Date:

A B C E
S F E S.
A D E E

---

$^9C_6$ $\frac{9!}{6!3!} = \frac{9\times8\times7}{3\times2} = 84$

$\frac{8!}{6!2!} = \frac{8\times7}{x} = 28$

$^3C_2 = \frac{3!}{2!} = 3$

# COURSE FAST-AI

## Lesson1: Getting Started

For every chapter
- Read Summary
- Watch Video
- Do exercises in video
- Read Related Chapters
- Solve chapter exercises
- Read Supporting Material

## Building a Bird Classifier

- Download first 4 bird images & resize & save them
- Remove broken links
- Create DataLoaders

Ch 8 What is fast AI

Date:

- train the data
- Verify Results

[CV models are easier because you can quickly verify results]

* SOTA - Twitter Bio Images using Dall E Midjourney Pathways Language Model - PaLM - explaining jokes & Math

- You dont need to be a google to create the best AI

• Why could we not build bird classifier before?
- Manually research a lot of features which takes time. Neural train doesn't use this. It uses forests - We don't give features, we ask it to learn it.
We don't need to handcode the features.

* Logic similar to Image Classifier can also be used for sounds
* Various other similar ideas can also be used.

* Deep learning Myths - You need 1) Lots of Math 2) Lots of Math 3) Lots of expensive computers.

* Pytorch is hairy - more of approach from scratch
FastAI - library on top of Pytorch which provides abstraction for things you don't need to worry about.

Date:

* Datablocks – Fast AI abstraction for building data.
- blocks = (Inpt, atpt) → Eg – (Img Block, Category Block)
- get_items = fn to where to get data from
splitter → for test, train split
getty → get correct category for input data
item_tfms → transform fn that runs on each item

↳ creates dataloaders which are passed to Pytorch

* learner – combines the model & data (dataloaders obj)

model → actual neural network fn
– there are some fixed models which are good for a wide variety of use cases
* TImm – library for TIMM SOTA Img Models for pytorch
* for some ready made models they also contain the best ~~wgts~~ weights to start with, instead of starting with some random weights
* & We may need to fine tune these weights for our use case – a process called as fine-tuning

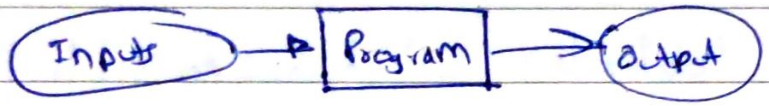
* Other OpenCV techniques
→ Segmentation → Identifying Segments
→ Tabular Analysis → Use tabular Dataloaders & tabular learners and use fit (Fit used when starting with random weights from scratch)

Date:

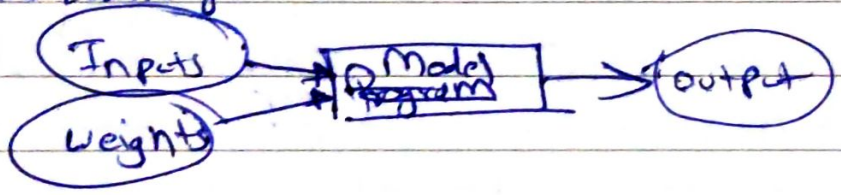
Collaborative filtering → Recommendation based on similar users.
using CollabDataLoaders & CollabLearners

(For range, start with a little value higher than min & max
eg (0.5, 5.5) for range in 0 to 5

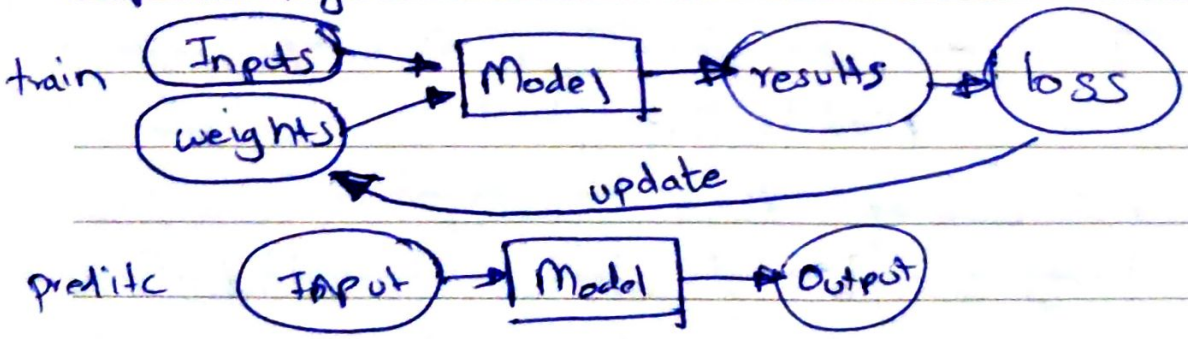
: Normal Program




Machine Learning




Deep learning




Todo 1) Build & Run bird example
2) Build something similar & post on forum & read others ideas
3) Read Chapter 1 & End of chapter Quiz & Exercise
4)