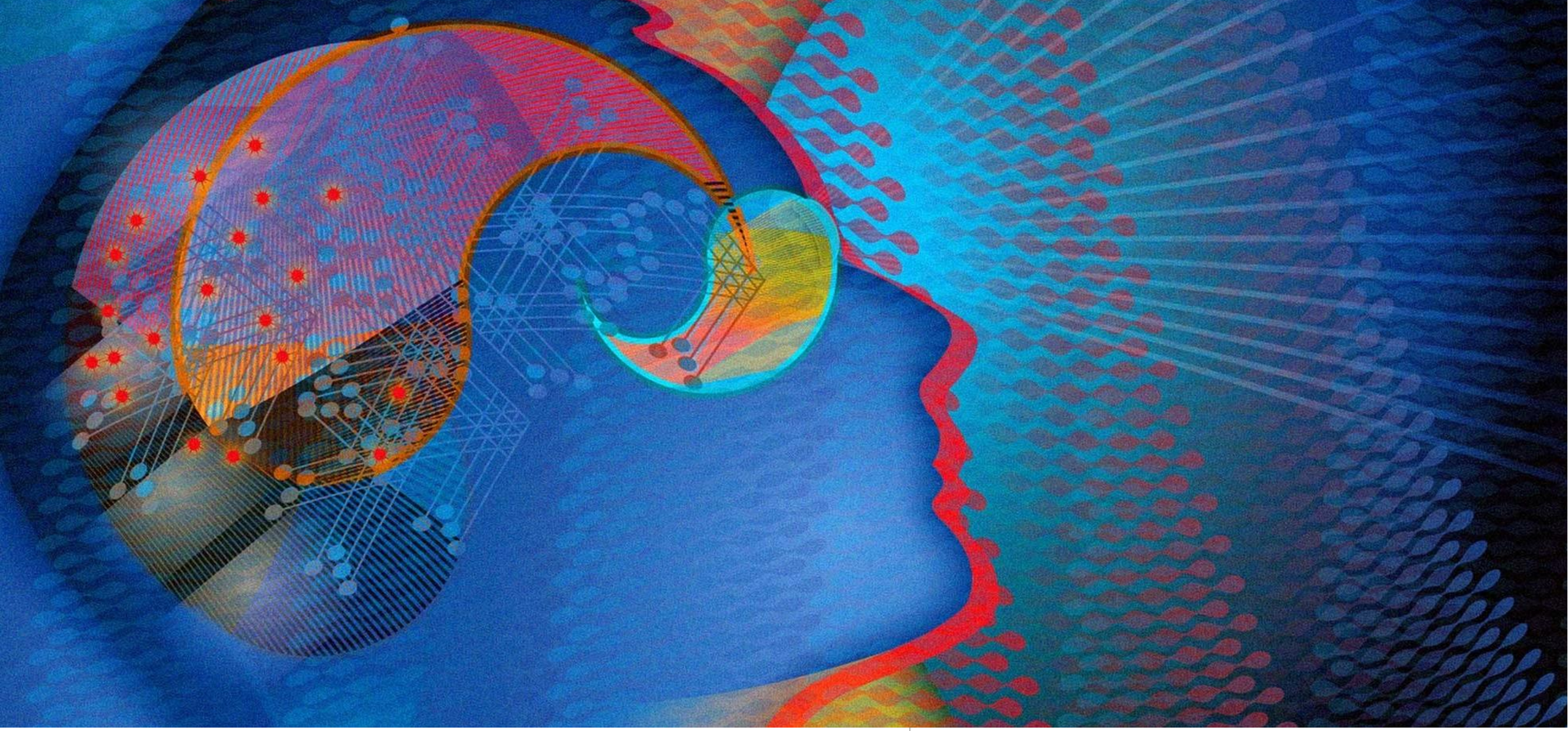
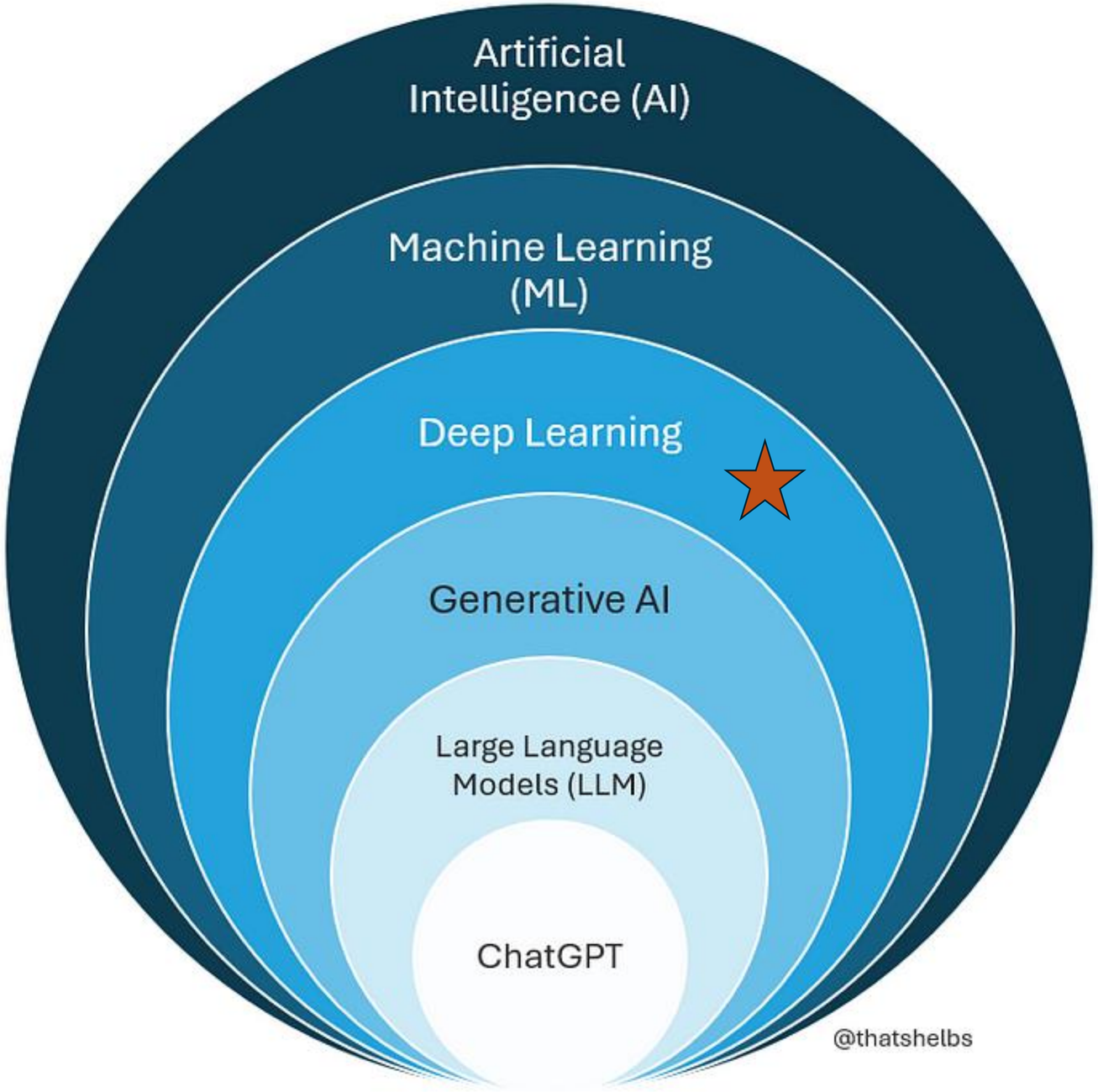


Introduction to Deep Learning (Computer vision)

Aaron Y. Lee MD MSCI
University of Washington


Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning
Generative AI
Large Language Models (LLM)
ChatGPT
@thatshelbs

Computer vision:
Giving machines the ability to “see.”

A NEW COMPUTING MODEL

Algorithms that Learn from Examples

Traditional Approach

- Requires domain experts
- Time consuming
- Error prone
- Not scalable to new problems

A NEW COMPUTING MODEL

Algorithms that Learn from Examples

Expert Written Computer Program

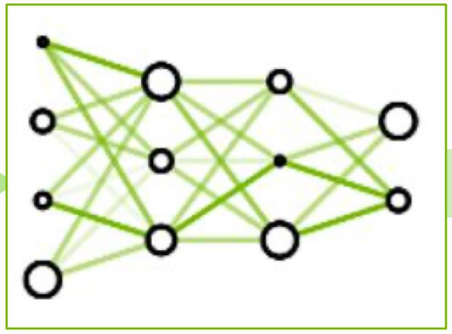
car

vehicle

coupe

Traditional Approach

- Requires domain experts
- Time consuming
- Error prone
- Not scalable to new problems



Deep Neural Network

car

vehicle

coupe

Deep Learning Approach

- ✓ Learn from data
- ✓ Easily to extend

Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

Supervised Learning
(We know the right answer)

Classification

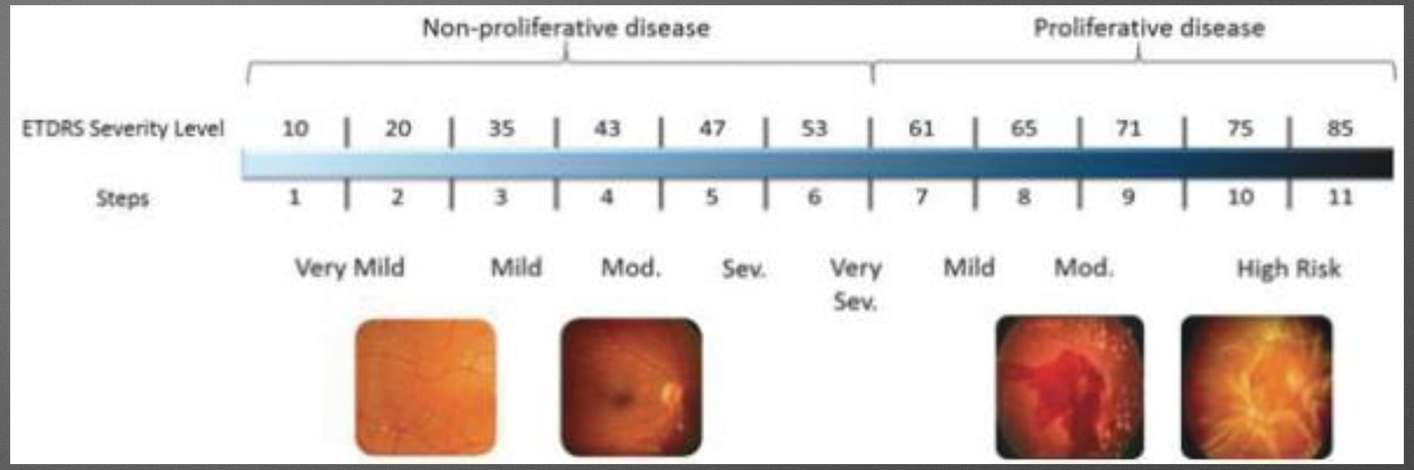
CAT

No Diabetic Retinopathy
Mild NPDR
Moderate NPDR
Severe NPDR
PDR

Given an image, output one of these 5 severities.

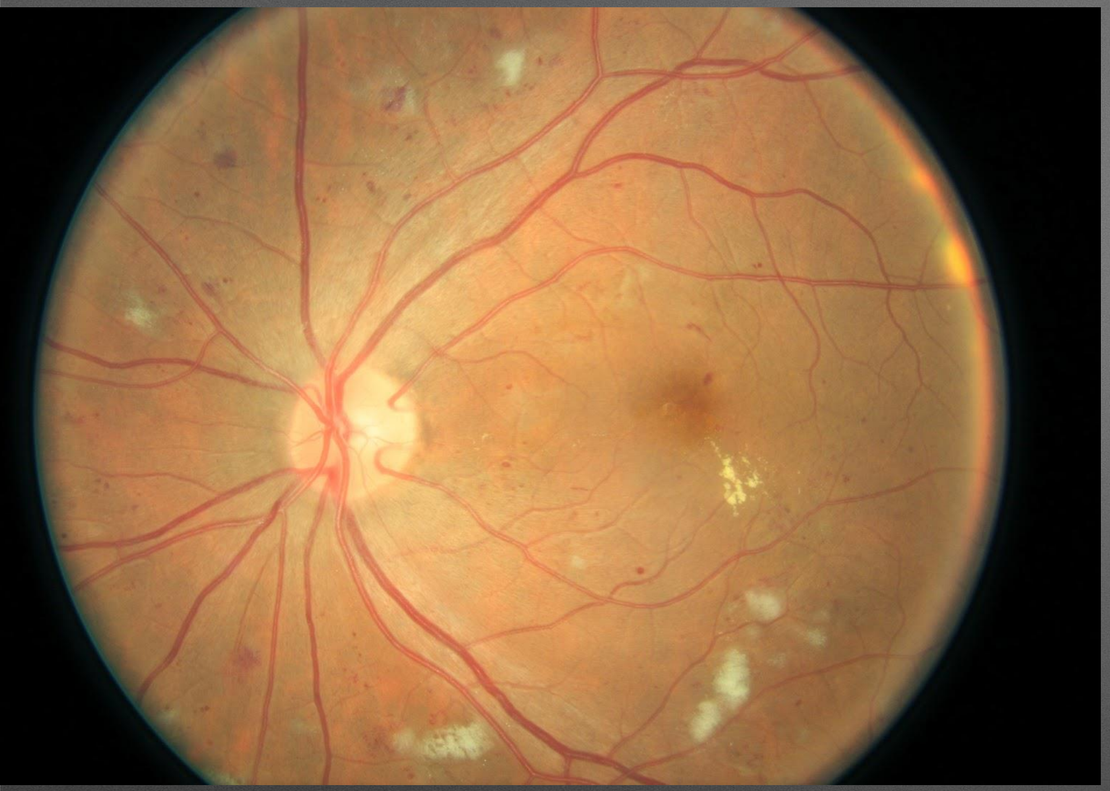
Classification

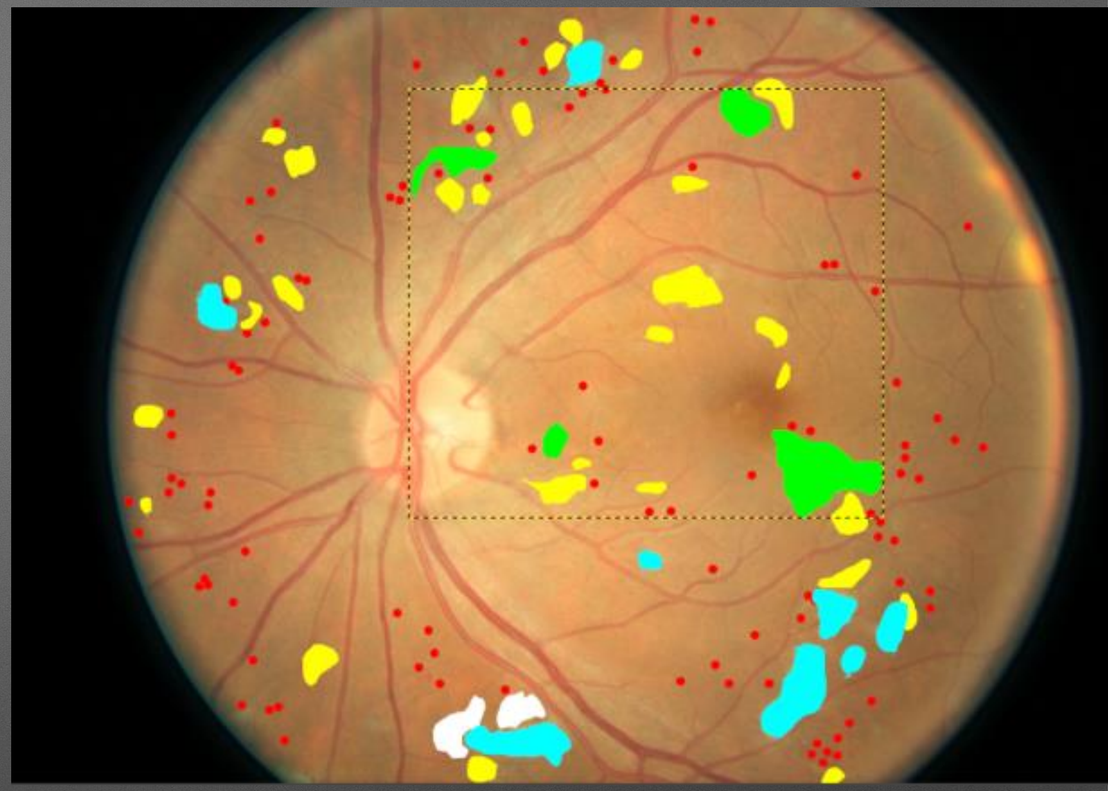
FDA



Given an image, output the ETDRS severity level (0-100).

Regression



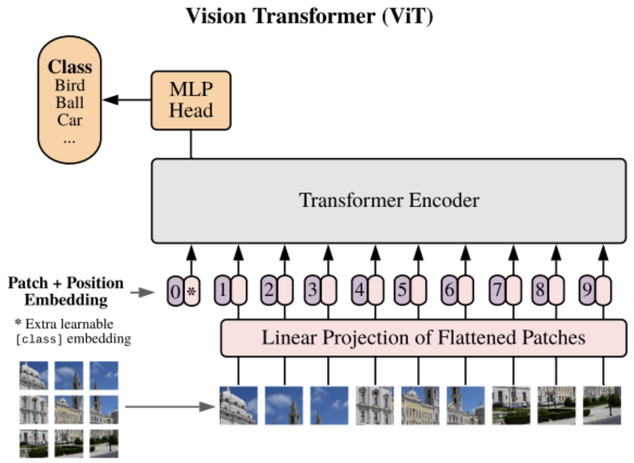


Given an image, output the locations of MA, HEx, CWS, and NV.

Feature Segmentation

Model choices (Classification)

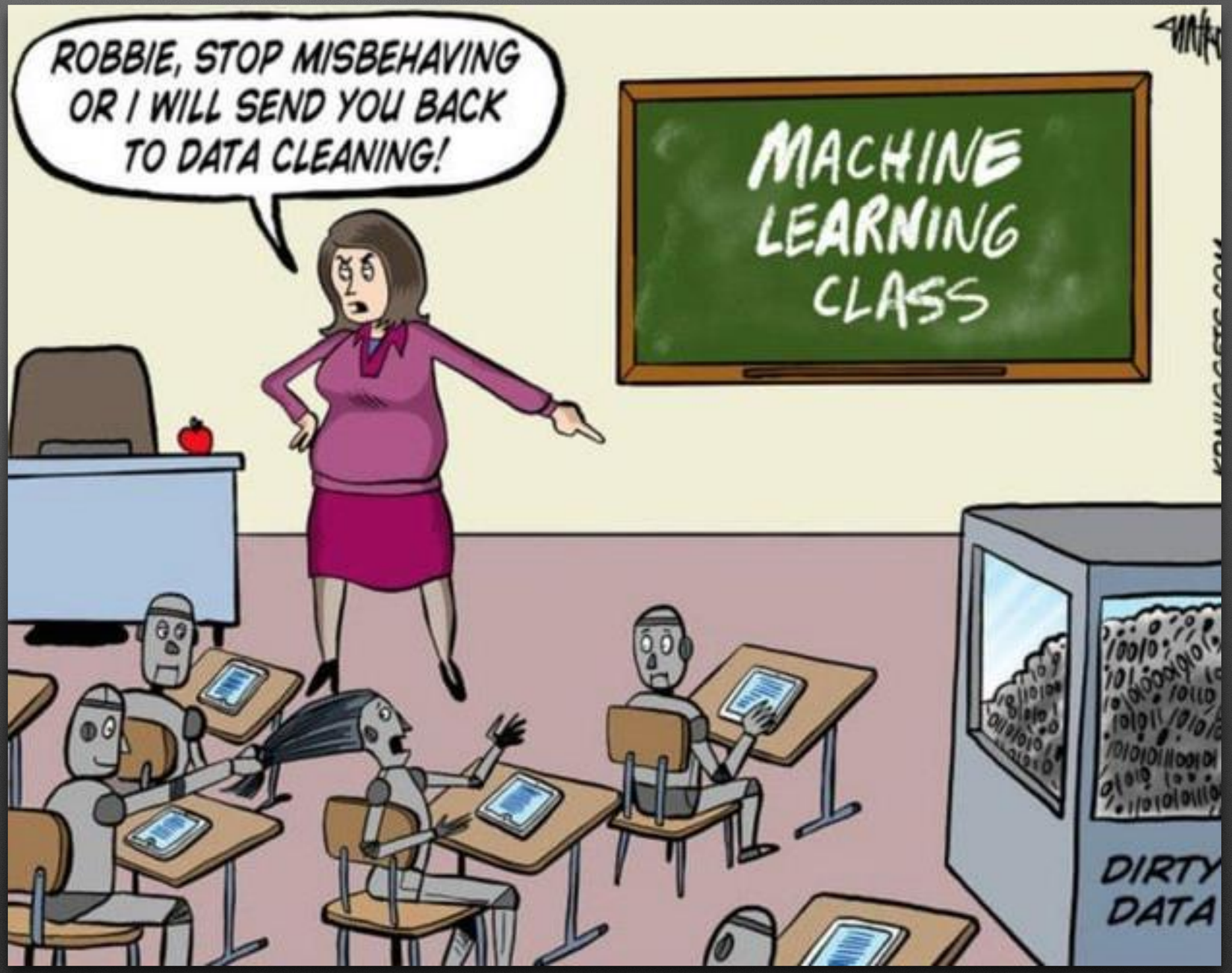
- Convolutional Models
  - VGG series
  - Resnet series
  - EfficientNet series
- Vision Transformer Models
  - Vit series
- Initialization:
  - Random weights
  - Imagenet pretrained weights
  - Foundation model weights



Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

Data Cleaning = 99% of the work



Preparing AI-Ready Data

OMOP

- Demographics
- Survey data
- Physical assessment data
- Medications
- Blood and urine lab values
- MOCA
- Vision testing

DICOM

- Fundus photography
- OCT
- OCTA
- FLIO

Waveform Database (WFDB)

- ECG

mHealth

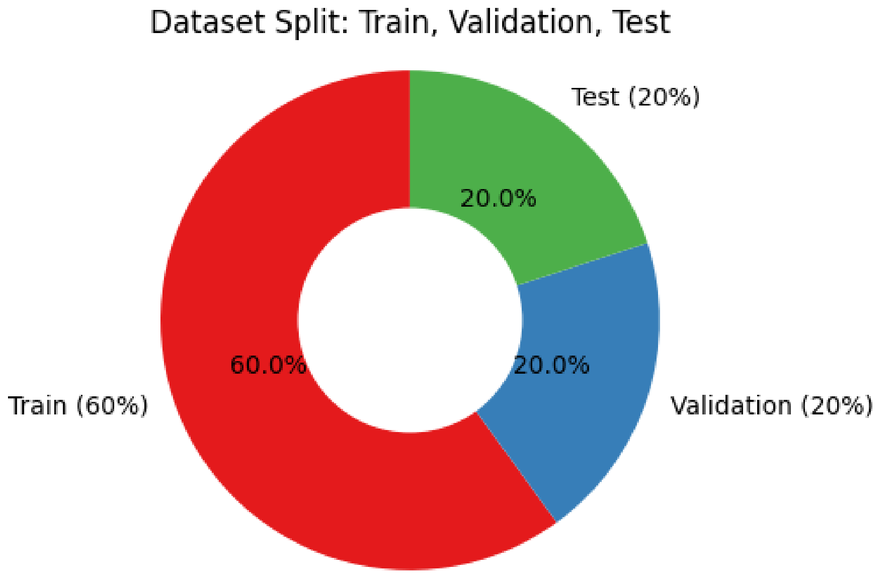
- Physical activity monitor
- Continuous glucose monitor

Earth Science Data Specification

- Environmental sensor

Partitioning data

- Training Data (60%)
  - Used to adjust the model to learn from the data
- Validation Data (20%)
  - Used to periodically check how well the model is doing.
  - Model weights are NOT adjusted
  - Used during training session and for hyperparameter tuning.
- Test Data (20%)
  - Used to evaluate the final performance of the final model (ideally once at the end)
  - Model weights are NOT adjusted
- Unit of partition: person



Suggested split

The suggested split for training, validating, and testing AI/ML models is included in the participants.tsv file that will be included in the dataset. A summary is provided in the table below.

	Train				Val				Test				Total			
	Hispanic	Asian	Black	White	Hispanic	Asian	Black	White	Hispanic	Asian	Black	White	Hispanic	Asian	Black	White
Race/ethnicity (count)	144	167	211	225	40	40	40	40	40	40	40	40	224	247	291	305
	Male	Female			Male	Female			Male	Female			Male	Female		
Sex (count)	302	445			80	80			80	80			462	605		
	No DM	Lifestyle	Oral	Insulin	No DM	Lifestyle	Oral	Insulin	No DM	Lifestyle	Oral	Insulin	No DM	Lifestyle	Oral	Insulin
Diabetes status (count)	292	162	235	58	40	40	47	33	40	40	41	39	364	242	331	130
Mean age (years ± sd)	60.3 ± 11.13				60.2 ± 10.5				60.4 ± 11.0				60.3 ± 11.1			
Total	747				160				160				1067			

Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

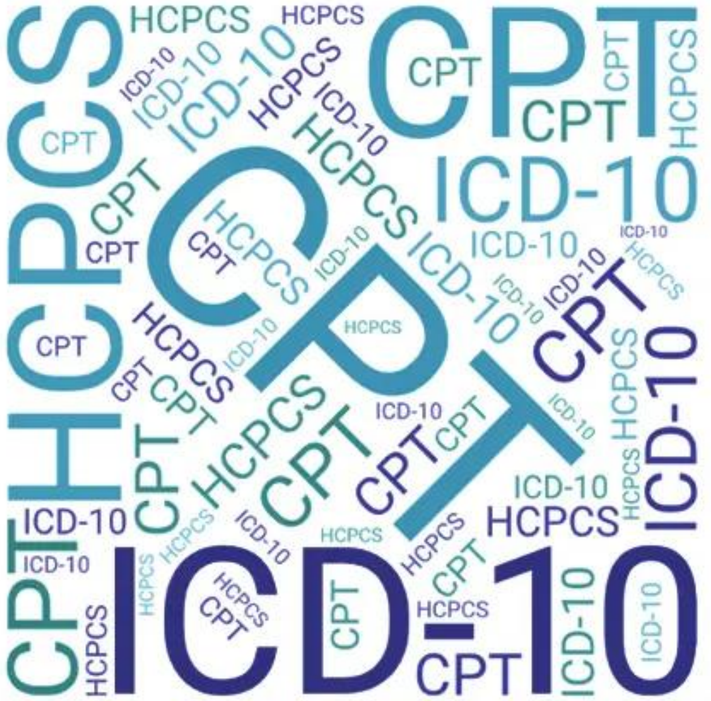
Tradeoffs

- Ideal setup:
  - 2 expert graders all grade the same images
  - A third arbiter (even more of an expert), masked to the identities of the graders, regrades and provides a final grade
  - Con: very labor intensive
- Most common setup:
  - Have *n* graders each grade on their own
  - Not everyone is the same level of expertise
  - Not everyone grades the same number of images
  - Con: model performance will likely be a mixed bag.

Leverage existing labels

- Derive labels from diagnosis codes, CPT codes, etc. from EHR
  - Noisy and can be unreliable
  - CPT procedure codes are better labels as they are true “events”

- Derive labels from lab tests, future events (eg readmission, need for surgery)
  - Can be acutely altered in extreme situations



Models will never do better than the average human error.

Moderate

Severe NPDR

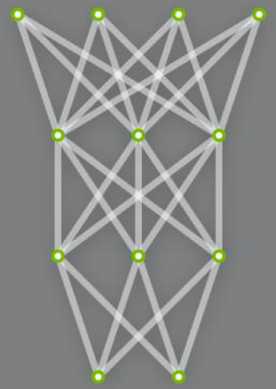
Try to use objective ground truth for labels (not derived from humans).

Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

DEEP LEARNING

Untrained
Neural Network
Model

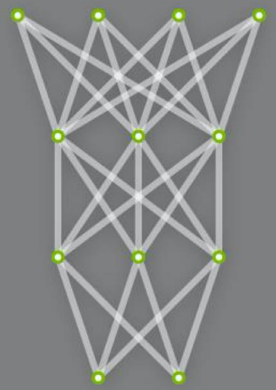


DEEP LEARNING

TRAINING

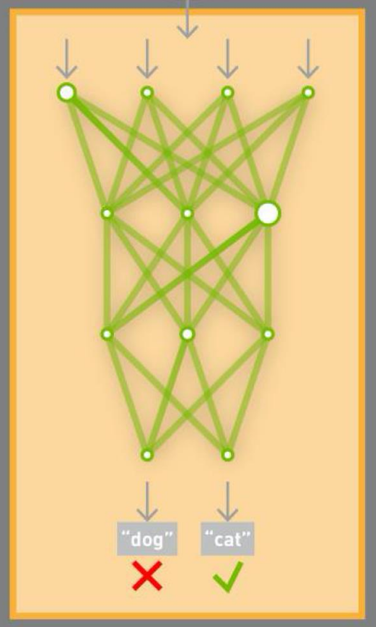
Learning a new capability
from existing data

Untrained
Neural Network
Model



Deep Learning
Framework

TRAINING
DATASET

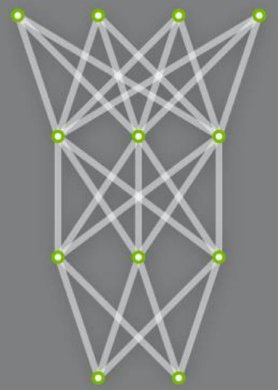


DEEP LEARNING

TRAINING

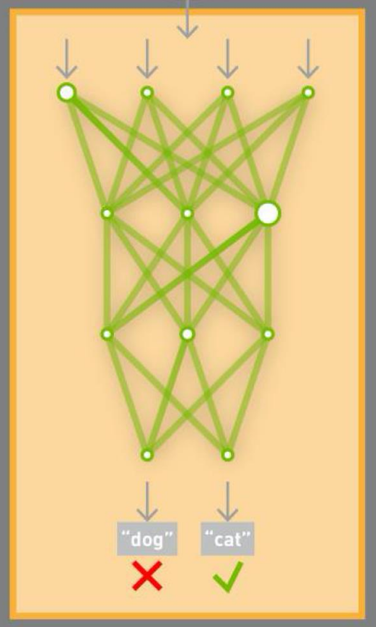
Learning a new capability
from existing data

Untrained
Neural Network
Model

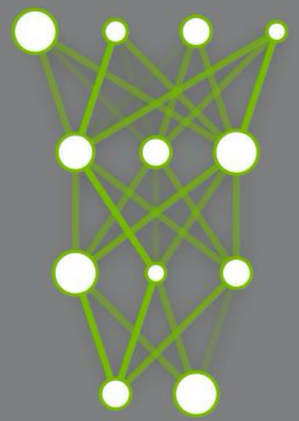


Deep Learning
Framework

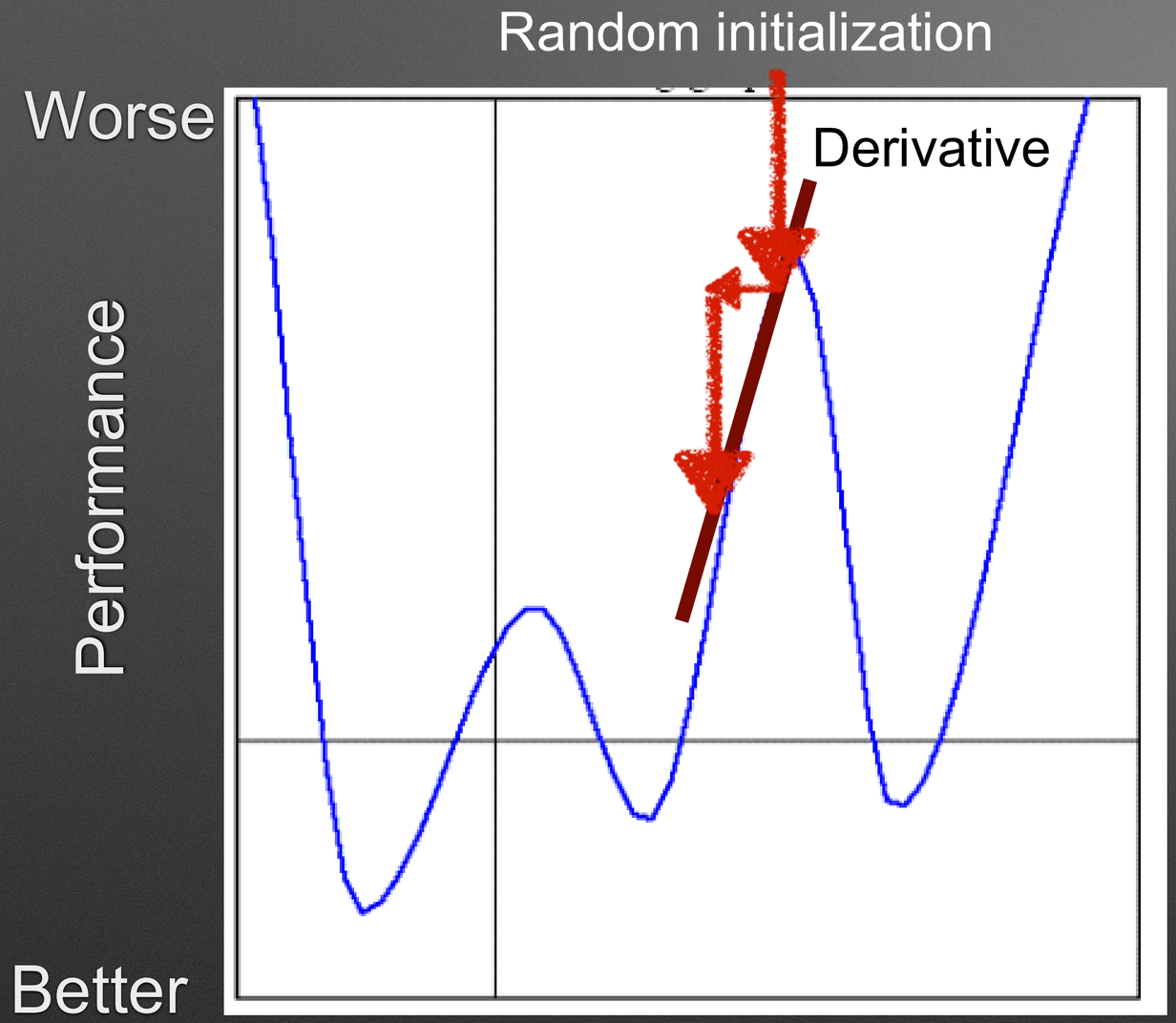
TRAINING
DATASET



Trained Model
New Capability



How does it actually learn?



Objective evaluation function

$$w := w - \eta \sum_{i=1}^{n} \frac{\nabla Q_i(w)}{n}$$

Random initialization

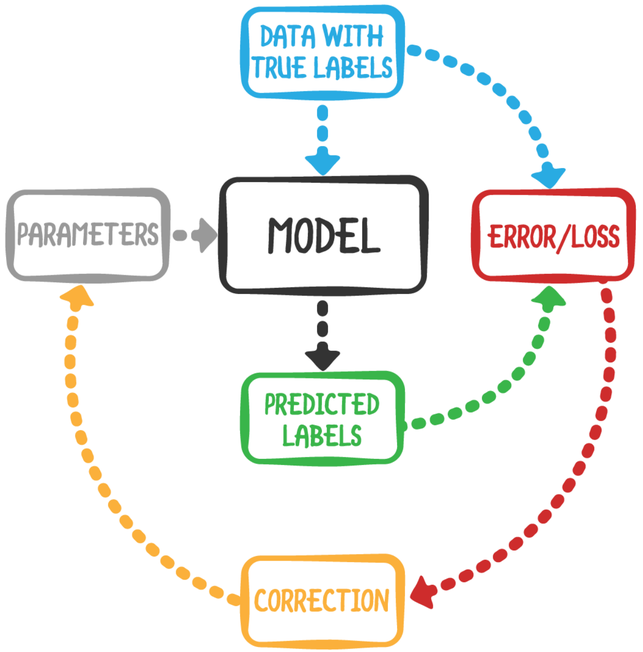
Learning rate

Gradient

Stochastic Gradient Descent

Training loop

- Epoch 1:
  - Shuffle training data
  - Divide into small batches
  - For each batch: adjust the model slightly
  - After going through all the training data, freeze the model and test on validation dataset
- Epoch 2:
  - Repeat
- Epoch 3:
  - Repeat

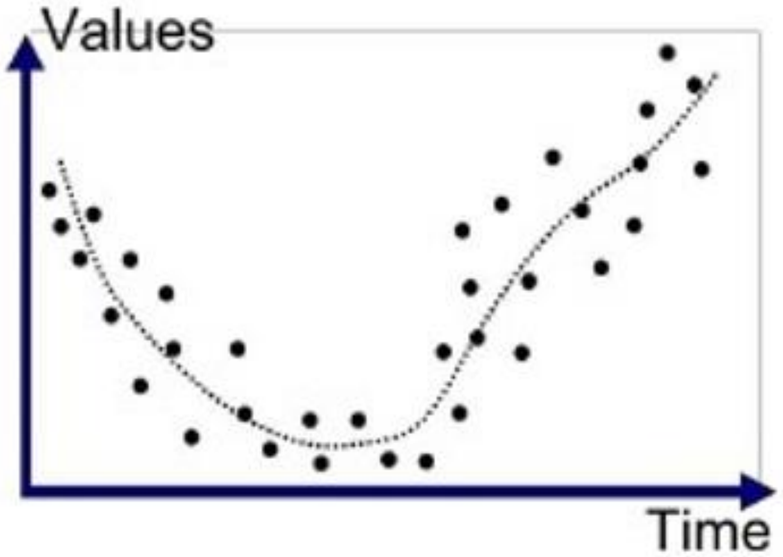


Lifecycle of a computer vision DL project

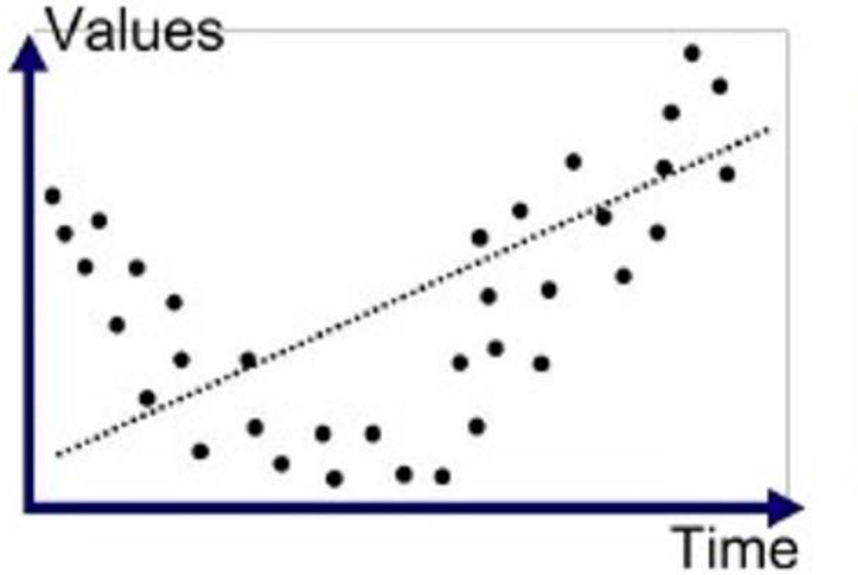
1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

What are hyperparameters?

- Parameters = how the model will behave for a given input.
- Hyperparameters = things we can adjust that will ultimately lead to better parameters
- Examples: batch size, learning rate, optimizer, number of epochs
- They affect how a training session behaves but not how a deployed model behaves.

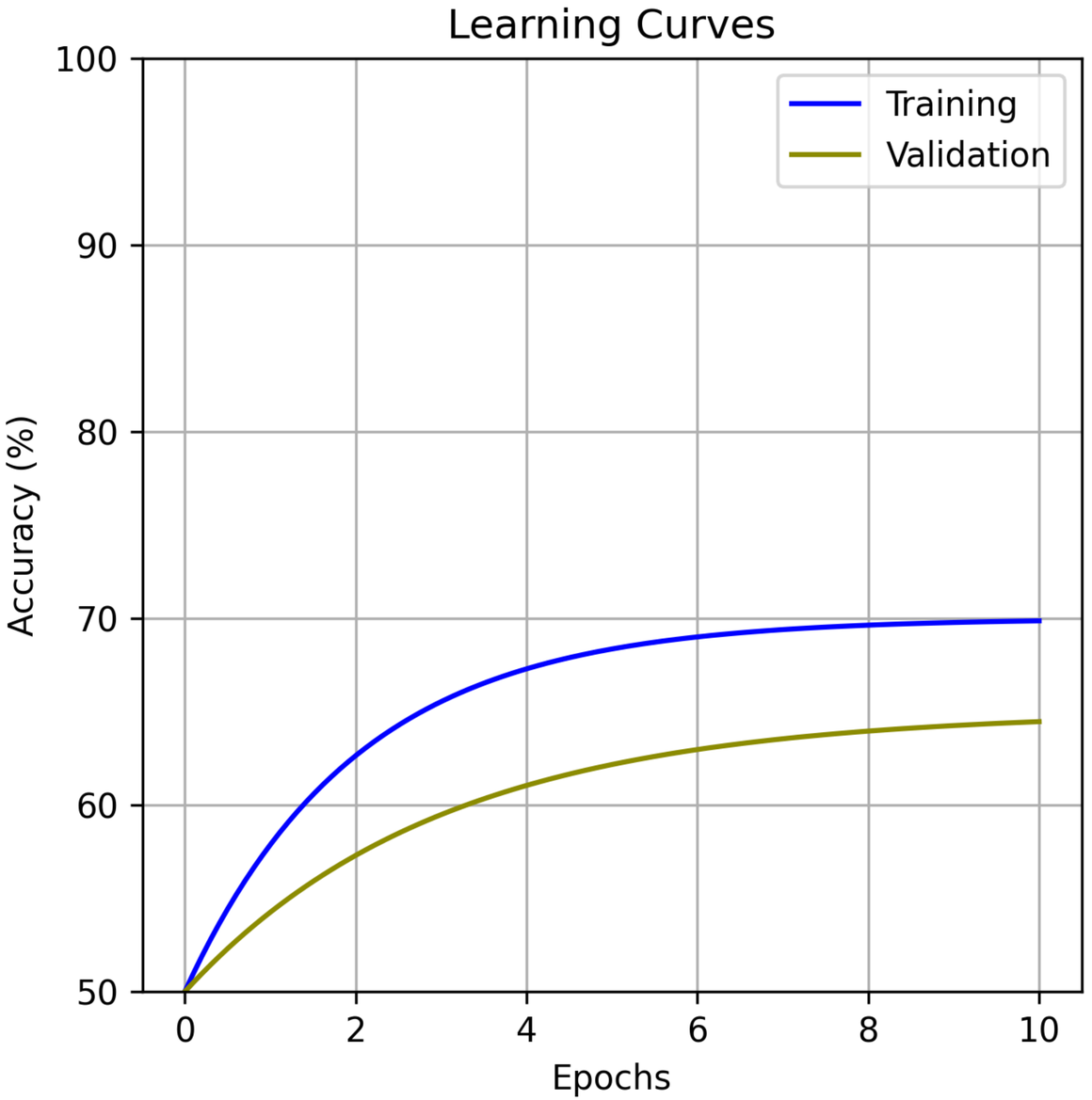


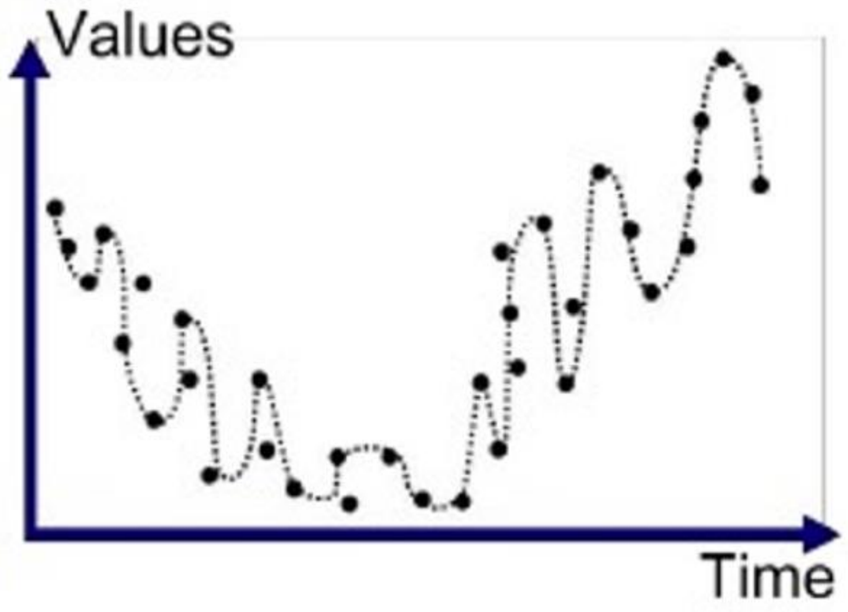
Good Fit/Robust



Underfitted

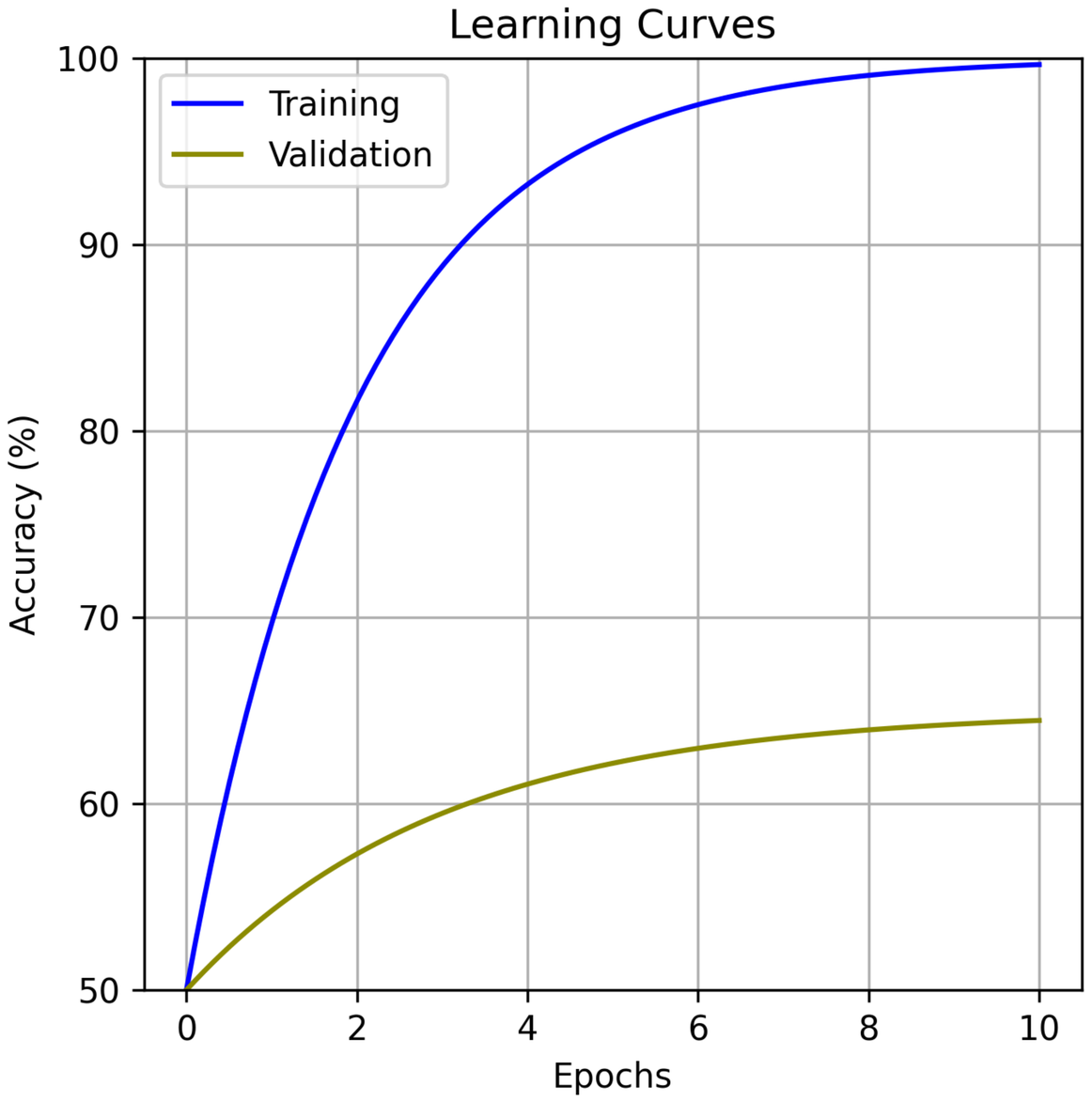
Increase model complexity

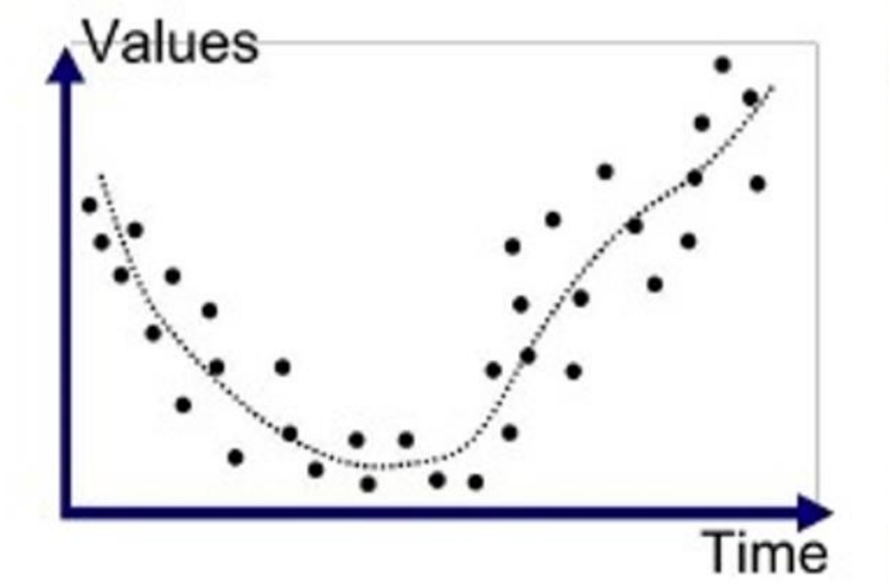




Overfitted

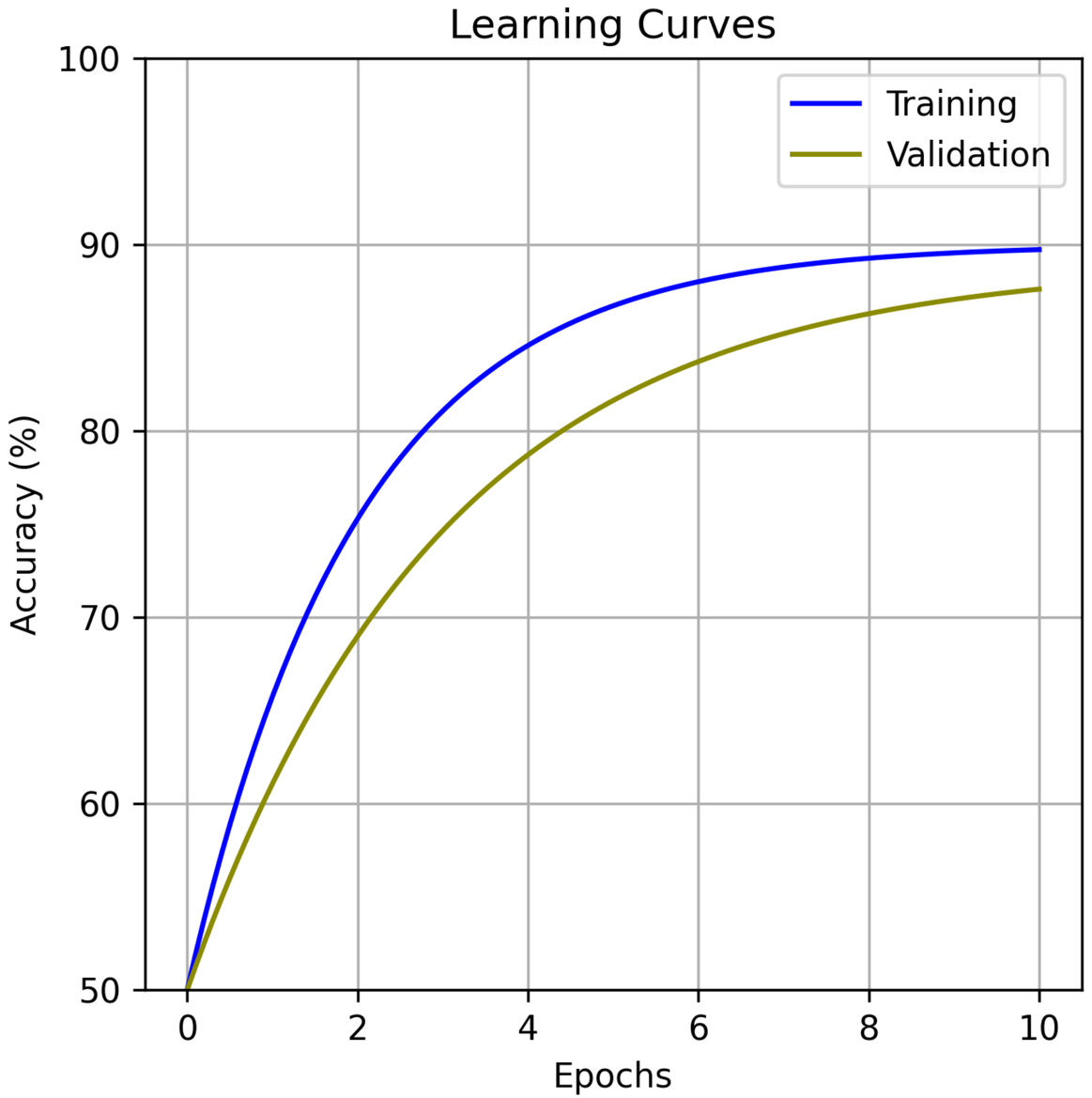
Get more data,
add regularization

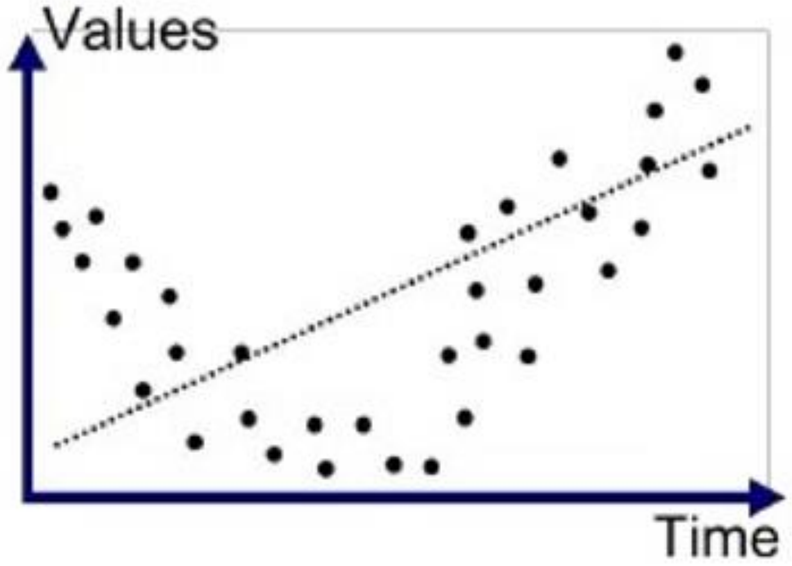


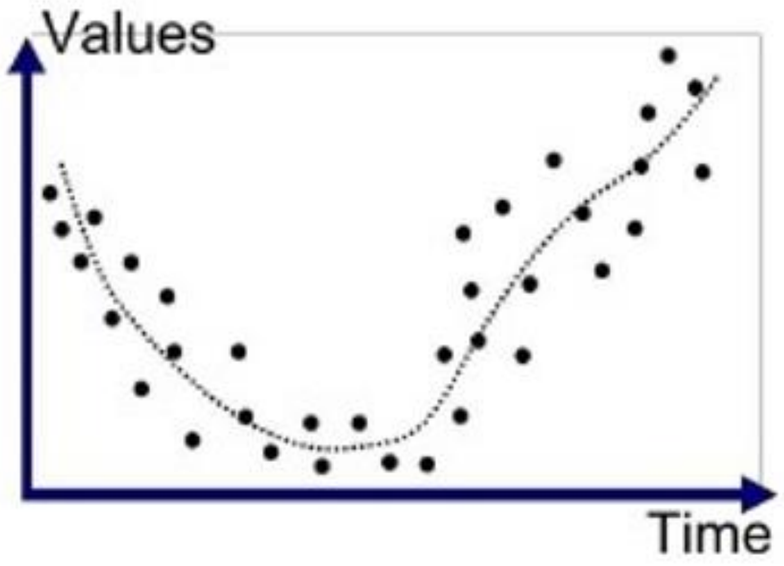


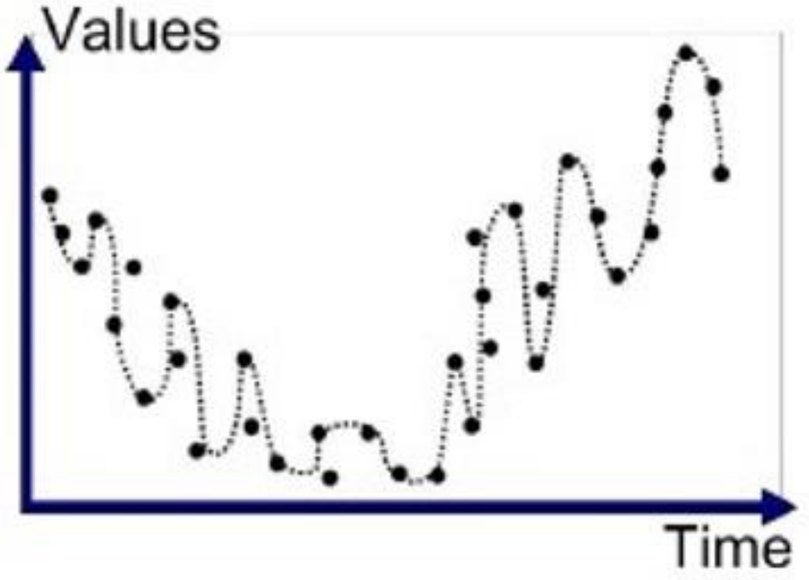
Good Fit/Robust

All done!

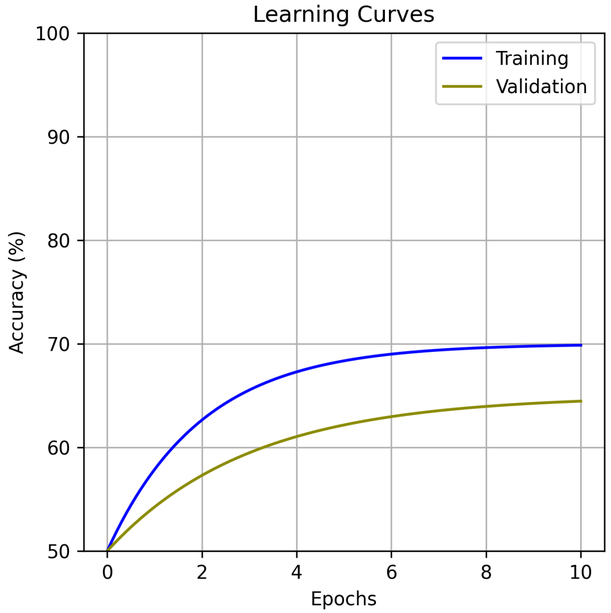




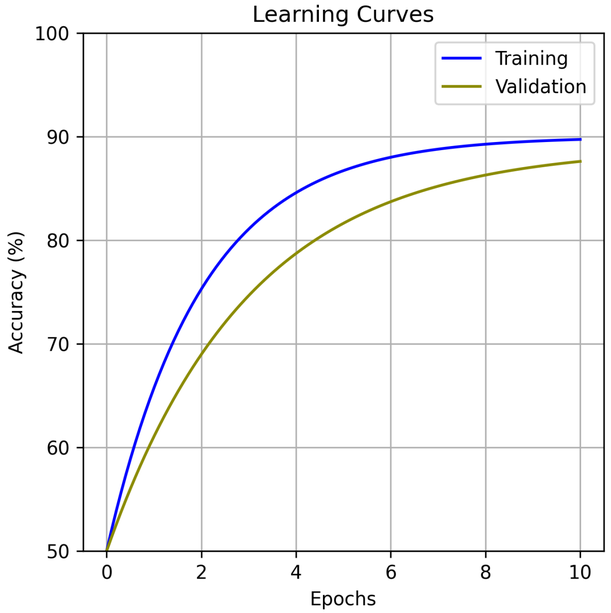




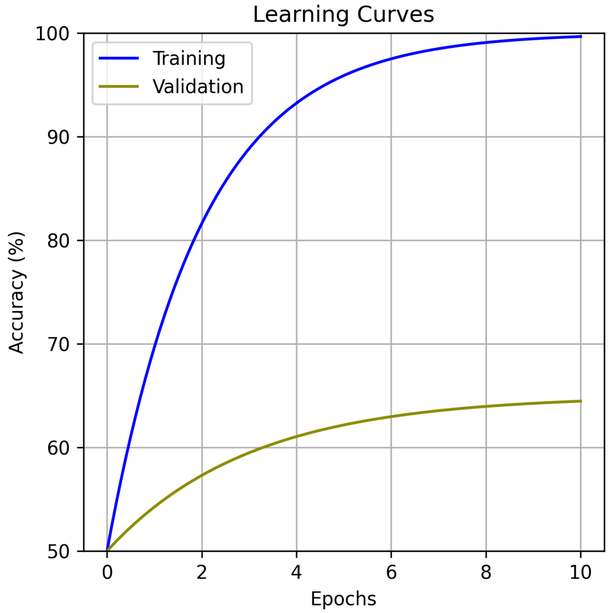
Underfitted



Good Fit/Robust



Overfitted



Tip: Take a small sample of your training set and try to overfit it.

Lifecycle of a computer vision DL project

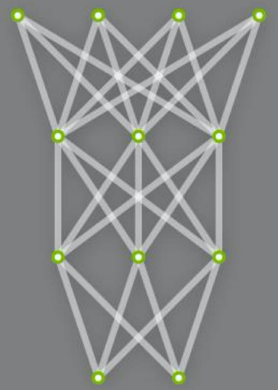
1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

DEEP LEARNING

TRAINING

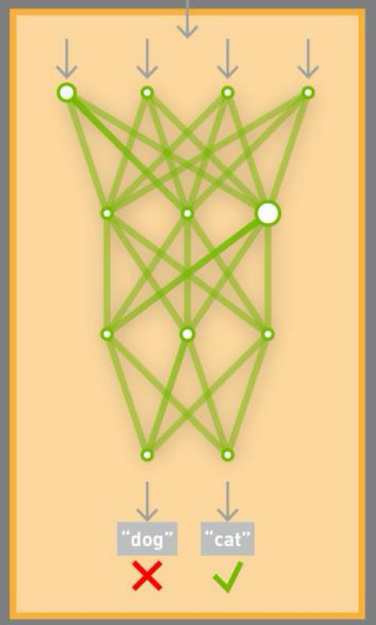
Learning a new capability
from existing data

Untrained
Neural Network
Model

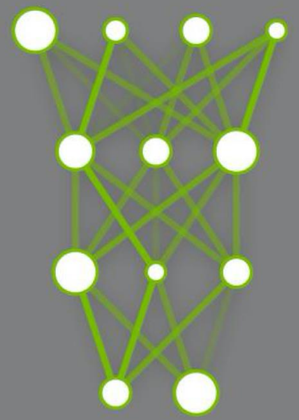


Deep Learning
Framework

TRAINING
DATASET



Trained Model
New Capability

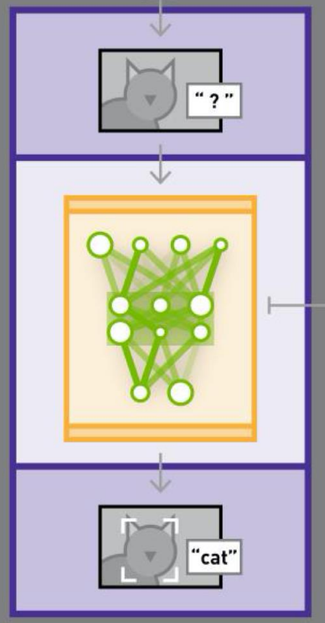


INFERENCE

Applying this capability
to new data

NEW
DATA

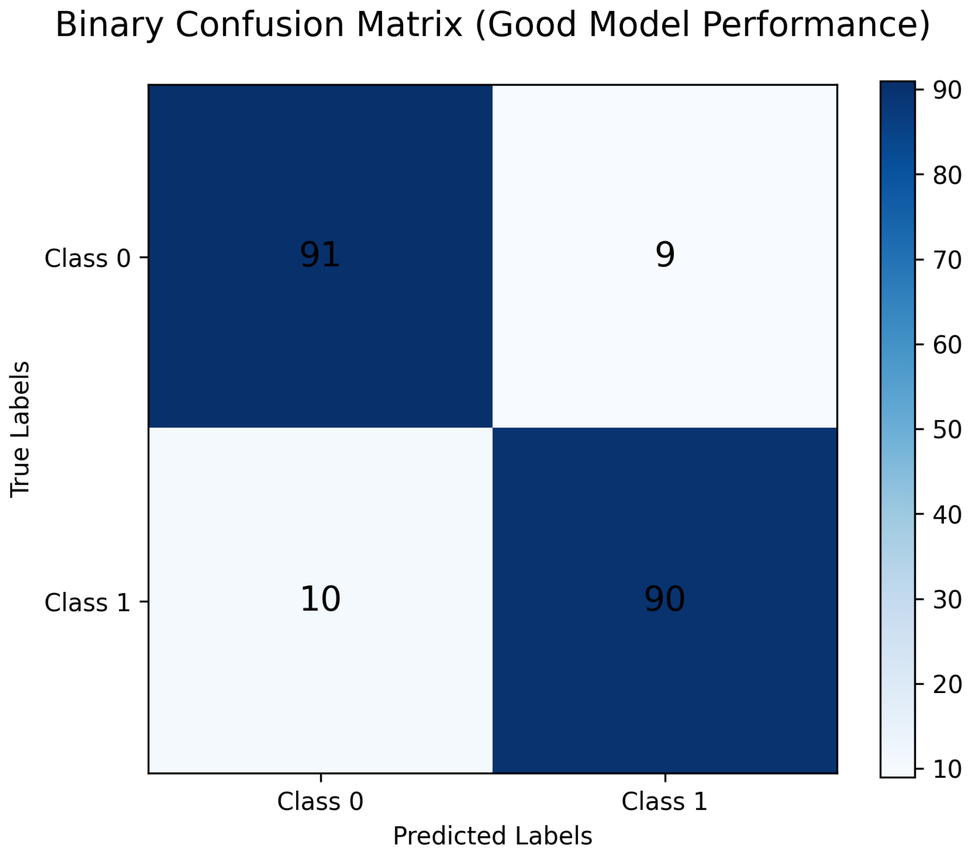
App or Service
Featuring Capability

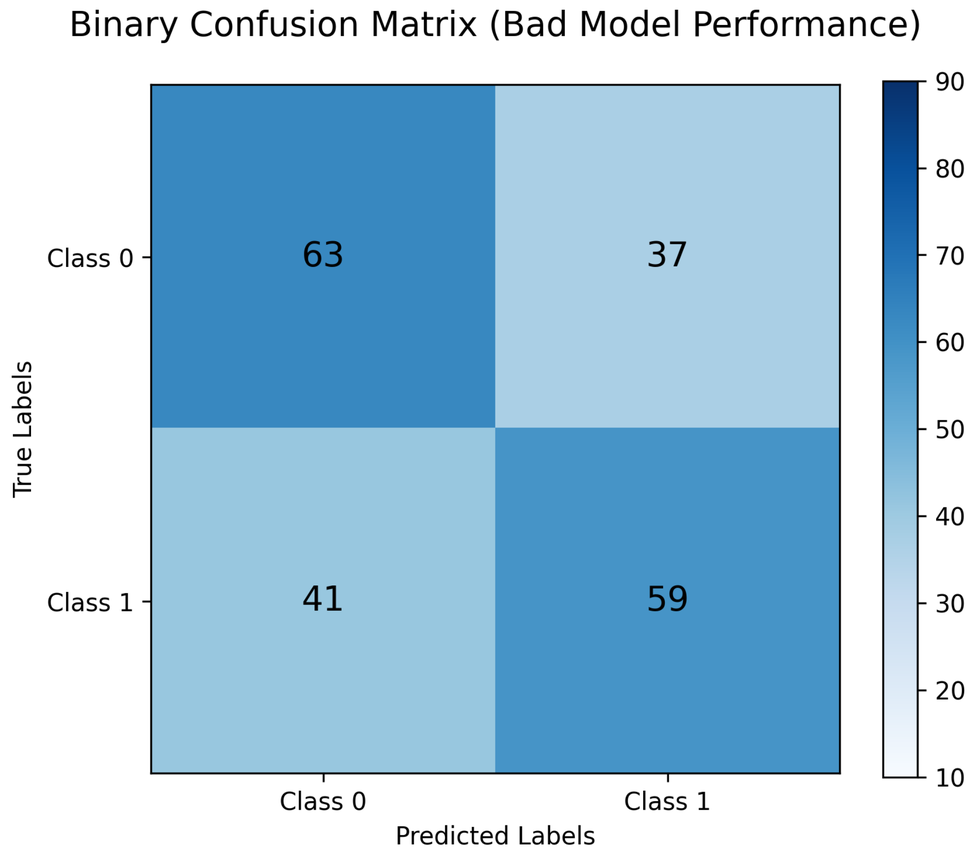


Trained Model
Optimized for
Performance

Confusion Matrix

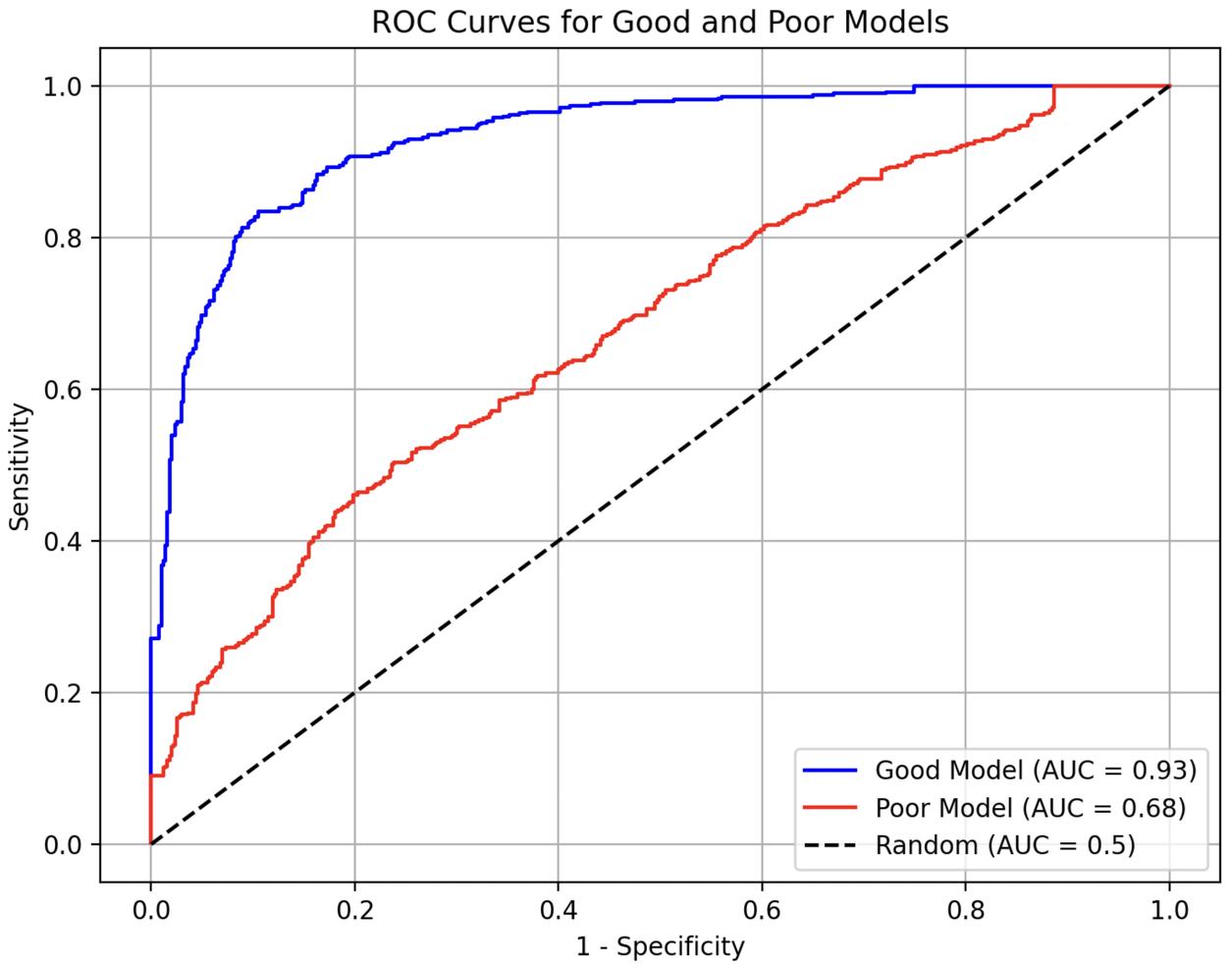
- (Basically a 2x2 contingency table for binary classification)





Receiver Operator Characteristic curve (for balanced datasets)

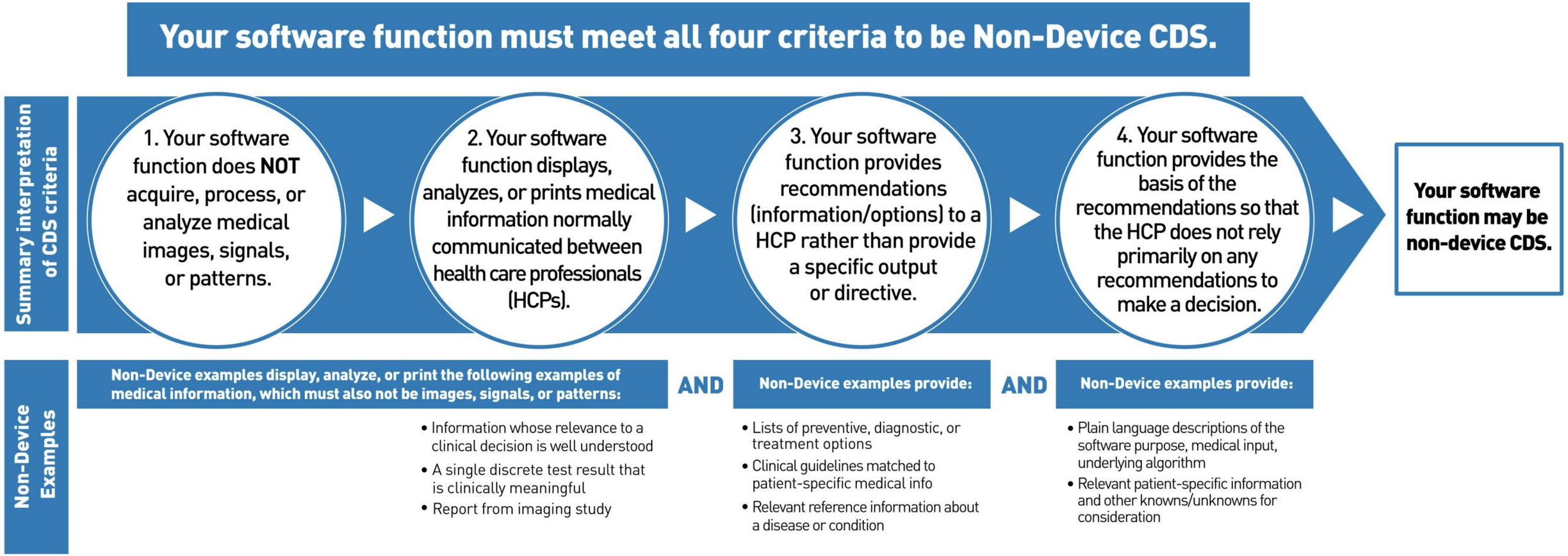
- The model will produce an output between 0 and 1.
- We can vary the threshold and recalculate the Sensitivity and Specificity for each threshold
- Random behavior = diagonal line
- Area under curve = AUROC



Lifecycle of a computer vision DL project

1. Define your problem (model)
2. Collect, clean, collate, partition data (input)
3. Torture humans to label the data (output)
4. Train model (hyperparameters)
5. Repeat 4, sometimes 2-4 (model parameters/weights)
6. Test the model a final time (test set performance)
7. Write a paper, publish a github repo (data + model + weights or at least model + weights)

Do you need FDA approval before you can use this in clinical care?



Acknowledgements

Funding Sources: NIH 1OT2OD032644, NIH/NIA R01 AG060942, NIH/NEI K23EY024921, NIA/NIH U19AG066567, Research to Prevent Blindness, Latham Vision Research Innovation Award, Latham Fund for Vision Research, Donors to Computational Ophthalmology Fund

Aaron Lee, MD MSCI

Cecilia Lee, MD MS

https://comp.ophthalmology.uw.edu

Megan Lacy MS
Julia Owen PhD
Yue Wu PhD
Scott Song BA

Missy Takahashi BS
Ashley Batchelor MS
Matthew Hunt BS
Theodore Spaide PhD

Emily Heindsmann MA
Christina Duong BS COA
Yelena Bagdasarova PhD

Marian Blazes MD
Jamie Shaffer MS
Yuka Kihara MS
Randy Lu BS