

BUILDING A MODEL

5 METHODS FOR BUILDING MODELS:

- 1: ALL-IN
- 2: BACKWARD ELIMINATION
- 3: FORWARD SELECTION
- 4: BIDIRECTIONAL ELIMINATION
- 5: SCORE COMPARISON

STEPWISE REGRESSION.

1) "ALL-IN" NOT A TECHNICAL TERM
THROUGH ALL VARIABLES.

YOU'D USE THIS METHOD IF:

- YOU KNEW THE VARIABLES WERE USEFUL & PREDICTIVE BASED ON PRIOR KNOWLEDGE; OR
- IF YOU HAVE TO (E.G., FOR COMPLIANCE REASONS); OR
- IF YOU'RE PREPARING FOR BACKWARD ELIMINATION

2) BACKWARD ELIMINATION (FASTEST.)STEP 1: SELECT A SIGNIFICANCE LEVEL TO STAY IN THE MODEL (E.G. $SL = 0.05$)

STEP 2: FIT THE FULL MODEL WITH ALL POSSIBLE PREDICTORS

STEP 3: CONSIDER THE PREDICTOR WITH THE HIGHEST P-VALUEIF $P > SL$, GO TO STEP 4 OR GO TO FIN

STEP 4: REMOVE THE PREDICTOR

STEP 5: FIT MODEL WITHOUT THIS VARIABLE.

3) FORWARD SELECTIONSTEP 1: SELECT A SIGNIFICANCE LEVEL TO ENTER INTO THE MODEL (E.G. $SL = 0.05$)STEP 2: FIT ALL SIMPLE REGRESSION MODELS. $y \sim x_1$

SELECT THE ONE WITH THE LOWEST P-VALUE.

STEP 3: KEEP THIS VARIABLE AND FIT ALL POSSIBLE MODELS WITH ONE EXTRA PREDICTOR ADDED TO THE ONE(S) YOU ALREADY HAVE.

STEP 4: CONSIDER THE PREDICTOR WITH THE LOWEST P-VALUE.
IF $P < SL$, GO TO STEP 3, OTHERWISE FIN

2023.12.10

4) BIDIRECTIONAL ELIMINATION

STEPWISE REGRESSION

STEP 1 SELECT A SIGNIFICANCE LEVEL TO ENTER & TO STAY IN THE MODEL

→ STEP 2 PERFORM THE NEXT STEP OF FORWARD SELECTION
(NEW VARIABLES MUST HAVE $P < SL$ ENTER TO ENTER)↓ STEP 3 PERFORM ALL STEPS OF BACKWARD ELIMINATION
(OLD VARIABLES MUST HAVE $P < SL$ STAY TO STAY)↓ STEP 4 NO NEW VARIABLES CAN ENTER AND
NO OLD VARIABLES CAN EXIT

YOUR MODEL IS READY

5) SCORE COMPARISON
ALL POSSIBLE MODELS

MOST THOROUGH, BUT MOST RESOURCE INTENSIVE

STEP 1 SELECT A CRITERION OF GOODNESS OF FIT (E.G., AKAIKE CRITERION)

STEP 2 CONSTRUCT ALL POSSIBLE REGRESSION MODELS.

 2^{n-1} TOTAL COMBINATIONS FOR N VARIABLES∴ 2^{n-1} TOTAL MODELS THERE CAN POSSIBLY BE.

STEP 3: SELECT THE MODEL WITH THE BEST CRITERION.

NO NEED TO APPLY FEATURE SCALING
FOR MULTIPLE LINEAR REGRESSION

□ WHY?

2023.12.15

```

IMPORT LIBRARIES [import numpy as np
                  "matplotlib"
                  "matplotlib as plt"
                  "pandas as pd"]

① [dataset = pd.read_csv('50_Startups.csv')
    X = dataset.iloc[:, :-1].values # everything before last column.
    y = dataset.iloc[:, -1]          # last column]

② [y = dataset.iloc[:, -1].values # last column]

```

From sklearn.compose import ColumnTransformer.

From sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder)])

import libraries
 ① import numpy as np
 import matplotlib as plt
 import pandas as pd

② Import Dataset
 dataset = pd.read_csv('filename.csv')
 X = dataset.iloc[:, :-1] # All columns except last
 y = dataset.iloc[:, -1] # Only last column

③ ENCODE CATEGORICAL DATA
 from sklearn.compose import ColumnTransformer
 from sklearn.preprocessing import OneHotEncoder
 ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])],
 remainder='passthrough')
 X = np.array(ct.fit_transform(X)) # output new matrix where index 3 is encoded
 and convert the matrix to a numpy array.

④ SPLIT DATASET INTO TEST & TRAINING.
 from sklearn.model_selection import train_test_split
 X_train, y_train, X_test, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

⑤ TRAINING MULTIPLE LINEAR REGRESSION MODEL ON THE TRAININGSET
 from sklearn.linear_model import LinearRegression
 regressor = LinearRegression()
 regressor.fit(X_train, y_train)

⑥ PREDICT THE TEST SET RESULTS
 y_pred = regressor.predict(X_test)
 np.set_printoptions(precision=2)
 print(np.concatenate((y_pred.reshape(len(y_pred), 1), y_test.reshape(len(y_pred), 1)), axis=1)) # print prediction inline with actual value

⑦ MAKE A SINGLE PREDICTION

8.6.

print(regressor.predict([[1, 0, 0, 160000, 130000, 300000]]))

⑧ GET FINAL LINEAR REGRESSION EQUATION

print(regressor.coef_)

print(regressor.intercept_)

CALIFORNIA	NEW YORK	FLORIDA	R&D SPEND	ADMIN. SPEND	MARKETING SPEND	MARKETING SPEND
0.66e+01	-8.73e+02	7.86e+02	7.73e+01	3.29e-02	3.66e-02	42467.52924853204

∴ profit = 8.66

× Dummy State 1 - 873

$$\begin{aligned} \text{∴ profit} = & 8.66 \times \text{Dummy State 1} \\ & - 8.73 \times \text{Dummy State 2} \\ & + 786 \times \text{Dummy State 3} \\ & + 0.773 \times \text{R&D Spend} \\ & + 0.0329 \times \text{Admin Spend} \\ & + 0.0366 \times \text{Marketing Spend} \\ & + 42467.53 \end{aligned}$$

SCIKIT LEARN > API > METRICS.
WEBSITE

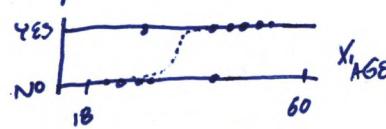
NOTE: COEFFICIENTS ONLY SPEAK TO THE ADDITIONAL
EFFECT OF ~~EACH~~ EVERY VARIABLE, GIVEN
THAT THE OTHER VARIABLES ARE IN PLACE.
 \therefore COEFFICIENT VALUES WILL CHANGE AS THE NUMBER
OF VARIABLES CHANGE.

SECTION 16

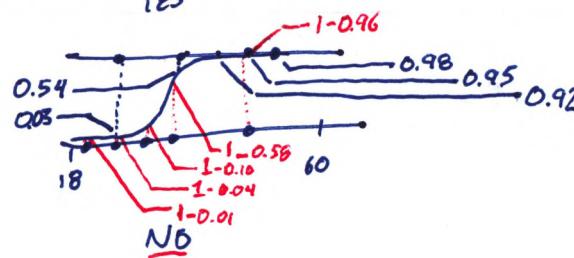
LOGISTIC REGRESSION

WILL PURCHASE HEALTH INSURANCE. AGE INCOME EDU LEVEL MARITAL STATUS

$$\ln \frac{P}{1-P} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 X_4$$



MAXIMUM LIKELIHOOD



$$\text{LIKELIHOOD} = 0.03 * 0.54 * 0.92 * 0.95 * 0.98 * \\ \text{FOR CURVE } (1-0.01) * (1-0.04) * (1-0.10) * (1-0.58) * (1-0.92) \\ = 0.000019939$$

DIFFERENT LIKELIHOODS FOR DIFFERENT
CURVES ARE CALCULATED AND THE BEST
CURVE WITH THE HIGHEST LIKELIHOOD
IS SELECTED.

CONFUSION MATRIX

```
From sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_true, y_pred)
# COMPARES THE PREDICTED VALUES TO OBSERVED VALUES
print(cm)
accuracy_score(y_true, y_pred) # will print the score.
```

SECTION 17

(K-NN) K-NEAREST NEIGHBORS

- 1: CHOOSE THE NUMBER, K, OF NEIGHBORS
- 2: GET THE K NEAREST NEIGHBORS OF THE NEW DATA POINT, (E.G., ACCORDING TO EUCLIDEAN DISTANCE)
- 3: AMONG THE K NEIGHBORS, COUNT THE NUMBER OF DATAPoints IN EACH CATEGORY
- 4: ASSIGN THE NEW DATA POINT TO THE CATEGORY THAT INCLUDES THE MOST OF THE NEIGHBORS.

WITH P=2 (DEFAULT)

NOTE USE A MINKOWSKI METRIC (DEFAULT) FOR EUCLIDEAN DISTANCE-BASED CLASSIFICATION.

NOTE WHEN USING KNearestClassifier WITH THE ALGORITHM SET TO AUTO, USE FIT METHOD ON CLASSIFIER.

MACHINE LEARNING AZ NOTES MISC

DATA PREPROCESSING

A WAY TO HANDLE MISSING DATA IS TO
SET THE VALUES TO THE EXPECTED MEAN.
OF THE COLUMN.

2023.12.29 MACHINE LEARNING A-Z

DECISION TREE INTUITION - NO NEED FOR FEATURE SCALING

RANDOM FOREST INTUITION AN ENSEMBLE LEARNING APPROACH
APPLIED TO REGRESSION TREES VS CLASSIFICATION TREES

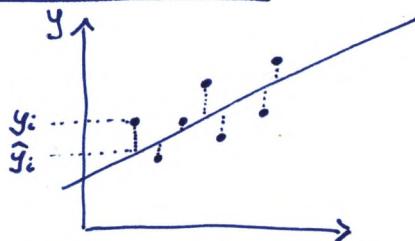
STEP 1 PICK, AT RANDOM, K DATA POINTS FROM THE TRAINING SET

STEP 2 BUILD THE DECISION TREE ASSOCIATED WITH THESE K DATA POINTS

STEP 3 CHOOSE THE NUMBER OF, NTREE, OF TREES YOU WANT TO BUILD AND REPEAT STEPS 1&2.
E.G., 500 NTREES, EACH HAVING A DIFFERENT K DATA POINTS.STEP 4 FOR A NEW DATA POINT, MAKE EACH ONE OF YOUR NTREE TREES PREDICT
THE VALUE OF Y FOR THE POINT IN QUESTION AND ASSIGN THE NEW
DATA POINT THE AVERAGE ACROSS ALL THE PREDICTED VALUES. Y VALUES.

ENSEMBLE LEARNING:

TAKING MULTIPLE ALGORITHMS, OR THE SAME ALGORITHM MULTIPLE TIMES, AND PUT THEM TOGETHER TO RESULT IN A MUCH MORE POWERFUL VERSION.

R SQUARED

$$SS_{\text{res}} = \sum (y_i - \hat{y}_i)^2$$

(RESIDUAL SUM SQUARED)

 R^2 WILL RANGE BETWEEN 0 & 1 R^2 : GOODNESS OF FIT (GREATER IS BETTER)

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$



$$SS_{\text{tot}} = \sum (y_i - y_{\text{avg}})^2$$

(RULE OF THUMB (FOR OUR TUTORIAL)*

1.0 : PERFECT FIT (SUSPICIOUS)

0.9 : VERY GOOD

0.7 : NOT GREAT

<0.4 : TERRIBLE

<0 : MODEL DOESN'T MAKE SENSE
FOR THIS DATA#THIS IS HIGHLY DEPENDENT ON CONTEXT
IN SOME INDUSTRIES, 0.4 MIGHT BE GREAT.ADJUSTED R SQUARED

HELPS DETERMINE WHETHER ADDITIONAL INDEPENDENT VARIABLES ARE WORTH WHILE
SOLVES PROBLEM: WITH R^2 , WHICH IS THAT SS_{tot} MAY REMAIN THE SAME
WHEN ADDING A NEW INDEPENDENT VARIABLE WHILE
 SS_{res} COULD DECREASE OR REMAIN THE SAME.

$$\text{ADJ } R^2 = 1 - (1 - R^2) \times \frac{n-1}{n-k-1}$$

INDEPENDENT VARIABLES
SAMPLE SIZE

ENSURES CONTRIBUTION OF EACH ADDITIONAL
IND. VAR. IS SUBSTANTIAL ENOUGH TO CONSIDER IT.

JUSTIFICATION : ONLY ADD VARIABLES WHEN THEY BRING SUBSTANTIAL IMPROVEMENT.

$$\hat{y} = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3$$

ADDITIONAL INDEPENDENT VAR.

by setting to 0,
the R^2 value would
remain the same
or even improve.THE ORDINARY LEAST SQUARES
METHOD IS USED TO CALCULATE
COEFFICIENT b_3 THAT WOULD IMPROVE
THE \hat{y}_i PREDICTED VALUE (E.G., 0)

MACHINE LEARNING A-Z SUPPORT VECTOR MACHINES

THE NATURE OF STATISTICAL LEARNING THEORY 1992 (VLADIMIR VAPNIK)

CH4 IN SUPPORT VECTOR REGRESSION by MARIETTE AWAD & RAHUL KHANNA

<https://core.ac.uk/download/pdf/81523322.pdf>

WHEN TO APPLY FEATURE SCALING:

WHEN VALUE RANGES IN DIFFERENT COLUMNS ARE
CONSIDERABLY HIGHER, NOT SCALING WOULD RESULT
IN THE LOWER VALUES' COLUMNS BEING
NEGLECTED IN THE ^{RESULTING} MODEL.

E.G., SALARY LEVEL \Rightarrow LEVEL VALUES ARE CONSIDERABLY

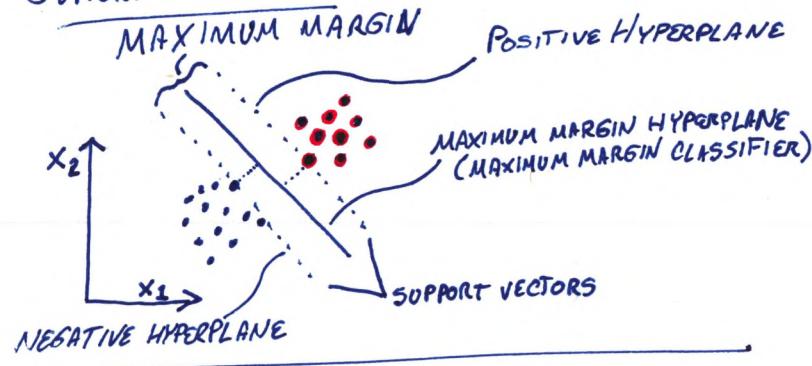
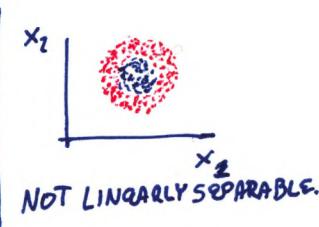
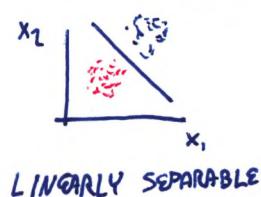
45000	1
60000	2
75000	3

 LOWER, SO SCALE THE SALARY

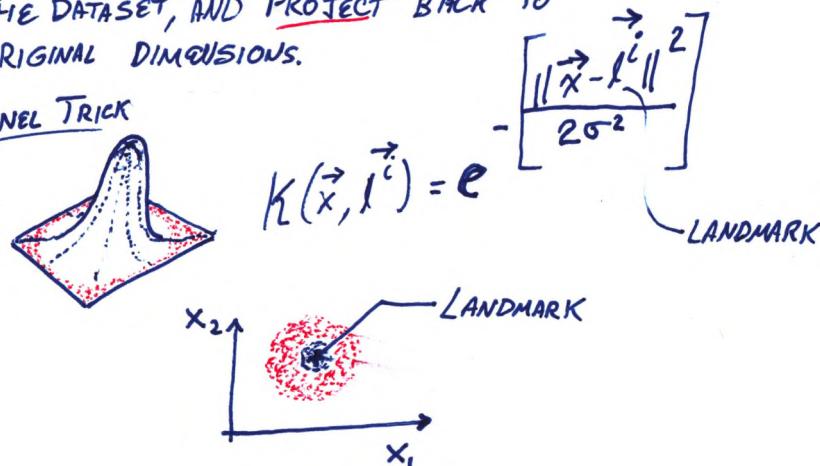
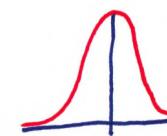
DON'T APPLY FEATURE SCALING TO:

BINARY VALUES

DUMMY VARIABLES RESULTING FROM ONEHOTENCODING.

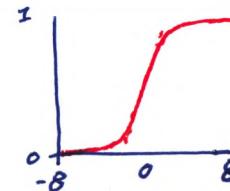
SUPPORT VECTOR MACHINE (SVM)KERNEL SVM INTUITIONHIGHER DIMENSIONAL SPACE

HOW TO TAKE A NON-LINEARLY SEPARABLE DATASET, MAP IT TO A HIGHER DIMENSION AND GET A LINEARLY SEPARABLE DATASET, INVOKE THE SVM ALGORITHM, BUILD A DECISION BOUNDARY FOR THE DATASET, AND PROJECT BACK TO ORIGINAL DIMENSIONS.

KERNEL TRICKTYPES OF KERNEL FUNCTIONS

GAUSSIAN RBF KERNEL $K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x} - \vec{l}^i\|^2}{2\sigma^2}}$

SELECT A LANDMARK DISTANCE FROM LANDMARK USED TO CLASSIFY



SIGMOID KERNEL $K(x, y) = \tanh(y \cdot X^T Y + r)$

SELECT A LANDMARK DISTANCE FROM LANDMARK USED TO CLASSIFY LINE INDICATES A DECISION BOUNDARY



POLYNOMIAL KERNEL $K(X, Y) = (r \cdot X^T Y + r)^d, r > 0$

SUPER DATA SCIENCE - MAKING THE COMPLEX SIMPLE

mlkernels.readthedocs.io/en/latest/Kernelfunctions.html

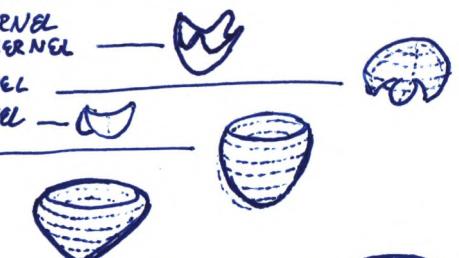
SCALAR PRODUCT KERNEL
SQUARED-DISTANCE KERNEL

SINE-SQUARED KERNEL

CHI-SQUARED KERNEL

GAUSSIAN KERNEL

LAPLACIAN KERNEL



Periodic Kernel

RATIONAL-SQUARE KERNEL

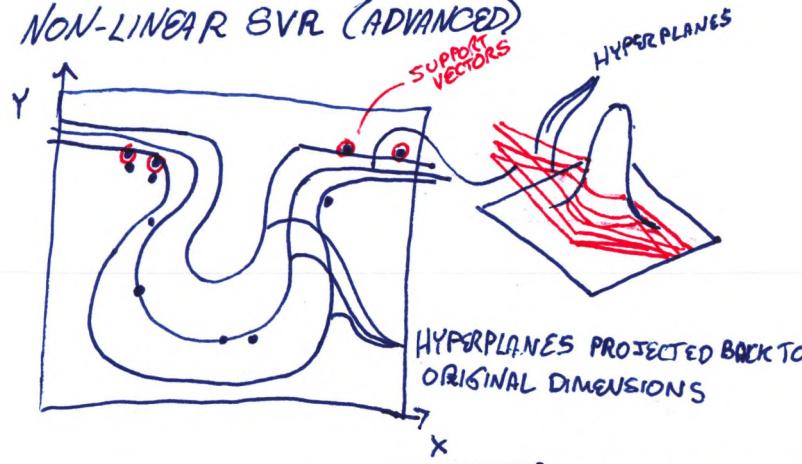
MATERN KERNEL

LINEAR&POLYNOMIAL KERNEL

SIGMOID KERNEL



NON-LINEAR SVR (ADVANCED)



BAYES THEOREM

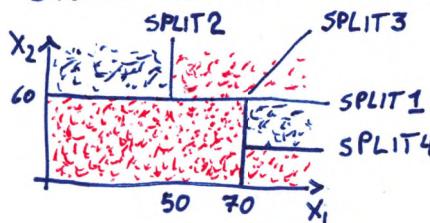
$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

PROBABILITY OF B, Given A.

PROBABILITY OF A, GIVEN B.

NAIVE BAYES.

DECISION TREE INTUITION

SPLITS TRY TO MINIMIZE INFORMATIONAL ENTROPYRANDOM FOREST CLASSIFIER
AN ENSEMBLE LEARNING APPROACH

DIFFERENT CONFUSION MATRICES

A) ACTUAL LABEL

		1	0
PREDICTED LABEL	1	TP	FP
	0	FN	TN

B) ACTUAL LABEL

		0	1
PREDICTED LABEL	0	TN	FN
	1	FP	TP

C) PREDICTED LABEL

		1	0
ACTUAL LABEL	1	TP	FN
	0	FP	TN

D) PREDICTED LABEL

		0	1
ACTUAL LABEL	0	TN	FP
	1	FN	TP

ACCURACY PARADOX

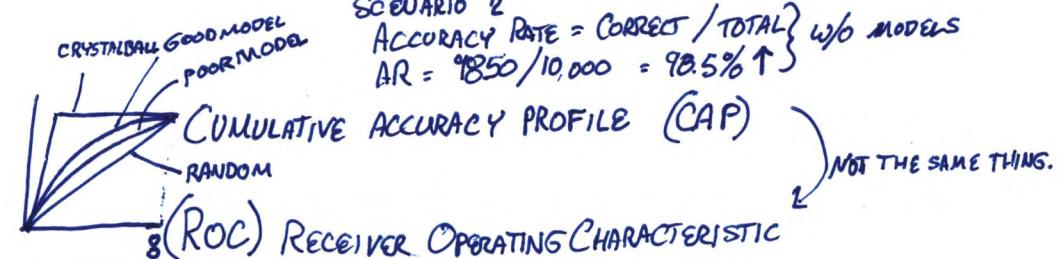
E.G., ACCURACY INCREASING BY ALWAYS ASSUMING THE SAME OUTCOME AND WHILE ABANDONING THE USE OF MODELS.

SCENARIO 1

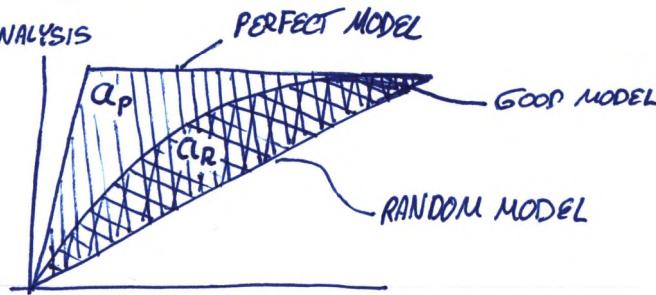
ACCURACY RATE = CORRECT / TOTAL } w/ MODELS
 $AR = 9,800 / 10,000 = 98\%$

SCENARIO 2

ACCURACY RATE = CORRECT / TOTAL } w/o MODELS
 $AR = 9850 / 10,000 = 98.5\% \uparrow$



CAP ANALYSIS

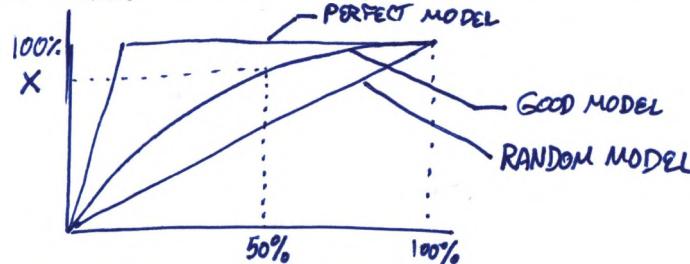


$$\text{ACCURACY RATIO} = \frac{\text{AREA UNDER GOOD MODEL} (a_R)}{\text{AREA UNDER PERFECT MODEL LINE} (a_p)}$$

~~GOOD MODEL~~

$$AR = \frac{a_R}{a_p}$$

ALTERNATE APPROACH TO ASSESS PERFORMANCE

RULE OF THUMB

$X < 60\%$ RUBBISH

$60\% < X < 70\%$ POOR

$70\% < X < 80\%$ GOOD

$80\% < X < 90\%$ VERY GOOD

$90\% < X < 100\%$ TOO GOOD (SUSPICIOUS)

↳ MAY BE OVERTFITTING THE MODEL.

POST FACTO VARIABLE

ONE OF YOUR INDEPENDENT VARIABLES MAY BE
SHOULDN'T BE IN THE TRAINING DATA
BECAUSE IT'S LOOKING INTO THE FUTURE.

CLUSTERING

K-MEANS CLUSTERING INTUITION

- 1) DECIDE HOW MANY CLUSTERS YOU WANT
- 2) FOR EACH CLUSTER, RANDOMLY PLACE A CENTROID (ON THE SCATTER PLOT)
- 3) K-MEANS WILL ASSIGN EACH DATAPoint TO THE CLOSEST CENTROID
- 4) CALCULATE THE CENTER OF MASS FOR THE CENTRO DATAPoints BELONGING TO EACH CENTROID (E.G. AVG OF ALL OF A CENTROID'S DATA POINTS)
- 5) MOVE THE CENTROIDS TO THE CENTER OF MASS CORRESPONDING TO THE DATAPoints
- 6) REASSIGN DATAPoints TO THE NEAREST CENTROID
- 7) REPEAT STEPS 4-6 UNTIL READING THE STEPS NO LONGER RESULTS IN CHANGES

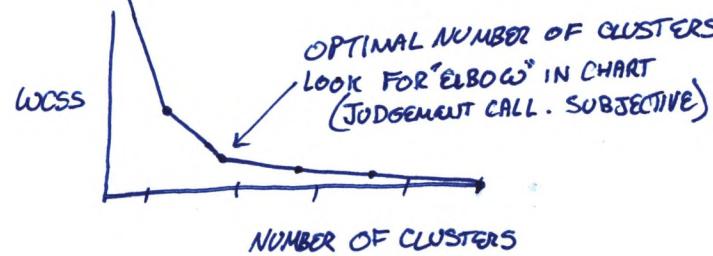
ELBOW METHOD (HELPS DECIDE HOW MANY CLUSTERS TO SELECT)

WITHIN CLUSTER SUM OF SQUARES (WCSS)

$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \dots$$

SQUARE SUM OF DISTANCE
SUM OF SQUARED DISTANCES FROM DATAPoints TO CENTROID 1] + ...

WCSS WILL REDUCE TO ZERO WHEN THE NUMBER OF CLUSTERS EQUALS THE NUMBER OF DATAPoints (I.E., THE MAXIMUM NUMBER OF CLUSTERS). EACH DATAPoint'S DISTANCE TO ITS CENTROID WOULD BE ZERO, SO WCSS WOULD BE ZERO.



K MEANS ++

- HELPS AVOID THE RANDOM INITIALIZATION TRAP
BECAUSE CENTROIDS ARE SELECTED IN A WEIGHTED RANDOM FASHION.
- 1) CHOOSE FIRST CENTROID AT RANDOM AMONG DATA POINTS
 - 2) FOR EACH OF THE REMAINING DATA POINTS, COMPUTE THE DISTANCE (D) TO THE NEAREST OUT OF ALREADY SELECTED CENTROIDS.
 - 3) CHOOSE NEXT CENTROID AMONG REMAINING DATA POINTS USING WEIGHTED RANDOM SELECTION - WEIGHTED BY D^2
 - 4) REPEAT STEPS 2 & 3 UNTIL ALL K CENTROIDS HAVE BEEN SELECTED
 - 5) PROCEED WITH STANDARD K-MEANS CLUSTERING.

HIERARCHICAL CLUSTERING INTUITION (HCI)

CAN RESULT IN SIMILAR RESULT AS K-MEANS CLUSTERING.

TYPES OF HIERARCHICAL CLUSTERING (HC):

- AGGLOMERATIVE (BOTTOM UP APPROACH)
- DIVISIVE (TOP DOWN)

AGGLOMERATIVE HC

1) MAKE EACH DATA POINT A SINGLE POINT CLUSTER
 \hookrightarrow THIS FORMS N CLUSTERS

2) TAKE THE 2 CLOSEST CLUSTERS AND DATA POINTS AND MAKE THEM ONE CLUSTER
 \hookrightarrow THIS FORMS N-1 CLUSTERS

3) TAKE THE 2 CLOSEST CLUSTERS AND MAKE THEM ONE CLUSTER
 \hookrightarrow THIS FORMS N-2 CLUSTERS.

⋮
4) REPEAT UNTIL THERE IS ONLY 1 CLUSTER

HOW IS THE DISTANCE BETWEEN TWO CLUSTERS DETERMINED:

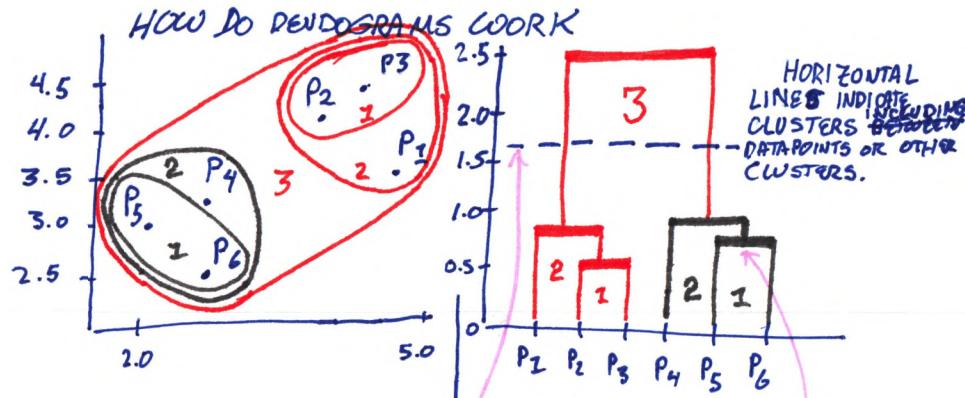
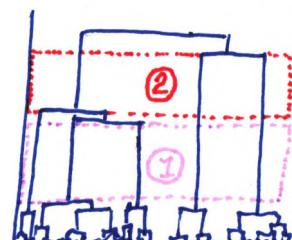
OPTION 1) DISTANCE BETWEEN TWO POINTS (E.G., EUCLIDEAN DISTANCE)

OPTION 2) " " FURTHEST POINTS BETWEEN CLUSTERS

OPTION 3) AVERAGE DISTANCE (OF ALL POINTS... SOMEHOW)

OPTION 4) DISTANCE BETWEEN CENTROIDS

OTHER... DETERMINE THE BEST APPROACH.

DISSIMILARITY/DISTANCE THRESHOLD:

A THRESHOLD LEVEL OF DISSIMILARITY, BASED ON DISTANCE, IS SET AND THE NUMBER OF CLUSTERS IS DETERMINED BASED ON THE NUMBER OF CLUSTERS THAT HAVE DISTANCES BETWEEN CLUSTERS THAT ARE BELOW THE THRESHOLD.

THE NUMBER OF CLUSTERS EQUALS THE NUMBER OF VERTICAL LINES THAT CROSS THE THRESHOLD.

HIGHER LEVEL BETWEEN P5-6 THAN P2-3 INDICATES A FARTHER DISTANCE BETWEEN P5-6 THAN BETWEEN P2-3.

DENDROGRAMS ARE BASED ON WITHIN-CLUSTER DENDROGRAMS.

HC PERFORMS WORSE THAN K-MEANS ON LARGE DATASETS

WARD'S MINIMUM VARIANCE METHOD

MOST RECOMMENDED CLUSTERING TECHNIQUE RESULTS IN CLUSTERS WHERE THE DATA/OBSERVATION POINTS DON'T VARY TOO MUCH. (I.E. THEY HAVE LOW VARIANCE)

"AT EACH GENERATION, THE WITHIN-CLUSTER SUM OF SQUARES IS MINIMIZED OVER ALL PARTITIONS OBTAINABLE BY MERGING TWO CLUSTERS FROM THE PREVIOUS GENERATION"

FINDING AN OPTIMAL NUMBER OF CLUSTERS

FIND THE TALLEST RECTANGLE, THAT EXTENDS HORIZONTALLY ACROSS THE ENTIRE DENDROGRAM, THAT TOUCHES & INCLUDES ONLY 2 HORIZONTAL CLUSTER SIMILARITY LEVELS.

OPTIMAL NUMBERS OF CLUSTERS WILL BE THE NUMBER OF VERTICAL LINES THAT PASS THROUGH THE TALLEST RECTANGLE(S). THERE MAY BE RECTANGLES THAT ARE SIMILAR IN HEIGHT, CHOOSE THE NUMBER BY USING THE SUGGESTED OPTIMAL NUMBERS AS A GUIDE.

ASSOCIATION RULE LEARNING. (ARL)

"People that bought X also bought..."

Two ASSOCIATION RULE METHODS:

- APRIORI • PREFERRED
- ECLAT

TABLE SHOULD BE HERE.

APRIORI ALGORITHM

3 PARTS: SUPPORT, CONFIDENCE, & LIFT

USER ID	MOVIES LIKED	POTENTIAL RULES:
1	MOVIE 1, MOVIE 2, MOVIE 3, MOVIE 4	$MOVIE 1 \rightarrow MOVIE 2$
2	MOVIE 1, MOVIE 2	$MOVIE 2 \rightarrow MOVIE 4$
3	" "	$MOVIE 1 \rightarrow MOVIE 3$
4	" ", " ", MOVIE 4	
5	MOVIE 2, MOVIE 3, MOVIE 4	
6	MOVIE 3, MOVIE 4	

E.G. MOVIE RECOMMENDATION:

$$SUPPORT(M) = \frac{\# \text{ USER WATCHLISTS CONTAINING } M}{\# \text{ USER WATCHLISTS}}$$

↳ HOW BIG PROPORTION
WHAT PERCENTAGE OF
PEOPLE HAVE SEEN A MOVIE

$$CONFIDENCE(M_1 \rightarrow M_2) = \frac{\# \text{ USER WATCHLISTS CONTAINING } M_1 \& M_2}{\# \text{ USER WATCHLISTS CONTAINING } M_1}$$

↳ MAKING/TESTING A HYPOTHESIS.

E.G., PEOPLE THAT HAVE SEEN MOVIE 1 ARE LIKELY TO HAVE SEEN MOVIE 2.

$$LIFT(M_1 \rightarrow M_2) = \frac{CONFIDENCE(M_1 \rightarrow M_2)}{SUPPORT(M_2)}$$

THE LIFT IS THE "IMPROVEMENT" IN YOUR PREDICTION.

E.G., IF THE CONFIDENCE VALUE IS 17.5% AND THE SUPPORT IS 10%, THE LIFT COULD BE 1.75.

IF 10% OF ANY POPULATION HAS SEEN MOVIE 2, WE ARE MORE LIKELY TO GUESS THAT A PERSON HAS SEEN ^{OR WILL SEE} MOVIE 2 BASED ON WHETHER THEY HAVE SEEN MOVIE 1.

- 1) SET UP A MINIMUM SUPPORT & CONFIDENCE
E.G., MAY NOT WANT TO CONSIDER SUPPORT BELOW 20%.
- 2) TAKE ALL THE SUBSETS IN TRANSACTIONS HAVING HIGHER SUPPORT THAN MINIMUM SUPPORT
- 3) TAKE ALL THE RULES OF THESE SUBSETS HAVING HIGHER CONFIDENCE THAN MINIMUM CONFIDENCE.
- 4) SORT THE RULES BY DECREASING LIFT.
HIGHEST LIFT WOULD BE MOST MEANINGFUL

Python Package:
apyori

ECLAT ALGORITHM ('T' IN ECLAT IS SILENT)

E.G. MOVIE RECOMMENDATION

$$\text{SUPPORT}(M) = \frac{\# \text{ USERS WATCHLISTS CONTAINING } M}{\# \text{ USER WATCHLISTS}}$$

M IS A SET OF MOVIES TOGETHER.

'HOW OFTEN ARE THESE MOVIES IN WATCHLISTS TOGETHER'

E.G. MARKET BASKET OPTIMISATION:

$$\text{SUPPORT}(I) = \frac{\# \text{ TRANSACTIONS CONTAINING } I}{\# \text{ TRANSACTIONS}}$$

~~SUPPORT(M) =~~

~~SUPPORT~~

- 1) SET A MINIMUM SUPPORT
- 2) TAKE ALL THE SUBSETS IN TRANSACTIONS HAVING HIGHER SUPPORT THAN MINIMUM SUPPORT
- 3) SORT THESE SUBSETS BY DECREASING SUPPORT.

REINFORCEMENT LEARNING.

UPPER CONFIDENCE BOUND (UCB)

"USING CONFIDENCE BOUNDS FOR EXPLOITATION
AND EXPLORATION OF TRADE OFFS" AVER

"IF YOU DON'T EXPLORE FOR LONG ENOUGH,
A SUBOPTIMAL MACHINE MAY APPEAR AS
AN OPTIMAL MACHINE..."

UPPER CONFIDENCE BOUND ALGORITHM
(USING ADS AS AN EXAMPLE)

STEP 1: AT EACH ROUND n , WE CONSIDER TWO NUMBERS
FOR EACH AD i :

- $N_i(n)$ - THE NUMBER OF TIMES THE AD, i , WAS SELECTED UP TO ROUND n ,
- $R_i(n)$ - THE SUM OF REWARDS OF THE AD, i , UP TO ROUND n .

STEP 2: FROM THESE TWO NUMBERS, WE COMPUTE:

- THE AVERAGE REWARD OF AD, i , UP TO ROUND n :

$$\bar{r}_i(n) = \frac{R_i(n)}{N_i(n)}$$

- THE CONFIDENCE INTERVAL $[\bar{r}_i(n) - \Delta_i(n), \bar{r}_i(n) + \Delta_i(n)]$ at round n WITH:

$$\Delta_i(n) = \sqrt{\frac{3}{2} \frac{\log(n)}{N_i(n)}}$$

STEP 3: WE SELECT THE AD, i , THAT HAS THE MAXIMUM UCB $\bar{r}_i(n) + \Delta_i(n)$.

MACHINE LEARNING A-Z 2024-02-02 ① 1530-1630

NATURAL LANGUAGE PROCESSING (NLP)

COURSE WILL NOT COVER SEQ2SEQ OR CHATBOTS.

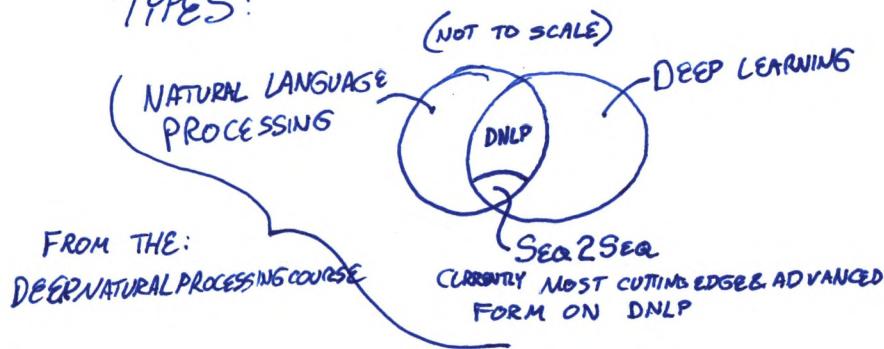
Will cover

TYPES OF NLP

CLASSICAL VS DEEP LEARNING MODELS

BAG-OF-COORDS. FOR SENTIMENT ANALYSIS.

TYPES:



CHATBOTS USED TO BE STRUCTURED BASED ON

- IF/ELSE STATEMENTS BASED ON PREDICTED QUESTIONS.
- AUDIO FREQUENCY COMPONENT ANALYSIS (SPEECH RECOGNITION)
- BAG-OF-WORDS MODEL (CLASSIFICATION) NLP/DNLP REGION.

COMMENT	PASS/FAIL
"GREAT JOB"	1
"AMAZING WORK"	1
"WELL DONE"	1
"VERY WELL WRITTEN"	1
"POOR EFFORT"	0
"COULD HAVE DONE BETTER"	0
"TRY HARDER NEXT TIME"	0

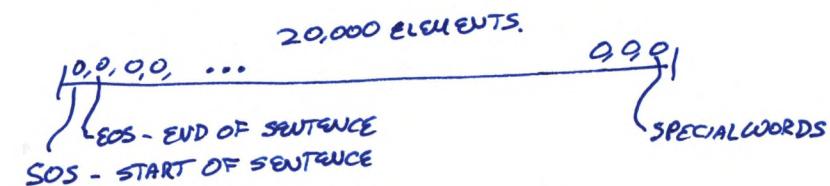
TERMS ARE SCORED
ASSOCIATE TERMS WORDS IN TERMS WITH SCORES.

- CONVOLUTIONAL NEURAL NETWORKS (CNN) FOR TEXT RECOGNITION (CLASSIFICATION)
- SEQ2SEQ

BAG-OF-WORDS MODEL

START WITH VECTOR/ARRAY THAT'S ABOUT 20,000 ELEMENTS LONG.
EACH NATIVE ENGLISH SPEAKER COMMONLY USES ABOUT 20K WORDS.
EACH ELEMENT CORRESPONDS TO A UNIQUE WORD.
THERE ARE ABOUT 171,476 WORDS IN THE ENGLISH DICTIONARY.

A COVERAGE OF ABOUT 3,000 WORDS COVERS 95% OF COMMON TEXTS. $\therefore 1.75\% \text{ OF TOTAL # OF WORDS.}$



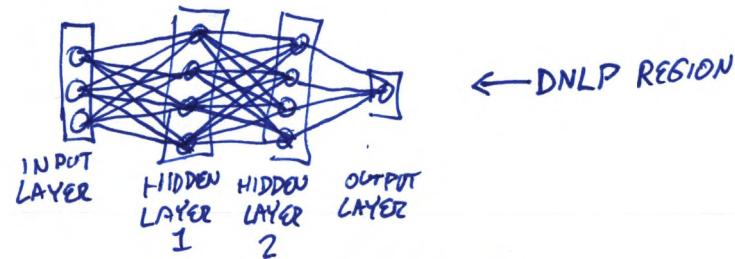
- COUNT EACH TIME EACH WORD APPEARS IN A GIVEN TEXT AND SET THE COUNTS TO THE CORRESPONDING ELEMENTS.
- SPECIAL WORDS, SUCH AS NAMES OR WORDS THAT AREN'T INCLUDED IN THE 20K ELEMENTS (FROM THE EXTENDED VOCABULARY OF 171,476 WORDS) ARE COUNTED AS SPECIAL WORDS ELEMENT.
- PUNCTUATION MARKS ARE COUNTED AS ELEMENTS AS WELL. E.G. COMMAS.
- THE ARRAY WILL BE SPARSE (MANY ZEROS) MOST OF THE TIME.

RESPONSES TO EMAILS CAN BE USED TO GENERATE VALUES OR SCORES PERTAINING TO DIFFERENT CHARACTERISTICS. & TO PREDICT RESPONSES.

E.G.
"DO YOU LIKE THE RECIPE I SENT YOU?" \rightarrow "NO"
SENTIMENT HAPPY? \rightarrow YES"



MAY APPLY A NEURAL NETWORK INSTEAD.



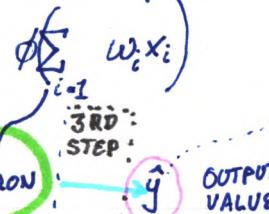
THE LAST TWO APPROACHES CAN ONLY
RESULT IN A YES/NO TYPE RESPONSE,
NOT A FULL CONVERSATION.

ARTIFICIAL NEURAL NETWORKS (ANN)

GEOFFREY HINTON
FATHER OF DEEP LEARNING. 80's.
YOUTUBE.

THE NEURON

yellow = input layer

analogy input layer for a person
is their senses.Activation Function
2nd Step

y means 'actual' value,
we use \hat{y} to mean
an 'output' value.

COST FUNCTION

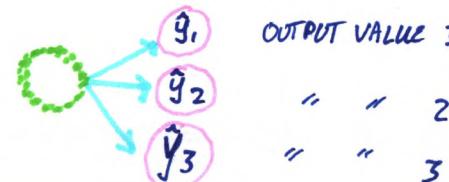
$$C = \frac{1}{2} (\hat{y} - y)^2$$

MOST COMMON COST FUNCTION,
BUT THERE ARE OTHERS.OUR GOAL IS TO MINIMIZE
THE COST FUNCTION BECAUSE
THE COST C REPRESENTS THE
ERROR IN YOUR PREDICTION.

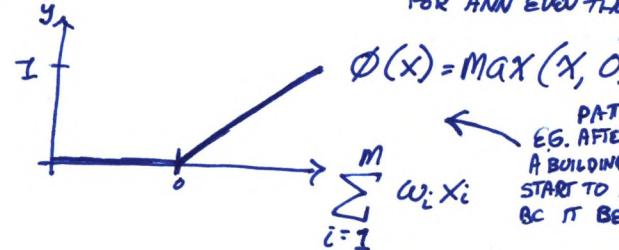
CAN BE:

- CONTINUOUS (E.G., PRICE)
- BINARY (E.G., WILL EXIT OR YES/NO)
- CATEGORICAL VARIABLE

IF THE OUT PUT IS CATEGORICAL
YOU WONT HAVE A SINGLE OUTPUT
VALUE, YOU'LL HAVE AN OUTPUT
FOR EACH DUMMY VARIABLE



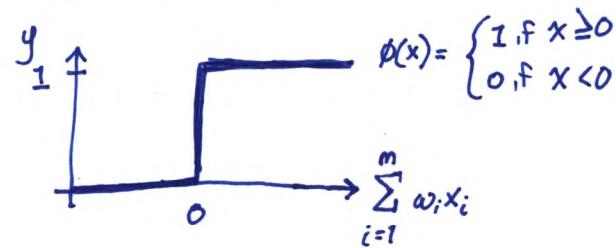
RECTIFIER FUNCTION

ONE OF THE MOST POPULAR FUNCTIONS
FOR ANN EVEN THOUGH IT HAS A KINK

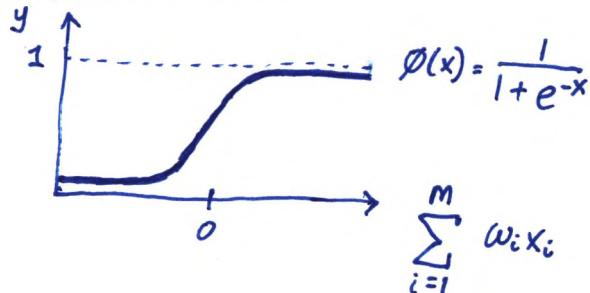
PATTERN
E.G. AFTER 100+ YEARS
A BUILDING'S VALUE MAY
START TO INCREASE RAPIDLY
BC IT BECOMES HISTORIC

COMMON
TO SEE A RECTIFIER
FUNCTION HERE (IN HIDDEN LAYER)
WHILE USING A
SIGMOID FUNCTION
AT THE OUTPUT LAYER.

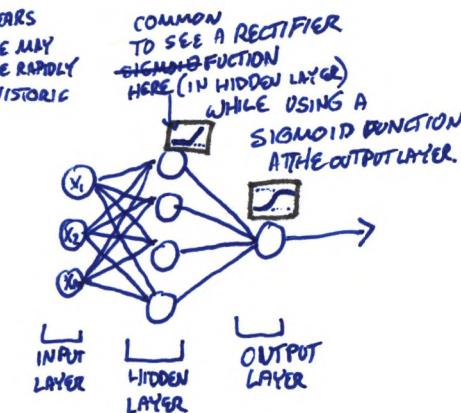
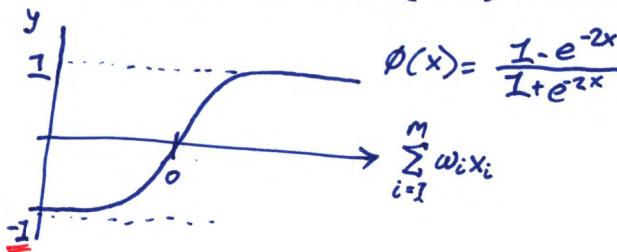
THRESHOLD FUNCTION



SIGMOID FUNCTION



HYPERBOLIC TANGENT (TANH) FUNCTION



GRADIENT DESCENT.

DETERMINISTIC ALGORITHM
(ALWAYS GET THE SAME RESULTS FOR THE SAME WEIGHTS ARE UPDATED)

BATCH GRADIENT DESCENT

PLUG ALL ROWS INTO NEURAL NETWORK & THEN UPDATE WEIGHTS AFTER CALCULATING THE COST FUNCTION

RANDOM ALGORITHM

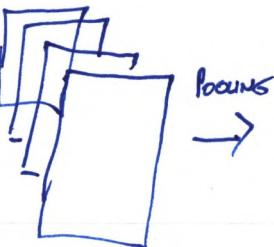
STOCHASTIC }
GRADIENT DESCENT }
ITERATIVELY UPDATES WEIGHTS
AFTER PLUGGING EACH ROW INTO THE ANN.
AND UPDATING WHILE CALCULATING THE COST FUNCTION.

HELPS AVOID LOCAL MINIMUMS AND ALSO FASTER

INPUT IMAGE

0	1	0	0	0
1	0	0	0	1
0	0	0	0	0
0	1	1	1	0
1	1	0	1	0
0	0	1	0	0

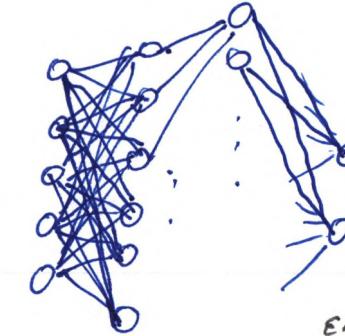
CONVOLUTION.



Pooling



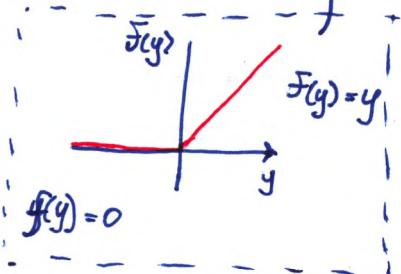
FLATTENING



EXPECTED CORRECT OUTPUTS

CONVOLUTION LAYER

POOLING LAYER



SOFTMAX FUNCTION.

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

SETS OUTPUT VALUES IN RANGE OF 0-1
AND TOTAL OF THE OUTPUT VALUES TO 1.

CROSS-ENTROPY FUNCTION

$$L_i = -\log \left(\frac{e^{z_i}}{\sum_j e^{z_j}} \right)$$

OR

$$H(p, q) = -\sum_x p(x) \log q(x)$$

SAME, BUT
BOTTOM IS
EASIER TO
CALCULATE.

SIMILAR TO THE MEAN-SQUARED
ERROR "COST" FUNCTION FOR
ASSESSING PERFORMANCE, THE
CROSS-ENTROPY FUNCTION IS A
"LOSS" FUNCTION FOR ASSESSING
PERFORMANCE OF A NETWORK
NEURAL NETWORK, AND SHOULD
BE MINIMIZED.

NN1		NN2	
DOG	CAT	DOG	CAT
0.9 ✓	0.1	1 ✓	0
0.1 ✓	0.9	0 ✓	1
0.4 ✗	0.6 ✗	1	0

$\frac{1}{3} = 0.33$ $\frac{1}{3} = 0.33$

0.25	0.71
0.38	1.06

CLASSIFICATION ERROR

WILL NOT TAKE INTO ACCOUNT
PROBABILITIES OF NETWORK OUTPUTS,
SO NETWORKS THAT WERE OFF BY THE
SAME NUMBER OF OUTPUTS, EVEN THOUGH
THE PROBABILITIES OF CORRECT OUTPUTS WERE
FURTHER OFF, WOULD BE REPORTED AS
HAVING THE SAME CLASSIFICATION ERROR.

MEAN SQUARED ERROR.

CROSS-ENTROPY ERROR.

STEP 1 - CONVOLUTION (RESULTS IN FEATURE MAP)

0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0

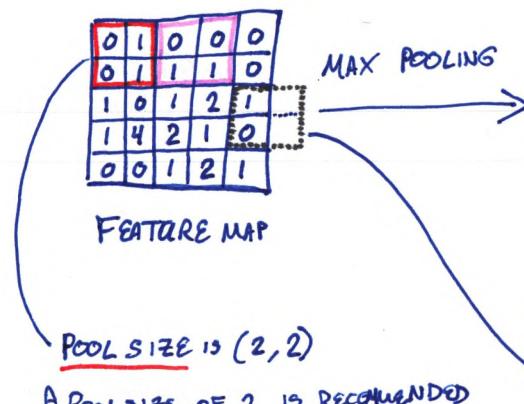
$$\otimes \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} =$$

FEATURE DETECTOR

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

FEATURE MAP

STEP 2 - MAX POOLING



POOL SIZE IS (2, 2)

A POOL SIZE OF 2 IS RECOMMENDED WHEN WORKING WITH MAX POOLING.

THE STRIDES VALUE IS 2 (RECOMMENDED), WHICH MEANS THAT THE 2x2 POOL THAT TRAVERSES THE FEATURE MAP SHIFTS 2 SPACES TO THE RIGHT AFTER EACH STEP RATHER THAN 1 SPACE TO THE RIGHT. THIS IS WHY THE PINK AREA POOL DOESN'T SURROUND THE SECTION PARTIALLY INCLUDED IN THE ORANGE POOL

PADDING

"VALID": IF PADDING IS SET TO 'VALID' THE EXTRA CELLS WOULD BE IGNORED WHEN CALCULATING THE POOLED FEATURE MAP VALUES.

- IGNORED BY SETTING TO ZERO

"SAME": IF PADDING IS SET TO 'SAME', THE EXTRACELLS ARE SET TO ZERO FOR THE PURPOSE OF CALCULATING THE POOLED FEATURE MAP VALUES.