Course 3

FUNDAMENTALS of MACHINE LEARNING for HEALTHCARE

Module 1

healthcare ecosystem
- AI developer
- tech company
- policy maker
- regulator
- health care system leadership
- pharma
- medical device
- frontline clinicians
- ethicist
- patients
- carers

standard unit — AS — aid

SUMEX - AIM — resulted in → images (retina), DG, to AI handbook projects

med experimental computer

1970s – New England Journal of Med → Computer Science displacing workforce

AI medicine ← AI algorithms for complex models
- access to large volumes of data
- powerful graphical processing unit or GPU's → massive parallel computation

AI can help healthcare systems → improve access
- workforce burnout
- ↑ quality

Unresolved Concerns about AI in healthcare

- Workforce displacement
- Skill atrophy
- algorithmic + over bias
- Patient privacy
- medico-legal responsibility
- oversight + regulation

early application of ML in healthcare

- care delivery (enhancing, optimising)
- info MGMT (improving)
- cognitive support for clinical care
- prediction
- risk assessment
- early detection

Machine learning

→ used by data scientist
→ statistical

→ statistical + mathematical modelling techniques that learn from data without explicit programming

algebra statistics probability	math foundations

AI

- used for dissemination

- development of the development of machine capabilities to simulate learning or other features of intelligence.

	Statistics	vs. ML
BG	statistics, data science	computer science, engineering
approach	hypothesis-driven model development	creating system that learn from data
goal	inferences relationships between variables	optimisation prediction accuracy
assumptions	need population info	—
success	discoveries that can be applied for new predictions learn new insights	resulting model produces accurate predictions without predefined characteristics ↓ more performance based

Programming

input → $f(x)$ [ ] → output

x → $f(x)$ → y

ML

Code all ≠ permutations and decisions with the input data for an accurate output

Just by leveraging existing data, ML models learn all components of a function (that would have had to be coded by someone)

↓

learn to assign proper values and calculations based on data

leverage → ignorance

MODULE 2

- Machine learning — family of statistical / mathematical } modelling

- Systems — improve performance — with ↑ experience data — uses based techniques / approaches

  ↳ automatically learn and improve PREDICTIONS (without programming)

⇒ Classified in a spectrum of supervision

Supervised learning:

given through which a program takes input-output pairs and "learns" the function that links them together

⇒ function → models

defining algorithms — models (functions) that matches well inputs - outputs

Training set

examples

How is the function right?

Objective → model works in unseen examples

Validation set

prueba / test

periodically

we learn from them and tweak layer parameters

hyperparameters
external configuration parameters that a data scientist sets before training a model.
They control various aspects of the learning process and model's architecture, influencing how well the model generalizes to new data

The model doesn't learn from validation set but we use that to tweak the parameters

Test set
Very end

TRAINING LOOP

hyperparameters tuning
↓
until performance is not better

How can unstructured data represent a model?

images → pixels are numerical

unstructured data → words → vectors
↓
word embedding

Regression — classification

Real numbers — Categories

Model training =
minimizing difference between output of the model's function and the true label for every sample in the training set

function output — True label = **Loss**

numerical value

↓

We average the loss

Then the model updates its function to say accurately

fitting / training the model

Adapts parameters → numerical values that are used to perform operations on the input data

Bias ← concept around model performance and algorithmic fairness
descriptor for additive parameters

"weights" → learnable parameters that determine the strength of connections between neurons or features in a model

LOGISTIC REGRESSION

used in → Categories → binaries

Sigmoid function
Non-linear transformation → squeezes all values between 0-1
→ probabilities

Probability cut off = threshold
decision boundary or operating point

Approaches to supervised learning

"No free lunch" theorem

- logistic regression
- linear " / polynomial
- decision trees
- support vector machines
- neural networks

Regression Tweaks → "regularization methods" - with approaches to better accomodate dataset challenge

Decision Trees

Random forest - ≠ criteria

Supervised Vector Machine (for classification)

Work better w/ crazy outliers
multidimensional features

Unsupervised learning

finds patterns / not labels

↓

in a dataset based on the structure of the data (not on pre-determined labels)

→ ReLU → Rectified Linear Unit

↓

"rectifies" or passes through the positive values → which are linear
negatives don't pass → 0

↓

not a line

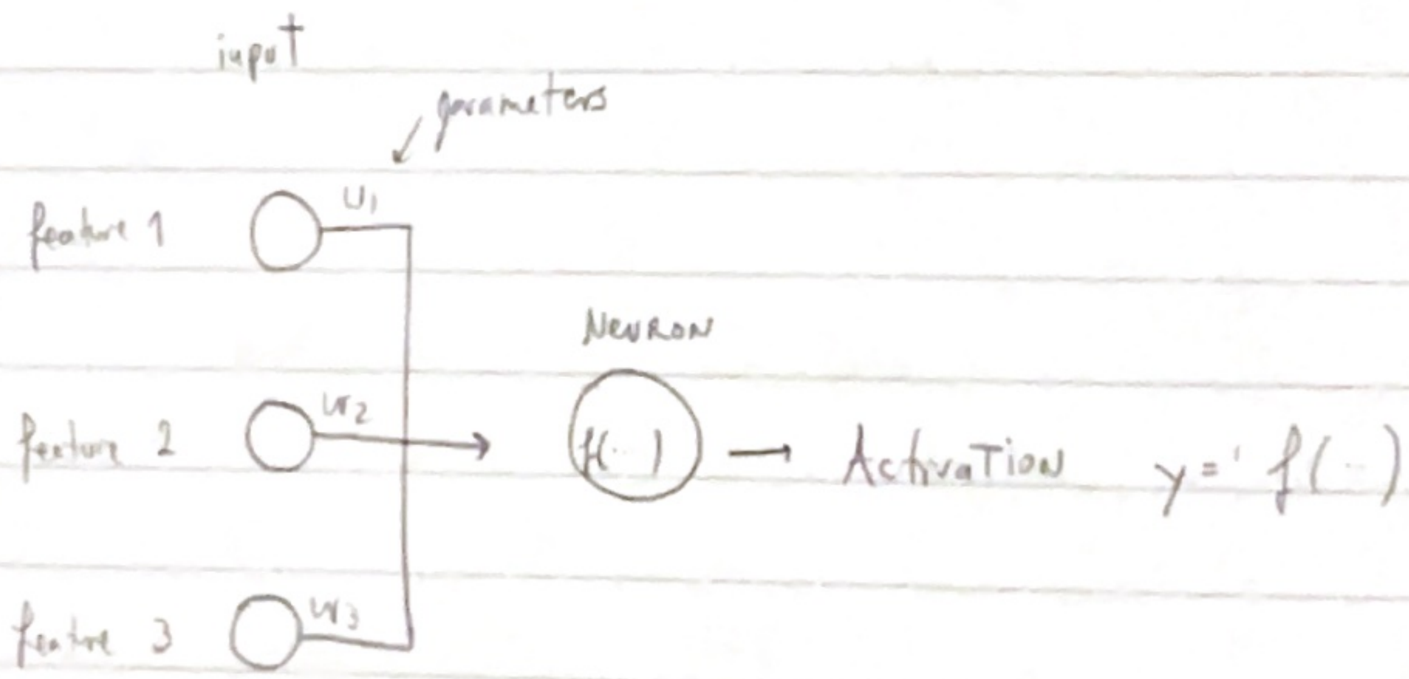
⇒ Traditional machine learning methods → linear combination of input features
+/− non linear transformation

↓

produce activations

↓

predictions



layers of neurons → they all take the same input → produce ⇒ output

↓

Complex function

→ if all the neurons in the layer take as input all the input to the layer

↓

fully connected layers = dense layer = linear layer

→ 2 fully connected layers in a neural network

layer

2° → model output

single neuron → represents binary classification probability that we want as output

⇓

Two-layer fully connected network

<u>1st layer</u>

linear combination → non linear transformation → first layers activations

<u>2nd layer</u>

Result:

first layer activations → linear combination → non linear transformation → number (prediction

→ <u>Neural Architecture</u> = how neurons are organized

going through training loop many times so the parameters are optimised

Training set
↓
Validation set
↓
Test set → N° we see commercialized

⇒ optimisation step

→ Goal of training is to minimize <u>Loss</u>
↓
the models calculates the loss
|
Adjusts itself
|
Minimizes loss

⇒ Loss function → Mathematical f(x) that processes all the individual losses
↓
tells us how well a model perform

Types of Loss

→ depends on data
→ task requested to the model

→ mean squared error loss (MSE)
" " logarithmic error loss
→ " absolute error loss (MAE)

binary cross entropy
hinge loss and squared hinge loss
multiclass cross entropy loss
sparse multiclass cross entropy loss

loss $f(x)$ =
⇒ Unique properties → helps model in a particular way

→ MSE = (model prediction − ground truth)^2
→ exponential
→ bigger value
no −

→ MAE = |model prediction − ground truth|
→ values are linear
no −

→ Cross entropy loss (= logistic loss / log loss)

classification problems → $\boxed{\text{CATEGORIES}}$

calculates the loss based on the probability of the predictions

negative log of error / natural log

allows more granular computation of error + closeness prediction to truth

$\boxed{\text{GRADIENT DESCENT}}$ → it's an algorithm

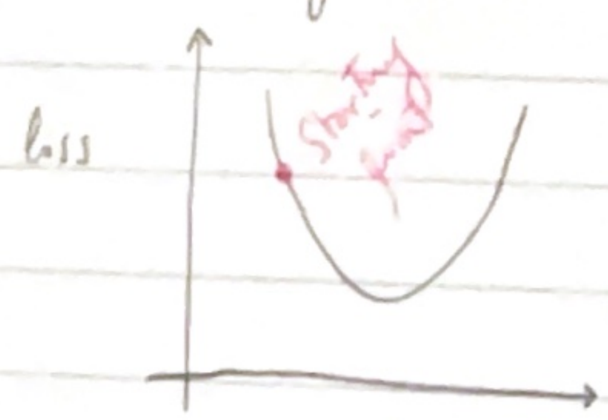
⇒ How to adjust weights / biases ?

→ which direction do we go to minimize loss ?

Training a machine learning model = running the numerical optimisation algorithm to find the best weights for the data set and minimise the loss

Ex

graphical look at gradient descent



→ this is every weight value for a function type

→ most algorithms set the initial guess for the value of the weights as 0 or pick a random number

Starting point → figure out the derivative

|

gradient → generalization of idea of derivative / slope that can be used for functions of multiple variables (parameters)

Backpropagation → technique to calculate gradients for complex $f(x)$

↓ and

simplify them into local gradient computations

↓ that can

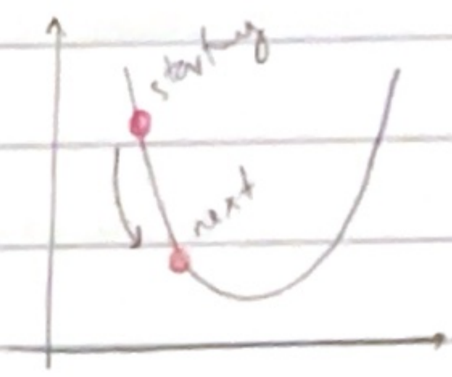
be combined together

parallell process makes it → possible / faster

Gradient → direction

→ magnitude

|



then the model takes a step into the negative gradient to minimize loss

⇒ Repeats itself aiming for the minimum

⇒ We can adjust the way we move towards minimal loss

adjusting → learning rate / step size | can be adjusted by specialist

↓

$\boxed{\text{tuning hyperparameters}}$

Representing unstructured image and text data

structured → there are numbers and "criteria"

unstructured → transform into numerical values (the pixels)

Colored images → 3 numerical values for each pixel → brightness of red, blue, green

↓

then they are RESIZED

aim = single size + resolution

word embedding

Texts → represented mathematically

word embedding = geometrical / spatial representation of a word

↓

2 dimensional

↓

need multidimensional (300 – 1024)

Types of Neural Networks

Convolutional Neural Networks (CNNs)

we need a layer that → one parameter efficient
→ preserve spatial information that's present in the pixel grids

↓

convolutional layer → key building block is comprised of convolutional → filter
→ kernel

↓

small group of parameters (multiplied, summed)

↓

a linear combination with one small patch or window of the image at a time

↓

the output of each linear combination

↓

passed through an activation function (eg sigmoid $f(x)$)

produces single value → placed relative to the location of the input patch in a new grid (something smaller)

→ filter ~ feature detector

→ in a train model → high number → something of interest is found on the patch
→ low № → nothing of interest

→ each filter → scanning pixel grid, looking for patterns / clues of interest — one patch at the time

- Receptive fields = amount of the image that a given filter looks information for

- Using random bound paradigm → instead of each processor being looked into a single good location

allows each filter — which is just a group of parameters to move freely about the image

→ each filter aggregate info → at a patch level

- to avoid can leverage from the spatial info of the image

NATURAL LANGUAGE PROCESSING (NLP)

⇒ ML approach to enable computers to understand and organize human languages

! Challenge is to understand the meaning ← computable way / interpretable / accurate

context
interpretation
nuance } add needed info

⇒ Speech algorithms → focus on digitalizing audio recordings → computable data

entonation
syllable emphasis } impact meaning

human speech → into → text then

healthcare apps ← dictation / transcription

NLPs → RNNs (Recurrent Neural Networks) → transformers

↓

used for: Unstructured data $\xrightarrow[\text{in}]{\text{comes}}$ sequences

↓

NLP tasks

Core component = RNN cell

⇒ Recall words = embeddings or high dimensional coordinates → capture semantic meaning

Read 1 word at the time

⇒ RNN cells can take as input
- output from earlier layers of neural network
- additional context set of values
  - ↓
  - can be used to pass info from one time step to the next

(≠ to dense layers from before) they ↓ don't have memory

⇒ vector with "context" (~ memory) → can learn to adjust values (words already seen)

⇒ parameters to predict the final classifier probability → sometimes used in the final step

⇒ ≠ variants of neural networks considering structure
- LDSM
- GRU

⇒ RNNs are fundamental to sequence based problems
- DNA sequencing
- dictation
- motion tracking

⇒ Difficulties
- reading 1 word at the time to process + memorize
  - ↓
  - loss effectiveness in long sequences
- technical = hard to parallelize processing of RNN
  - ↓
  - computations are sequentially
    - O → O

earlier layers of
neural network
set of values

l to pass info
one time step to
:

adjust values
already seen)

ues used in
the final step

M

sequencing
on
racking

TRANSFORMER

→ built on → self attention layer → processes in parallell while knowing the sequence

⇒ fundamental concept = the meaning of a word changes based on context

Self attention layers =

→ calculate weighting scores for the strength of contextual signal from any word to another

⇒ can produce ≠ weighted outputs → to encode ≠ types of context

→ transform inputs $\xrightarrow{\text{into}}$ outputs that → incorporate context at each element position in a variable way

⇒ Challenges → keeping positional information about how far away input elements are from each other

Summary

Major classes of Neural Network Architectures

Fully connected neural network
→ structured data
(→ DNN / MLP)

Convolutional Neural Network
→ image

RNNs
→ sequence inputs

→ Some other are hybrid

AlexNet (2012) → CNN → 8 layers
VGG → 19 layers
GoogLeNet (2014) → 22 "
inception network
ResNet (2015) Microsoft → 150 layers

⇒ SEGMENTATION → outputs one category prediction for each pixel

pixel mask

⊕ gives granularity

⊖ don't ≠ between distinct instances of a category object in the image (eg tumors / lung nodules)

⇒ INSTANCE SEGMENTATION → both semantic segmentation, object detection

output

boundary box of each instance

+

pixel mask corresponding to the segmentation of the object within each boundary box

contain ≠ types of neural network architectures or structures to produce each type of output

→ classification branch → predicts category of objects

→ regression branch → outputs numerical values ← location, extent } of object boundary box

↓

allow multiple classification regression outputs corresponding to ↓ multiple objects that can be detected in an image

Detection → eg. individual cells

Reinforcement learning

→ eg Gaming
trial + error

data in medicine is:

inferential
discontinuous } which makes it difficult

learning accurate functions $\xrightarrow{\text{that}}$ map from desired inputs
↓ to
outputs
| from

features describing the environment
observation to a value or
"reward"
|

of how good any particular action is given the observation features ← (based on minimising a loss function via gradient descent)

4

Model performance evaluation

Training / test - evaluation

80 - 20

70 - 30

Validation = Dev

Development

Cross validation
k fold } divide the data in k (amount of times) folds

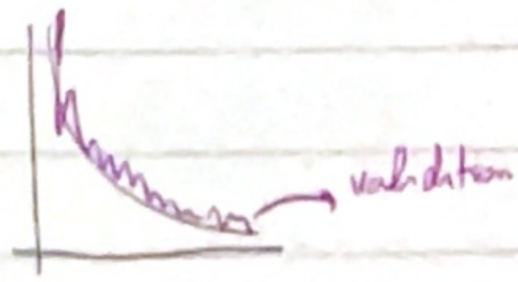
train in k - 1 of them

→ Stop before overfitting

when model fits to more than just the signal that is useful about the features in a dataset and also begin to memorize other stuff

Underfitting ← can't capture pattern in data
model / algorithm are to simple for complex data

learning curves → plots of model performance over time

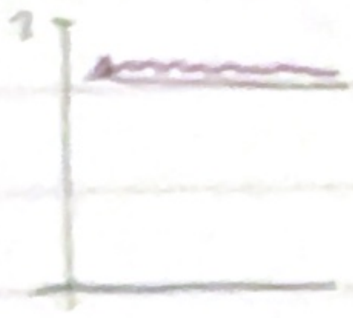


plot both curves → useful for debugging

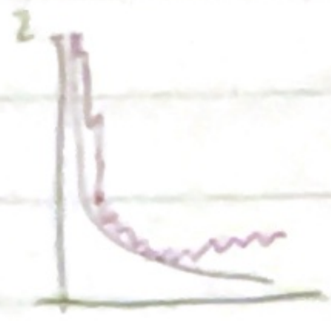
→ Computing loss ← training set → free / doesn't take time → done frequently
validation " → costly / takes time → periodically

epoch ← After every swing through training data

Underfitting



overfitting

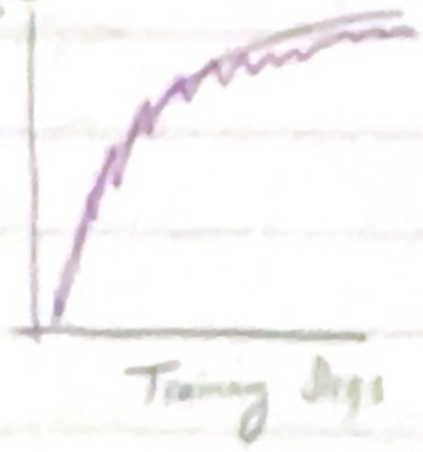


→ Not learning from model → performs poorly $\begin{cases} T \\ V \end{cases}$ sets

→ Good fit → Reaching a plateau

→ Generalization gap → Training loss − Validation loss

Model Accuracy Curve



STRATEGIES

DNNodes → universal function approximators

⚠ if model doesn't have enough info → Ex medical images } reduce pixels
↓
less info

Strategies for Underfitting
→ ↑ time
→ ↑ parameters → ↑ complexity ex deep neural network
↑ layers or ↑ neurons / layers

↓ Overfitting → Regularization
→ Weight decay L1 / L2 → penalise model for any ↑ parameters
↓ "

→ Dropout → drops out values randomly
→ model builds redundancy → ↓ complexity

→ Data augmentation – transforms the data → so the model learns ↑ generalizable features rather than memorising

Statistical Approaches to Model Evaluation

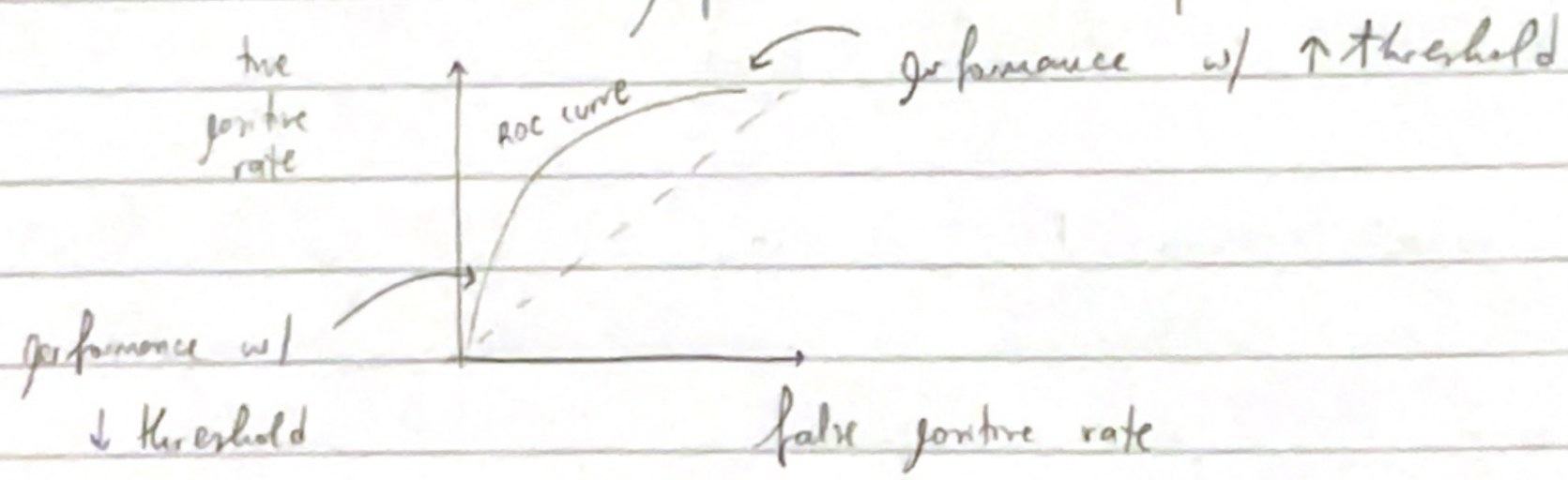
threshold to convert probability to → 1, → 0

↓

typically 0.5

Receiving operating Characteristic (ROC) Curves

evaluate → model performance → considering a range of thresholds



ROC curves → each category vs another

CONFUSION MATRIX

		predicted values	
		positive	negative
Actual values	positive		
	negative		

eg

actual		predicted ♡ attack +	−
♡ attack	+	TP	FN
	−	FP	TN

true / false ← +, −

→ Accuracy = $\frac{\text{all correct}}{\text{total data}}$ (0-1)

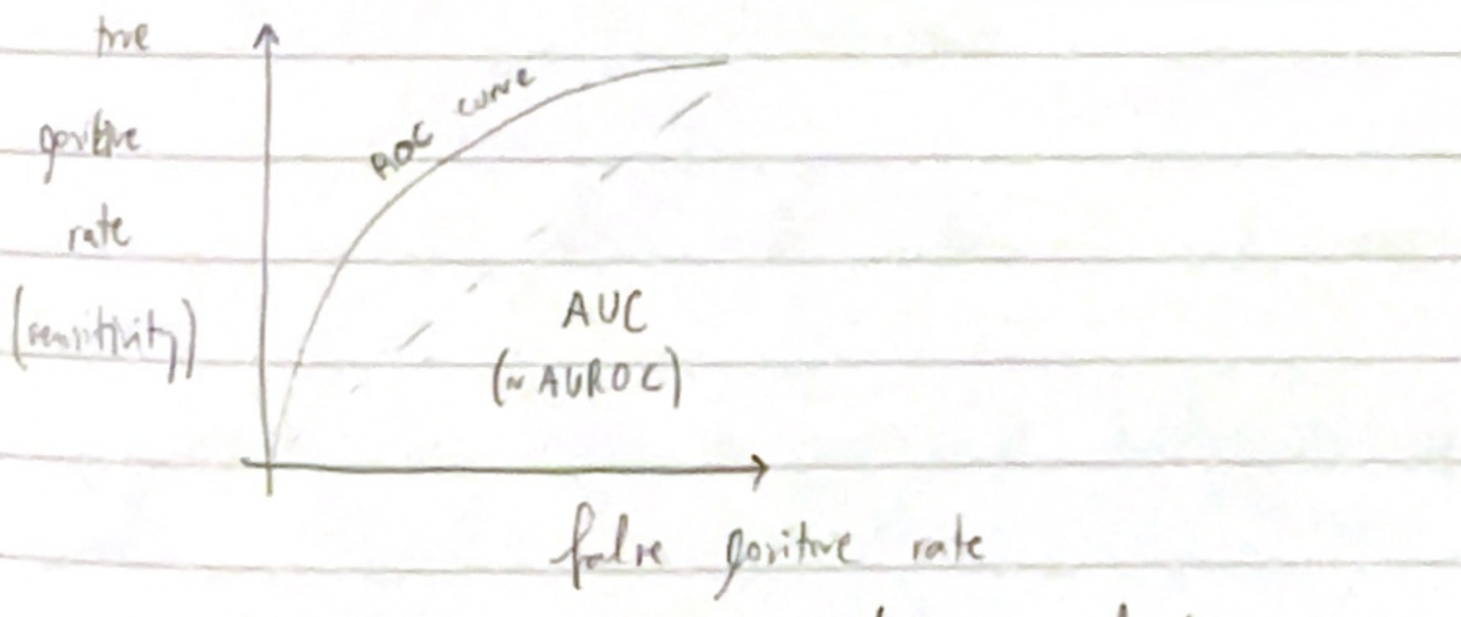
→ Sensitivity or recall = + predictions out of all +

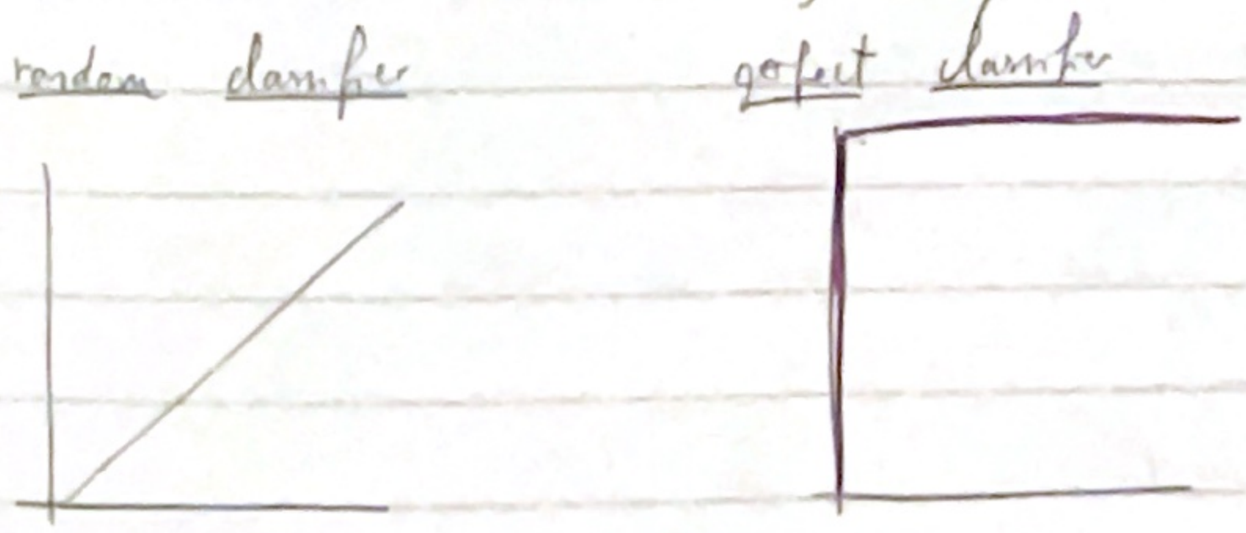
→ Specificity = " out of all −

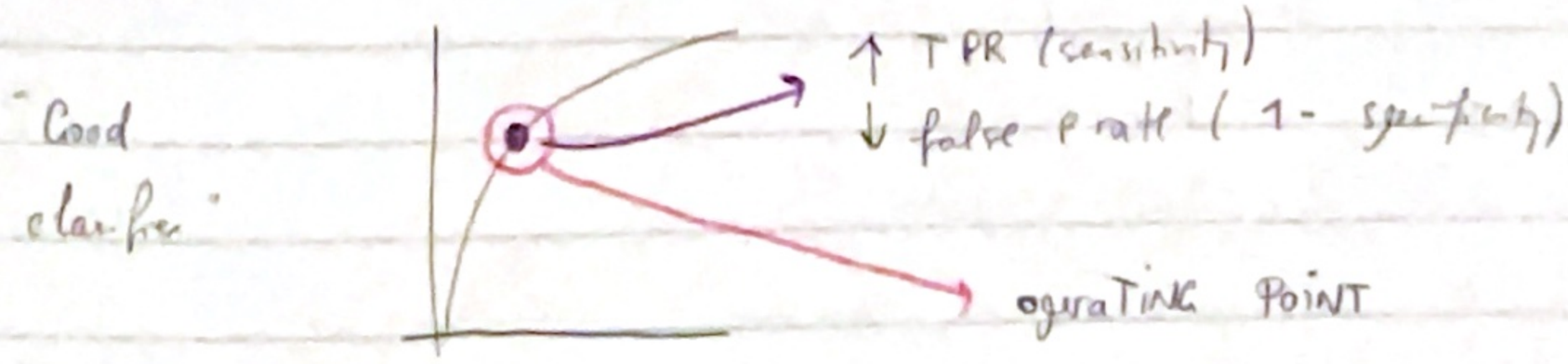
→ Precision (positive predictive value) → % of corrects predicting + } influence by prevalence

→ NPP → % " " " −

Receiver operator and Precision Recall Curves as evaluation metrics







Choosing operating point

→ If I prefer ↑ sensitivity (eg. screening) → ↑ False Positive Rate

→ If I ↑ specificity →

By minimising misclassification → choose operating point → minimises Σ false ⊕ + false ⊖

By optimising utility → assign a ≠ utility value for each type of outcome and optimize total utility

↓

subjective

⇒ Shape of the curve can be ↑ useful than the area

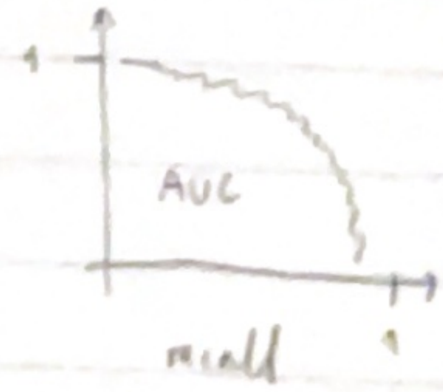
⇒ Misleading if the # of true positive labels are very ≠ than # of true negative labels

↓

IMBALANCED DATASETS

Alternative to general ROC :

Precision - recall curves



doesn't need/use

↓

key # true negative results

5 Strategies and Challenges in Machine learning in healthcare

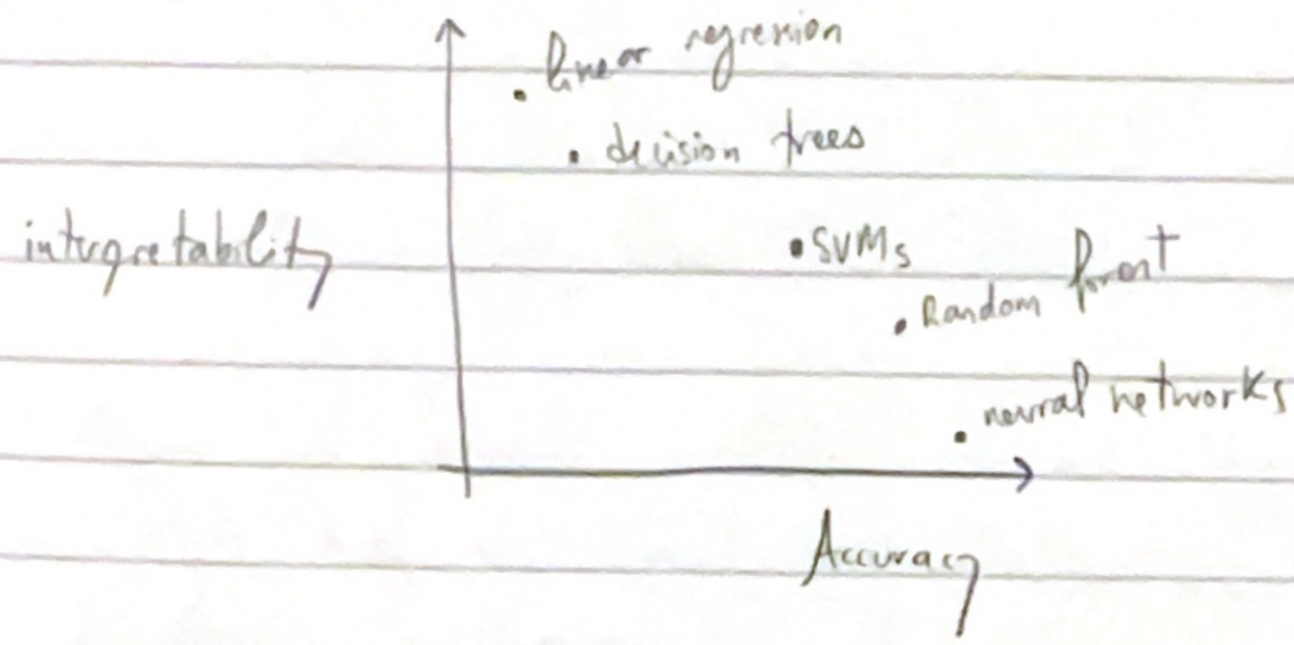
Unforseen variables:

→ Patterns can be — correlations — (- common response variables)
confounding factors
lurking variables

→ Reconfigure model

Context in Clinical ML

→ T between black box and interpretable algorithms



Black boxes → we don't understand the algorithm
↳ we observe Inputs / Outputs → but not internal workings

⇒ efforts to bring more intergretability
→ leveraging MDT to review false +
false −
→ test the model to external datasets → try to gain insight into causal vs correlative features

⇒ SALIENCY → get that matters the most to the model to make prediction

Class activation maps (CAMs) → based on analising the neurons in the final layer of some types of neural networks to compute how much the neurons that are important for any particular class are firing at every spatial location

1° firing neurons 2° importance → produce heatmaps

→ Computing change in prediction score from changing something

→ Transparent → can lead to overconfidence

→ explainable

→ inspectable

Intrinsic interpretability → simple, self explanatory models

Post-hoc " → used to understand decisions by complex models that don't have prescriptive declarative knowledge representations or features

Medical Data Challenges in ML

data types:

- EHR
- structured / unstructured
- imaging
- genomics
- peripheral sources

→ small representative sample (of the type) → understand how it's generated $\xrightarrow[\text{how}]{\text{consider}}$ it fits into ML workflow

get a sense of

how much data is needed $\xleftarrow{\text{gauge}}$ METrics

Sample
Preliminary analysis
Discussion
} should be done before investing many resources

→ data used to train model → should be available in real world

what preprocessing
feature engineering of data will be needed

→ Data in medicine can be → heterogenous
→ incomplete
→ duplicated

⚠ Missing data

→ Class imbalance → in amount of labels

↓

accuracy paradox problem → ↑ accuracy → doesn't reflect "real" accuracy

they just choose the label with ↑ amount (over represented class)

→ Choose proper metrics + reevaluate performance of classifier

→ Artificially sample a small hold out test set from your data that contains more of a balance of classes

→ Collect ↑ data

→ Resample data set

Sample w/ replacement

Remove from over-represented data

→ Algorithmic approaches (forest, trees) that tolerate imbalance

When ANALYSING, rely on:

- ROC curves → if balanced
- precision recall → if unbalanced

How much data is needed?

→ ↑ n° training data at logarithmic rythm and see
→ models like RegRESSION it grows and then reaches a PLATEAU

"1 in 10" → 10 examples of each level class

→ for NEURAL Networks → ~ 1000 examples

factors that impact on the amount of data needed
→ N° of features in dataset relative to n° of uncorrelated / weakly correlated attributes of dataset

Retrospective data + "Shelf life"

⇒ Adding ↑ info ≠ ↑ data

Quantity vs Quality

You can overcome data label noise with ↑ volume

10 % Noise → 50 % data
15 % " → double "

⇒ Weak supervision – relative strength of the label → incomplete / inaccurate

→ the test set should have ↓ NOISE as possible
↓
representing the gold standard

6 Designing and evaluating clinical ML application

→ Finding problems worth solving

Science | Practise | Delivery

Classify
Predict
Act / treat

⇒ begin with the end in mind

⇒ OAP = Output Action pairing

⇒ OAP framework

↓

minimum acceptable performance

↓

overall utility?

(Eden) ⇒ ≠ factors to consider

Computer Scientist / engineer

ML techniques
co-designing model
workflow pipeline
coding

↓

knows ML techniques

Data Scientist

data mining
feature engineering → pre processing etc
analytics
metrics of model performance

↓

can do simple models

engineering can optimise this

FOUNDATION MODELS (FM)

def → learned from unlabelled data / unsupervised

adaptability → learn ↑ → with ↑ data

↑ # of parameters

create vision of world from ↑ data
self supervised

TRANSFER LEARNING

fine tuning
fine tweak

few shot → def that is asked → FM adapts to new tasks with small training → e.g. skin lesions

zero shot → now using new recipe → FM transfers knowledge to new tasks → e.g. likelihood of a pt developing a condition (eg rare conditions)

Other (+)

→ enabling ← modular, reusable, robust models

→ open new AI interfaces

→ useful where data ← scarce, diverse

Narrow AI

1 task

programmed by human

doesn't "evolve"

General AI

many tasks

learns and teaches itself

emergent behaviours

this system { ↑ data ⟷ ↑ parameters at scale (interrelationship)

↓

exhibits behaviours not predictable from data alone

↓

eg. → flock → collective behaviour consequence of relations between individual elements

hallucinations → generation of false / unrealistic output

↓

to ↓ its incidence ← FM should be trained with data ← diverse, representative

→ predictions validated through rigorous testing and experimentation

↓

develop post governing capabilities — eg AI to check AI

How do FM work

key components → transformer architecture → 1st introduced in paper "attention is all you need"

Neural Model
↓
→ Attention mechanism

→ Self-attention : input data $\xrightarrow{\text{applies}}$ attention mechanism → itself

→ Considers → relations $\xrightarrow{\text{between}}$ elements → All at once

LLM
↓
high level = Deep Neural Network

⇒ core = layers interleaved w/ attention mechanisms

⇒ sequential data

⇒ Tokenisation $\xrightarrow{\text{breaks}}$ text to units
↳ can be applied to other things ≠ than text

⇒ Subtypes of transformer models

→ transformer encoder

sequence → vector representation } captures semantics

Self supervised training with large data sets — unlabeled data — eg BERT → there are unlabeled words → model trains to find them and predict them (trained w/ GRADIENT DESCENT)

then fine tuned w/ small amounts of labeled data

⇒ Model that generates its own labels based on inherent structure in that unlabeled data instead of requiring human labels

→ transformer decoder

vector → tokens / words — EG ChatGPT

Self supervision
Pre-trained ⟶ predicts words given past words

Objective function:

→ Probability distribution over word sequences

ChatGPT based on GPT3 foundation LMM

↓

Reinforced learning (trained receiving human feedback)

3 key steps → supervised fine tuning
→ mimic human preferences → Reward model
→ proximal policy optimization

⇒ PROMT = input model uses
starting point

control what generates
direct attention

→ Prompt engineering → select / → data → to train a model
constructing /
organizing /
setting
constraints

goal → maximize accuracy / efficiency
↓ hallucinations

Types:

⇒ Instruction prompt → we ask a task

→ Role assigned prompt → we assign the model a role

→ few shot standard prompt → includes examples of the task
the prompt is trying to solve

→ Chain of thought prompt → model explains reasoning

⇒ zero shot chain of thought prompt → when you don't have any idea of the answer

→ Self consistency prompting → aims at improving the accuracy of the model's response

↓

By Generating multiple chains of thought

+

taking the majority answer as the final answer

|

Could improve =

DG / Pt management / TTOs planning

Generative knowledge → Generation → facts

→ Integration → Based on the generated fact and then chooses the + probable

→ MULTIMODAL Models → DALI family

→ DIFFUSION MODELS → generates realistic images

⇒ Tokenisation

- Name entity recognition → drugs / diseases
- negation detection → T / F on entities
- relation extraction → pieces of puzzles together
- de-identification → protect privacy + confidentiality

} require purpose built-solutions

FM can do it easier

Also generalisation

⇒ FM would be trained with ↑ data to ↑ accuracy

⇒ Genomic → FASTA format