2023.12.03 MACHINE LEARNING A-Z

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) THROW IN 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-VALUE

IF P > SL, GO TO STEP 4 OW GO TO FIN

STEP 4: REMOVE THE PREDICTOR

STEP 5: FIT MODEL WITHOUT THIS VARIABLE.

3) FORWARD SELECTION

STEP 1: SELECT A SIGNIFICANCE LEVEL TO ENTER INTO THE MODEL (E.G. SL=0.05)

STEP 2: FIT ALL SIMPLE REGRESSION MODELS. $y \sim x_n$

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, OW 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

$\therefore 2^N - 1$ TOTAL MODELS THERE CAN POSSIBLY BE.

STEP 3: SELECT THE MODEL WITH THE BEST CRITERION.

2023.12.10 MACHINE LEARNING A-Z

NO NEED TO APPLY FEATURE SCALING
FOR MULTIPLE LINEAR REGRESSION

☐ WHY?

2023.12.15

IMPORT LIBRARIES

① import numpy as nm
" ~~matplotlib~~ matplotlib as plt
" pandas as pd

IMPORT DATASET

② dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values # everything before last column.
~~y = dataset.iloc[:, :-~~
y = dataset.iloc[:, -1].values # last column

From sklearn.compose impose ColumnTransformer.
From sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder()

2023.12.15 MACHINE LEARNING A-Z MULTIPLE LINEAR REGRESSION

#E.6.

Import libraries ①

```
import numpy as nm
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))
```

TRANSFORMATION TYPE · ENCODER TYPE · INDICES TO ENCODE

DON'T TOUCH DATA THAT WON'T BE TRANSFORMED.

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_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

⑤ TRAINING MULTIPLE LINEAR REGRESSION MODEL ON THE TRAINING SET

```
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))
```

row to column · row to column · # print prediction inline with actual value

⑦ MAKE A SINGLE PREDICTION

CALIFORNIA · NEW YORK · FLORIDA · R&D SPEND · ADMIN. SPEND · MARKETING SPEND

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

⑧ GET FINAL LINEAR REGRESSION EQUATION

```
~~print(regressor._coef~~
print(regressor.coef_)
print(regressor.intercept_)
```

```
[ 8.66e+01,
 -8.73e+02,
  7.86e+02,
  7.73e+01,
  3.29e-02,
  3.66e-02]
42467.52924853204
```

~~∴ profit = 8.66 × DummyState1 − 873~~

∴ profit = 866 × DummyState1
− 873 × DummyState2
+ 786 × DummyState3
+ 0.773 × R&D Spend
+ 0.0329 × Admin Spend
+ 0.0366 × Marketing Spend
+ 42467.53

2023.12.29 MACHINELEARNING A-Z

SCIKIT LEARN > API > METRICS.
WEBSITE

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

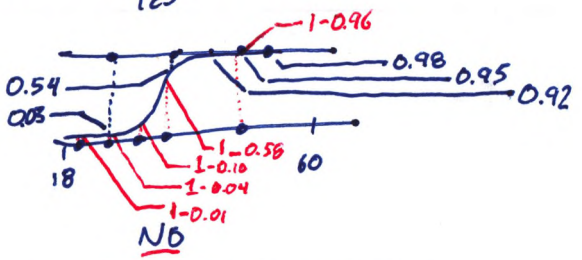
SECTION 16

LOGISTIC REGRESSION

WILL PURCHASE HEALTH INSURANCE. AGE INCOME EDU LEVEL MARITAL STATUS

$$\ln \frac{p}{1-p} = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4$$

MAXIMUM LIKELIHOOD



LIKELIHOOD = 0.03 * 0.54 * 0.92 * 0.95 * 0.98 *
FOR CURVE (1-0.01) * (1-0.04) * (1-0.10) * (1-0.58) * (1-0.96)

= 0.00001993

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_test) # 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 DATAPOINT, (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.

NOTE USE A MINKOWSKI METRIC (DEFAULT) WITH P=2 (DEFAULT) FOR EUCLIDEAN DISTANCE-BASED CLASSIFICATION.

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

MACHINE LEARNING A-Z 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

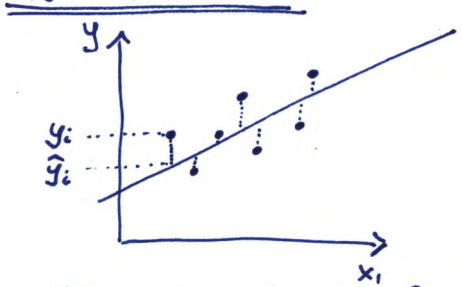
STEP 2 BUILD THE DECISION TREE ASSOCIATED WITH THESE K DATA POINTS

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.

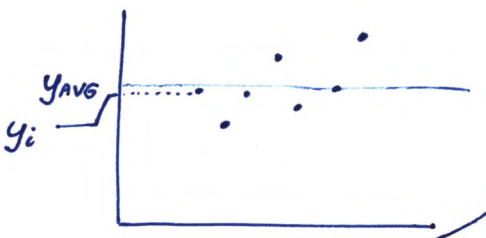
ENSEMBLE LEARNING:
TAKING MULTIPLE ALGORITHMS, OR ~~THE~~ A SAMPLE ALGORITHM MULTIPLE TIMES, AND PUT THEM TOGETHER TO RESULT IN A MUCH MORE POWERFUL VERSION.

R SQUARED



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

RESIDUAL SUM SQUARED



TOTAL SUM SQUARED.

$$SS_{tot} = SUM(y_i - y_{AVG})^2$$

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

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.

R^2 WILL RANGE BETWEEN 0 & 1

R^2 : GOODNESS OF FIT (GREATER IS BETTER)

ADJUSTED R SQUARED

HELPS DETERMINE WHETHER ADDITIONAL INDEPENDENT VARIABLES ARE WORTHWHILE

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.

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

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

JUSTIFICATION : ONLY ADD VARIABLES WHEN THEY BRING SUBSTANTIAL IMPROVEMENT.

k = # INDEPENDENT VARIABLES

n = SAMPLE SIZE

THE ADJUSTED R^2 FORMULA ~~TRIES~~ IS AN APPROACH TO RESOLVE THE PROBLEM OF ADDING VARIABLE THAT ARE IRRELEVANT OR THAT ARE OF LITTLE SIGNIFICANCE.

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + b_3X_3$$

ADDITIONAL IND EPENDENT 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
45000	1
60000	2
75000	3

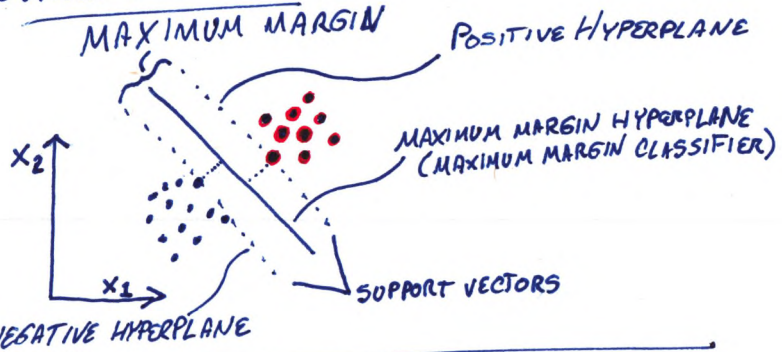
=> LEVEL VALUES ARE CONSIDERABLY LOWER, SO SCALE THE SALARY

DON'T APPLY FEATURE SCALING TO:

- BINARY VALUES
- DUMMY VARIABLES RESULTING FROM ONEHOTENCODING.

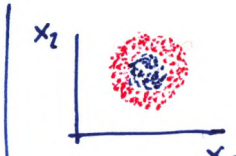
MACHINE LEARNING A-Z 2021-01-05 A

SUPPORT VECTOR MACHINE (SVM)



KERNEL SVM INTUITION

LINEARLY SEPARABLE

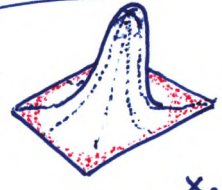


NOT LINEARLY SEPARABLE.

HIGHER DIMENSIONAL SPACE

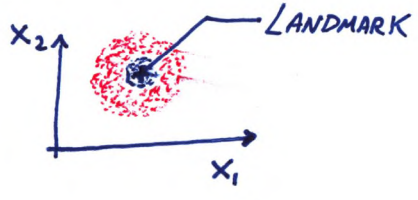
HOW TO TAKE A NON-LINEARLY SEPARABLE DATASET, MAP IT TO A HIGHER DIMENSION ~~AND~~ TO GET A LINEARLY SEPARABLE DATASET, INVOKE THE SVM ALGORITHM, BUILD A DECISION BOUNDARY FOR THE DATASET, AND PROJECT BACK TO ORIGINAL DIMENSIONS.

KERNEL TRICK

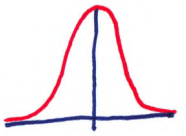


$$K(\vec{x}, \vec{l^i}) = e^{-\left[\frac{\|\vec{x} - \vec{l^i}\|^2}{2\sigma^2}\right]}$$

LANDMARK

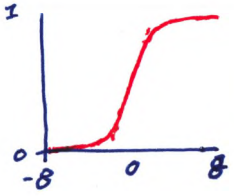


TYPES OF KERNEL FUNCTIONS



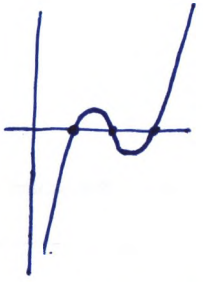
GAUSSIAN RBF KERNEL $K(\vec{x}, \vec{l^i}) = e^{-\frac{\|\vec{x} - \vec{l^i}\|^2}{2\sigma^2}}$

SELECT A LANDMARK
DISTANCE FROM LANDMARK USED TO CLASSIFY



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

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

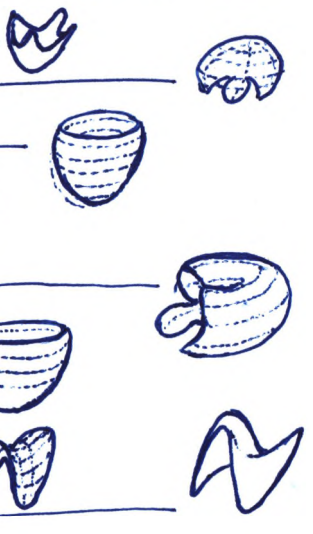


POLYNOMIAL KERNEL $K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$

SUPER DATASCIENCE - MAKING THE COMPLEX SIMPLE

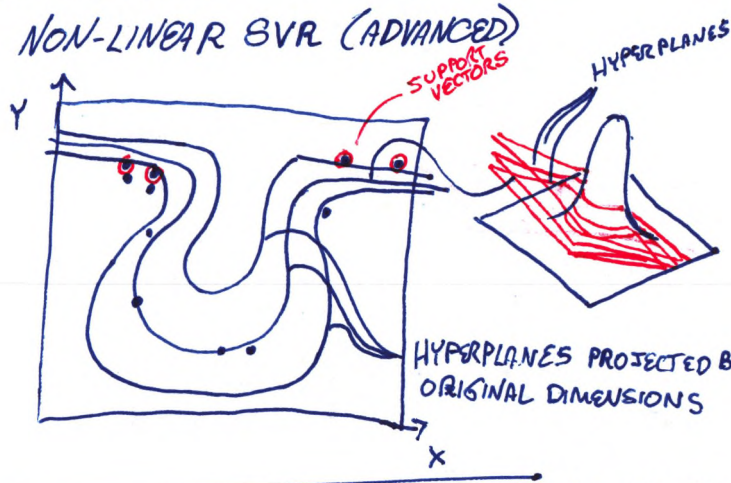
m1kernels.readthedocs.io/en/latest/kernelfunctions.html

SCALAR PRODUCT KERNEL
SQUARED-DISTANCE KERNEL
SINE-SQUARED KERNEL
CHI-SQUARED KERNEL
GAUSSIAN KERNEL
LAPLACIAN KERNEL
PERIODIC KERNEL
RATIONAL-QUADRATIC KERNEL
MATERN KERNEL
LINEAR&POLYNOMIAL KERNEL
SIGMOID KERNEL



MACHINE LEARNING A-Z 2024-01-05B

NON-LINEAR SVR (ADVANCED)



BAYES THEOREM

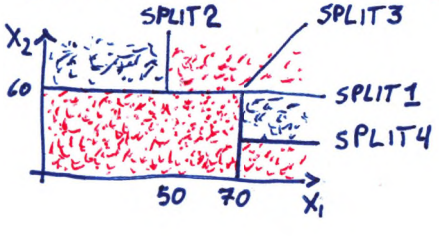
$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- $P(A|B)$: PROBABILITY OF A, GIVEN B.
- $P(B|A)$: PROBABILITY OF B, GIVEN A.

NAIVE BAYES.

DECISION TREE INTUITION

SPLITS TRY TO MINIMIZE INFORMATIONAL ENTROPY



RANDOM FOREST CLASSIFIER

AN ENSEMBLE LEARNING APPROACH

DIFFERENT CONFUSION MATRICES

A)

PREDICTED LABEL \ ACTUAL LABEL	1	0
1	TP	FP
0	FF	TN

B)

PREDICTED LABEL \ ACTUAL LABEL	0	1
0	TN	FN
1	FP	TP

C)

ACTUAL LABEL \ PREDICTED LABEL	1	0
1	TP	FN
0	FP	TN

D)

ACTUAL LABEL \ PREDICTED LABEL	0	1
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% ↑

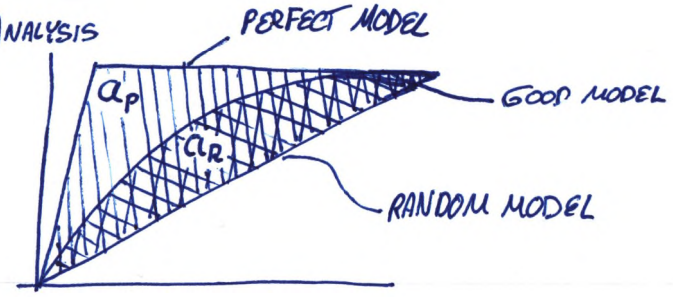
CUMULATIVE ACCURACY PROFILE (CAP)

NOT THE SAME THING.

(ROC) RECEIVER OPERATING CHARACTERISTIC

MACHINE LEARNING A-Z 2024-01-05 C

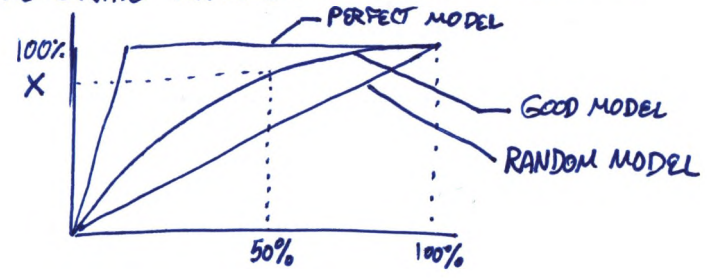
CAP ANALYSIS



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

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

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)
  - I MAY BE OVERFITTING 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.

MACHINE LEARNING A-Z 2024-01-05 D

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 ~~EACH~~ DATAPOINTS BELONGING TO EACH CENTROID (E.G. AVG OF ALL OF A CENTROIDS DATA POINTS THEIR)
5) MOVE THE CENTOIDS TO THE CENTER OF MASS CORRESPONDING TO ~~IT~~ DATAPOINTS
6) REASSIGN DATAPOINTS TO THE NEAREST CENTROID
7) REPEAT STEPS 4-6 UNTIL REPEATING THE STEPS NO LONGER RESULTS IN CHANGES

ELBOW METHOD (HELPS DECIDE HOW MANY CLUSTERS TO SELECT)

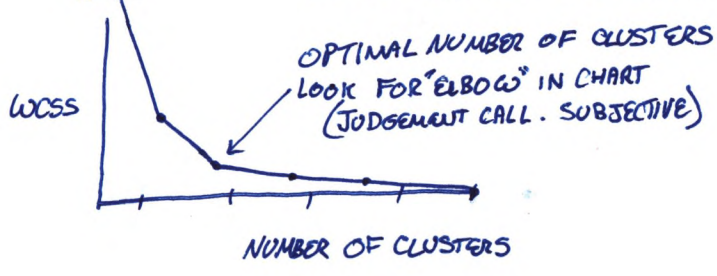
WITHIN CLUSTER SUM OF SQUARES (WCSS)

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

~~SQUARE SUM OF DISTANCE~~

~~SOME~~ 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 DATAPOINTS ~~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 DATAPOINTS 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.

MACHINE LEARNING A-Z 2024.01.16 A

HIERARCHICAL CLUSTERING INTUITION. (HCI)

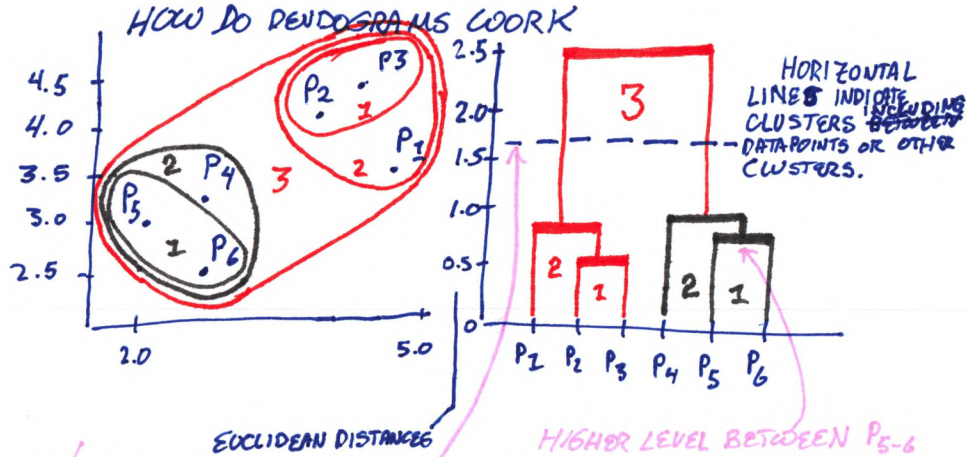
CAN RESULT IN SIMILAR RESULT AS K-MEANS CLUSTERING.

TYPES OF HIERARCHICAL CLUSTERING (HC):
- AGGLOMERATIVE (BOTTOM UP APPROACH)
- DIVISIVE (TOP DOWN)

AGGLOMERATIVE HC

1) MAKE EACH DATA POINT A SINGLE POINT CLUSTER
   ↳ THIS FORMS N CLUSTERS
2) TAKE THE 2 CLOSEST ~~CLUSTERS~~ ~~DATA POINTS~~ DATA POINTS AND MAKE THEM ONE CLUSTER
   ↳ THIS FORMS N-1 CLUSTERS
3) TAKE THE 2 CLOSEST CLUSTERS AND MAKE THEM ONE CLUSTER
   ↳ THIS ~~MAKES~~ FORMS N-2 CLUSTERS.
   ⋮
4) REPEAT UNTIL THERE IS ONLY 1 CLUSTER

HOW DO DENDOGRAMS WORK



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 P_{5-6} THAN P_{2-3} INDICATES A FARTHER DISTANCE BETWEEN P_{5-6} THAN BETWEEN P_{2-3}.

DENDOGRAMS ARE BASED ON WITHIN-CLUSTER DENDOGRAMS.

HC PERFORMS WORSE THAN K-MEANS ON LARGE DATASETS

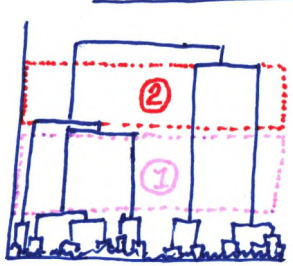
HOW IS THE DISTANCE BETWEEN TWO CLUSTERS DETERMINED:

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.

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 DISSIMILARITY 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 • PREFERED
- ECLAT

TABLE SHOULD BE HERE.

APRIORI ALGORITHM

3 PARTS: SUPPORT, CONIDENCE, & LIFT

USER ID	MOVIES LIKED
1	MOVIE 1, MOVIE 2, MOVIE 3, MOVIE 4
2	MOVIE 1, MOVIE 2
3	" " " ", MOVIE 4
4	" " " ",
5	MOVIE 2, MOVIE 4
6	MOVIE 7, MOVIE 3

POTENTIAL RULES:
- MOVIE 1 → MOVIE 2
- MOVIE 2 → MOVIE 4
- MOVIE 1 → MOVIE 3

E.G. MOVIE RECOMMENDATION:

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

HOW MANY PROPORTION WHAT ~~PERCENTAGE~~ OF PEOPLE HAVE SEEN A MOVIE

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

↳ MAKING/TESTING A HYPOTHESIS.
E.G., PEOPLE THAT HAVE SEEN MOVIE 1 ARE LIKELY TO HAVE SEEN MOVIE 2.

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

THE LIFT IS THE "IMPROVEMENT" IN YOUR PREDICTION.
E.G., IF THE CONFIDENCE VALUE IS 17.5% AND THE SUPPORT IS 10%, THE LIFT WOULD BE 1.75.

IF 10% OF ~~AN~~ A 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

MACHINE LEARNING A-Z 2024.01.16C

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.

MACHINE LEARNING A-Z 2024.01.19 A

REINFORCEMENT LEARNING.

UPPER CONFIDENCE BOUND (UCB)

"USING CONFIDENCE BOONDS FOR: EXPLOITATION AND EXPLORATION OF TRADE OFFS" AUER

"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 NUMBES, 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 $\left[\bar{r}_i(n) - \Delta_i(n), \bar{r}_i(n) + \Delta_i(n)\right]$ 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 @ 1530-16:30

NATURAL LANGUAGE PROCESSING (NLP)

COURSE WILL NOT COVER SEQ2SEQ OR CHATBOTS.

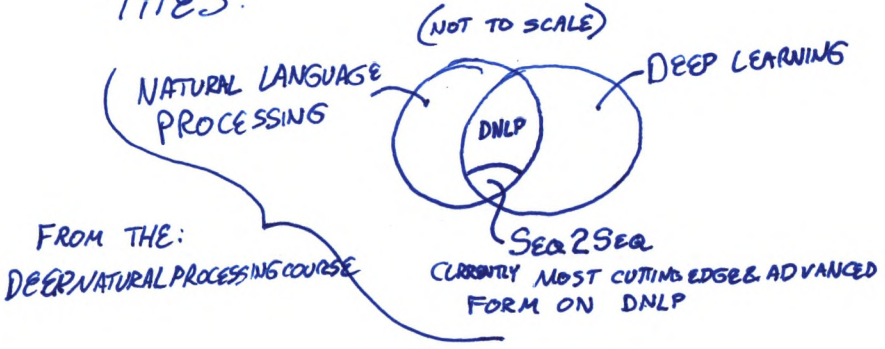
WILL COVER

TYPES OF NLP

CLASSICAL VS DEEP LEARNING MODELS

BAG-OF-WORDS. FOR SENTIMENT ANALYSIS.

TYPES:



CHATBOTS USED TO BE STRUCTURED BASED ON

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

EXAMPLE:

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

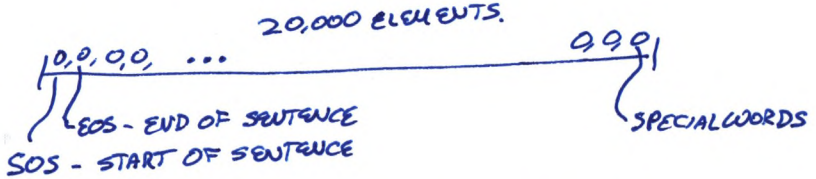
START WITH VECTOR/ARRAY ~~WITH~~ THAT'S ABOUT 20,000 ELEMENTS LONG

~~EACH~~ NATIVE ENGLISH SPEAKERS COMMONLY USE 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. ∴ ~1.75% 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~~ IN THE SPECIAL WORDS ELEMENT.
- PUNCTUATION MARKS ARE COUNTED AS ELEMENTS AS WELL. E.G. COMMAS.
- THE ARRAY WILL BE SPARSE (MANY ZEROES) MOST OF THE TIME.

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

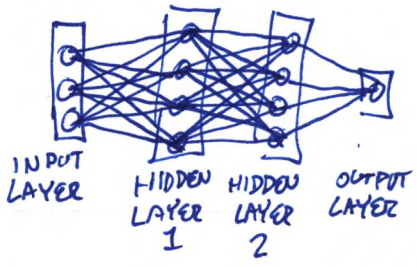
E.G. "DID YOU LIKE THE RECIPE I SENT YOU?" → "NO" / → "YES"

SENTIMENT HAPPY?

↑ APPLY LOGISTIC REGRESSION TO BAG-OF-WORDS.

L NLP REGION.

MAY APPLY A NEURAL NETWORK INSTEAD.



← DNLP REGION

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

MACHINE LEARNING A-Z 2024.02.09 (A)

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.

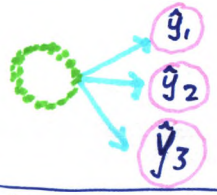
Activation Function
2nd Step

$$\phi\left(\sum_{i=1} w_i x_i\right)$$

3RD STEP

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

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



COST FUNCTION

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

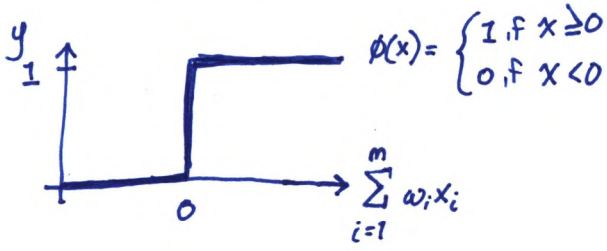
MOST COMMON COST FUNCTION, BUT THERE ARE OTHERS.

OUR GOAL IS TO MINIMIZE THE COST FUNCTION BECAUSE THE COST IS REPRESENTS THE ERROR IN YOUR PREDICTION.

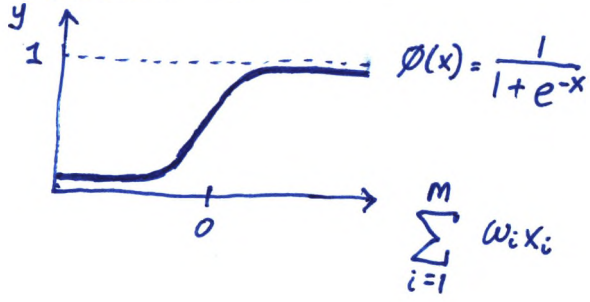
"EFFICIENT BACKPROP" BY YANN LECUN ET AL. ~~(1988)~~ 1998

THE ACTIVATION FUNCTION

THRESHOLD FUNCTION

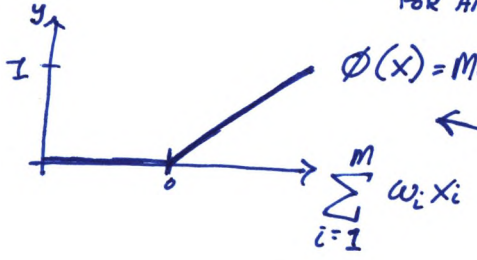


SIGMOID FUNCTION



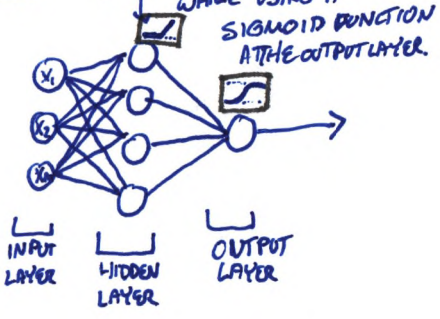
RECTIFIER FUNCTION

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

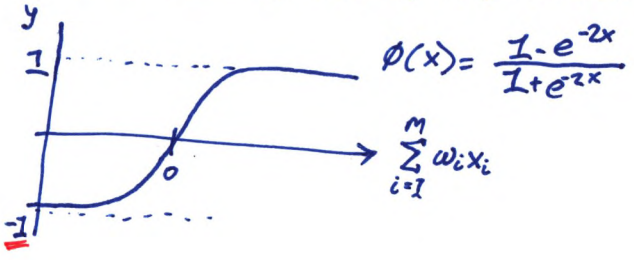


PATERN
E.G. AFTER 100+ YEARS A BUILDING'S VALUE MAY START TO INCREASE RAPIDLY BC IT BECOMES HISTORIC

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



HYPERBOLIC TANGENT (TANH) FUNCTION



MACHINELEARNING A-Z 2024.02.09 (B)

GRADIENT DESCENT.

BATCH GRADIENT DESCENT

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

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

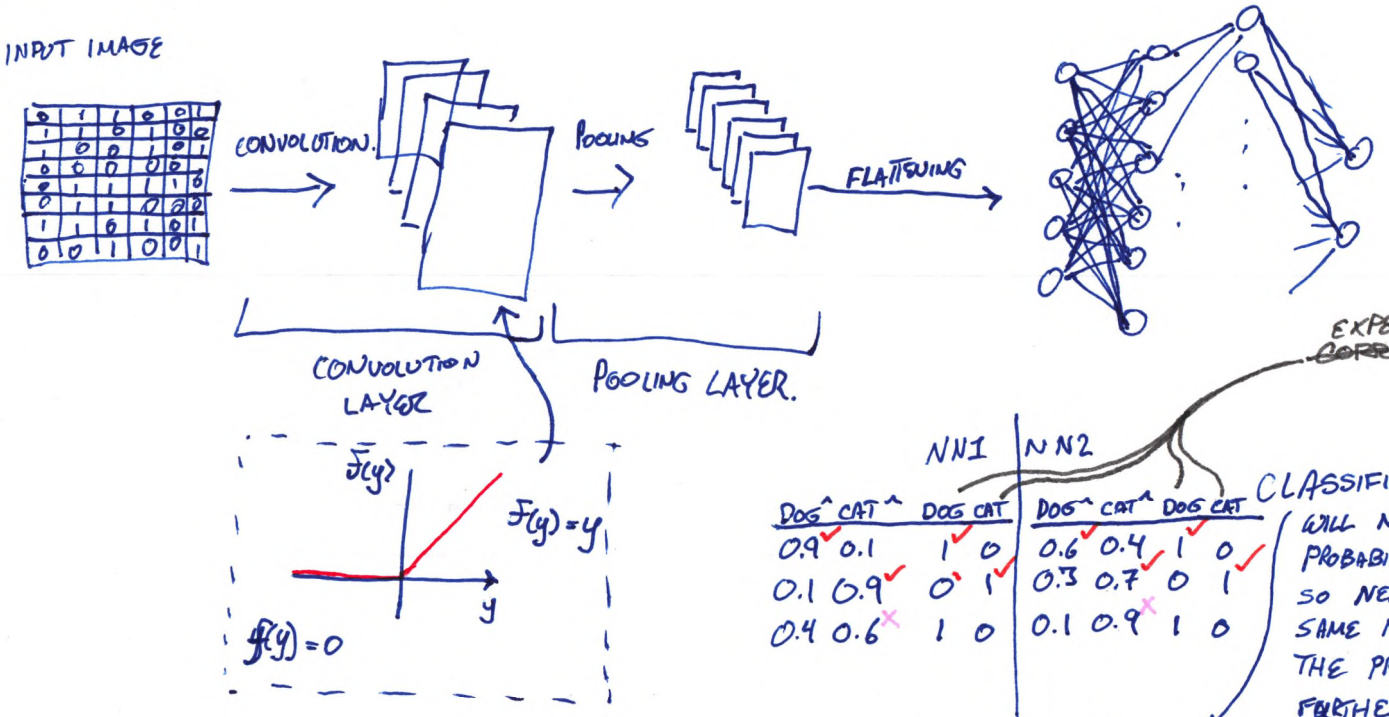
STOCHASTIC GRADIENT DESCENT

RANDOM ALGORITHM

HELPS AVOID LOCAL MINIMUMS AND ALSO FASTER

ITERATIVELY UPDATES WEIGHTS AFTER PLUGGING EACH ROW INTO THE ANN.
AND ~~UPDATING WEIGHTS~~ CALCULATING THE COST FUNCTION.

MACHINE LEARNING A-Z 2024.02.23 (A)



EXPECTED ~~CORRECT~~ OUTPUTS

NN1				NN2			
DOG	CAT	DOG	CAT	DOG	CAT	DOG	CAT
0.9 ✓	0.1	1 ✓	0	0.6 ✓	0.4	1 ✓	0
0.1	0.9 ✓	0	1 ✓	0.3	0.7 ✓	0	1 ✓
0.4	0.6 ✗	1	0	0.1	0.9 ✗	1	0
1/3 = 0.33				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.

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

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

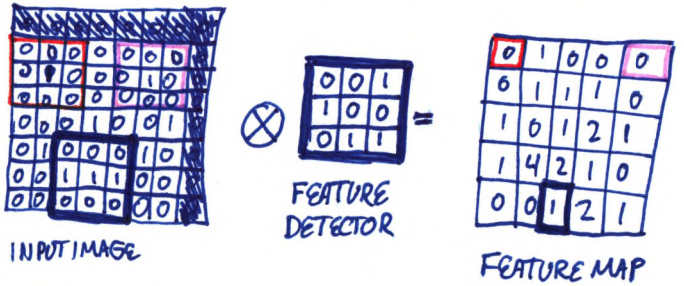
SAME, BUT BOTTOM IS EASIER TO CALCULATE.

OR

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

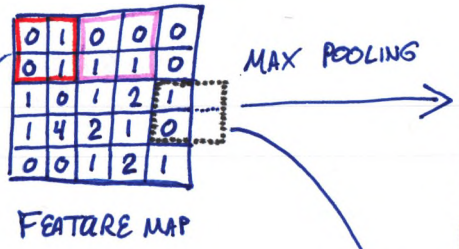
SIMILAR TO THE MEAN-SQUARED ERROR "COST" FUNCTION FOR ASSESSING PERFORMANCE, THE CROSS-ENTROPY FUNCTION IS A "LOSS" FUNCTION FOR ASSESSING PERFORMANCE OF A NEURAL NETWORK, AND SHOULD BE MINIMIZED.

STEP 1 - CONVOLUTION (RESULTS IN FEATURE MAP)



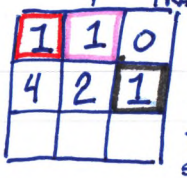
MACHINE LEARNING A-Z 2024.02.23 (B)

STEP 2 - MAX POOLING



FEATURE MAP

MAX POOLING



POOLED FEATURE MAP

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

POOL SIZE IS (2, 2)

A POOLSIZE OF 2 IS RECOMMENDED WHEN WORKING WITH MAX POOLING.

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 EXTRA CELLS ARE SET TO ZERO FOR THE PURPOSE OF CALCULATING THE POOLED FEATURE MAP VALUES.