# Twitter Thread by Sahil Sharma

**Sahil Sharma**
@sahil_vi

■■■■ ■

**Artificial Inteligence = ■■ + ■**

**When can you expect your fridge & mobile phone to revolt against you? (Spoiler Alert: Not in the foreseeable future).**

**Please Do 're-tweet' the Thread & help me spread the knowledge to more people.**



1. The Language
In order to understand Artificial intelligence, machine learning, deep learning, we need to start at the very basics.

Humans speak and understand a wide variety of 'natural' languages (this is what computer science community calls them): like English, Hindi, Marathi, Bengali, Tamil etc.

But have you seen a peculiarity of human languages? They do not have a precise meaning associated with a given sentence. On an average, every English word has 2 meanings. This makes human conversations confusing, with the need for multiple turns & clarifications. Example ■:

Person 1 conversing with Person 2 face to face:

Person 1: Lets go watch The Dark Knight. Its our marriage anniversary!
Person 2: I dont like going out at night. Its too scary & depressing.
Person 1: Oh, what i meant was watching the movie "The dark Knight".

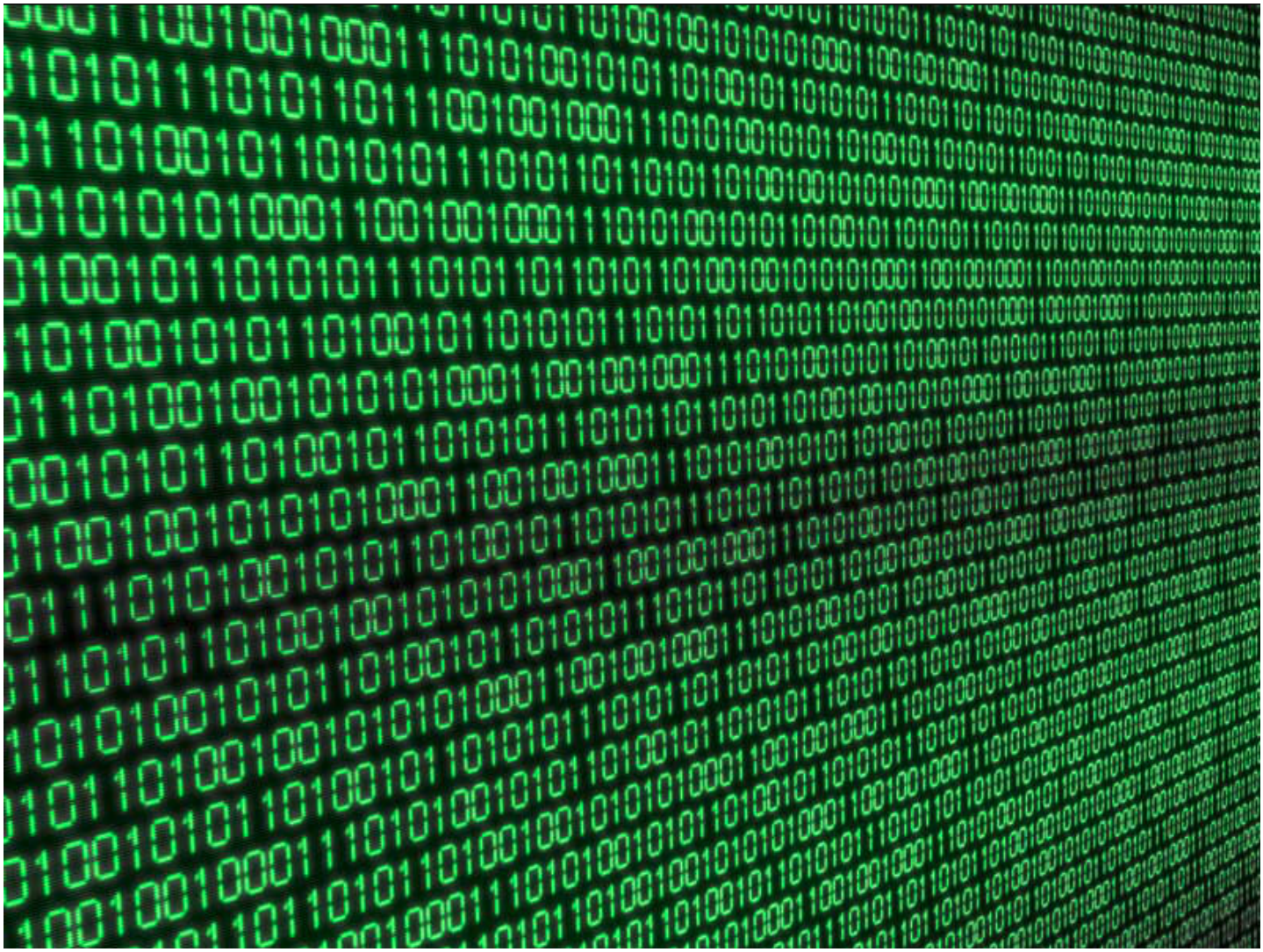
Person 2: Yeah yeah, sure. Let's all get together and watch ""the dark knight"". Because that is what everyone loves watching on their anniversary.
Person 1: Thanks honey, you're the best :D.

Person 2: Are you serious right now?
Person 1: As serious as Harvey Dent. As serious as Gautam Gambhir's surname.
Person 2: Okay, I dont know who they are, and obviously I was being sarcastic.

I hope the reader can appreciate that misunderstandings & miscommunications are a part & parcel of human language. We certainly did not & really more importantly did not know how to build our computers in a way that they understand all this ambiguity.

So we gave them the most basic language possible. Binary.

But its more than that. This language, in which computers talk, every string has a unique precise way to understand the 'sentence' (also known as a computer program). No ambiguity. No Ambiguity.



print("Hello World!")

This is the stereotypical 1st program which any person new to Computer Science writes in one of the computer programming languages: Python.

Summary: Humans have an ambiguous uncertain way to communicate. Computers communicate in a language where each 'sentence' or 'conversation' (also known as a computer program) has a unique meaning.

2. The two schools.
As old as our tryst with computers is our yearning to see them do 'intelligent' things. This was the 1st time that humans have turned God. And we wanted to build our prodigy in our own image. Give it the gift of intelligence.

This gave rise to the field of Artificial intelligence (AI).

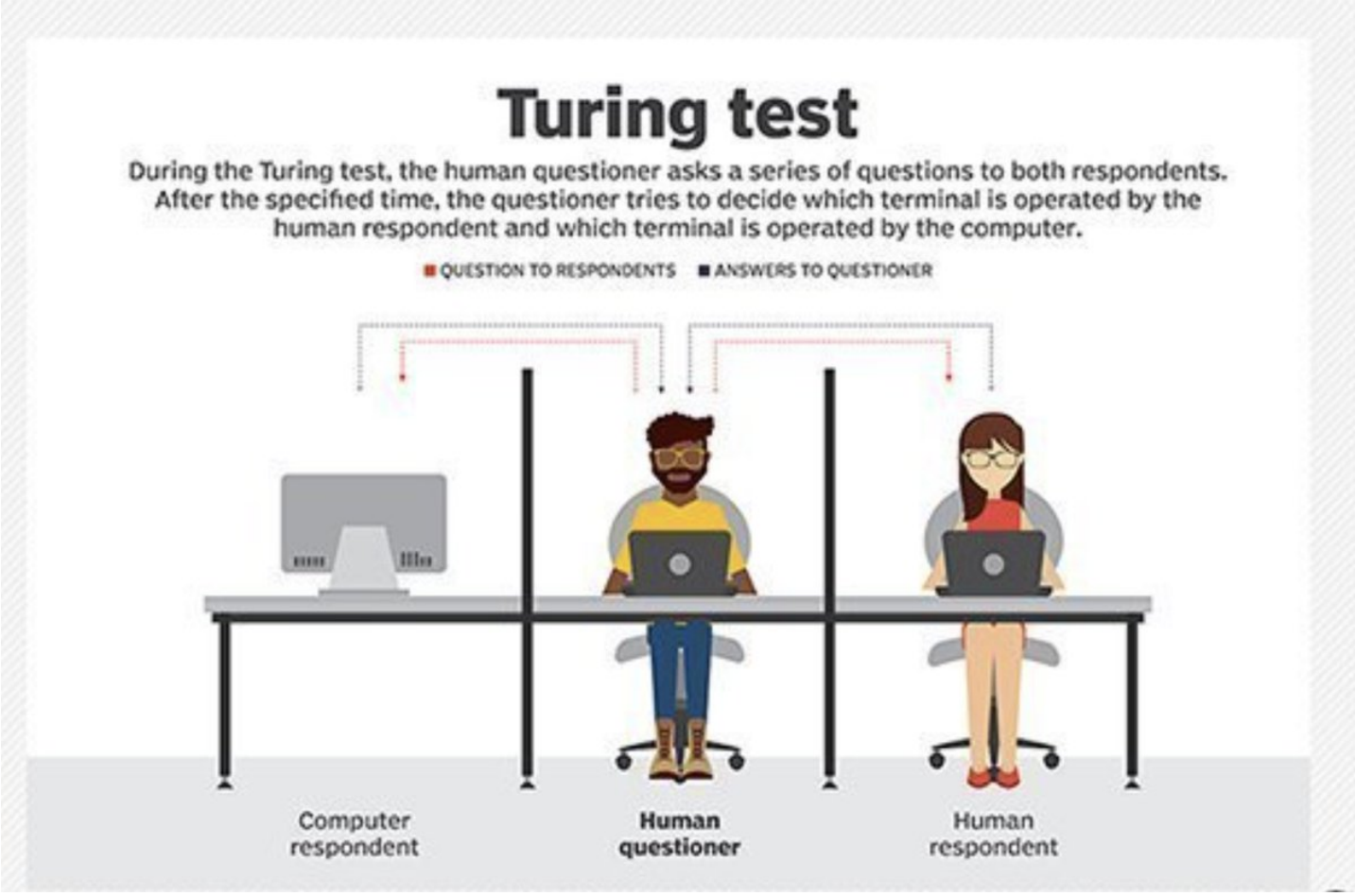
Artificial Intelligence: Any style of programming which enables computers to mimic humans.

The academic circles were somewhat divided in opinion in how they would go from computers with simple programs to computers which could demonstrate 'human like' behavior.

As an aside, one of the accepted tests of human like behavior (an Agni pariksha if you will) is what is known as a 'Turing test' named after the Father of computer science Sir Alan Turing.

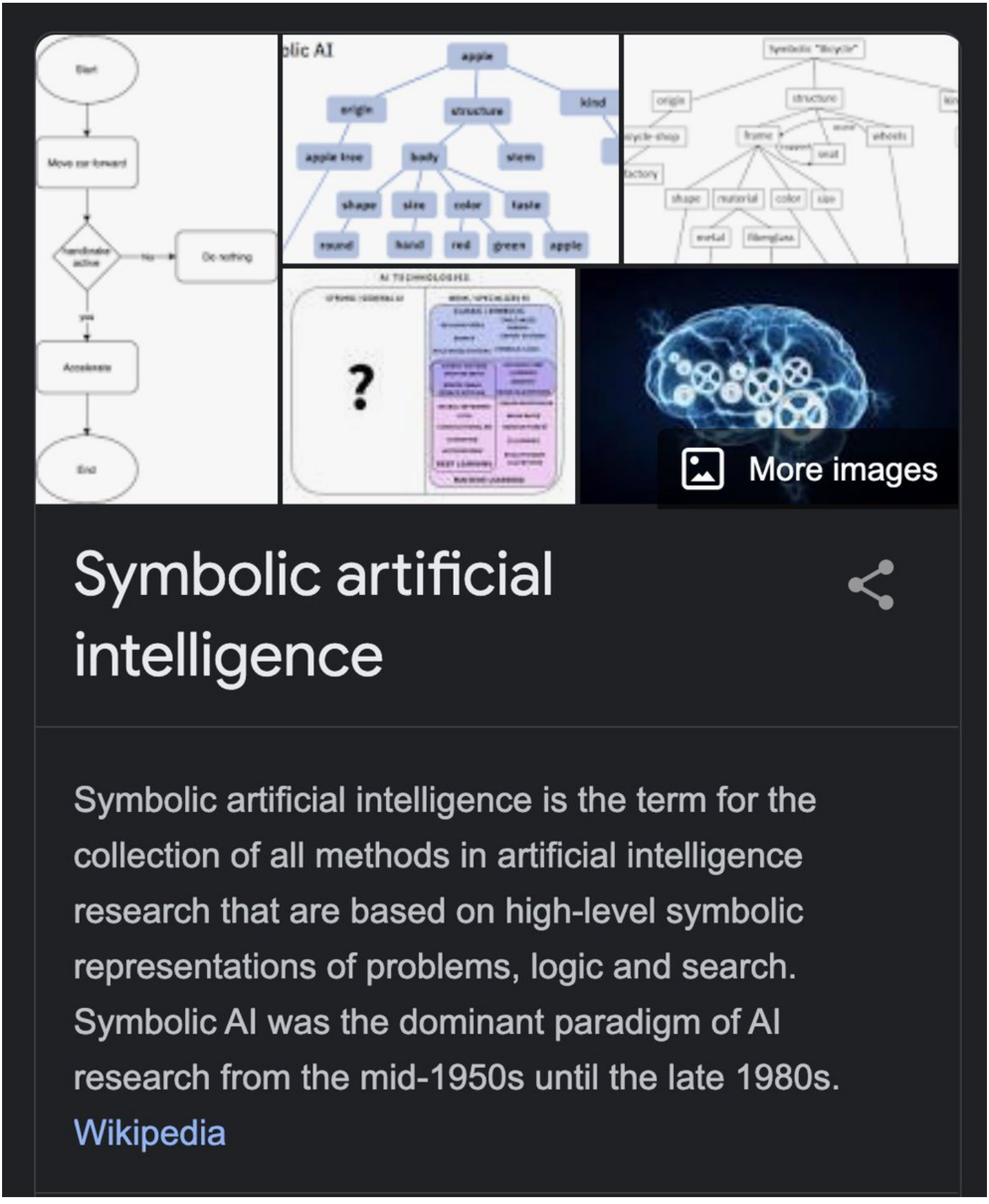


In Short, a turing test is successful if a computer can convince a human evaluator that it is indistinguishable from a human. They communicate through a medium like web chatting.



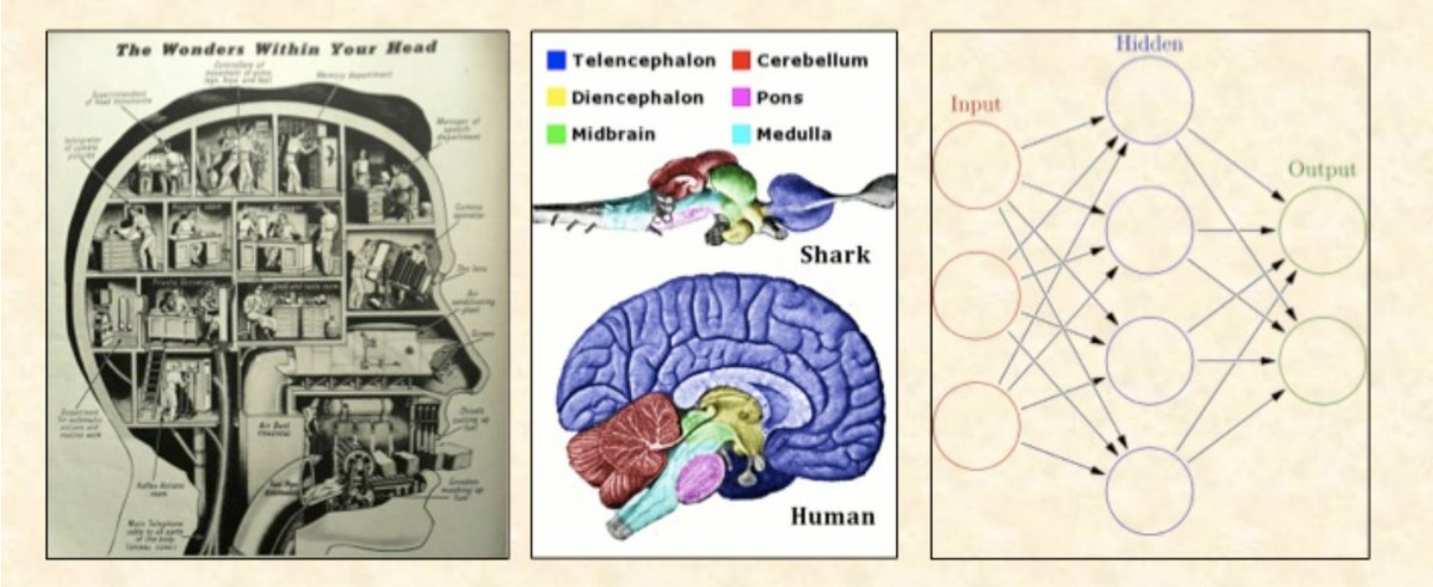

Back to the division. There were two schools of thoughts on how AI would be created.

1. The Symbolism school of thought: Cutting out all the jargon, they thought that AI would be created by programmers by codifying explicitly in a program exactly how it should respond to each situation it is presented with.

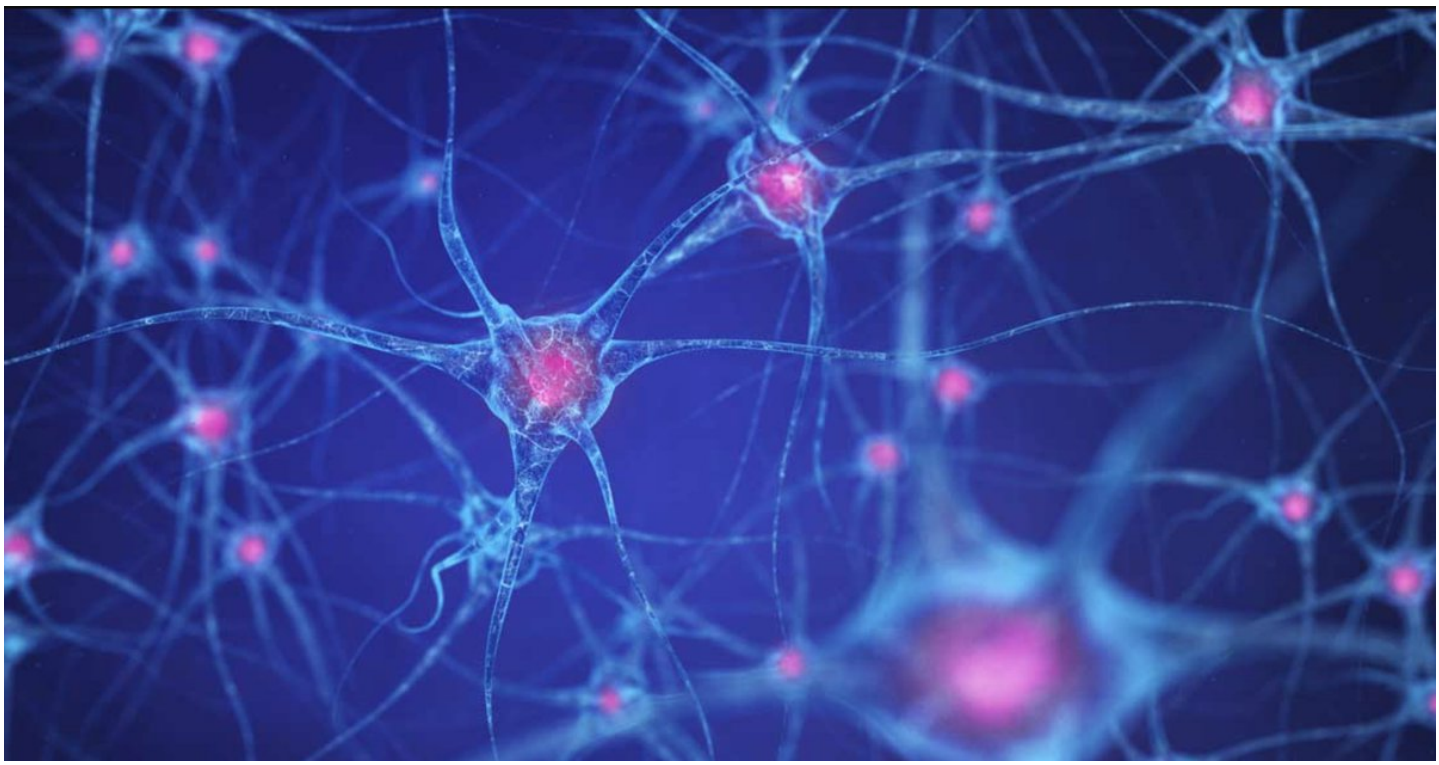



This is not as bad as it sounds. Computer programs compose, just as math functions.
so, through a limited set of atomic or basic programs we can represent an exponential set of outcomes.

2. The connectionist school of thought: This school of thought argued we already have a working example of intelligence: The human brain. They wanted to mimic the human brain & anatomy into the computer systems. Make the prodigal child in the parent's image.



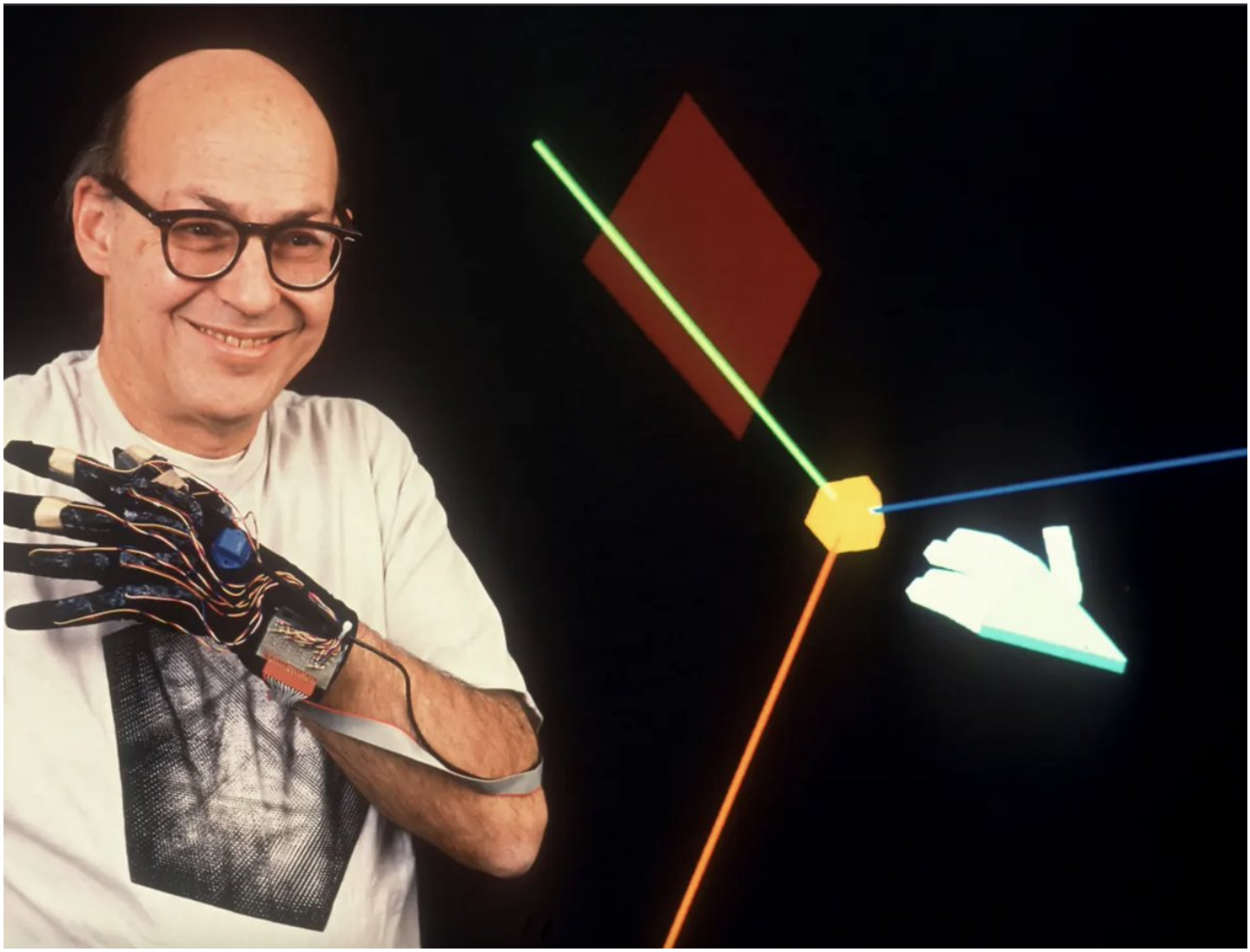

If we look at the human brain, it seems to be a network of atomic units called Neurons. What is a network? An web of nodes interconnected by edges. Neuron cell bodies are the nodes, the dendrites are the edges.



The connectionists had a Eureka moment. 'Why don't we try to mimic/simulate this in computers'! One of the very first advances here was 'Perceptron' by Frank Rosenblatt.



As an aside, the one of the pioneers of the field, Marvin Minsky who also started the AI lab at MIT wrote a book on perceptrons.



This book is the center of a long-standing controversy in the study of AI. It is claimed that pessimistic predictions made by Marvin were responsible for a change in the direction of research in AI & contributed to the AI winter of the 1980s, when AI's promise was not realized

Expanded Edition

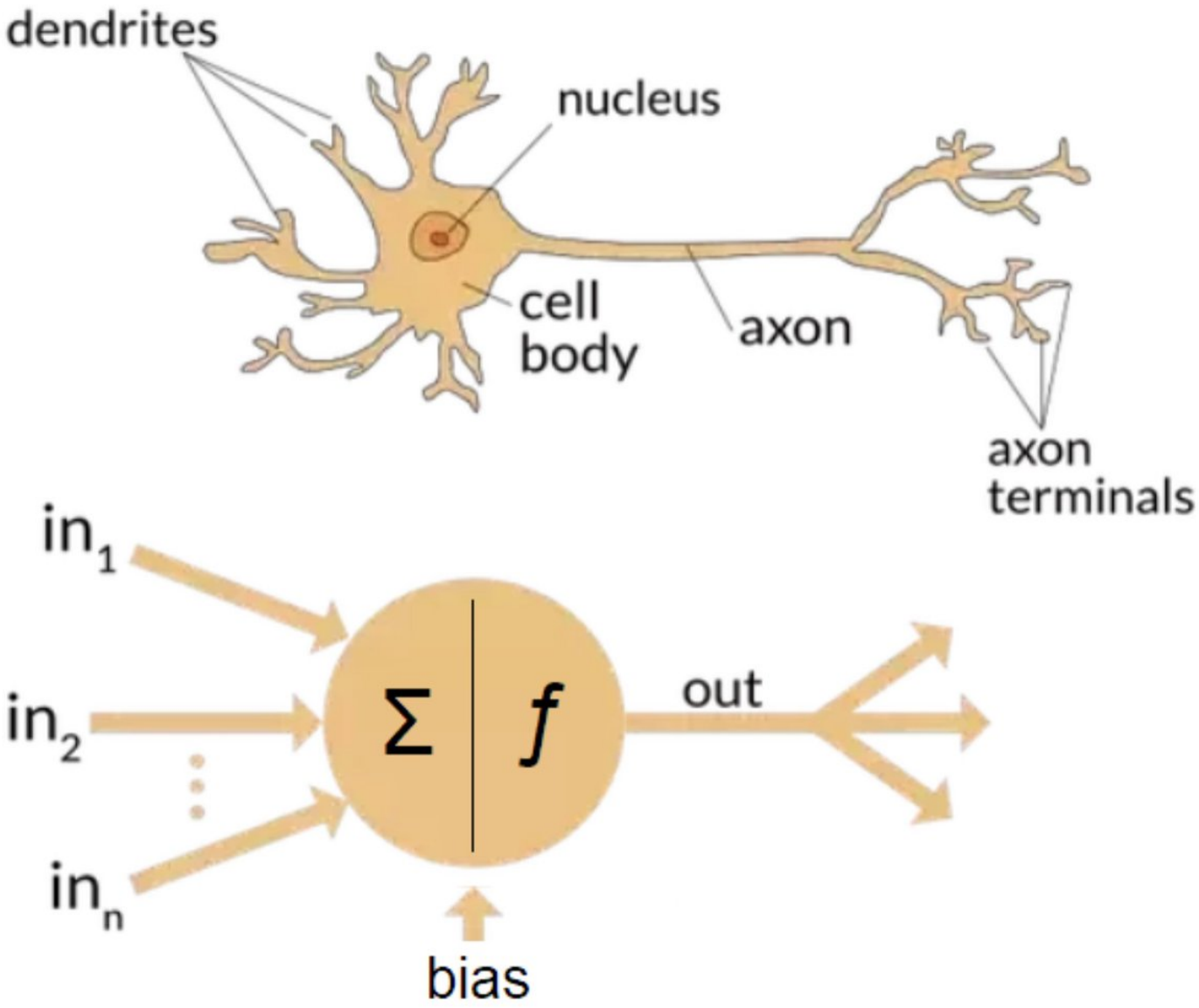


# Perceptrons

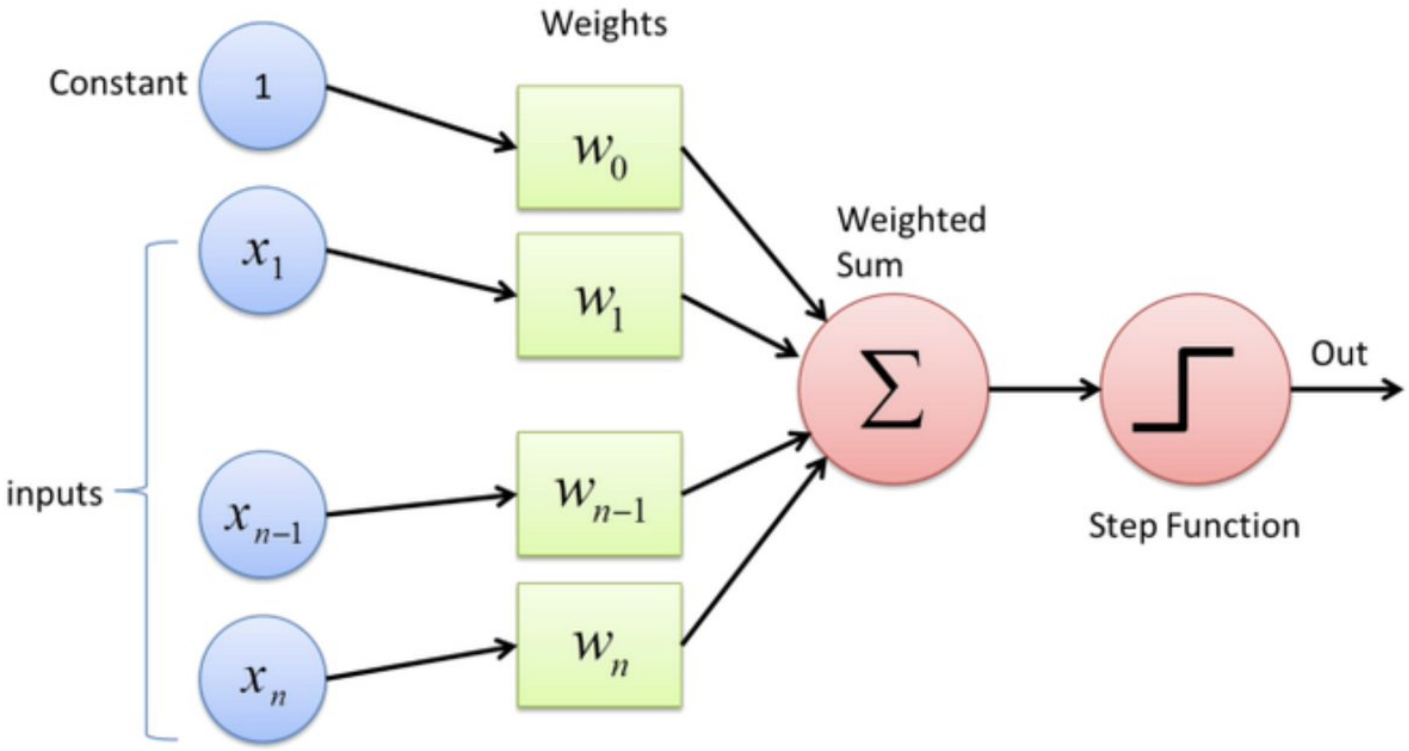


Marvin L. Minsky
Seymour A. Papert

Perceptron is literally a mathematical model of the human neuron. Of course, the human neuron is much more complex than a perceptron (How much more? Read on to find out).



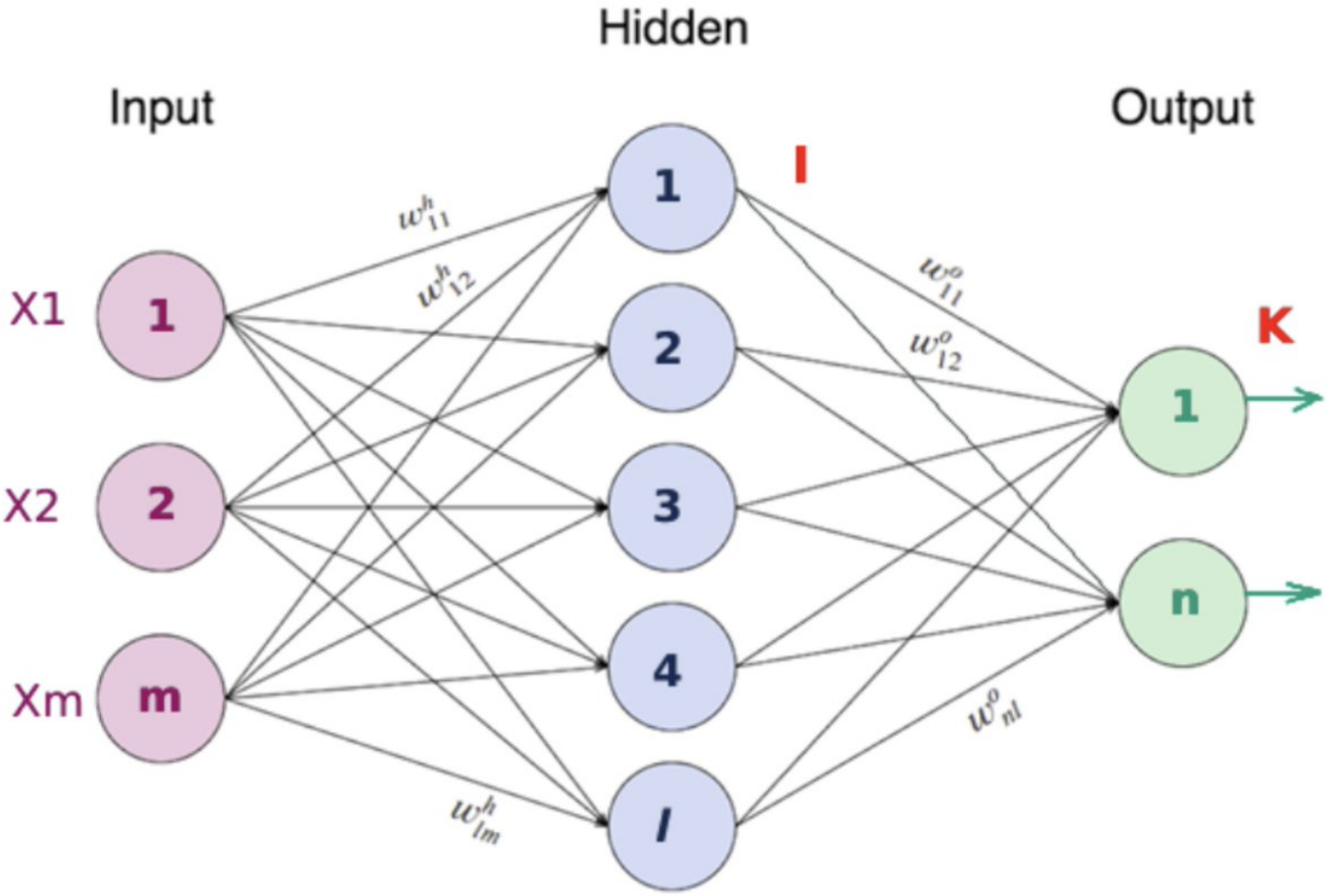

A perceptron literally just sums its inputs after weighing them, applies a function (typically a non-linear function in the modern era) on top of it, and voila, you have a simulated, or artificial neuron.




What you see in front of you is a mathematical model. How do you go from here to a working code? You need something called an algorithm. Which can help us decide the 'weights' of the perceptron. This is the perceptron learning algorithm. Good resource: https://t.co/Yn7pV45yip

3. From one to many

Now that we can simulate a neuron, what is the next step? Of course, to simulate an interconnected web of neurons! These are what are known as artificial neural networks.




These early multi-layer artificial neural networks are also known as feedforward neural networks because they, 'feed forward'. if you know some math, you might see vectors and matrices inherent in these networks.

These networks needed a new training algo, because they are a different, more powerful mathematical model than a perceptron due to the composition. How much more powerful? Glad you asked. :)

Enter, the universal approximation theorem.

It turns out, single hidden layer feed forward neural networks can approximate ANY* function to arbitrary degree of precision.

*terms & conditions applied.
https://t.co/WFRooX53Sx

This is as powerful as it gets. A single hidden layer FF neural network is all you need. You're done. Not so soon :)
While the theorem proves that such networks exist for each function and degree of precision, it says NOTHING about how to find them.

And that, remains the hard part. The learning algorithms for these things is essentially something a 11th class math student would understand. Its the good old differential calculus. This has a fancy name called back-propagation.

4. The backprop Algo.

Let me describe this algorithm in few easy steps. I will try to keep it as non-mathematical as possible since many readers might have a non-mathematical background. Some math is unavoidable thought.

1. First, you need to define a problem to apply FF neural networks on. The canonical example here is the MNIST example.https://t.co/jKn5cs6t0d

MNIST consists of a dataset of hand written digits. Model needs to identify which digit this is. You can imagine that optical character recognition programs are built on top of models like these.



2. With the problem defined, you need to define an encoding for the output. In this case, we represent the output as a 1-hot vector. If the digit was 3, the vector is: [0,0,0,1,0,0,0,0,0,0]
https://t.co/AfxlveCec6

3. Now you can imagine, that the model outputs a probability distribution over the 10 possible outputs. Eg: Model might say: with 90% probability it is the digit 1. With 10% probability is is 7. This

4. Now what we need is a way to judge whether model output was correct or not. We have 2 probability distributions in front of us: First, the model output. Second, the 'ground truth' distribution: the one-hot vector.

Enter the 'loss function'. This is also called the objective function. Because this function defines how the model behaves. Since we have 2 probability distribitions, maths has many functions to measure the 'distance' between these distributions.

Pick any one such function, then you have your loss function. The 'loss' is the distance between true probability distribution & model predicted one. Objective is to minimize it.

5. As long as all our functions were differentiable, differential calculus can tell us how to compute the derivative of any 'weight' in the model with respect to the final loss function.

This is the back-prop step. Because you converted a final loss function into derivatives of the weights.

5. Then. we make a small step in the direction of the derivative, adjusting the weight of the network ever so slightly in the direction of the derivative. What this does is 'tunes' the weight and model output moves towards true distribution ever so slightly.

This is known as a training step in ML parlance. Do this enough number of times, & voila! You have a trained neural network which recognizes digits.

There are readymade programs available on Github which do this. If you have a laptop, you can tinker with these & play with them:
https://t.co/yTZ9W55ncy

5. From ANN to modern Deep learning.

We have known about ANNs for many decades. Why then did we have to wait until 2010s before seeing wide ranging applications?

Well, it turns out that for these systems to be useful, 3 ingredients were needed:
1. Lot of data. If we're learning from data (this is the core of machine learning), we need lot of data! To capture all the variations in data. https://t.co/9x4qnTAWB4

2. Lot of compute. Training a neural network is so intensive (matrix multiplications are heavy duty lifting after all) that companies how build custom computer hardware just to train neural networks. https://t.co/GzAR6udF3P

3. We needed the undying human spirit which refuses to give up even in the face of near certain doom. The pioneers of deep learning spent all their lives being ridiculed for focussing on something which never really seemed to work. They did not give up.
https://t.co/NZDi4fYpU6

The first 'aha' moment of modern Deep learning happened in 2010 in the ImageNet competition. A competition which was held every few years to advance the state of image recognition.
https://t.co/tudLnUAhHh

THis moment was not a triumph for science alone. It was a triumph for engineering. The science was known since 1960s perfected over 50 years. But it took a giant engineering effort to make it work in reality.

Science dreams how our world should be. Engineers work hard to realize that dream. Society has a need for both kinds of contributions.

Near 2010 they reduced the State of the Art (SOTA) error rates from 25% to 15% which was one small step for the authors, but one GIANT leap for humankind & deep learning.

## 6 Results

Our results on ILSVRC-2010 are summarized in Table 1. Our network achieves top-1 and top-5 test set error rates of **37.5%** and **17.0%**[5]. The best performance achieved during the ILSVRC-2010 competition was 47.1% and 28.2% with an approach that averages the predictions produced from six sparse-coding models trained on different features [2], and since then the best published results are 45.7% and 25.7% with an approach that averages the predictions of two classifiers trained on Fisher Vectors (FVs) computed from two types of densely-sampled features [24].

We also entered our model in the ILSVRC-2012 competition and report our results in Table 2. Since the ILSVRC-2012 test set labels are not publicly available, we cannot report test error rates for all the models that we tried. In the remainder of this paragraph, we use validation and test error rates interchangeably because in our experience they do not differ by more than 0.1% (see Table 2). The CNN described in this paper achieves a top-5 error rate of 18.2%. Averaging the predictions

| Model | Top-1 | Top-5 |
|---|---|---|
| *Sparse coding [2]* | *47.1%* | *28.2%* |
| *SIFT + FVs [24]* | *45.7%* | *25.7%* |
| CNN | **37.5%** | **17.0%** |

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

6. Wait but what about ML?
I'm been untruthful when i talked about 2 schools of thought in AI. There are several. What predates modern ML and developed alongside the other schools are traditional ML methods. Everybody loves talking about the winners.

That is why this thread is about connectionists. This is what really really works very well. Traditional ML techniques work, but DL trumps them in almost all ways. Still for someone curious enough, here is the bible of ML:
https://t.co/ylraMZQips

We did parts of this in college.

7. What has Deep learning/Machine learning done for me?

It wouldn't be an overstatement to say, everything.
Google, Facebook, Instagram, YouTube, Google Photos, Twitter everything is powered by Deep Learning. Deep Learning is eating software, quite literally.

1. Recently, OpenAI launched a thing Called Codex.
https://t.co/SXBoEXQuma

You can provide this program english description of what a program should do, and it codes that program. That day might not be far when standard programming is 'deprecated' by Deep Learning models.

2. One of the biggest problems in Computational biology is to figure out the structure of proteins. This is called the Protein folding problem. This will enable great advances in the field of drug discovery.
https://t.co/7yz7ydUL0a

3. Another space Deep learning has really excelled is Games. Chess, go. These are really complex games. Deep learning techniques are now so advanced, nothing even comes a close second. Not traditional programming methods, not humans, by a mile.
https://t.co/AprUFaYKJE

4. Tesla Driving AI. If you havent seen Tesla Full Self Driving beta demos, you HAVE to watch them. Self driving cars are far far nearer than anyone imagines. Definitely in the west. I wouldnt be surprised if that is the case in India as well.
https://t.co/SQOQAPUvBL

8. Where can I learn ML/DL?
So many great places. But in my opinion, best one is courses. There are many of them, all of them great. My Masters guide Ravindran sir has free NPTEL courses.
https://t.co/aIjSK13v0F

There are also live versions of the source where one can do assignments, get graded, learn more closely with other people. Learning has come to you, knocking your doors. Now its up to you whether you get it come inside or not :D

There is of course the Stanford coursera ML course by Andrew Ng :
https://t.co/LwE94cwzoe

for those that prefer the gentler, simpler, easier version.

For those that want to listen to more philosophical discussions & history of AI, I cannot recommend @lexfridman podcast enough. It has technical details too, but is more advanced.

9. Can AI revolt against us?

This is something which has worried many people. The short answer is, not in the foreseeable future. Computers are still quite dumb. They do what they're told. Remember objective functions? That is how you tell a DL model what to do.

The concept of revolting against humans does not make any sense unless someone has explicitly programmed them for that. In which case, it isnt a revolt :). We discovered nuclear power & for a while tried to blow ourselves up.

AI needs clear government regulations which prevent that. At the same time, the potential of AI is literally unlimited. It is going to help us discover new materials, new drugs, new ways of tackling climate change, new microorganisms which eat plastic.

Software is eating the world. And Deep learning is eating software.
But thanks to awesome tools and platforms, everyone has the ability to make every day, an AI day. :)




If you like the thread, please retweet the very 1st tweet so that this interesting information reaches as many people as possible.