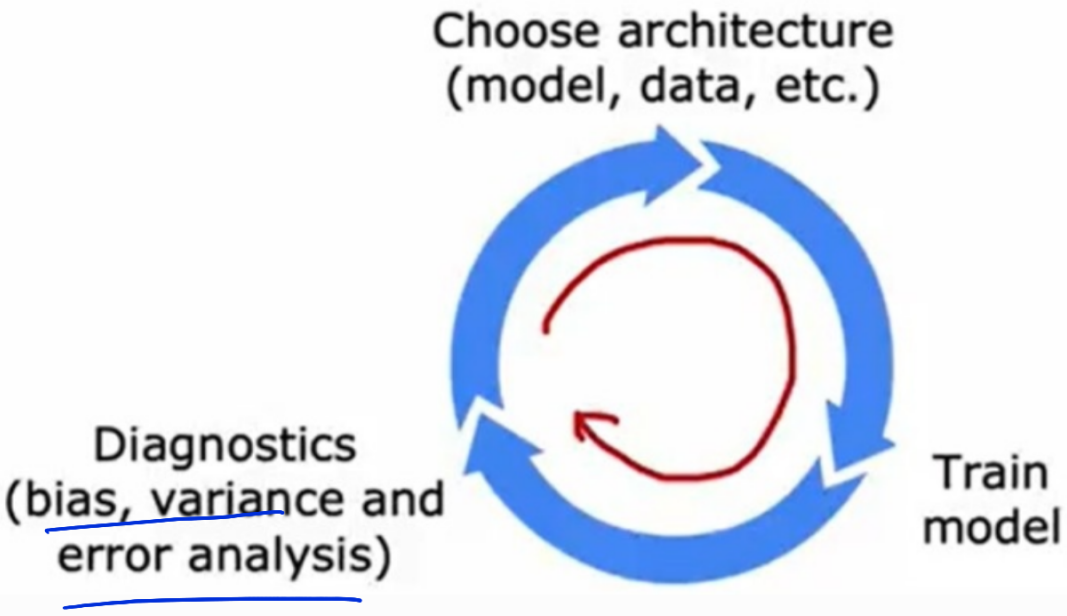
Iterative loop of ML development



Build a spam classifier;

$\vec{x}$ = features of email

top 10,000 words

no. of times of u.

y

spam (1)

not spam (0)

model → logistic reg. / neural network

How to try to reduce your spam classifier's error?

- Collect more data. E.g., "Honeypot" project. ← bias/var
- Develop sophisticated features based on email routing (from email header).
- Define sophisticated features from email body. E.g., should "discounting" and "discount" be treated as the same word.
- Design algorithms to detect misspellings. E.g., w4tches, med1cine, m0rtgage.

Error analysis;

check misvalidated examples & categorized based on common traits;

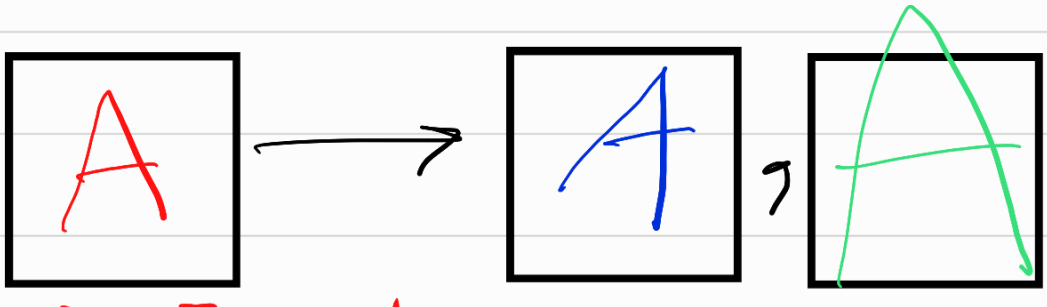
ex; mispellings deliberate
unusual email routing
steal p.w → phishing
spam msg in embeded img

or → check for a sample subset

Adding Data

→ Add data of type error analysis helped to ind.

Data Augmentation → sp. images ex: / speech



or ↔ some letters

or adding distortions;

Speech

Original audio

+ Noisy background: Crowd /Car

+ on bad cellphone connection

Increase data artificially.

* does not help to add purely random/meaning less noise to data

Synthesis:- create brand new data

mostly for computer vision

Artificial data synthesis for photo OCR

Real data

Synthetic data

Adam Coates and Tao Wang]

from text editor different fonts

Conventional model-centric approach;

↓

AI = Code + Data

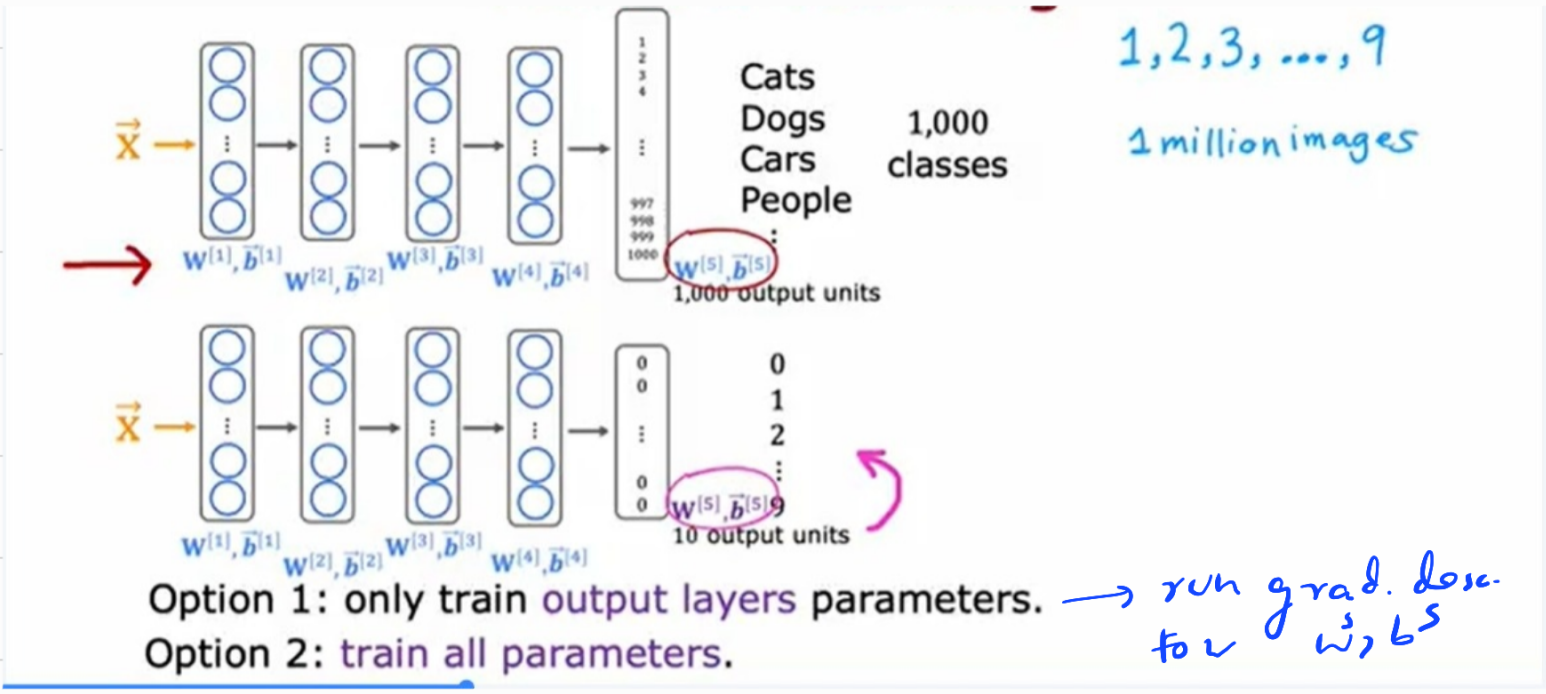
(model, algorithm)

↑

Data-centric Approach

Transfer Learning

- use data from different task



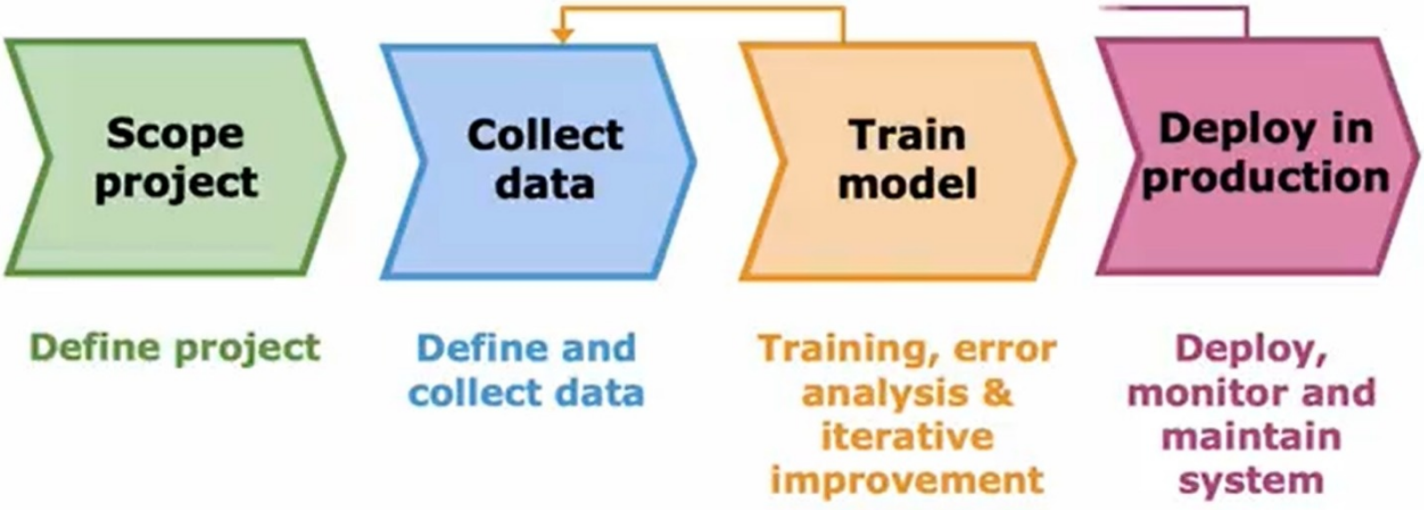
~~1st step~~ download a neural network

→ replace output layer &

fine tune with option 1 or option 2

1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text) as your application (or train your own). 1 million images
2. Further train (fine tune) the network on your own data. 10 00 images

Full cycle of a machine learning project



Inferense server

ML Model

API call

(x)

Mobile app

$\hat{y}$

Inference

Scaling

Logging

System monitering

Model updates

MLOPS

machine learning operations

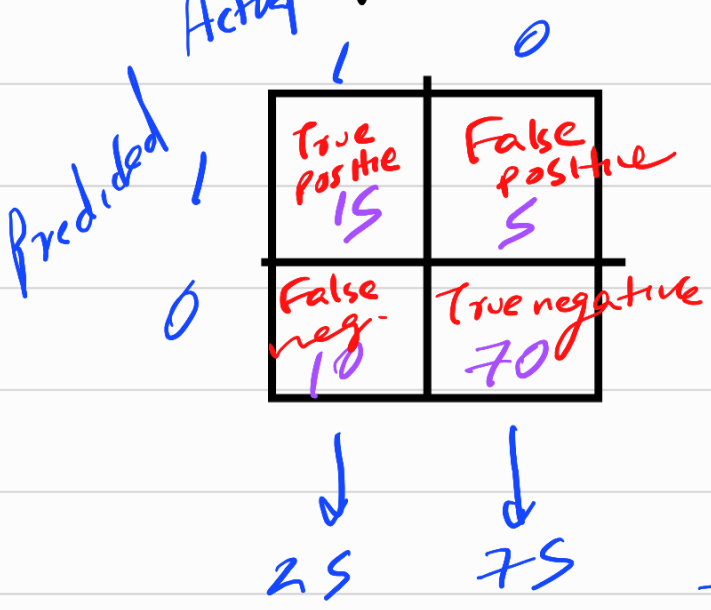
Skewed data;

1% error is not always good;
think a disease rare 0.5%.
then $y=0$ has 0.5% error (patients)

↓

Error metrics for this

precision / recall



Precision

fraction actually have rare disease?

$$\frac{\text{True posit}}{\text{predicted post.}} = \frac{\text{True pos.}}{\text{True pos} + \text{false pos}}$$

$$= \frac{15}{15+5} = 0.75$$

Recall

fraction have disease and correctly detect

$$\frac{\text{True post}}{\text{Actual post.}} = \frac{\text{True pos.}}{\text{True pos} + \text{false neg.}} = \frac{15}{15+10} = 0.6$$

Trading off precision and recall

Logistic regression: $0 < f_{\vec{w},b}(\vec{x}) < 1$

→ Predict 1 if $f_{\vec{w},b}(\vec{x}) \geq$ ~~0.5~~ ~~0.7~~ ~~0.4~~ 0.3

→ Predict 0 if $f_{\vec{w},b}(\vec{x}) <$ ~~0.5~~ ~~0.7~~ ~~0.4~~ 0.3

$$\text{precision} = \frac{\text{true positives}}{\text{total predicted positive}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{total actual positive}}$$

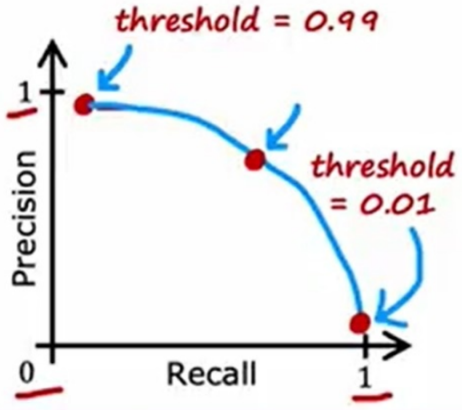
Suppose we want to predict $y = 1$ (rare disease) only if very confident.

higher precision, lower recall

Suppose we want to avoid missing too many case of rare disease (when in doubt predict $y = 1$)

lower precision, higher recall

More generally predict 1 if: $f_{\vec{w},b}(\vec{x}) \geq$ threshold.



F1 score;

↳ combine precision & recall

$$F_1 \text{ score} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)}$$

$$= 2\frac{PR}{P+R}$$ ← Harmonic mean

=

if precision & recall lot smaller then low sore (y=0 useless ones)