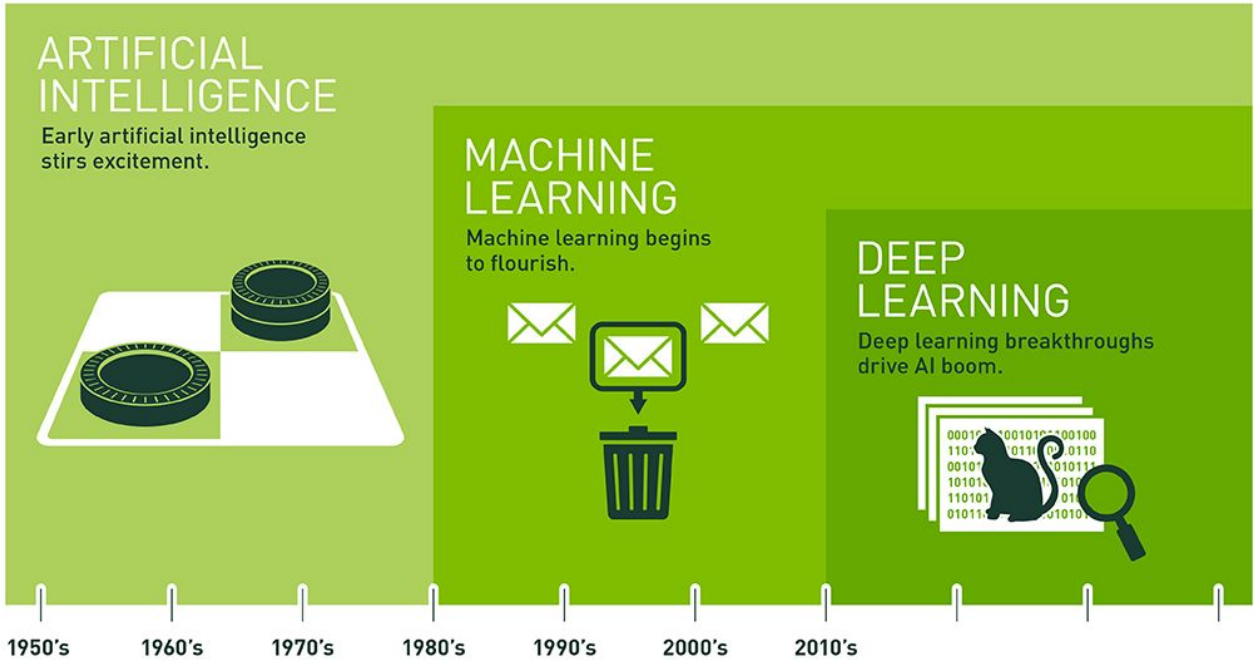
Applied Deep Learning

Nitish Bhardwaj

Overview of Session-1:

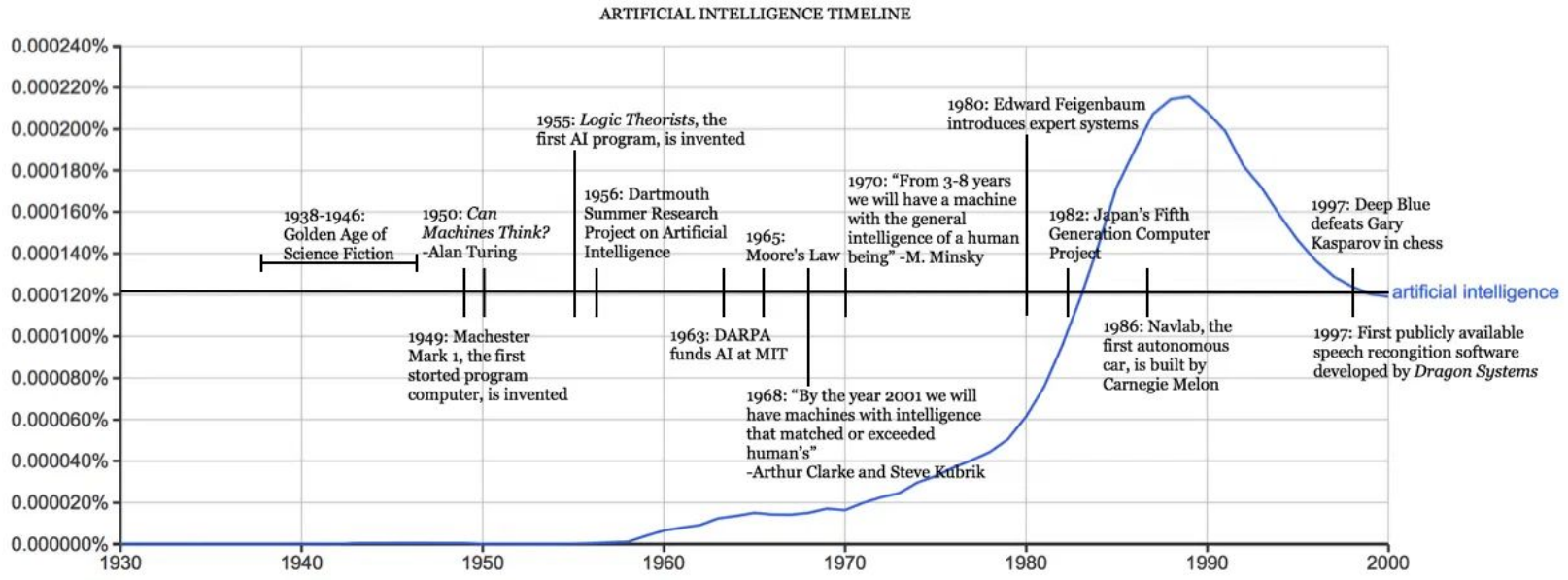
- Evolution of AI
- Introduction to Deep Learning
- Industrial Use-case

Evolution of Artificial Intelligence

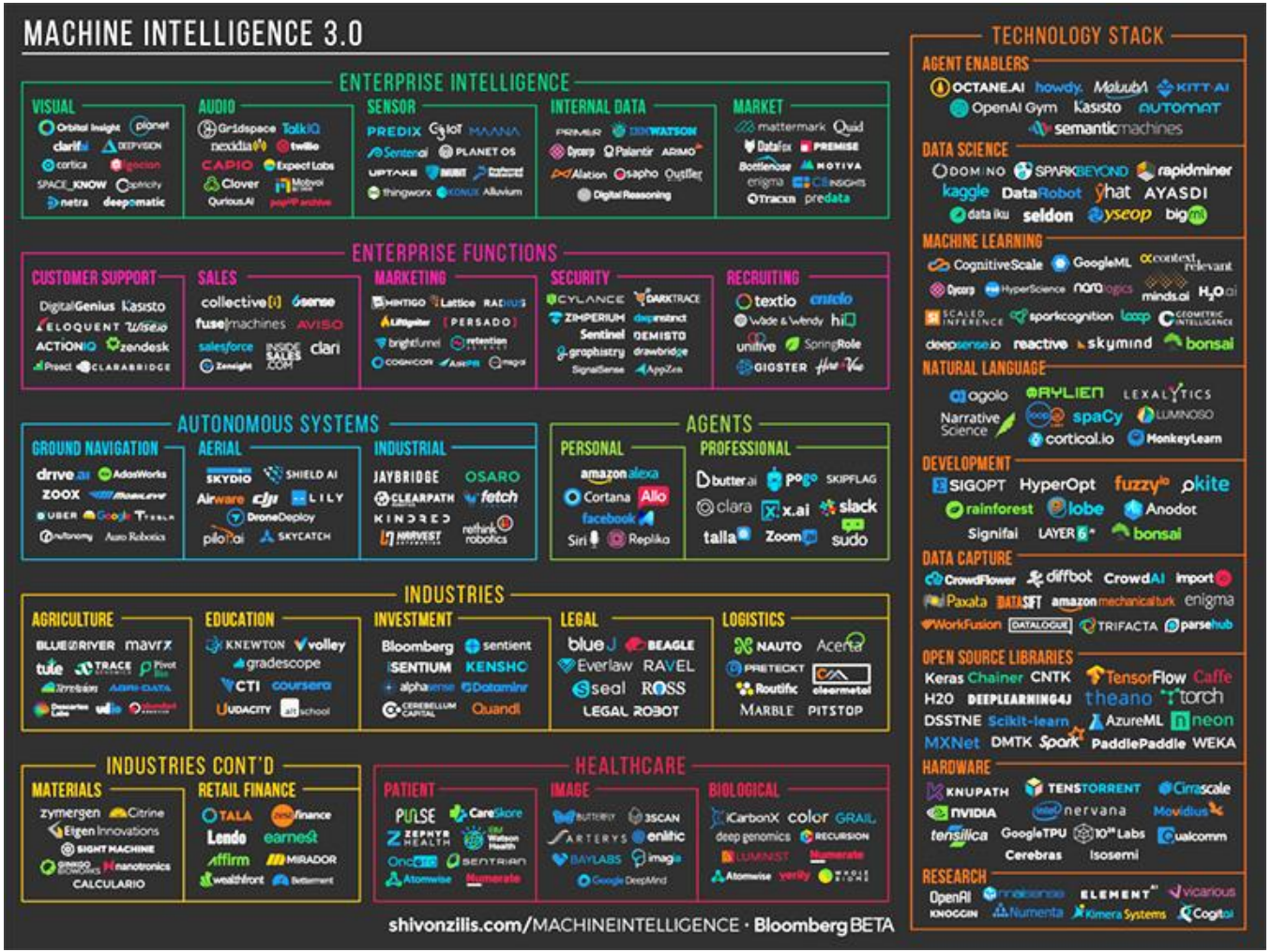


Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Evolution of Artificial Intelligence



Applied Artificial Intelligence

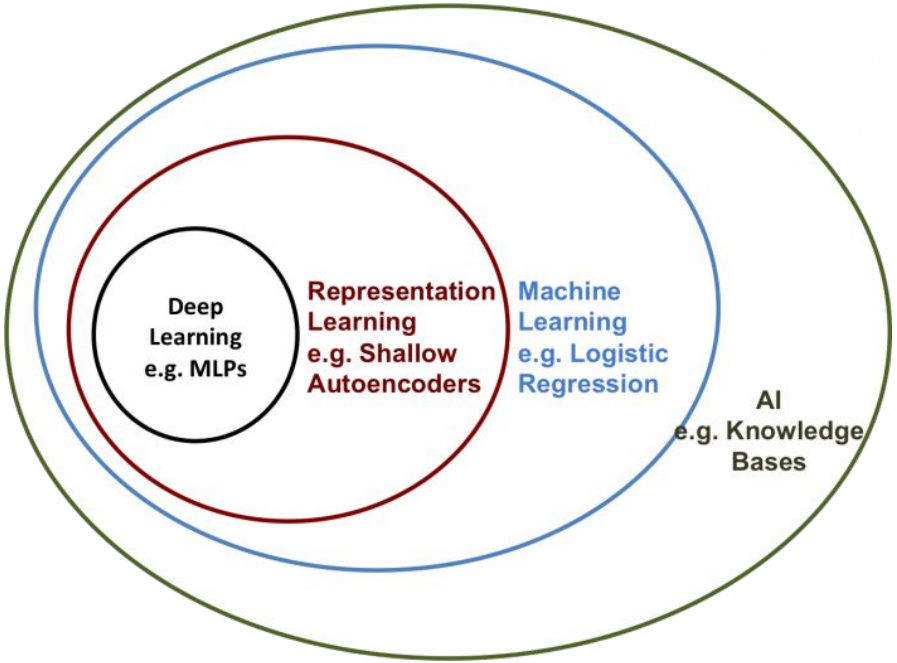


AI vs Machine learning vs Deep learning

AI (*technique to enable machine to mimic humans*) is super-set of ML

ML (*program machines to learn and improve with experience*) and **ML** is the super-set of **DL**.

DL (*achieve great power and flexibility by learning to* ***represent the data as nested hierarchy of concepts****, i.e, more depth features with multiple processing layers, hence "deep"*).



Why is everyone talking about **Deep Learning**?

- DL is working really well (with awesome *accuracy numbers*) to solve some big problems like Image classification, Face Recognition, Recommendation system, Sentiment Analysis and a lot more.

- And it's easier to implement in "**10 lines of code**"

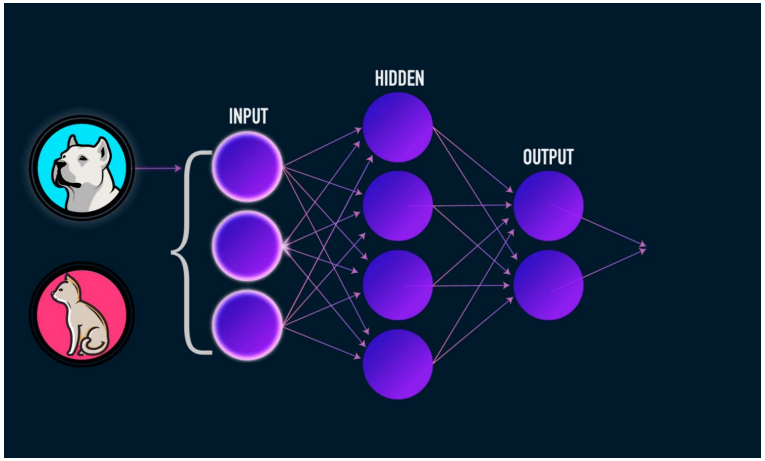
Why is everyone talking about **Deep Learning**?

- **These problems are solved in 90s, aren't they ??** Then why Deep Learning? Because accuracy matters a lot and really a lot.
- DL has transformed the industry by solving these *already solved problems* with ***great accuracy numbers***.
  - For example, *detection of hand-written digits, accuracy on test data*
    - ***88%*** *(LeCun et al. 1998) to*
    - ***99.7%*** *(Ciresan et al. CVPR 2012).*
- DL has also brought solutions to many unsolved problems.
- Open-source + Academic Industry collaboration + Fight for Accuracy

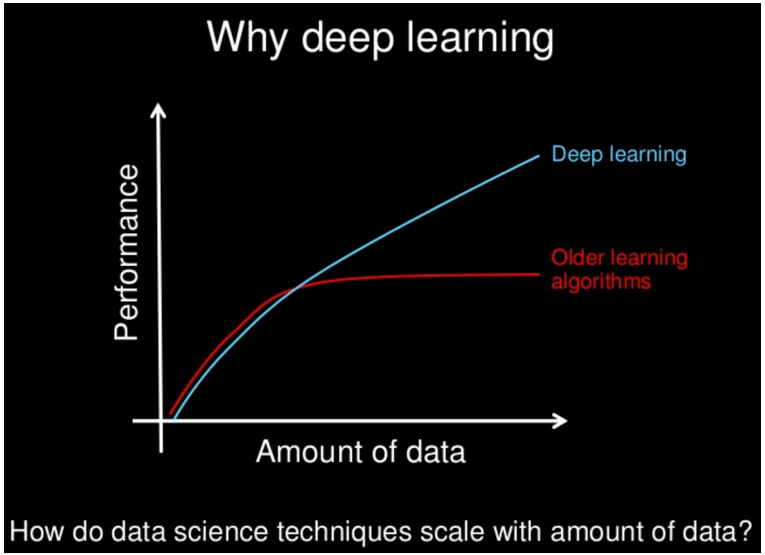
Introduction to Deep Learning

Deep Learning

- Based on Concept on Artificial Neural Networks



- "*The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones. If we draw* ***a graph showing how these concepts are built on top of each other****, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning.*" - Ian GoodFellow



Deep Neural Networks

Dataset

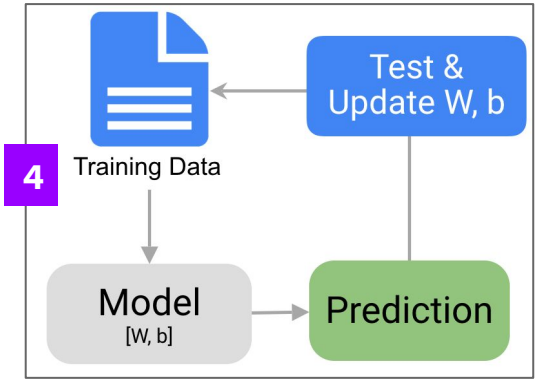
Inputs			Output
x1	**x2**	**x3**	**y**
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

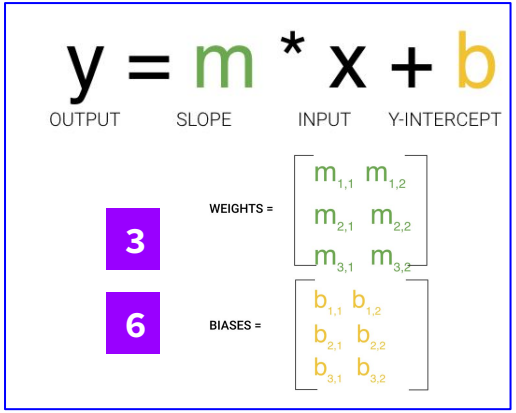
Problem solving in 7 steps

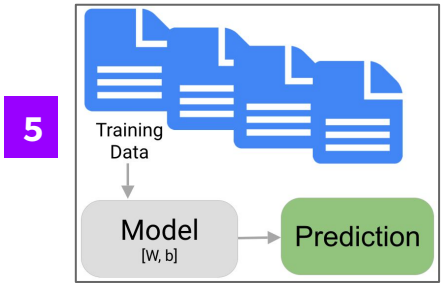
1. Problem Statement
2. Data Preparation
3. Choosing a **model, y=f (x)**
4. **Training**
5. Evaluation / **Validation**
6. Hyperparameter tuning
7. Prediction/**Testing**

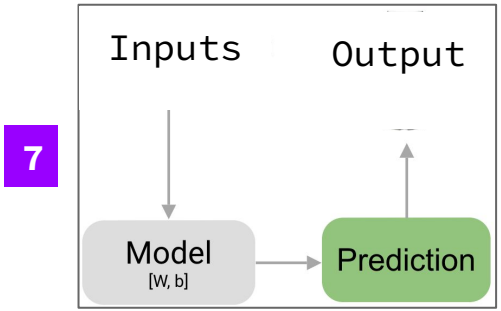
1 Classification Problem

2 Data Preparation





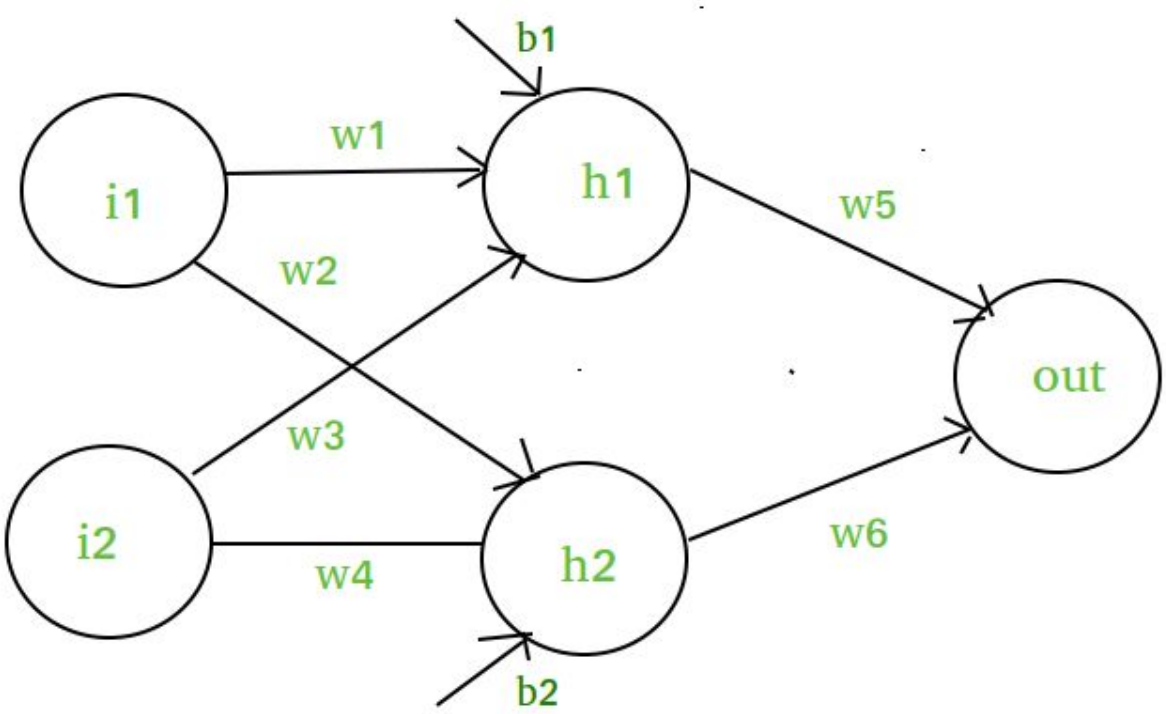




Problem Solving :

Inputs			Output
x1	**x2**	**x3**	**y**
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

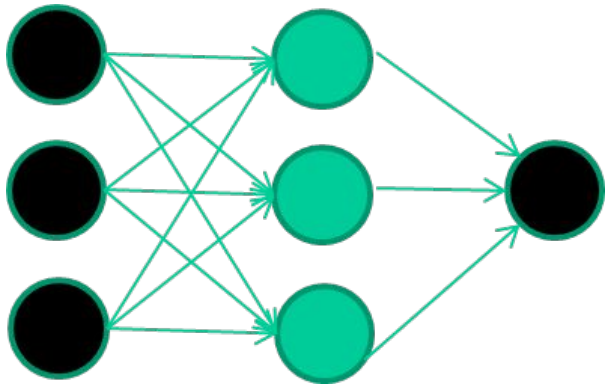
Weights and Biases in Neural network



A dataset

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

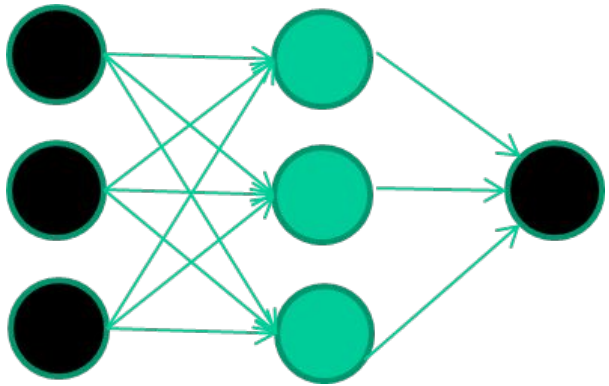
etc …



Training the neural network

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

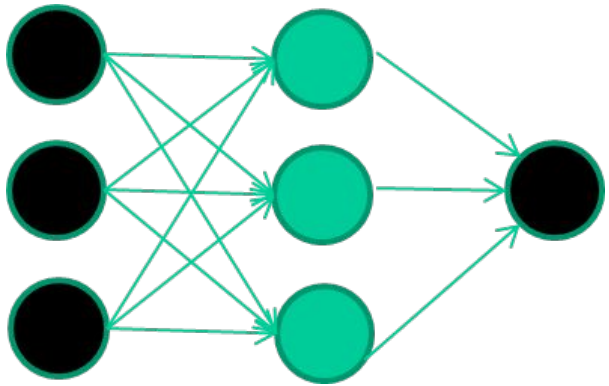


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

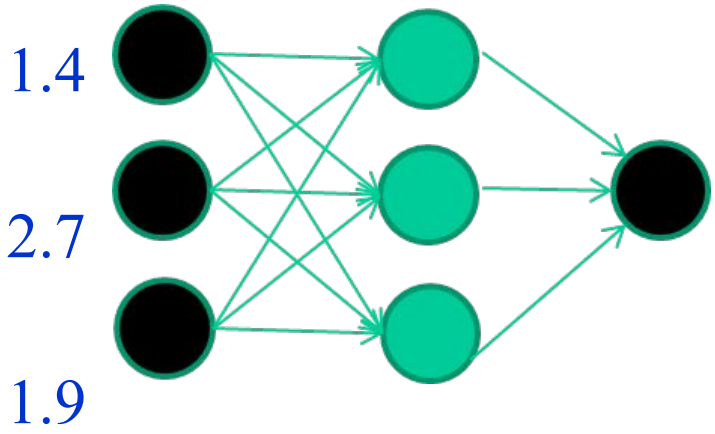
Initialise with random weights and bias



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

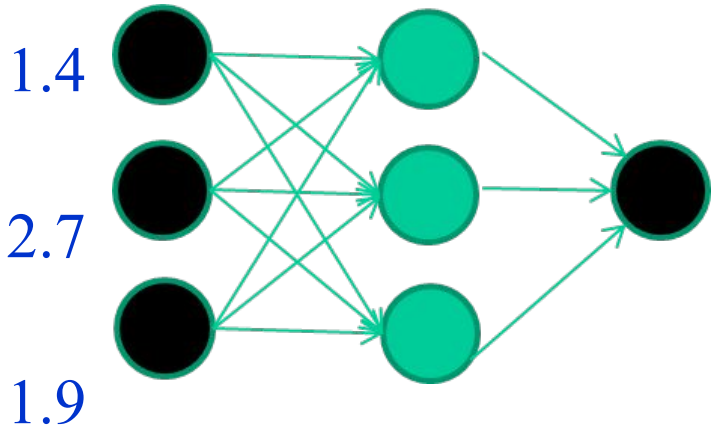
Present a Training Pattern



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

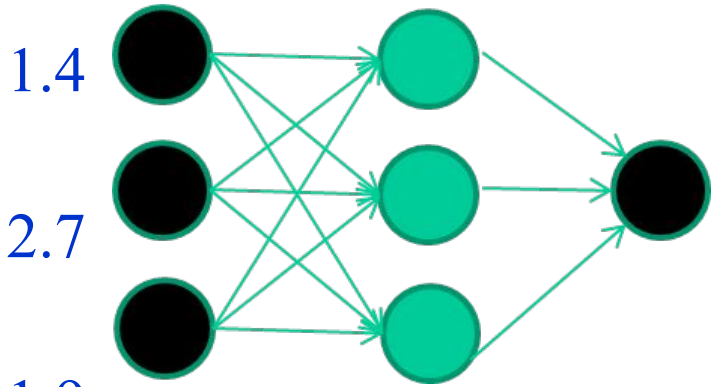
Feed it through the output



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Compare with target output



0.8

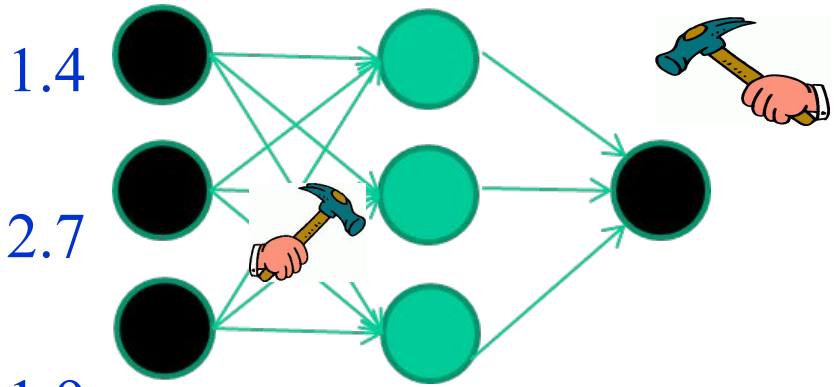
0

error 0.8

Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

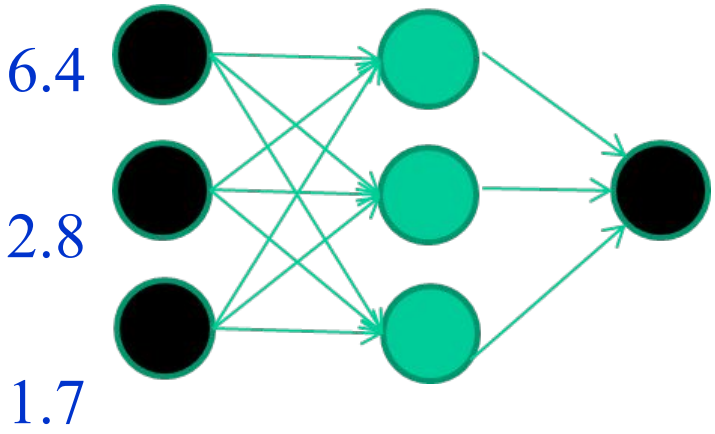
Adjust weights based on error



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

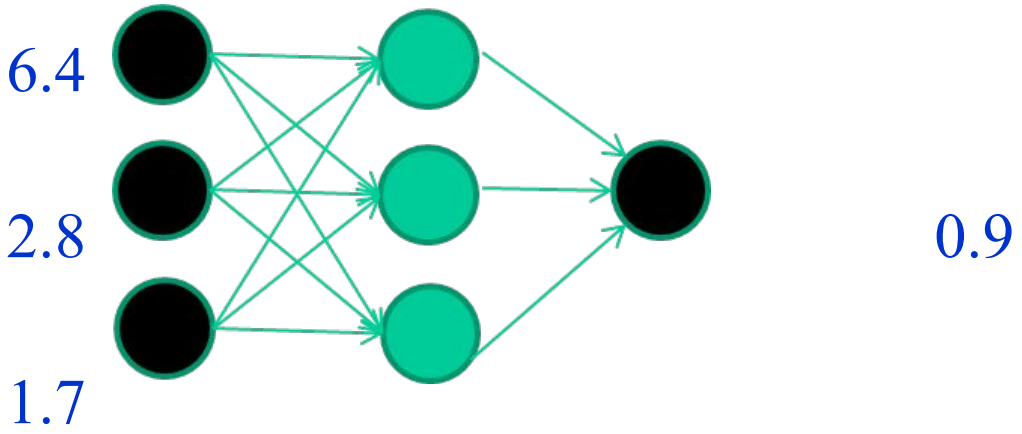
Present a Training Pattern



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

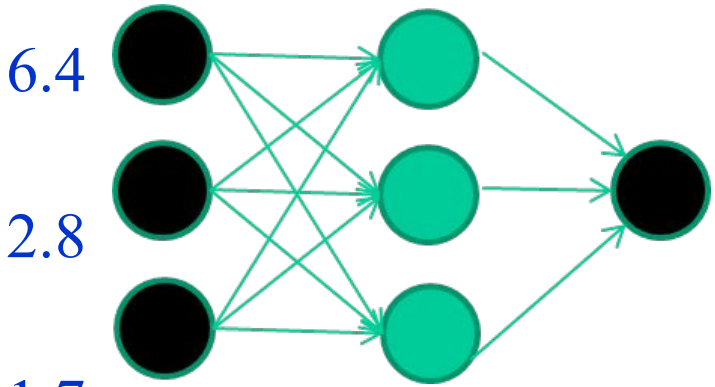
Feed it through the output



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

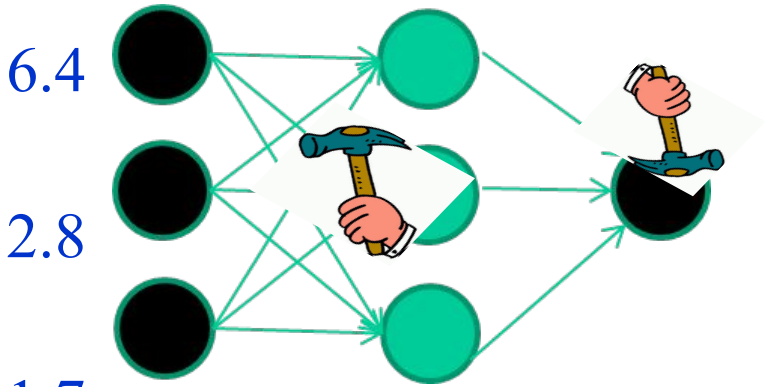
Compare with target output



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Adjust weights based on error

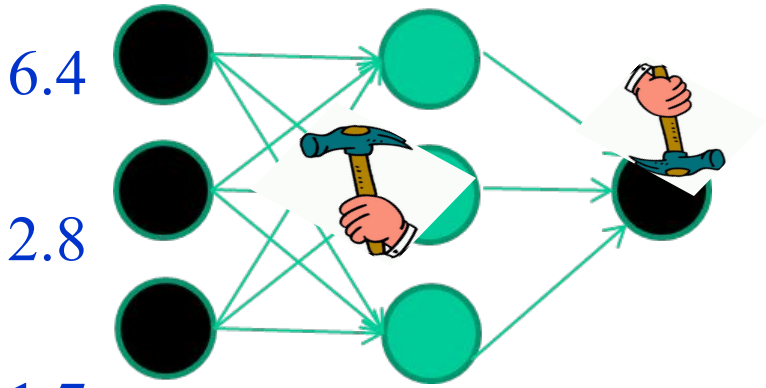


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

And so on ……..

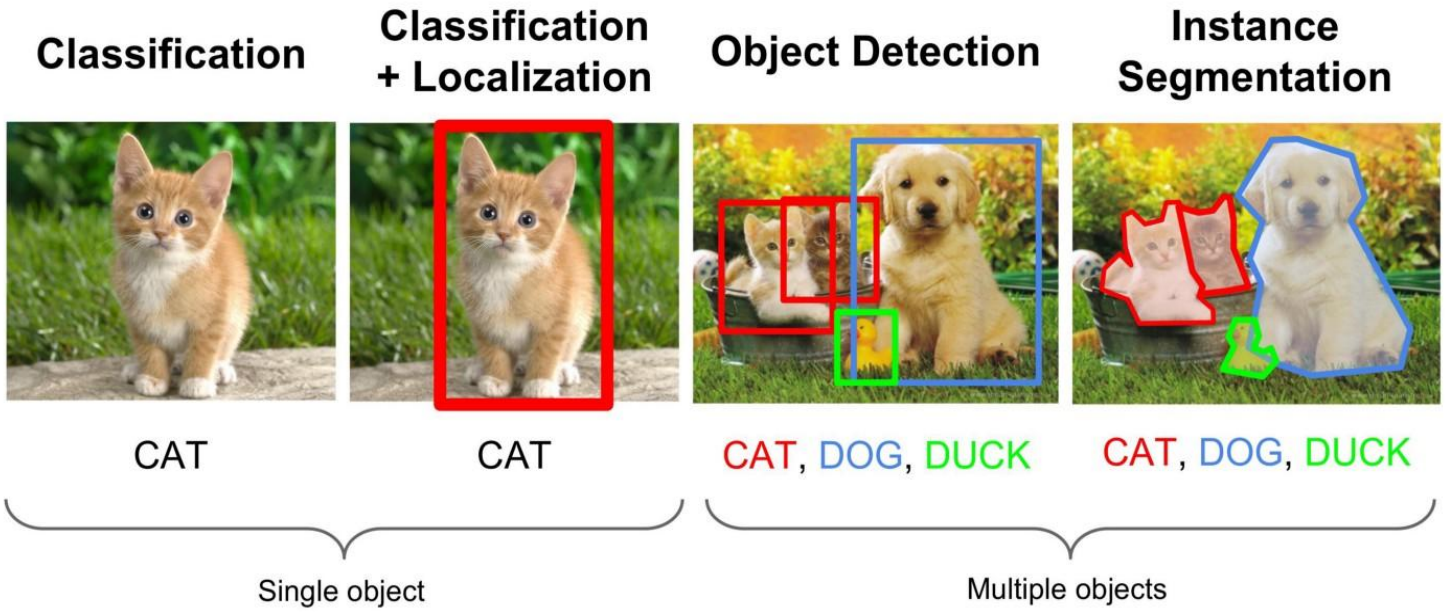


Repeat this thousands, maybe millions of times – each time taking a random training instance, and making slight weight adjustments

Algorithms for weight adjustment are designed to make changes that will reduce the error

Business use-cases

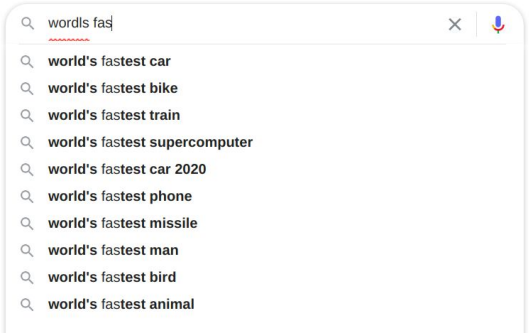
Deep learning with Images : **Computer Vision**



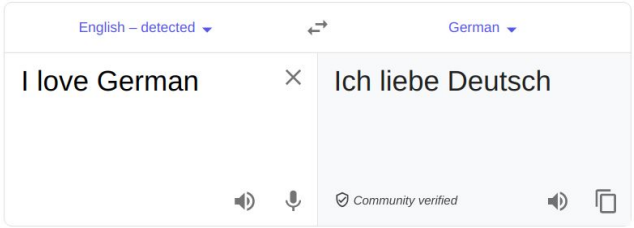
- Classification
- Detection
- Segmentation
- Generation
- And Lot more

Deep Learning with Text : **Natural Language Processing**

Search Autocorrect and Autocomplete



Language Translation



- Text Classification
- Chatbots
- Voice Assistants
- Social Media Analytics
- Advertisements
- Summarization
- And lot more

Deep Learning with Number data :

Stock Market Prediction

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

- House Price Prediction
- Recommendation system
- Disease Identification
- Medical Reporting
- And lot more

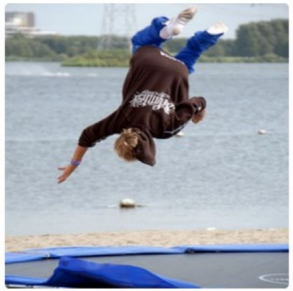
Complex Deep Learning

Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."

Images

Text

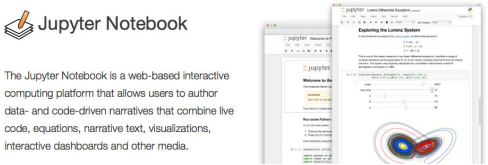
Overview of Session :

- AI / ML / DL
- Applications of CV, NLP

Let's code...

https://www.youtube.com/watch?v=inN8seMm7UI

https://colab.research.google.com/



Google Colaboratory

References

-
- Steps of Machine Learning : https://www.youtube.com/watch?v=nKW8Ndu7Mjw
- Tensorflow Playground : https://playground.tensorflow.org/
- Google Colab : https://www.youtube.com/watch?v=inN8seMm7UI
-

Thank you!

Compiled By Nitish Bhardwaj
https://www.linkedin.com/in/nitish-bhardwaj/