Amazon ML Summer School

Matrix multiplication as composition

It is my experience that proofs involving matrices can be shortened by 50% if one throws the matrices out.

— Emil Artin

Linear transformation is determined by the basis of vector space $\hat{i}$, $\hat{j}$ — 2-D

Grid lines remain parallel and evenly spaced has a wonderful consequence.

2×2 matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = x\begin{bmatrix} a \\ c \end{bmatrix} + y\begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

$f(g(x))$ Composition of a rotation and a shear

↳ Record where $\hat{i}$ and $\hat{j}$ land

$$\underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\text{Shear}}\left(\underbrace{\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}}_{\text{rotation}}\begin{bmatrix} 3 \\ 4 \end{bmatrix}\right) = \underbrace{\begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}}_{\text{composition}}\begin{bmatrix} x \\ y \end{bmatrix}$$

Eigenvectors and Eigenvalues.

→ Vector gets knocked off span

but sometimes remain

Transformation

$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ → stretch by 1 / stretch by 3 — Eigenvalues, Eigen vectors

Matrix vector multiplication

Matrix

$A\vec{v} = \lambda\vec{v}$ → scalar multiplication

Eigenvalues

Eigenvectors

$(A - \lambda I)\vec{v} = \vec{0}$ → we need $\det(A - \lambda I) = 0$

↑ solutification

Supervised learning

Appln
email → spam (0/1) = spam filtering

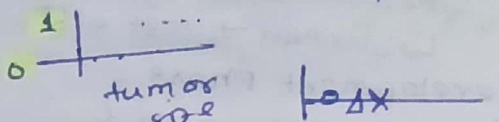
i/p → output labels

House pricing prediction

Learns from being given right answers
↳ algorithm ⇒ regression : predict a number
↳ infinitely many possible o/p

2. classification algorithm
↳ breast cancer classification → malignant / benign
↓
class
predict category.

size	diagnosis
2	0
5	1
1	0
7	1



o benign
x malignant
Δ malignant type 2

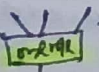
Unsupervised learning
↳ Find something interesting in unlabeled data.
clustering algorithm → google news : similar words.
↳ DNA microarray

Anomaly detection
↳ find unusual data points

dimensionality reduction
↳ compress data

Naive Bayes ⇒ $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$
→ Gaussian : contin var following normal distribution
→ Multinomial : discrete data
Bernoulli's : Binary data features
↓ → Probabilistic machine learning algorithm
assumes features are conditionally independent given the class label

x regression
↳ Decision trees → mean / median of the group
step func. ← ↙ disadv. small change in data ? completely changed decision tree.
methods to combat overfitting
1. pre-pruning (early stopping) → max depth
2. post pruning

Random forest
↳ ensemble method
Bootstrapping

→ Decision trees
↙ o/p = ~~per~~ prediction
↳ Less prone to overfitting
more hyper parameters.
can handle dataset with high dimensionality

majority vote ←
+ Distance metric
K-~~nearest neighbour~~ nearest -neighbours (KNN)
↑ hyperparameter
↳ lazy learning algo
↙ ↑
scaling is very important training data
↖ assign weights.

Support vector Machines (SVM) → high performance, computationally slow.
↓
margin
↓
found using Lagrange multipliers. → C = allows some misclassification
↓
Kernel trick : algorithm in hyperplane
(RBF)

DevOps lifecycle.

Methodology/Process → development + operation team → deliver services/product more effⁿ

1. Operation Phase
minimizing plan downtime
schedule or maintainance
↳ business → cost effⁿ

2. Development Phase
↳ continuous process
write code whenever modification are required.

3. Integration Phase
↳ Development of new features + functionality / Updating of existing tool

4. Testing phase
continuous loop in which code is tested before each commit
S/W changes are verified

5. Monitoring Phase
detect: performance, bugs, issues
Understanding causes.

6. Feedback phase

7. Deployment phase.
configuration management
process of smooth deployment of code on the servers + schedule updaradation

Data Encryption for Beginners

For protecting sensitive information
↳ stored, transmitted.

converting readable data → encoded format (cipher text)
↑ cryptographic key management algo + keys

AWS KMS system service → integrated with AWS
↳ Access is governed by the key policy → control → when, how, who

1. databases
2. analytics
3. storage
4. workflow
5. producivity tools.

Encryption & Transit
↳ protects data while it is moving between system across network
↓ prevents
1. eavesdropping
2. data breaches.
(HTTPS, TLS)

cloudtrail → record all API req. → read your data → conduct forensic analysis

DevOps CI/CD explanation

continuous integration → automate things ✓ → higher code quality + small fault detection early

3. commands: test, build, deploy → Github actions (workflow ↑ push to a master branch)

How to evaluate ML models

↳ evaluation metrics

1. classification task : Accuracy = $\frac{\text{correctly classified instances}}{\text{all instances}}$

2. precision and recalls → TP = True +ve, FP = False +ve, TN = True -ve, FN = False -ve

[Binary classification]

→ $\frac{TP}{TP+FP}$ = % of correctly labelled +ve instances out of all +ve labelled instances.

→ $\frac{TP}{TP+FN}$ = % of correctly labelled +ve instances out of all +ve instances.

3. F1 score → harmonic mean = $\frac{2}{(1/P)+(1/R)}$

↳ ROC curve compare TP with FP

↳ AUC → ✓ performance maximize area (Area under curve)

4. Cross entropy → Binary, Category [Keras library]

5. Regression evaluation metric

i) Mean Absolute error (MAE) = $\frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$

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

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

↓

R^2↑ → coeff^n of determination

iv) Cosine similarity : vector similarity.

Machine Learning cheatsheet

1. formal learning theory
   + vc-dimension

1 PAC learning : a framework : how well an algorithm can → learn from algorithm
↳ high confidence : learned hypothesis has low error

2. vc-dimension
↳ max no of points
↳ can be shattered by a hypothesis class

2. Regression
+ logistic model:

Linear → predicts contin values.
logistic → " for classification using a sigmoid.

Cross-entropy
↳ difference
1. predicted probability
2. actual class labels.

3. ensemble learning

Bagging → ↓ variance on bootstrapped samples.

Boosting → sequentially trains : misclassified samples

Adaboost
↳ ↑ weights : hard to classify points to focus subsequent layers.

4. SVM & kernel methods.

↓ maximize margins

↓ maps : data to higher dimensional space
w/o explicit transformation.

RBF : similarity : distance

5. Neural N/W

Activation func : non-linearity : (ReLU, sigmoid, tanh)
→ mitigate vanishing gradient + speed up convergence.

Backpropagation : to update weights ↳ gradient descent