Fastai v2 (2020) [Jeremy Howard, Sylvain Gugger and Rachel Thomas]

fastbook -> fastai book

Lecture1

What you don't need, to do deep learning

Myth (don't need)	Truth
Lots of data	Not true always
Lots of math	High school math is sufficient
Lots of expensive computers	We can get what we need for state of art work for free

AGI -> Artificial General Intelligence

Where is deep learning the best known approach

NLP, CV, Medicine, Biology, Image generation, Recommendation systems, Playing games, Robotics

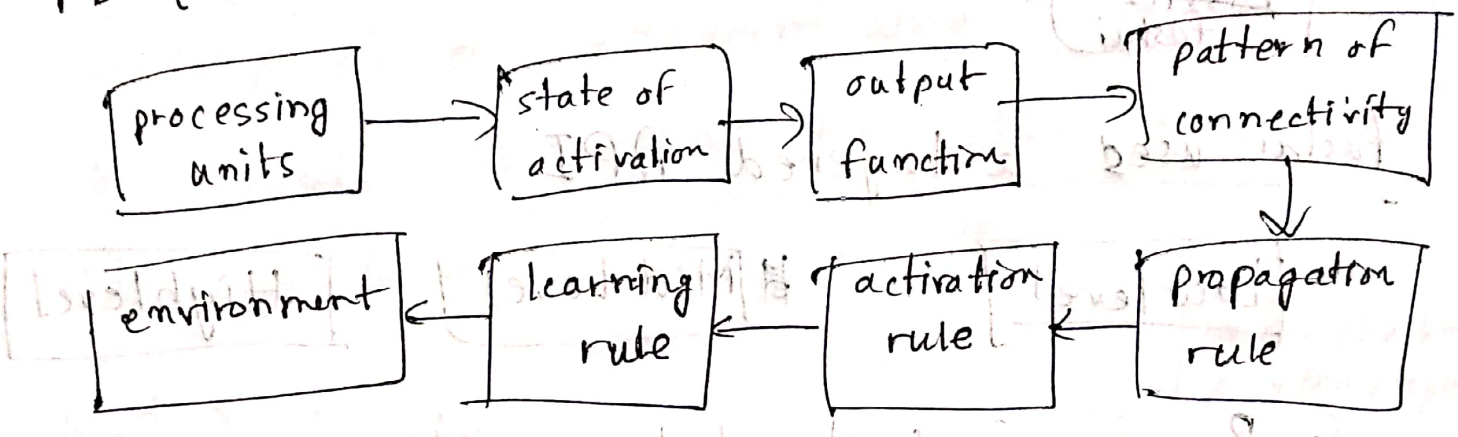
Neural Network: (origin)

1943 Warren McCulloch, a neurophysiologist, and Walter Pitts, a logician, teamed up to develop a mathematical model of an artificial neuron

Mark-I (Frank Rosenblatt)

Marvin Minsky (multilayer pectron) but people ~~misinterp~~ misinterpret

PDP (Parallel Distributed Processing) 1986 MIT



The age of deep learning (1980s)

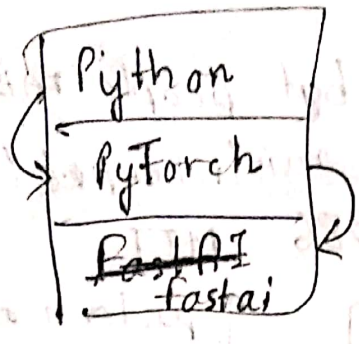
universal approximation theorem: stacking up layers with non-linearity in between can allow any model to ~~learn~~ be approximated

Now we have "a machine capable of perceiving, recognizing and identifying its surroundings without any human training or control."

Course strategy

- Play the whole game
- Make the game worth playing
- Work on the hard parts

[top-down approach]



~~PyTorch~~

fastai used a layered API

Low level | Mid Level | High level

- Removes boilerplate

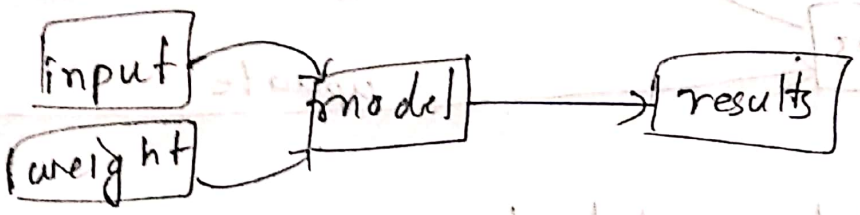
→ Getting a GPU (first learn and do it in colab)

~~Ren~~

Fine-tuning: A transfer learning technique where the params of a pretrained model are update by training for additional epochs → using a different task to that used for pretraining

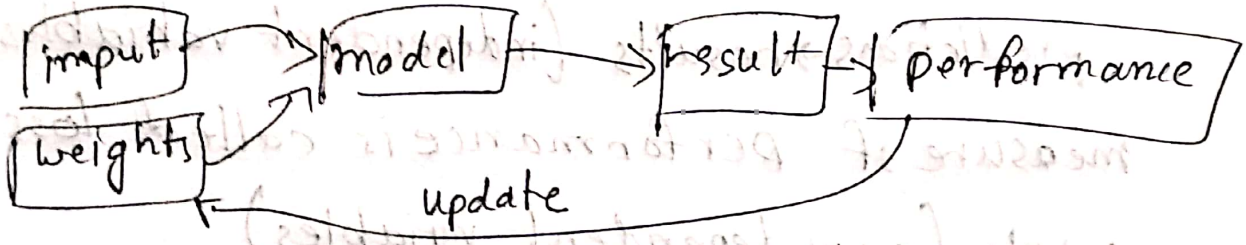
Right back at the dawn of computing in 1949, an IBM researcher named Arthur Samuel started working on a different way to get computers to complete tasks, which he called machine learning. ~~In his classic way 1962~~

1962 →

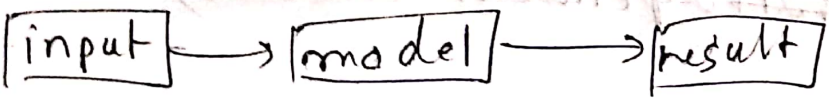


his checkers playing program bebeted Connecticut state champion

Training a machine learning model

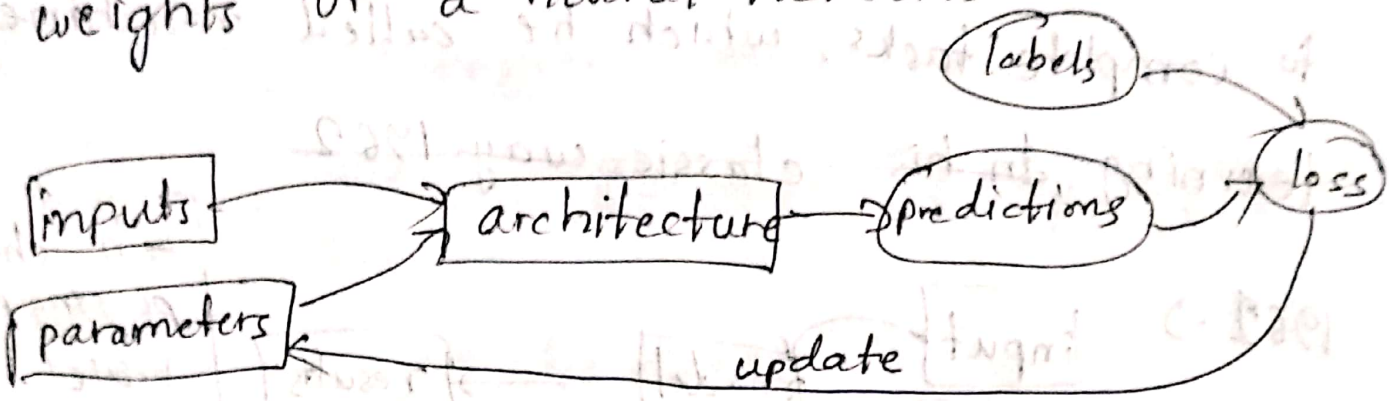


Later



Machine learning: training programs developed by allowing a computer to learn from experience, rather than through manually coding the individual steps.

- SGD (Stochastic Gradient Descent) is a completely general way to update the weights of a neural network



~~para~~ model → architecture

~~p~~weights → parameters

predictions → results (independent variables)

measure of performance is called loss

labels (~~var~~ dependent variables)

Limitations

- ~~Dat~~ need data
- can only learn from input data (creates bias)
- creates predictions → not recommended actions
- Not enough to just have examples of input data we need labels for that data too