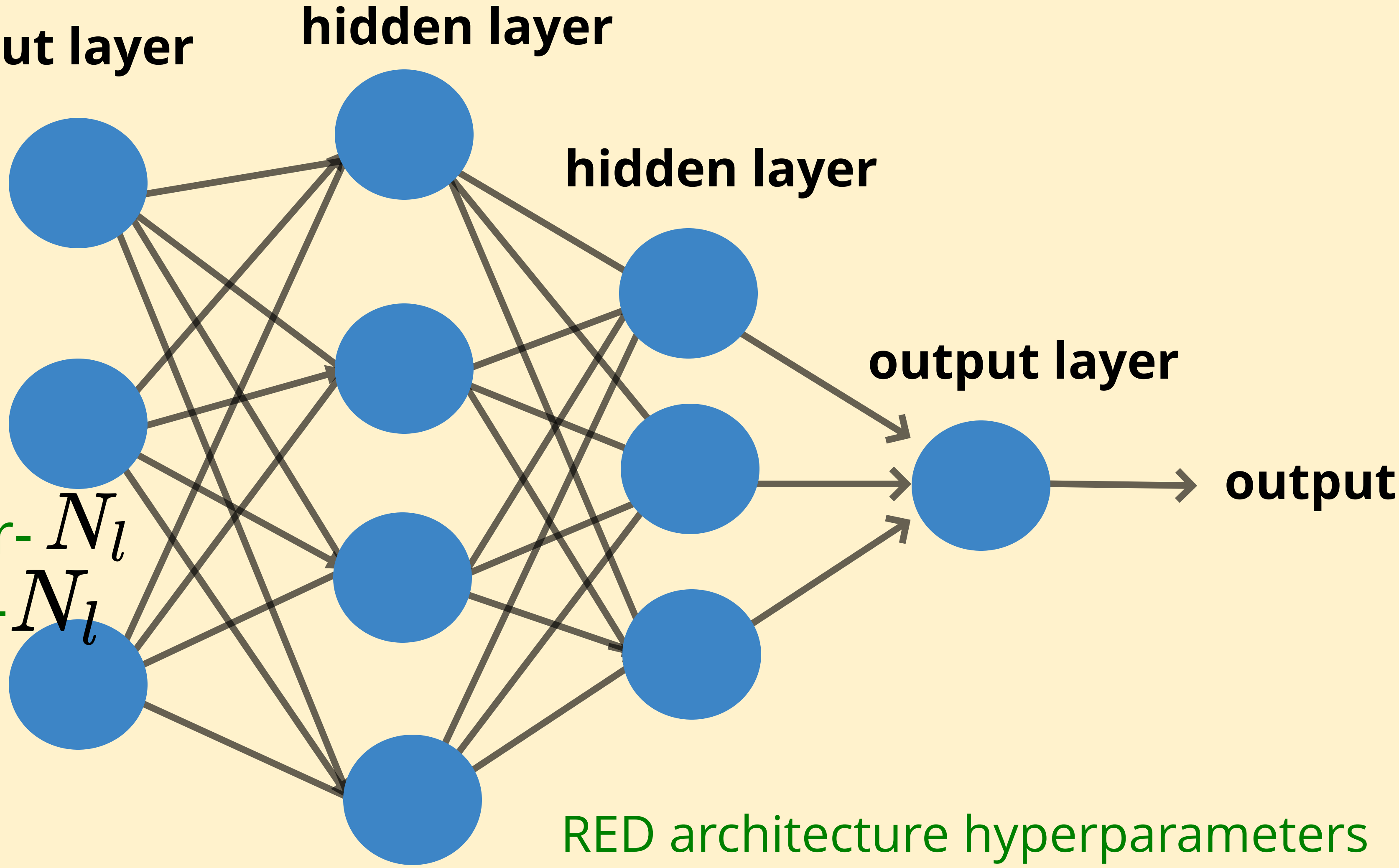
http://bit.ly/DSPSnnhp



EXERCISE

how many hyperparameters?

http://bit.ly/DSPSnnhp



1. number of layers- *1*
2. number of neurons/layer- N_l
3. activation function/layer- N_l
4. layer connectivity- $N_l^{??}$
5. optimization metric - *1*
6. optimization method - *1*
7. parameters in optimization- *M*

RED architecture hyperparameters

RED training hyperparameters

deepneural
networks

6

Punch Line

Deep Neural Net are not some fancy-pants methods, they are just linear models with a bunch of parameters

Black Box?

Because they have many parameters they are difficult to "interpret" (no easy feature extraction)

tha is ok becayse they are prediction machines

deep neural net

1986: Deep Neural Nets

Fully connected: all nodes go to all nodes of the next layer.

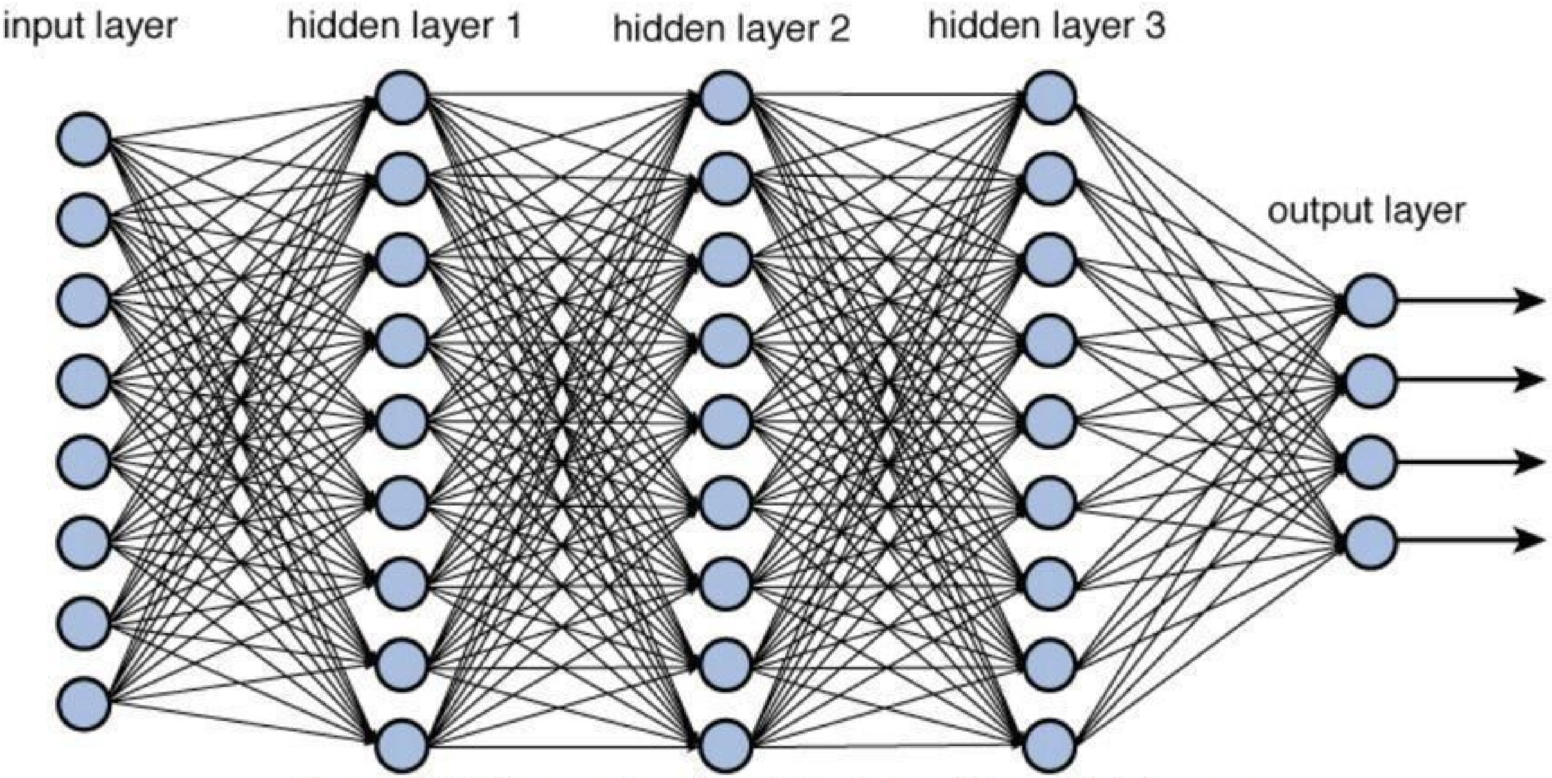
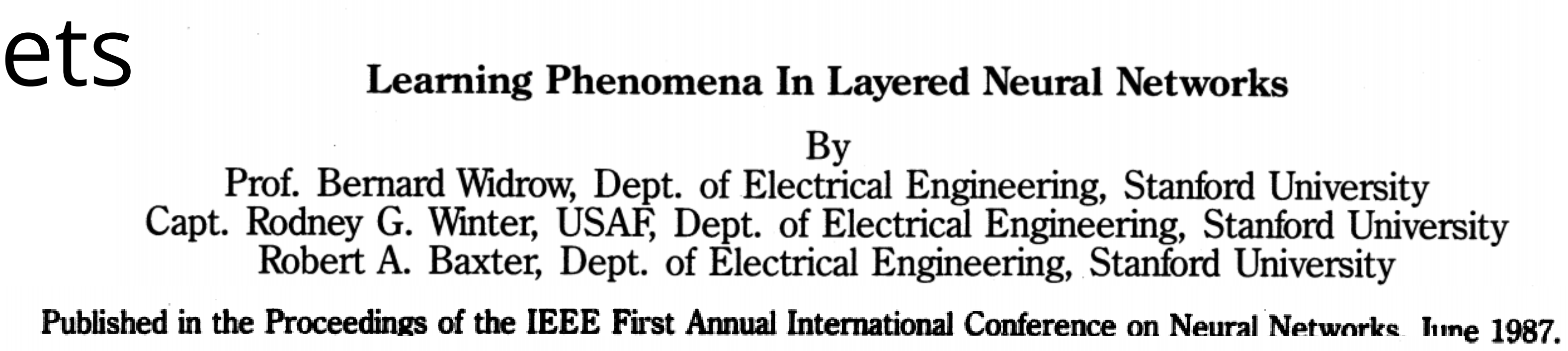


Figure 12.2 Deep network architecture with multiple layers.

deep neural net

1987: Deep Neural Nets learning procedure

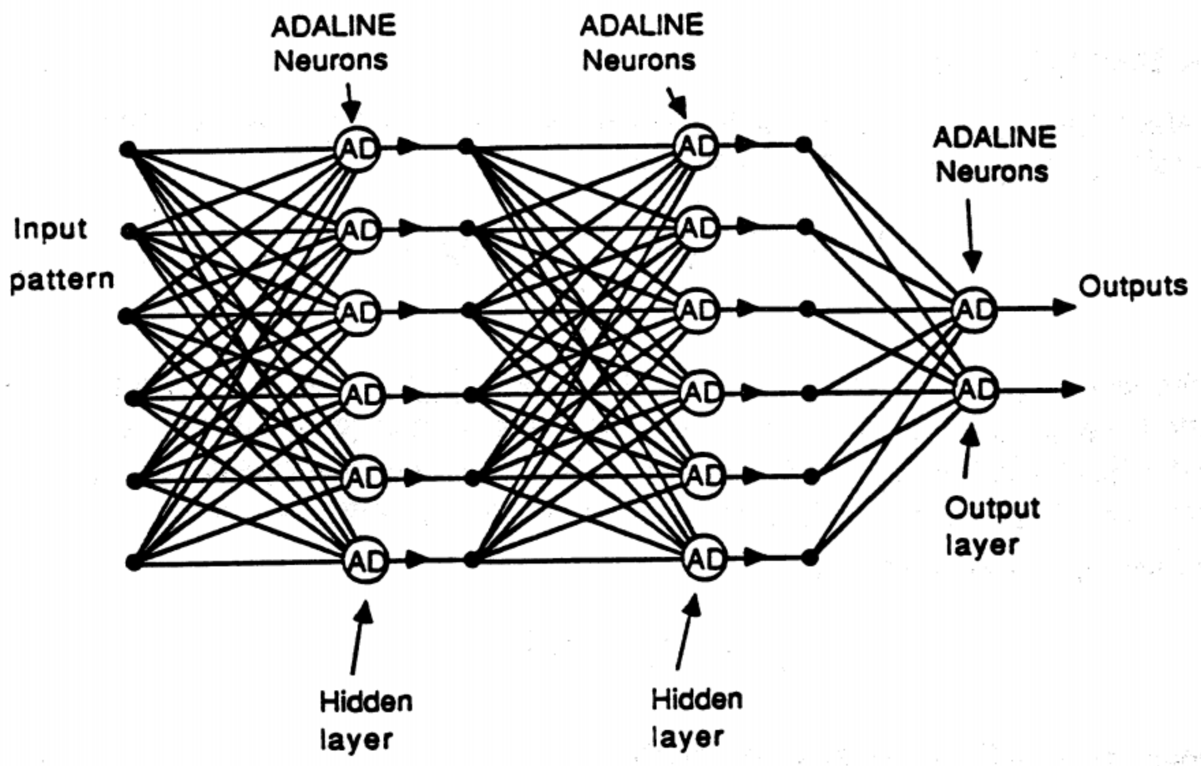


Learning Phenomena In Layered Neural Networks

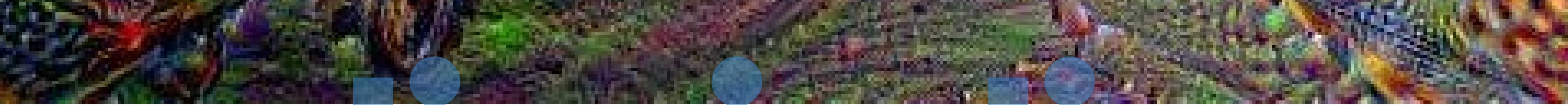
By
Prof. Bernard Widrow, Dept. of Electrical Engineering, Stanford University
Capt. Rodney G. Winter, USAF, Dept. of Electrical Engineering, Stanford University
Robert A. Baxter, Dept. of Electrical Engineering, Stanford University

Published in the Proceedings of the IEEE First Annual International Conference on Neural Networks. June 1987.

How do you propagate Widrow-Hoff ?



deep dreams



optimization schemes

7

Deep Learning

backpropagation

8

Deep Learning

preprocessing (minibatch)

9

1. Architecture components: perceptron, layers, activation function
2. Optimization
3. Single layer NN
4. Deep NN

key concepts

Partition clustering:

Hard: K-means *O(KdN)* , needs to decide the number of clusters, non deterministic simple efficient implementation but the need to select the number of clusters is a significant flaw

Soft: Expectation Maximization *O(KdNp)* , needs to decide the number of clusters, need to decide a likelihood function (parametric), non deterministic

Hierarchical:

Divisive: Exhaustive $O(2^N);\ O(N^2)$ at least non deterministic

Agglomerative: $O(N^2d + N^3)$**,** deterministic, greedy. Can be run through and explore the best stopping point. Does not require to choose the number of clusters a priori

Density based

DBSCAN: Density based clustering method that can identify outliers, which means it can be used in the presence of noise. Complexity $O(N^2)$. Most common (cited) clustering method in the natural sciences.

encoding categorical variables:

variables have to be encoded as numbers for computers to understand them. You can encode categorical variables with integers or floating point but you implicitly impart an order. The standard is to ***one-hot-encode*** which means creating a binary (True/False) feature (column) for each category of a categorical variables but this *increases the feature space and generated covariance.*

model diagnostics for classifiers: Fraction of True Positives and False Positives are the metrics to evaluate classifiers. Combinations of those numbers include Accuracy (TP/ (TP+FP)), Precision (TP/(TP+FN)), Recall ((TP+TN)/(TP+TN+FP+FN)).

ROC curve: (TP vs FP) is a holistic metric of a model. It can be used to guide the choice of hyperparameters to find the "sweet spot" for your problem

Neural Network and Deep Learning

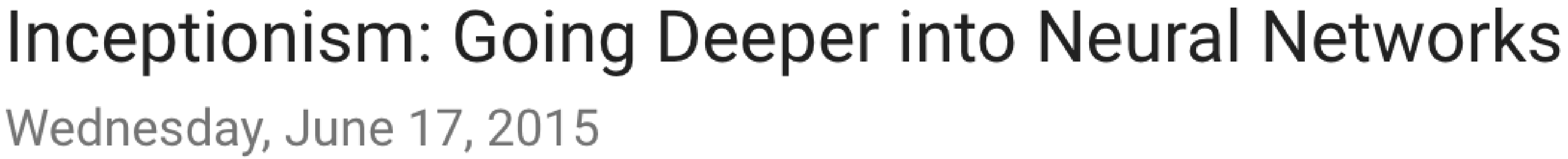
an excellent and free book on NN and DL

http://neuralnetworksanddeeplearning.com/index.html

History of NN

https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html

resources



Inceptionism: Going Deeper into Neural Networks

Wednesday, June 17, 2015

https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

reading