INTRO TO ML REVISION

CURSE OF DIMENSIONALITY - INCREASED COMPUTATIONAL COMPLEXITY AND MEMORY
- DATA SPARSITY IN SOME REGIONS
- OVERFITTING

EAGER LEARNERS - PREMADE MODEL
LAZY LEARNERS - STORE DATA, EG K-NEAREST NEIGHBORS

K-NEAREST NEIGHBORS - K NEAREST DATA POINTS VOTE, CHOOSE K USING VALIDATION DATA
MANHATTAN DISTANCE = L1 NORM
EUCLIDEAN DISTANCE = L2 NORM
HAMMING = # BITS DIFFERENT
MAHALANOBIS = WEIGHTED EUCLIDEAN
DISTANCE-WEIGHTED KNN = LIKE KNN VOTE, VALUES K FARTHER LESS IF CLASSIFIER
ROBUST TO NOISY TRAINING DATA
KNN FOR REGRESSION - AVERAGE OF K VALUES OR WEIGHTED AVERAGE (SMOOTH)

BINARY CLASSIFICATION - DETECT ONE THING
MULTI-LABEL CLASSIFICATION - MULTIPLE BINARY DETECTION OF SOME THINGS
MULTI-CLASS CLASSIFICATION - PUT INTO ONE OF MANY CLASSES
BALANCED VS UNBALANCED DATASET - SOME DATA CLASSES MIGHT BE UNDER-REPRESENTED
BIAS/VARIANCE TRADEOFF ⇒ UNDERFIT = TOO SIMPLE = HIGH BIAS, LOW VARIANCE
OVERFIT = TOO COMPLEX = LOW BIAS, HIGH VARIANCE

DECISION TREES - SPLIT RULE = GINI = PROBABILITY OF CORRECT LABEL FOR RANDOM POINT
= INFORMATION GAIN = REDUCE ENTROPY

ENTROPY OF GROUP $= H_{group} = -\sum p_i(x)\log_2(p(x))$ i = EACH TYPE OF DATA POINT IN GROUP

ENTROPY OF SYSTEM $H_{tot} = \frac{N_{group}}{N_{total}} H_{group}$

INFORMATION GAIN = REDUCTION IN ENTROPY = $H_{old} - H_{new}$

p(G) =	(1/2, 1/2)	(1/4, 3/4)	(1/4, 3/4)	(1/5, 4/5)	(1/6, 5/6)	(1/7, 6/7)	(1/8, 7/8)	(1/9, 8/9)	(1/10, 9/10)
H =	1	0.9183	0.8114	0.7219	0.65	0.592	0.544	0.503	0.469
			(3/5) 0.971			(3/7) 0.863	(2/7) 0.954	(3/8) 0.764	(2/10) 0.881
						(4/7) 0.985		(5/8) 0.991	

PRUNING - PRUNE EACH FILAL NODE IF IT IMPROVES VALIDATION ACCURACY

EVALUATION **CONFUSION MATRIX**

REAL	PREDICTED +	PREDICTED −
+	TP	FN
−	FP	TN

$ACCURACY = \frac{\text{TOTAL RIGHT}}{N} = \frac{TP+TN}{TP+FP+TN+FN}$

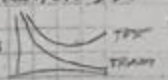
$PRECISION = \frac{TP}{TP+FP}$ ⇒ PUNISH FALSE POS

$RECALL = \frac{TP}{TP+FN}$ ⇒ PUNISH FALSE NEG

F-MEASURE ⇒ $F_1 = \frac{2\cdot PRECISION\cdot RECALL}{PRECISION+RECALL}$ $F_\beta = \frac{(1+\beta^2)\cdot PRECISION\cdot RECALL}{(\beta^2\cdot PRECISION)+RECALL}$

MULTI-CLASS
MACRO-AVERAGE = COMPUTE STAT FOR EACH CLASS AND AVERAGE
MICRO-AVERAGE = SUM UP TP, TN, FP, FN FOR ALL CLASSES AND COMPUTE STAT

OVERFITTING - INCREASED TRAINING RESULTS, DECREASING TEST RESULTS



TOO COMPLEX
- DROPOUT (RANDOM 50% DURING TRAINING)
- REDUCE COMPLEXITY OF MODEL (FEWER LAYERS, FEWER NEURONS)
- EARLY STOPPING (USE VALIDATION SET TO KNOW WHEN)
- GET MORE DATA
- L1/L2 REGULARIZATION EARLY TERMS (α = LEARNING RATE)

L1 = PENALIZE NUMBER OF NON-ZERO WEIGHTS $L1 = \lambda\sum|w_i| \Rightarrow w \leftarrow w - \alpha\left(\frac{\partial Loss}{\partial w} + \frac{\lambda w}{|w|}\right)$
L2 = RIDGE REGRESSION $L2 = \lambda\sum w_i^2 \Rightarrow w \leftarrow w - \alpha\left(\frac{\partial Loss}{\partial w} + 2\lambda w\right)$

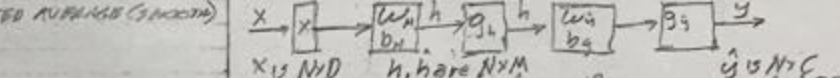
CONFIDENCE INTERVALS = WE ARE N% CERTAIN PERFORMABLE WILL BE IN INTERVAL

N% =	50%	68%	80%	85%	90%	95%	98%	99%	99.5%	99.9%
Z_n =	0.67	1	1.28	1.44	1.64	1.96	2.33	2.58	2.807	3.291

$ERROR\ RATE \pm Z_N\sqrt{\frac{(ERROR\ RATE)(1-ERROR\ RATE)}{N}}$ OR $ERRORS \pm Z_N\sqrt{\frac{ERRORS\cdot CORRECT}{N}}$

P-VALUE = CHANCE THAT THE RESULTS ARE JUST RANDOM CHANCE
STATISTICALLY SIGNIFICANT = p-VALUE < 5%

NEURAL NETS - K FEATURES, N DATA POINTS, C OUTPUT LABELS



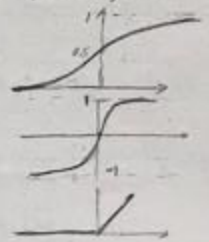
g_h, g_g = ACTIVATION FUNCTIONS
X is $N\times D$ (DATA POINTS, FEATURES)
$h, \hat{h}$ are $N\times M$ (ACTIVATION)
$\hat{y}$ is $N\times C$ (ESTIMATE CLASSIFICATION DATA POINT)
W_h is $D\times M$ b_h is $N\times M$
W_g is $M\times C$ b_g is $N\times C$

$h = X W_h + b_h$
$\hat{h} = g_h(h) = g_h(X W_h + b_h)$
$\hat{y} = g_g(\hat{h} W_g + b_g)$

LOSS FUNCTION = MSE $= \frac{1}{N}\sum_i^N (\hat{y}(i) - y(i))^2 = \frac{1}{N}\sum_i^N (w x(i) + b - y(i))^2$

UPDATES $w \leftarrow w - \alpha\frac{\partial L}{\partial w} = w - \alpha 2\sum_i^N (\hat{y}(i) - y(i))^1 x(i)$
$b \leftarrow b - \alpha\frac{\partial L}{\partial b} = b - \alpha 2\sum_i^N (\hat{y}(i) - y(i))^1$
α = LEARNING RATE

ACTIVATION FUNCTIONS

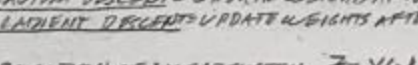


SIGMOID $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$ — ROBUST BINARY CLASSIFICATION — $0 \le f(x) \le 1$, $f'(x) = \sigma(x)(1-\sigma(x))$
TANH $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ — BINARY CLASSIFICATION — $-1 \le f(x) \le 1$, $f'(x) = 1 - f(x)^2$
RELU $f(x) = \text{ReLU}(x) = \begin{cases} 0 & x \le 0 \\ x & x \ge 0 \end{cases}$ — COMMON IN DEEP LEARNING — $0 \le f(x)$, $f'(x) = \begin{cases} 0 & x \le 0 \\ 1 & x > 0 \end{cases}$ = STEP
SOFTMAX $(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}}$ ⇒ ACCENTUATES CHOICES — MULTI-LABEL CLASSIFICATION — $0 \le f(x) \le 1$, SUM OF OUTPUTS FOR EACH DATAPOINT = 1
$\frac{\partial L}{\partial z} = \frac{1}{N}(\hat{y} - y)$ AS MATRIX

LOGISTIC REGRESSION = CLASSIFICATION, NOT REGRESSION

GRADIENT DESCENT = UPDATE WEIGHTS FROM FULL DATASET ⇒ SLOW
STOCHASTIC GRADIENT DESCENT = UPDATE WEIGHTS AFTER EACH DATA POINT ⇒ NOISY
MINI-BATCH GRADIENT DESCENT = UPDATE WEIGHTS AFTER BATCHES OF DATA ⇒ GOLDILOCKS

BACK-PROPOGATION OF LINEAR SYSTEM $Z = XW + b$



$\frac{\partial Loss}{\partial X} = \frac{\partial Loss}{\partial Z} W^T$

$\frac{\partial Loss}{\partial W} = X^T \frac{\partial Loss}{\partial Z}$ $\frac{\partial Loss}{\partial b} = \begin{bmatrix} 1 \\ \vdots \end{bmatrix} \frac{\partial Loss}{\partial Z}$ ALL ROWS OF b ARE TOGETHER

ALL ROWS OF b ALSO $\frac{\partial Loss}{\partial b}$ ARE IDENTICAL

DERIVATIVE IDENTITIES

$z = wx \Rightarrow \frac{\partial z}{\partial x} = w$
$z = xw \Rightarrow \frac{\partial z}{\partial x} = w^T$
CHAIN RULE

$\frac{\partial}{\partial x}[f(x)g(x)] = f'(x)g(x) + f(x)g'(x)$
$\frac{\partial}{\partial x}\left[\frac{f(x)}{g(x)}\right] = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}$
$\frac{\partial}{\partial x}[f(g(x))] = f'(g(x))\,g'(x)$

$\frac{\partial}{\partial x} e^x = e^x$
$\frac{\partial}{\partial x} a^x = a^x \ln(a)$
$\frac{\partial}{\partial x}(\ln(x)) = \frac{1}{x}$
$\frac{\partial}{\partial x}(\log_b(x)) = \frac{1}{x\ln(b)}$

CROSS-ENTROPY = LOSS FOR CLASSIFICATION = CHANCE OF MISTAKES
= NEGATIVE LOG-LIKELIHOOD

y = ACTUAL, $\hat{y}$ = PREDICTED

BINARY CROSSENTROPY $\Rightarrow L = -\frac{1}{N}\sum^{N}(y(i)\log\hat{y}(i)) + (1-y(i))\log(1-\hat{y}(i))$

CATEGORICAL CROSS-ENTROPY $\Rightarrow L = -\frac{1}{N}\sum^{N}\sum_{c} y_c(i)\log(\hat{y}_c(i))$ N=SAMPLES, C=CATEGORY CLASSES

FOR MULTI-CLASS CLASSIFICATION

$\prod^{N} p(y_i|x_i;\theta)$

FOR DERIVATIVE, SEE LOGISTIC REGRESSION: $\frac{dL}{dy} = -\frac{1}{N}\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)$

PROBLEM	TYPE	OUTPUT ACTIVATION	LOSS
FUTURE STOCK PRICES	LINEAR REGRESSION	LINEAR	MSE
TUMOR DETECTION	BINARY CLASSIFICATION	SIGMOID	BINARY CROSS-ENTROPY
CHEMICAL COMPOSITION	MULTI-LABEL CLASSIF.	MULTIPLE SIGMOID	BINARY CROSS-ENTROPY
CHARACTER RECOGNITION	MULTI-CLASS CLASSIF.	SOFTMAX	CATEGORICAL CROSS-ENTROPY

ADAPTIVE LEARNING RATES - DIFFERENT LEARNING RATES FOR EACH PARAMETER
RECOMMEND ADAM, ADAD, RMS
- RULE OF THUMB $LR \propto \frac{1}{m}$
- ADJUST LEARNING DOWN IF BOUNCE, UP IF STUCK

WEIGHT INITIALIZATION $b=0$ ⇒ DON'T INTRODUCE BIAS
W = RANDOM EG NORMAL = $N(x|0,1)$ OR $N(x|0,\sigma_i)$
XAVIER GLOROT = UNIFORM $\left(-\sqrt{\frac{6}{N_i+N_{i+1}}}, \sqrt{\frac{6}{N_i+N_{i+1}}}\right)$

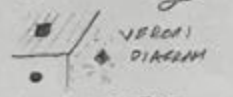
N_i = NEURONS IN PREVIOUS LAYER, N_{i+1} = NEURONS IN FOLLOWING

DATA NORMALIZATION
MIN-MAX = SCALE BETWEEN $[a,b]$ EG $[0,1]$, $[-1,1]$

$x' = a + \frac{(x-x_{min})(b-a)}{(x_{max}-x_{min})}$

STANDARDIZATION = Z-NORMALIZATION = 0 MEAN, STD $\sigma = 1$ $x' = \frac{x-\mu}{\sigma}$

UNSUPERVISED LEARNING (CLUSTERING)
K-MEANS
- INITIALIZE K RANDOM CENTROIDS
- ASSIGN EVERY POINT TO A CENTROID
- RE-EVALUATE AVERAGE CENTROID AND REPEAT UNTIL CONVERGENCE


VORONOI DIAGRAM

TO CHOOSE K, USE THE ELBOW METHOD OR CROSS-VALIDATION
INCREASE K UNTIL THE LOSS FUNCTION FLATTENS OUT $L(\theta) = \sum_i^N (x_i - c_i)^2$ CENTROID

K-MEANS WEAKNESSES = SENSITIVE TO INITIAL CENTROIDS
- SENSITIVE TO OUTLIERS
- NOT SUITABLE WHEN NOT LINEARLY SEPARABLE


EG X IS IN UNIFORM CIRCLE

PROBABILITY DENSITY FUNCTIONS
NON-PARAMETRIC
KERNEL DENSITY ESTIMATION $\hat{p}(x) = \frac{1}{N}\sum^{N}\frac{1}{h^D}H\left(\frac{x-x(i)}{h}\right)$ $H(u) = \begin{cases} 1 & |u| < 1/2 \\ 0 & \text{else} \end{cases}$

h = SMOOTHING HISTOGRAM BIN WIDTH, D = # DIMENSIONS, WIDER h = SMOOTHER

GAUSSIAN KERNEL $\hat{p}(x) = \frac{1}{N}\sum^{N}\frac{1}{(2\pi h^2)^{D/2}}\exp\left(-\frac{\|x-x(i)\|^2}{2h^2}\right)$

PARAMETRIC GMM = GAUSSIAN MIXTURE MODEL ⇒ EVERY POINT BELONGS PARTIALLY TO MULTIPLE CLASSES, NOT ONE CLASS AS IN K-MEANS "SOFT CLUSTERING"

ASSUME $p(x) = \sum^{K}\pi_k N(x|\mu_k, \Sigma_k)$

EXPECTATION MAXIMIZATION TO GET VALUES
1) RANDOMLY INITIALIZE π_k, μ_k, Σ_k
2) COMPUTE RESPONSIBILITIES OF EACH GAUSSIAN (k) FOR EACH TRAINING SAMPLE $x(i)$

$r_{ik} = \frac{\pi_k N(x(i)|\mu_k, \Sigma_k)}{\sum^{K}\pi_j N(x(i)|\mu_j, \Sigma_j)}$

3) UPDATE THE MEAN AND COVARIANCE

$\mu_k = \frac{1}{\sum r_{ik}}\sum^{N} r_{ik}x(i)$ $\Sigma_k = \frac{1}{\sum r_{ik}}\sum_{i=1}^{N} r_{ik}(x(i)-\mu_k)(x(i)-\mu_k)^T$

4) UPDATE MIXING PROBABILITY $\pi_k = \frac{1}{N}\sum_i^N r_{ik}$

HOW TO PICK K? CROSS-VALIDATION OR MINIMIZE $BIC \approx f(k) + \frac{k}{2}\log(N)$ (NEGATIVE LOG-LIKELIHOOD, NUMBER OF PARAMS)

# EVOLUTIONARY ALGORITHMS
PHENOTYPE = PHYSICAL EXPRESSION OF GENOTYPE (GENES), EG. RECESSIVE TRAIT OF REDHAIR

FITNESS FUNCTION = PERFORMANCE AGAINST TASK

GENETIC ALGORITHM
GENOTYPE - REPRESENT SOLUTION AS BINARY STRING
CROSSOVER - SELECT TWO PARENTS, COMBINE CHUNKS FROM BOTH
MUTATION - RANDOMLY FLIP BITS

SELECTION OPERATORS
BIASED ROULETTE WHEEL - EACH INDIVIDUAL HAS A CHANCE TO BE SELECTED PROPORTIONAL TO THEIR FITNESS
$p_i = \frac{f_i}{\sum f_k}$
TOURNAMENT — RANDOMLY DRAW TWO INDIVIDUALS AND ADD THE WINNER TO THE POOL OF PARENTS

ELITISM - KEEP THE TOP 10% OF THE "PARENT" GENERATION
AS OPPOSED TO GENERAL EVOL ALGORITHMS, WHERE CHILD ENTIRELY REPLACE PARENTS ENTIRELY

STOPPING CRITERIA - SET NUMBER OF GENERATIONS
- ACHIEVE FITNESS GOAL
- FITNESS STAGNATES, STOPS IMPROVING

$(\mu+\lambda)$-EVOLUTIONARY STRATEGY
1) RANDOMLY GENERATE POPULATION
2) SELECT μ BEST TO BE PARENTS
3) ADD GAUSSIAN NOISE TO MAKE λ OFFSPRING $Y = X + N(0,\sigma)$ $\frac{\lambda}{\mu} \approx 5$
4) MIX TO CREATE POPULATION SIZE $\mu+\lambda$ AND REPEAT

NOVELTY SEARCH = KEEP ALL SOLUTIONS
BEHAVIOURAL DESCRIPTOR - CHARACTERIZES TYPE OF SOLUTION
OPTIMIZE THE NOVELTY AS THE DISTANCE IN BEHAV. DESCR. SPACE TO OTHER INDIVIDUALS

QUALITY-DIVERSITY OPTIMIZATION - USE FITNESS FUNC + BEHAVIORAL DESCRIPTOR
NOVELTY SEARCH W/ LOCAL COMPETITION - KEEP THE BEST IN EACH "NEIGHBORHOOD"
MAP-ELITES - DIVIDE BEHAVIORAL DESCRIPTOR SPACE INTO GRIDS
- SELECT A PARENT AND MUTATE IT TO MAKE NEW INDIVIDUAL
CAN CHOOSE BASED ON FITNESS, NOVELTY, CURIOSITY, RANDOM
- KEEP THE BEST IN EACH GRID
EASY TO IMPLEMENT, BUT MAYBE NOT UNIFORMLY DIVERSE, SOME BRITTLENESS

QUALITY-DIVERSITY SCORE = SUM OF ALL FITNESS