CHA 2555
Artificial Intelligence

PREPROCESSING, PERFORMANCE MEASURES AND INTRO TO NN

DR. EMMANUEL PAPADAKIS

University of HUDDERSFIELD

Outline

Data preparation

Classification performance metrics

- Accuracy
- Confusion matrix

Regression performance metrics

- Measuring errors
- Coefficient of determination

Limitations of traditional Machine Learning

Neural Networks

Data preparation

ID	Sweetness	Colour	Apple
1	**-20**	Green	Yes
2	5	Red	Yes
2	5	Red	Yes
3	5	Red	**No**
4	Yes	**null**	No

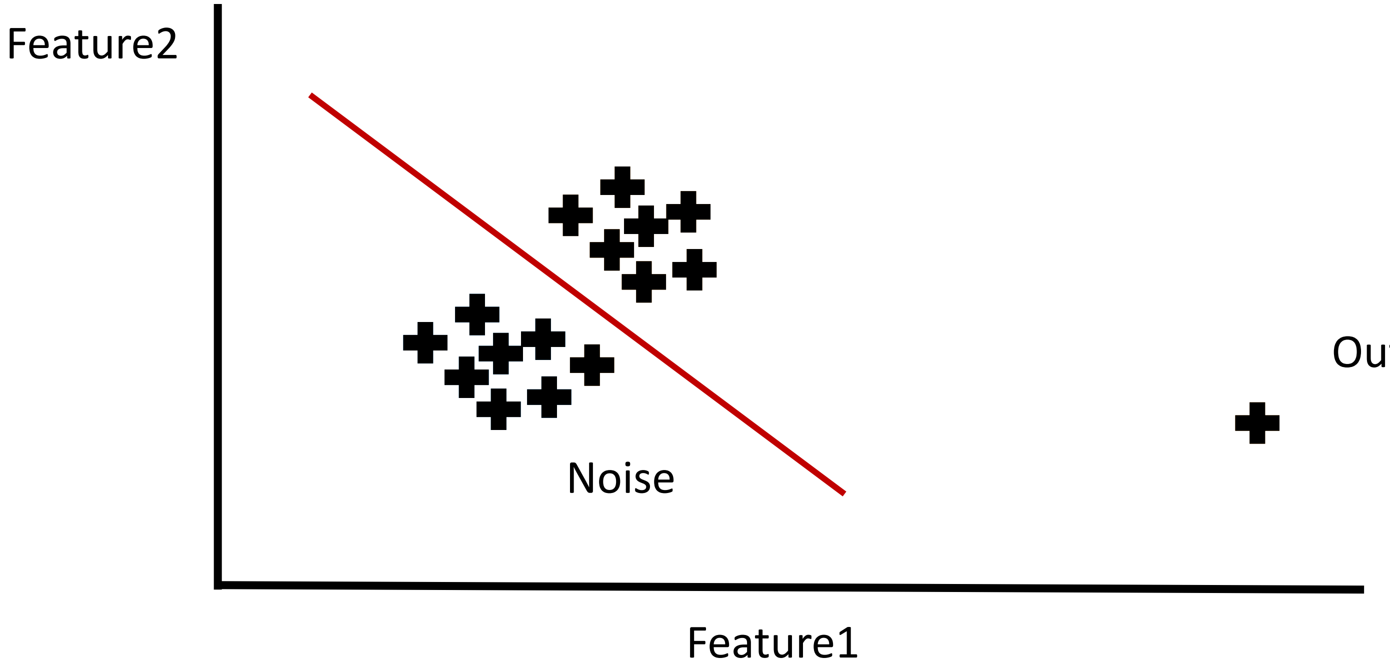
Data-driven learning is **sensitive** to data quality

- “bad” training data => “bad” predictions

Challenges of Training Data

- Insufficient quantity – More data is always better
- Nonrepresentative (e.g. missing observations) – What about Green fruits?
- Irrelevant data: garbage in, garbage out
  - Feature selection – most important & relevant data
  - Feature extraction – dimensionality reduction (merge the information of multiple features)
- Poor quality – errors in the data
  - Missing data, type errors, outliers and noise

Noise Vs Outliers



Common practices to preprocess data

Drop rows with missing values

Drop dublicates (e.g. rows with the same ID)

Check features for consistency

- Numerical features must contain **ONLY** numbers
- Categorical features must have consistent values

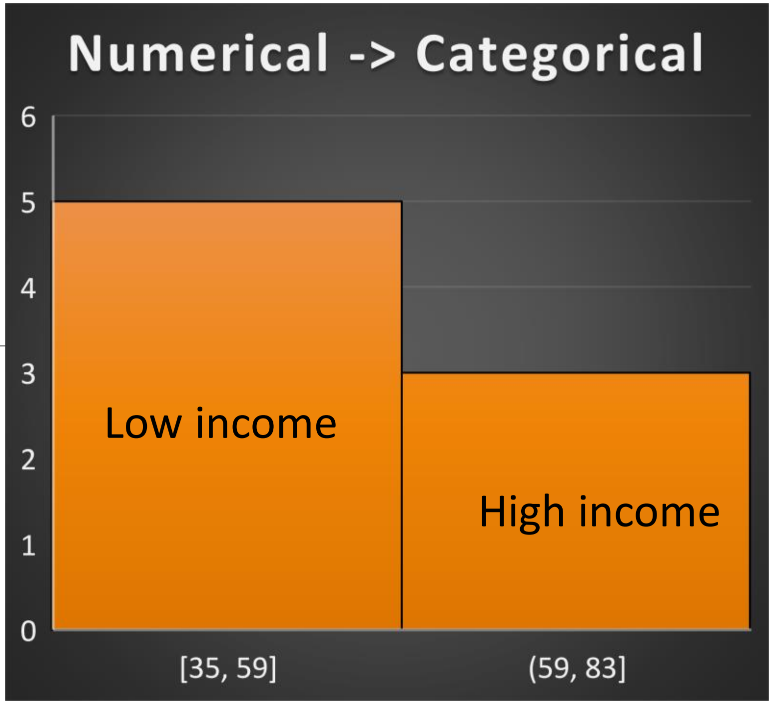
Feature engineering

- Deal with outliers
- Normalization or Standardization of numerical features
- Feature extraction by processing existing features
- Remove features with near-zero predictive power
  - Variable independency

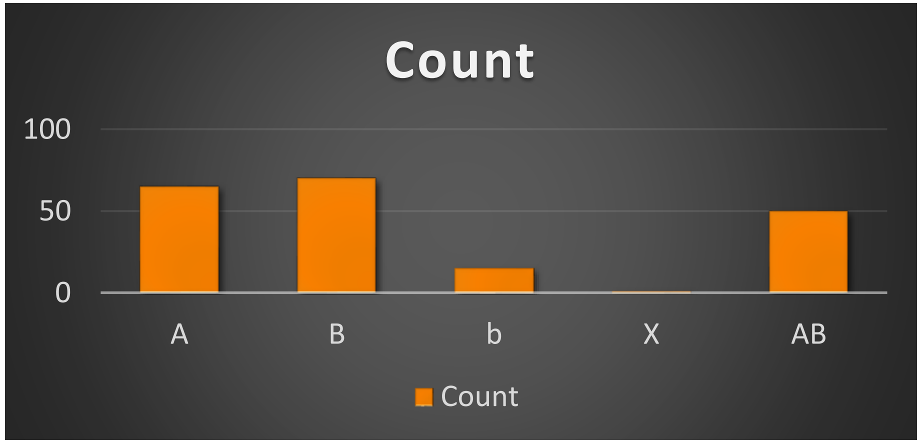
Cleaning data

40000

ID	Name	BT Full	Blood Type	Rhesus	Country	Income	Device	Class
1	John	Alpha	A	Yes	US	35000	Linux	Yes
2	Mary	Alpha	A	Yes	US	67000	iOS	No
3	Alice	Beta	B	No	US	68500	Android OS	Yes
4	Bob	Beta	b	No	US	40K	MacOS	No
5	Daniel	Zero	X	No	US	70000	Windows10	No
6	Stacy	AlphaBet	AB	Yes	US	45000	Android 12	No
7	~~Helen~~	~~H~~	~~O~~	~~Yes~~	~~US~~	~~1000000~~	~~WindowsXP~~	~~Yes~~



Blood types: A, B, AB, O (+/-)



Country has zero predictive power

Blood Type = merge Blood Type and Rhesus

Device has great variance that creates noise

- Cluster based on OS family

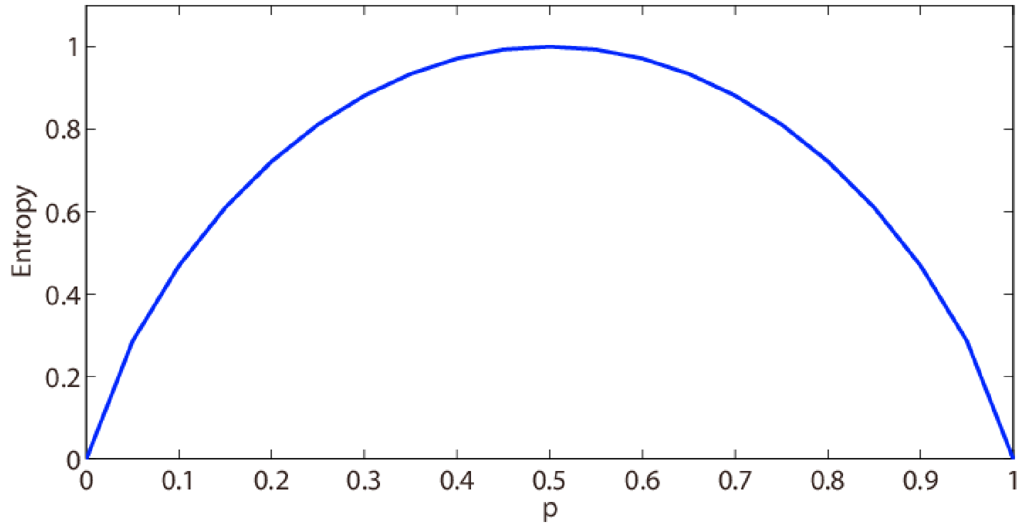
BT Full and Blood Type show great dependency

- One of them must be dropped

Feature Importance using information gain

- Entropy of a variable = measure of uncertainty of the possible outcomes
- Information gain = amount of information obtained about a variable given another variable

Size	Number of Leaves	Green Index	Age
1	20	15	2
1.5	40	30	1



Information Gain:
- 0, the variables are independent
- > 0, level of correlation

Important features: Features with high predictive power

Feature Selection in Weka

1. Calculate the Information Gain for each Feature against the Class

2. Normalize every value by scaling it between 0 and 1 $x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$

3. Select a threshold (> 0.1) and drop all the features below

Viewer

Relation: pima_diabetes

No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: **class** Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive

Feature	Information Gain	Normalized
plas	0.1901	1
mass	0.0749	0.34
age	0.0725	0.33
insu	0.0595	0.25
skin	0.0443	0.17
preg	0.0392	0.14
pedi	0.0208	0.03
pres	0.014	0

Classification performance metrics

ID	preg	plas	pres	skin	insu	mass	pedi	age	actual value	predicted value
1	6	148	72	35	0	33.6	0.627	50	positive	positive
2	1	85	66	29	0	26.6	0.351	31	negative	negative
3	8	183	64	0	0	23.3	0.672	32	positive	negative
4	1	89	66	23	94	28.1	0.167	21	negative	negative
5	0	137	40	35	168	43.1	2.288	33	positive	positive
6	5	116	74	0	0	25.6	0.201	30	negative	positive
7	3	78	50	32	88	31	0.248	26	positive	positive
8	10	115	0	0	0	35.3	0.134	29	negative	negative
9	2	197	70	45	543	30.5	0.158	53	positive	positive
10	8	125	96	0	0	0	0.232	54	positive	positive

How can we measure the performance of a classifier?

- Compare the predicted data with the test data
- Measure the
  - Precision
  - Recall
  - Sensitivity
  - Specificity
  - Accuracy

Performance measures

Confusion Matrix

		Actual Value	
		Positive	Negative
Predicted Value	Positive	**T**rue **P**ositives	**F**alse **P**ositives
	Negative	**F**alse **N**egatives	**T**rue **N**egatives

$$Sensitivity/Recall = \frac{TP}{TP + FN}$$

True Positive Rate

$$Specificity = \frac{TN}{TN + FP}$$

True Negative Rate

$$Precision = \frac{TP}{TP + FP}$$

Class agreement on positives

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Overall effectiveness

$$Error\ rate = \frac{FP + FN}{TP + FP + TN + FN}$$

Classification Error

Performance Example

		Actual Value	
		Diabetes	No diabetes
Predicted Value	Diabetes	**160**	**93**
	No diabetes	**108**	**407**

$$Sensitivity\ or\ Recall = \frac{TP}{TP + FN} = \frac{160}{160 + 108} = 0.59$$

$$Specificity = \frac{TN}{TN + FP} = \frac{407}{407 + 93} = 0.814$$

$$Precision = \frac{TP}{TP + FP} = \frac{160}{160 + 93} = 0.63$$

$$Accuracy = \frac{160 + 407}{160 + 93 + 108 + 407} = 0.73$$

$$Error\ rate = \frac{93 + 108}{160 + 93 + 108 + 407} = 0.26$$

Accuracy Vs Recall Vs Precision Vs F1 score

Accuracy is a simple evaluation metric

- Estimate how well the model operates against all observations
- Usefull to quantify and compare the effectiveness of different classifiers

Precision and Recall depend on the problem

- High precision : when we want to eliminate false positives
  - Examples?
  - Internet reccomendations, investments – it is better to correctly capture a subset of users' preferences instead of showing irrelevant ads
- High recall : when we want to eliminate false negatives
  - Examples?
  - Diagnosis of any disease, risk assessment – it is better to dismiss falsely identified cancer rather than ignore tumors

Why not maximize both?

- F1 score = $2 * \frac{Precision * Recall}{Precision+Recall}$ - higher score resembles a better balance between precision and recall

Performance metrics for Regression

Predictions occur over continuous values ➔ impossible to have clean match of values

- Residual: Estimate how close are the predicted with the actual values

Month	Inflation (%)	Predicted (%)	Residual (%)
1	0.5	1.8	-1.3
2	2.7	1.8	0.9
3	2.4	1.8	0.6
4	2.2	1.9	0.3
5	1.9	1.9	0.0
6	1.6	1.9	-0.3
7	1.7	2.0	-0.3
8	1.9	2.0	-0.1
9	2.2	2.0	0.2
10	2.0	2.1	-0.1
11	2.2	2.1	0.1
12	2.1	2.2	0.0

Source: Displayr

Residual = actual value – predicted value

Model performance based on:

- Mean Absolute Error
- Mean Squared Error
- Root Mean Squared Error
- Coefficient of determination R^2

Measuring … errors

Mean absolute error:

Actual Value

Predicted Value

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|$$

Total number of data points

Actual	Predicted
3	2
2	5
7	8
5	5

$$MAE = \frac{1}{4}\left(|3-2| + |2-5| + |7-8| + |5-5|\right) = \frac{1}{4} * (1+3+1) = \frac{5}{4} = 1.25$$

The bigger the MAE, the more critical the error is.

- It is robust to outliers.
- Comparing the MAE of different models requires the values to be on the same scale

Measuring … errors

Mean Squared error:

Predicted Value

Actual Value

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Total number of data points

Actual	Predicted
3	2
2	5
7	8
5	5

$$MSE = \frac{1}{4}\left((3-2)^2 + (2-5)^2 + (7-8)^2 + (5-5)^2\right) = \frac{1}{4} * (1 + 9 + 1) = \frac{12}{4} = 3$$

- It punishes outliers more – which sometime is bad… why?
  - A model with many small errors Vs A model with a single outlier
- Lower value means better regression model

Measuring … errors

Root Mean Squared error:

Actual Value

Predicted Value

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}$$

Total number of data points

Actual	Predicted
3	2
2	5
7	8
5	5

$$RMSE = \sqrt{MSE} = \sqrt{3}$$

- It punishes outliers more but also normalizes the final value to make comparison easier
- A Higher RMSE indicates that there are large deviations between the predicted and actual value

Coefficient of determination R^2

R^2 explains to what extent the variance of one variable explains the variance of a second value

- Explain how well changes in dependent variables are explained by the independent variables

Estimate how close are the data points to the fitted regression algorithm

- Estimate the ration of the sum of squares and the total sum of squares

$$R^2 = 1 - \frac{SSE}{SST}$$

$$SSE = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Actual Value

Predicted Value

$$SST = \sum_{i=1}^{m} (y_i - \bar{y})^2$$

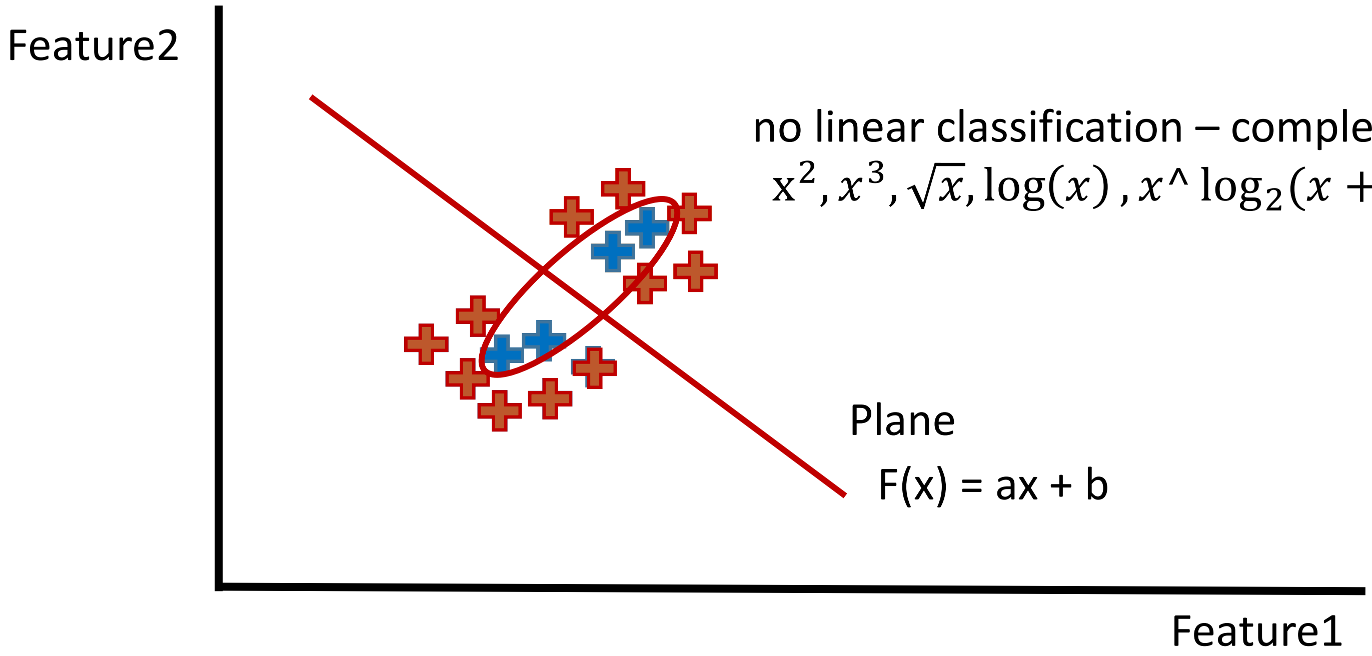
Number of observations

Mean predicted Value

R^2 ranges from 0 to 1.

- If 0 the model does not perform better than a random model
- Higher the better
- Easy to compare models with different units

Problems with linear classification



no linear classification – complex models!

$\mathrm{x}^2, x^3, \sqrt{x}, \log(x), x\hat{}\ \log_2(x + 2x^2)\ \ldots$, or a combination of distributions

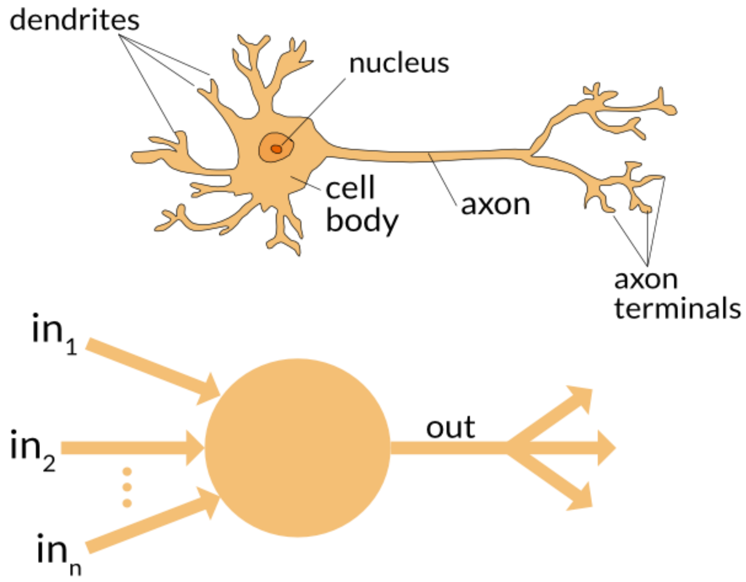
Computationally expensive if we have many features and sometimes impossible with simple models.

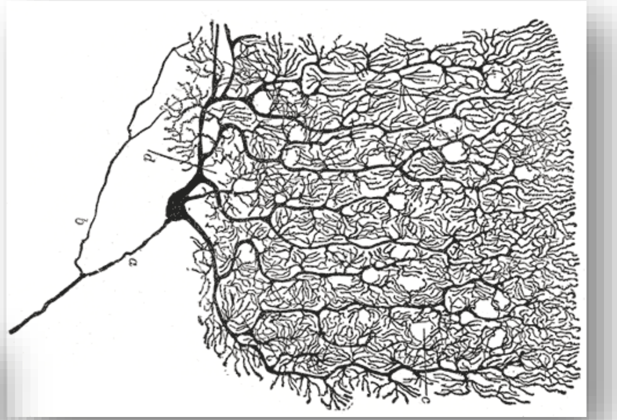
What if we use a combination of simple models to approximate complex patterns of data?

Artificial Neural networks – main idea

The idea originates from 1943

- A computational model using propositional logic to represent how simple structures can work together to perform complex tasks – mathematicians and neurophysiologists
- Similar to how combined neurons in our brain achieve complicated tasks





Source: Wikimedia commons

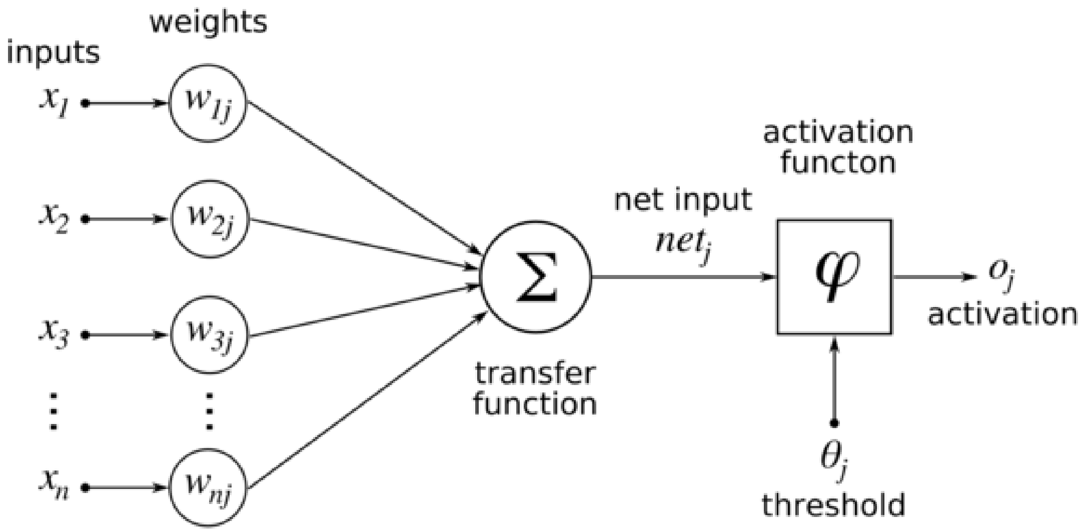
Artificial Neural networks – first steps

The belief of computationally modelling brain functions was way too ambitious

- Insufficient computational power and data scarcity
- Now we may employ GPUs for fast computations and Data are widely available!

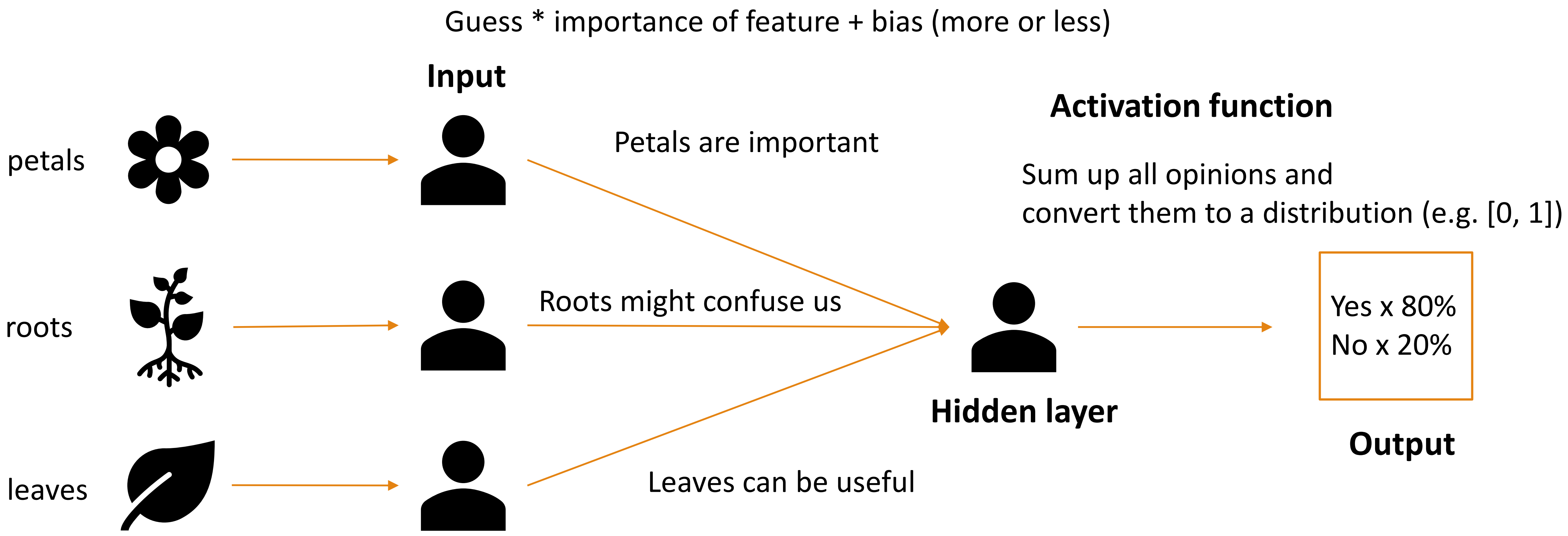
Perceptron – Frank Rosenblatt, 1957

- Simplest ANN architecture

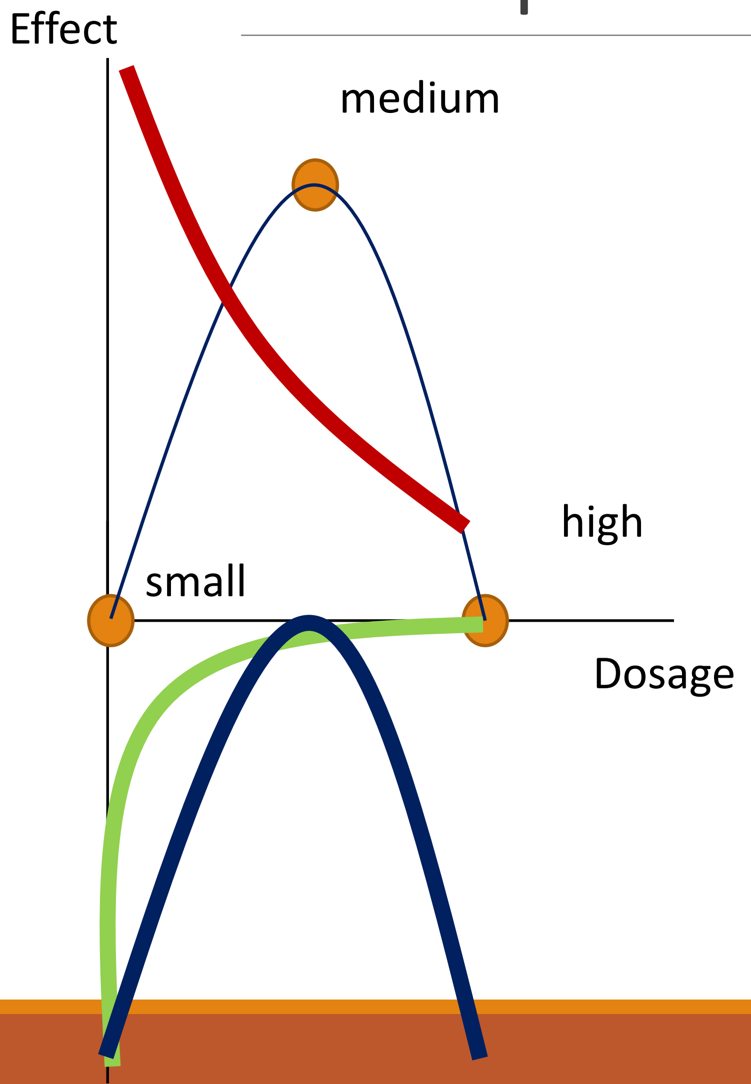


Mathematical model of a Simple ANN

4 students cooperate to learn how to identify daisies

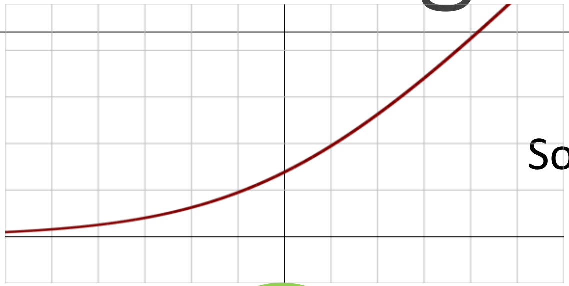


Example: how effective is a drug?



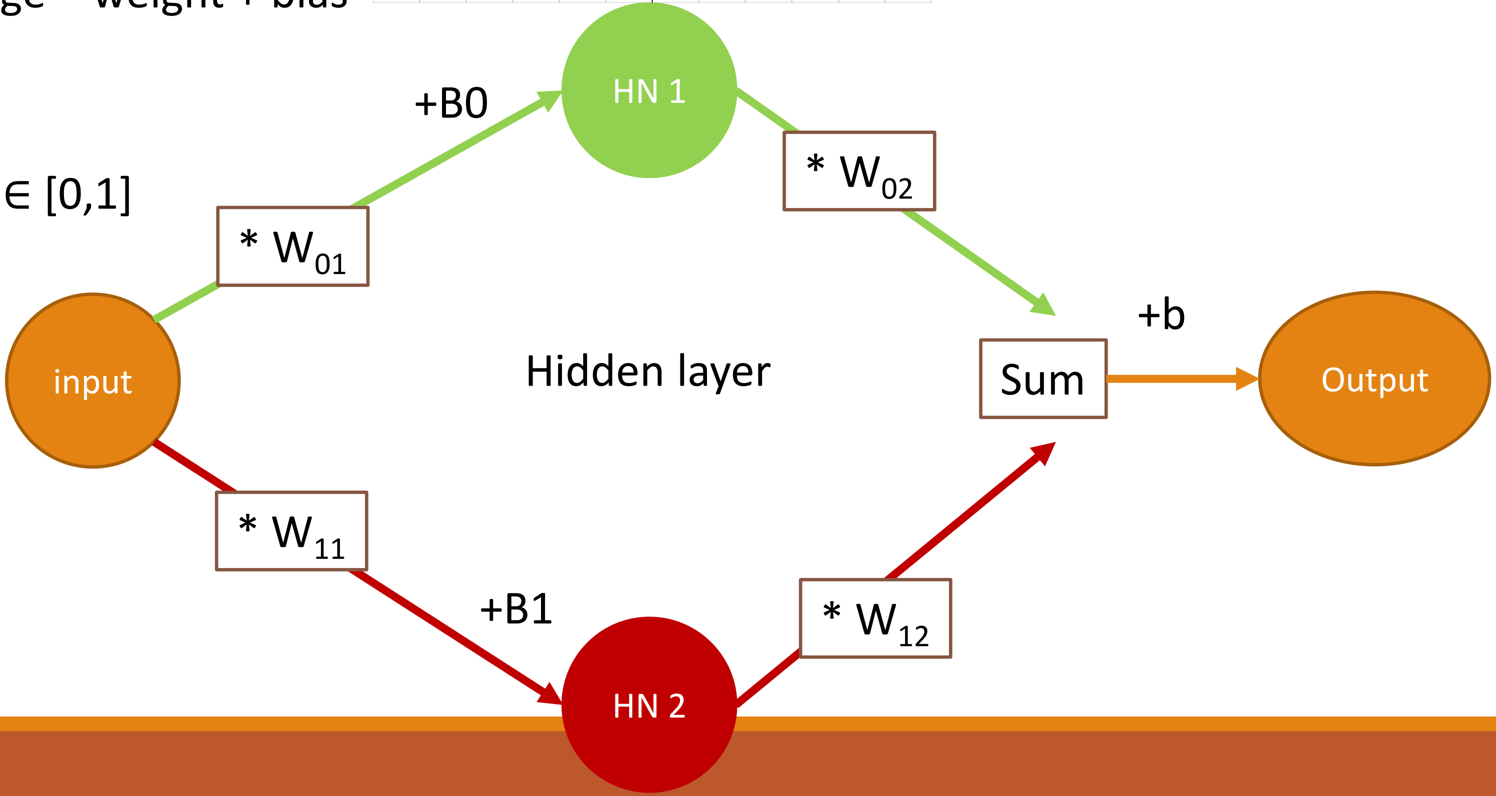
y = x * slope + intercept

effect = dosage * weight + bias



Soft-plus – activation function

Dosage x ∈ [0,1]

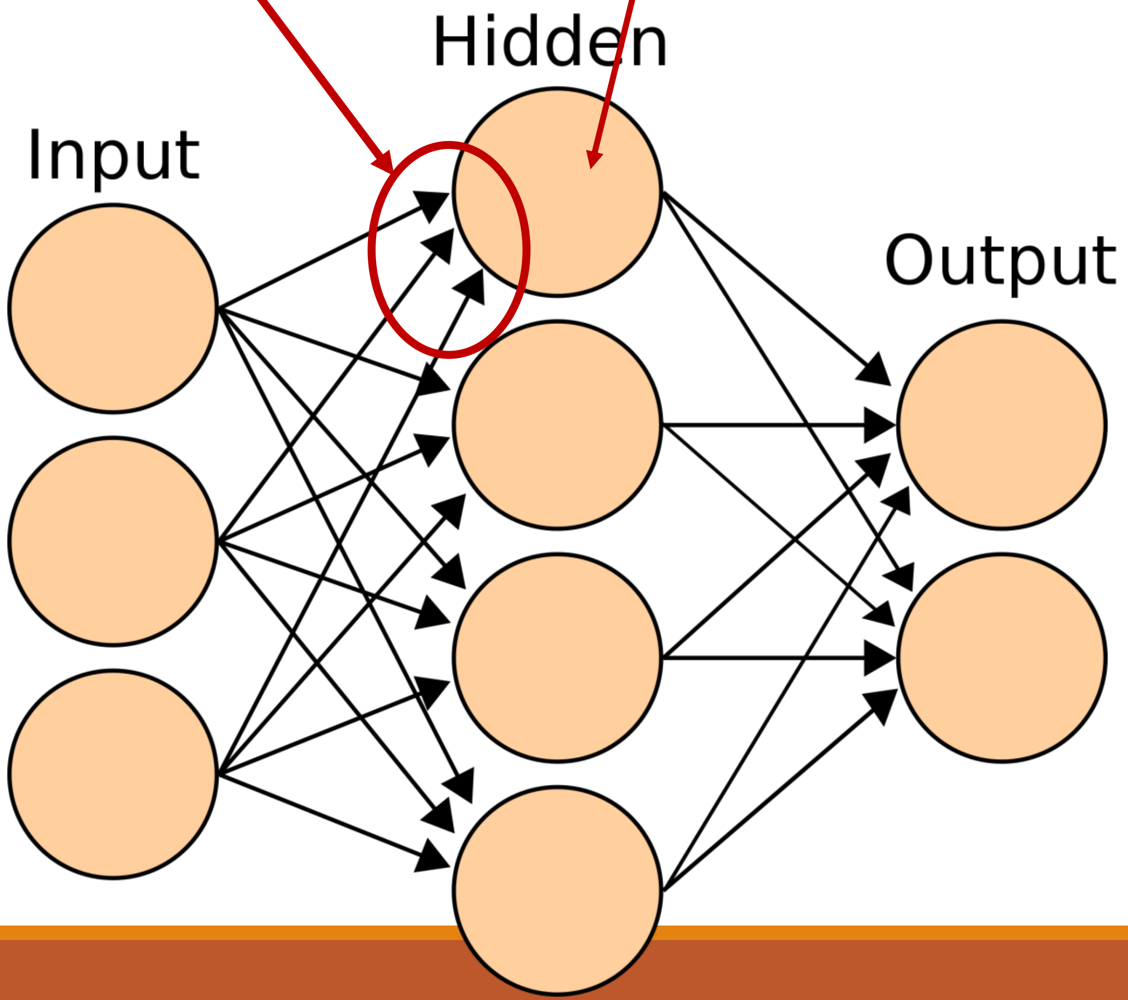


Preactivation:
Linear transformation of inputs
(weighted sum of inputs and biases)

Activation:
Non-linear transformation

Of course, that was a toy example!

Forward propagation → Inference



Naming conventions:

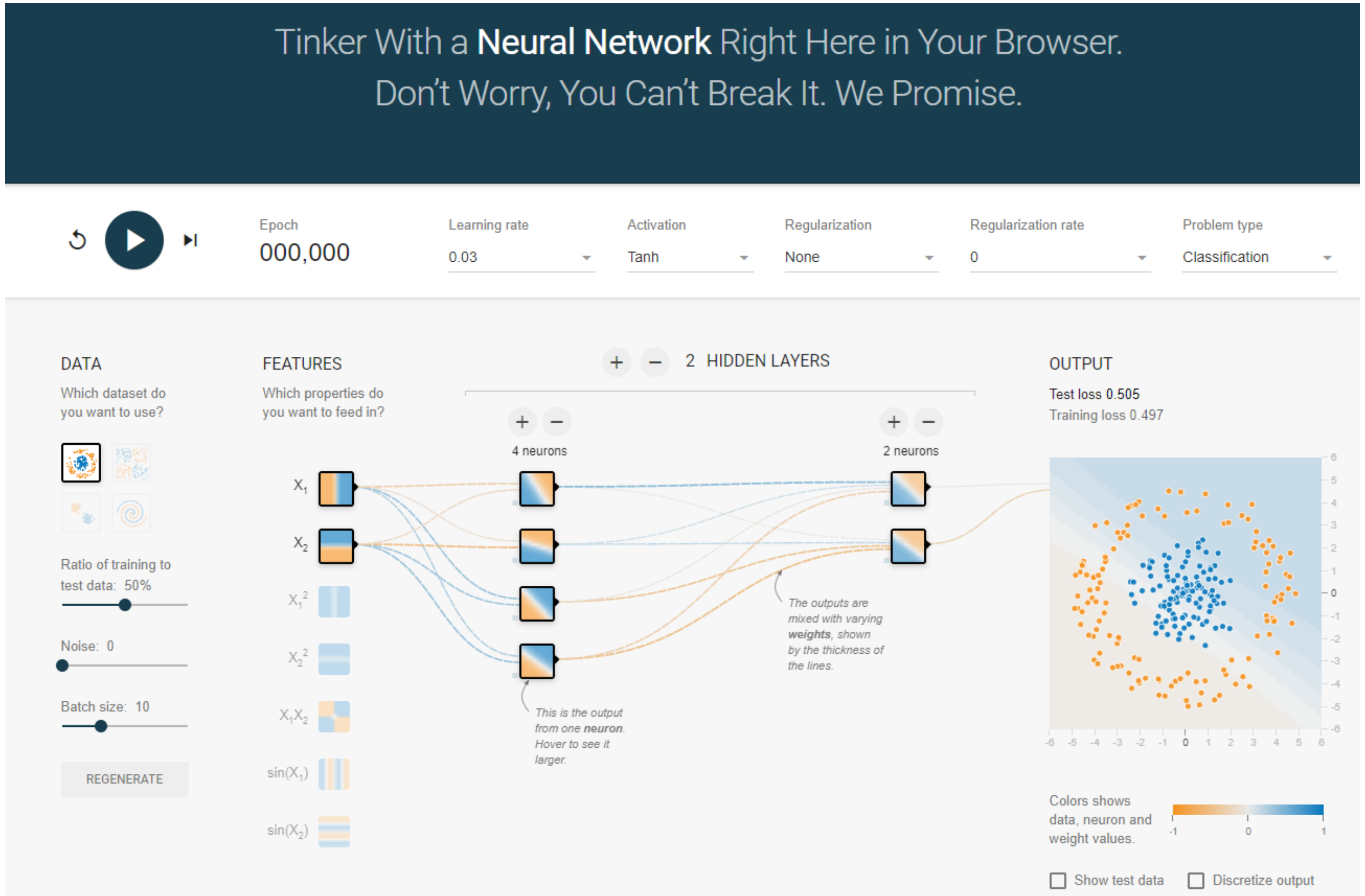
input layer is ignored, 2-layer network

More that 2 layers => Deep Neural Network

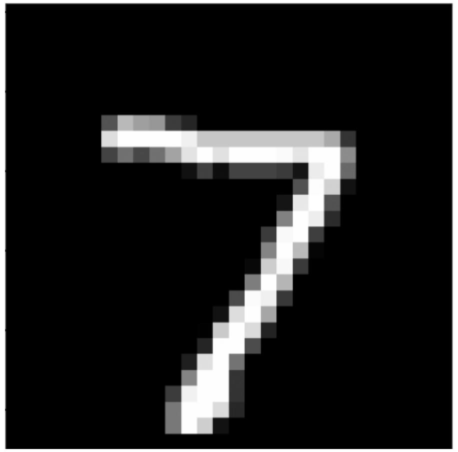
ANN with at least one hidden layer can represent any function

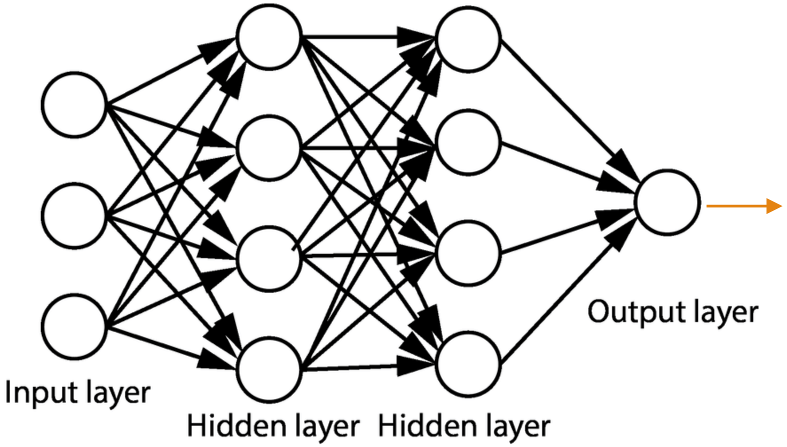
Cybenko, 1989

Fun with ANNs



Example Image recognition



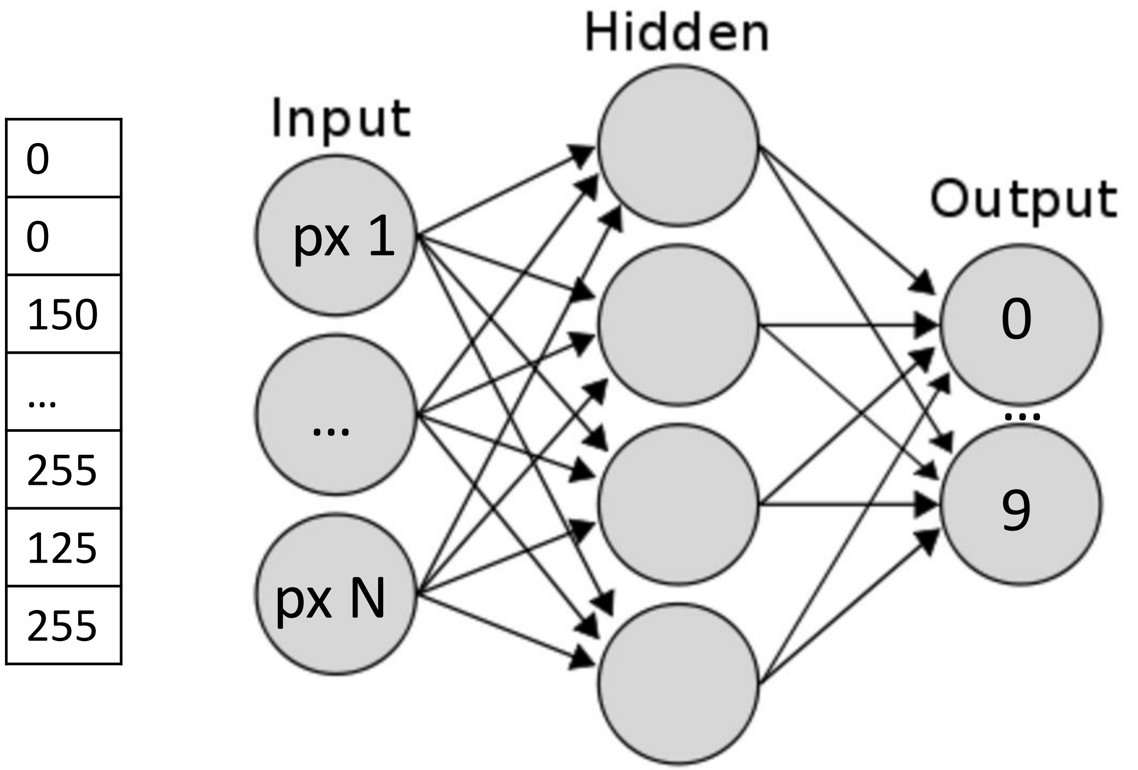


7 - 98%

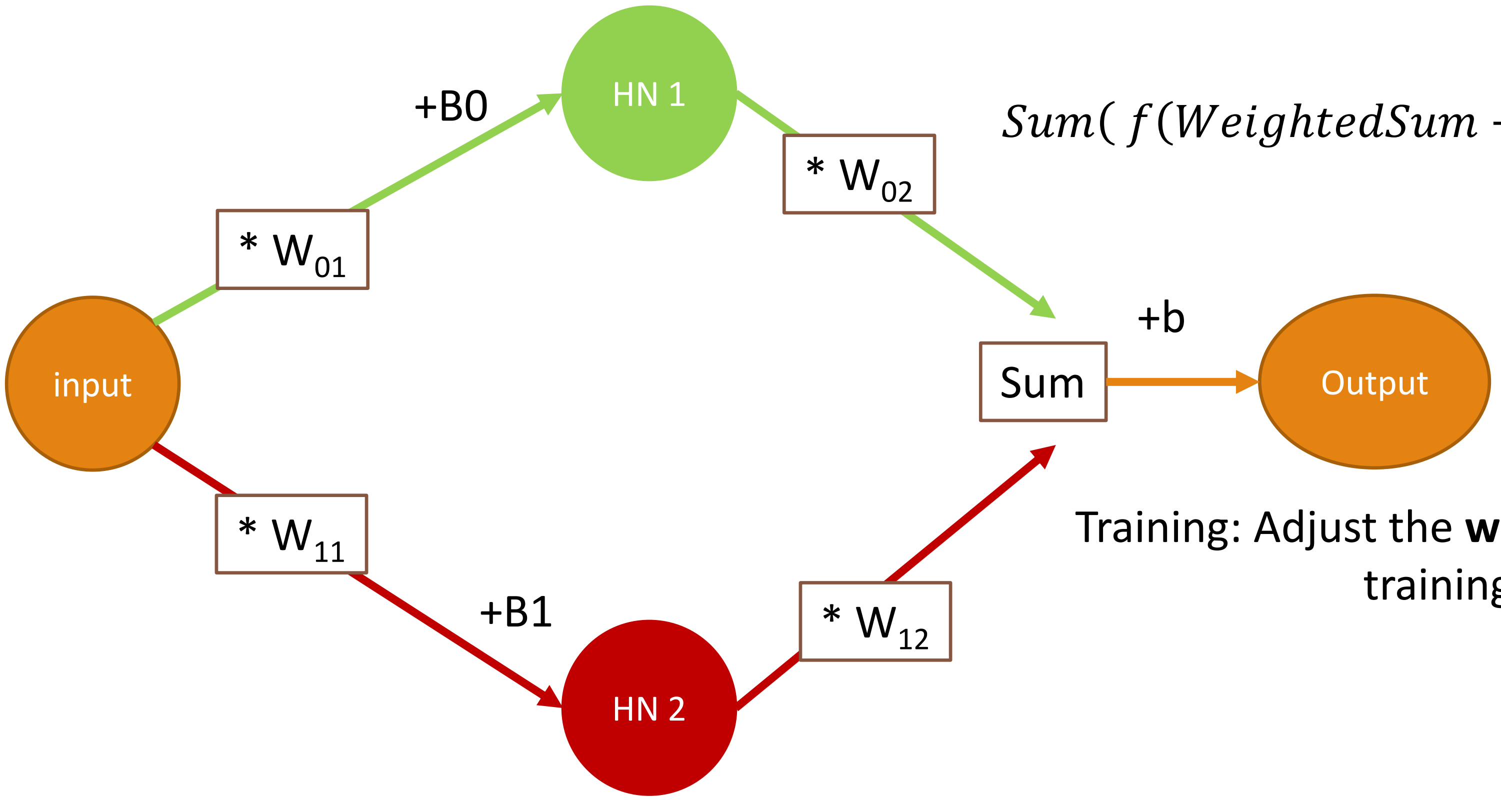
Example Image recognition

An image is a 2-dimensional array of pixels

- e.g. 28 x 28 = 784 inputs, each representing the value of the color [0 , 255]



How do ANNs learn?



$$Sum(\, f(WeightedSum + biases)\,) + Bias$$

Training: Adjust the **weights** and **biases** based on the training data.

ANN training

Backward propagation

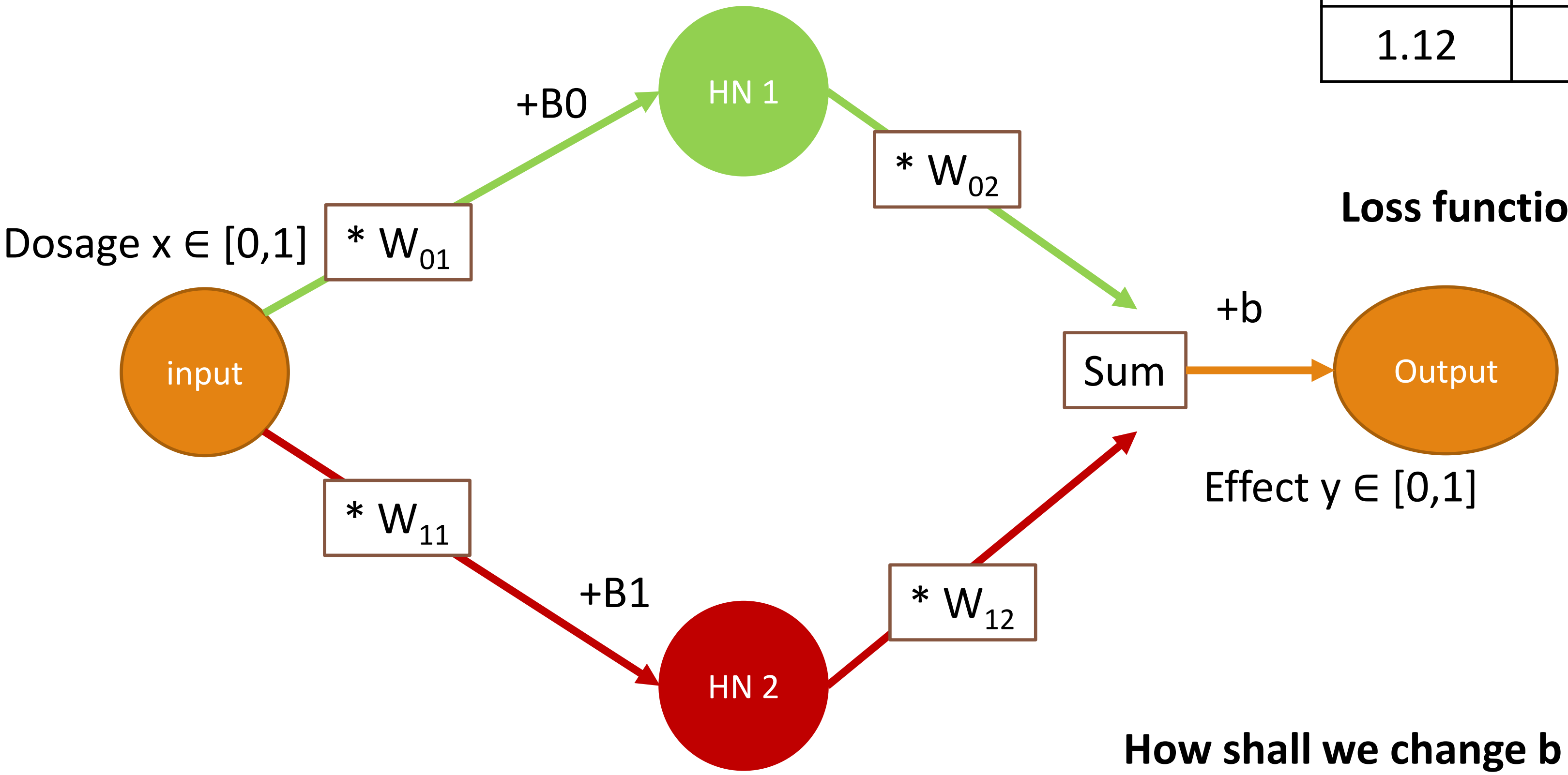
- A method that uses gradient descent to optimize the weights in each layer of the network

Loop until satisfied (predefined epochs or acceptable error)

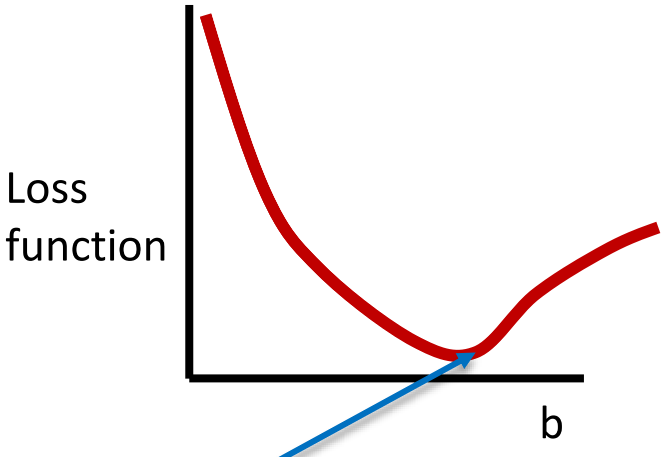
- Get the training data set
- Apply forward propagation to get predictions
- Estimate the error of classification – loss function
- Apply backward propagation to estimate how each hidden node affects the error using the chain rule
- Update each weight using gradient descent

How do ANNs learn?

	Training dataset		Squared Errors
b	Predicted	Actual	
0	0	1	1
0.7	0.5	0.5	0
-0.3	0.8	0.2	0.49
1.12	0.6	0.4	0.4



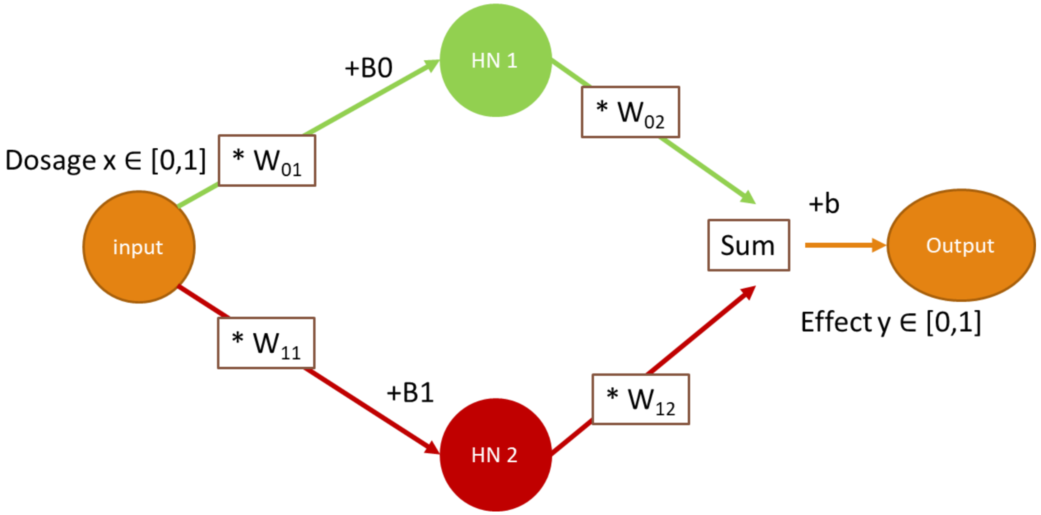
Loss function- Sum Squared Errors: (Actual – Predicted)^2

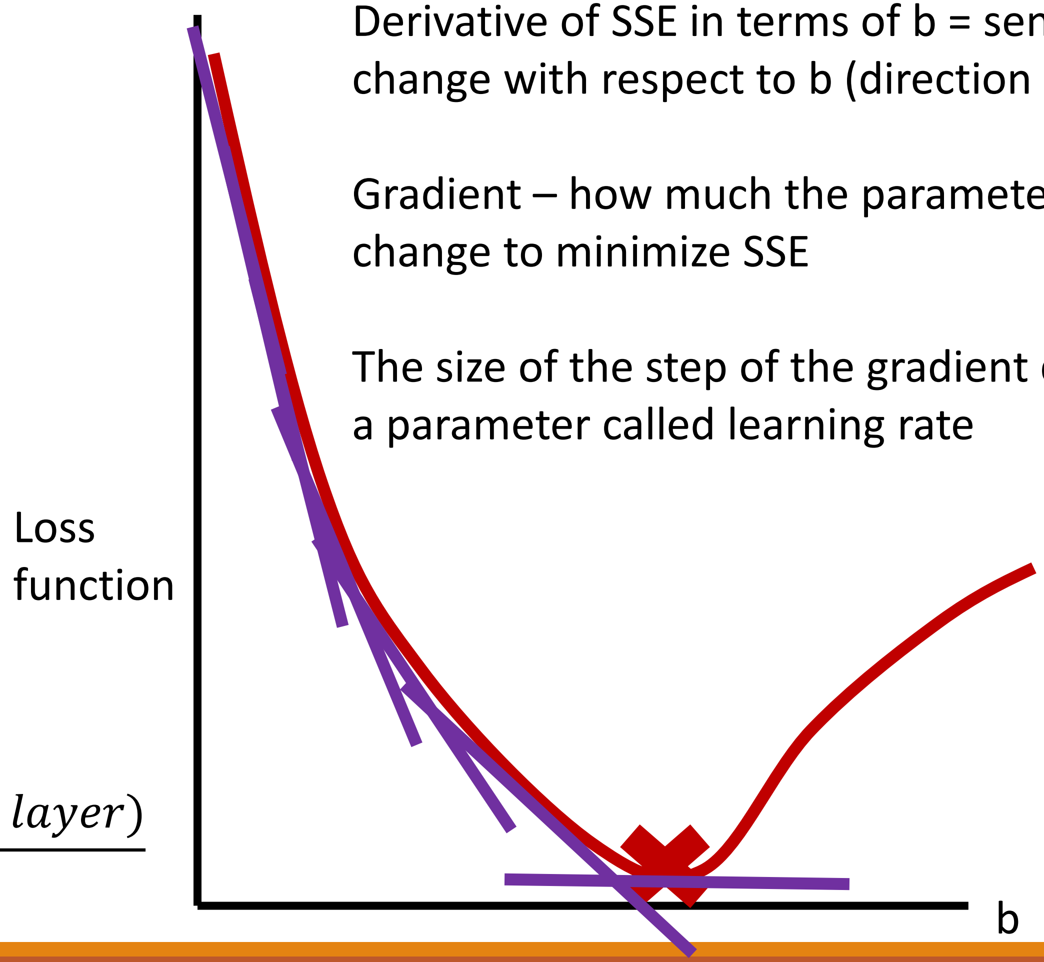


How shall we change b to minimize LF?

How do ANNs learn?

Sum Squared Errors: (Actual – Predicted)^2





Derivative of SSE in terms of b = sensitivity of change with respect to b (direction of SSE)

Gradient – how much the parameter b has to change to minimize SSE

The size of the step of the gradient descent is a parameter called learning rate

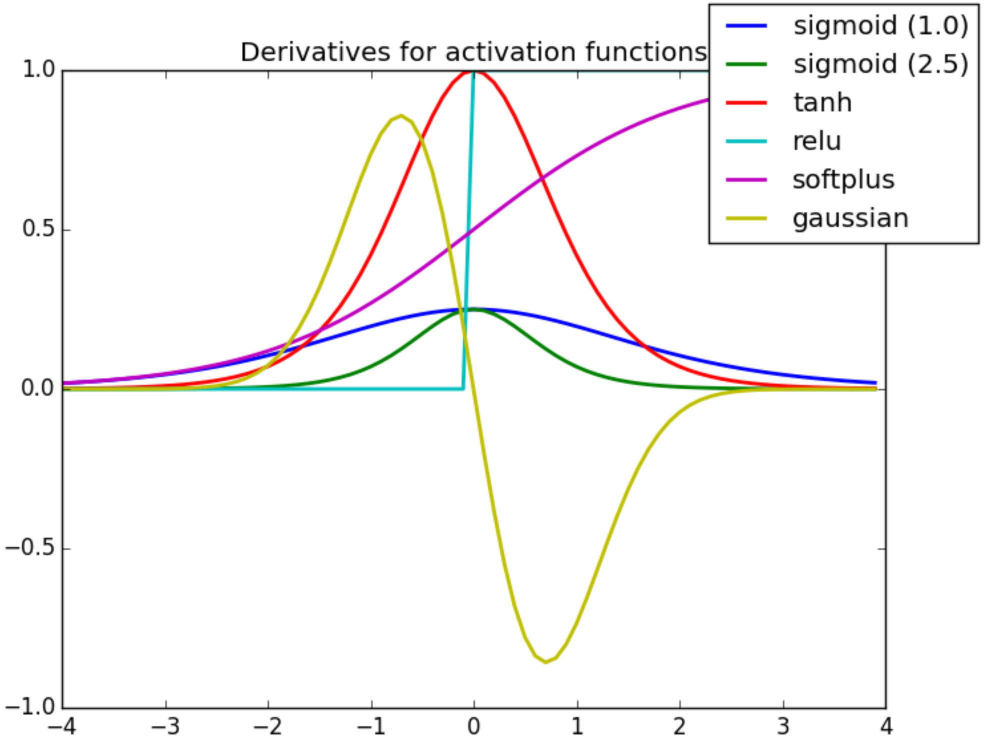
Calculate the gradients using the Chain Rule!

$$\frac{D(Error)}{dW_{02}} = \frac{D(Error)}{d(weights\ of\ prev.layer)} * \frac{D(weights\ of\ prev.layer)}{dW_{02}}$$

How to choose activation functions?

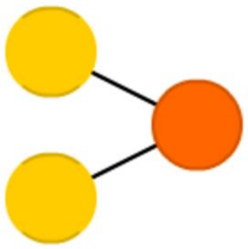
Hidden layers act as the “brain” of the neural network.

- There is no good and bad activation function, it depends on the problem
- E.g. computational power, output layer type etc…
  - Sigmoid and Tanh are computationally expensive
  - ReLU prevent nodes to fire if the value is negative
  - Softplus is more gentle



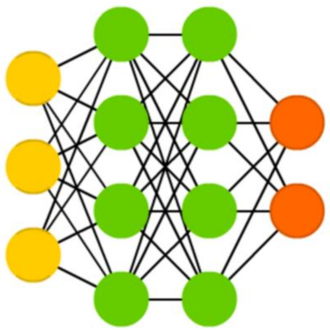
ANN flavours

Perceptron (P)



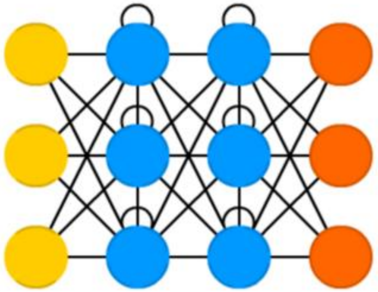
Simple and Old

Deep Feed Forward (DFF)



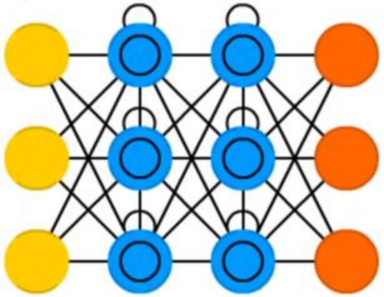
Computationally expensive
Really good results

Recurrent Neural Network (RNN)



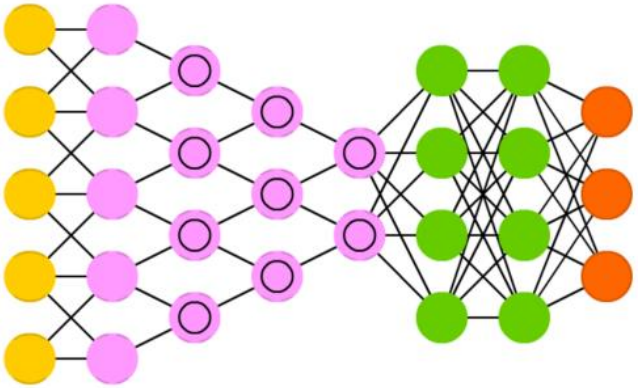
Hidden nodes form loops.
Widely used in NLP
Preferred when context is important (sequential)

Long / Short Term Memory (LSTM)



RNNs with memory cells
Used in Video applications
Preferred when “keep in mind what happened 10 seconds ago”

Deep Convolutional Network (DCN)



Semantically partitions data
The stars of ANN
Used when spatial or semantic relations are necessary.