VISUAL RECOGNITION

ANUSH SANKARAN
IBM RESEARCH

RECAP: VISUAL RECOGNITION – WHAT IS THE PROBLEM?

- You are a genius …
- Only if a machine or an algorithm could do it …

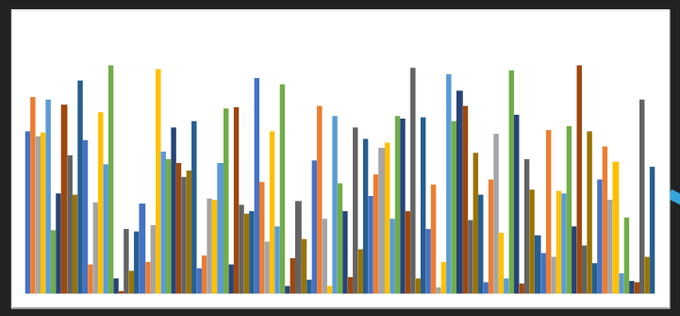
- Make a computer understand the image …
- Understand the background & environment …
- Classify various objects in the image …

RECAP: HOW DO WE MAKE A MACHINE CLASSIFY IMAGES?

Image Intensities

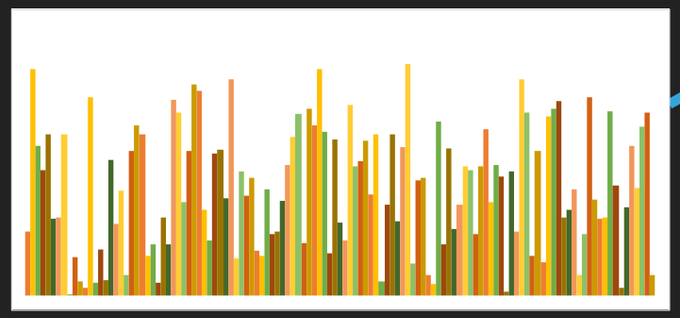
Extract Features

195	2	4	213	132	196	81	202	124	186	199
3	164	113	209	36	88	70	221	189	147	247
191	8	88	118	181	182	30	192	50	27	129
114	191	71	249	230	129	248	121	167	61	150
9	133	217	73	251	108	157	232	177	109	102
252	171	125	200	26	38	249	240	93	242	217
101	111	22	118	203	140	27	229	215	106	128
1	192	148	206	199	97	106	64	12	76	170
232	239	254	131	9	180	35	217	22	46	163
111	33	187	135	254	46	57	5	173	48	15
6	59	235	182	153	22	235	106	65	189	15



Learn how to match similarity

180	170	100	28	240	148	109	72	13	45	127
219	32	35	42	124	209	133	121	127	181	164
174	101	76	105	58	83	162	7	178	41	104
179	202	250	104	180	8	168	35	68	114	147
215	144	158	145	75	197	83	243	17	111	23
71	254	149	205	233	122	197	191	248	186	85
111	17	185	33	8	92	194	225	198	75	14
210	3	145	207	39	19	91	209	11	254	12
154	72	129	99	103	185	251	82	150	54	216
110	25	136	89	60	49	17	156	116	180	41
237	69	192	92	15	172	196	110	65	34	141



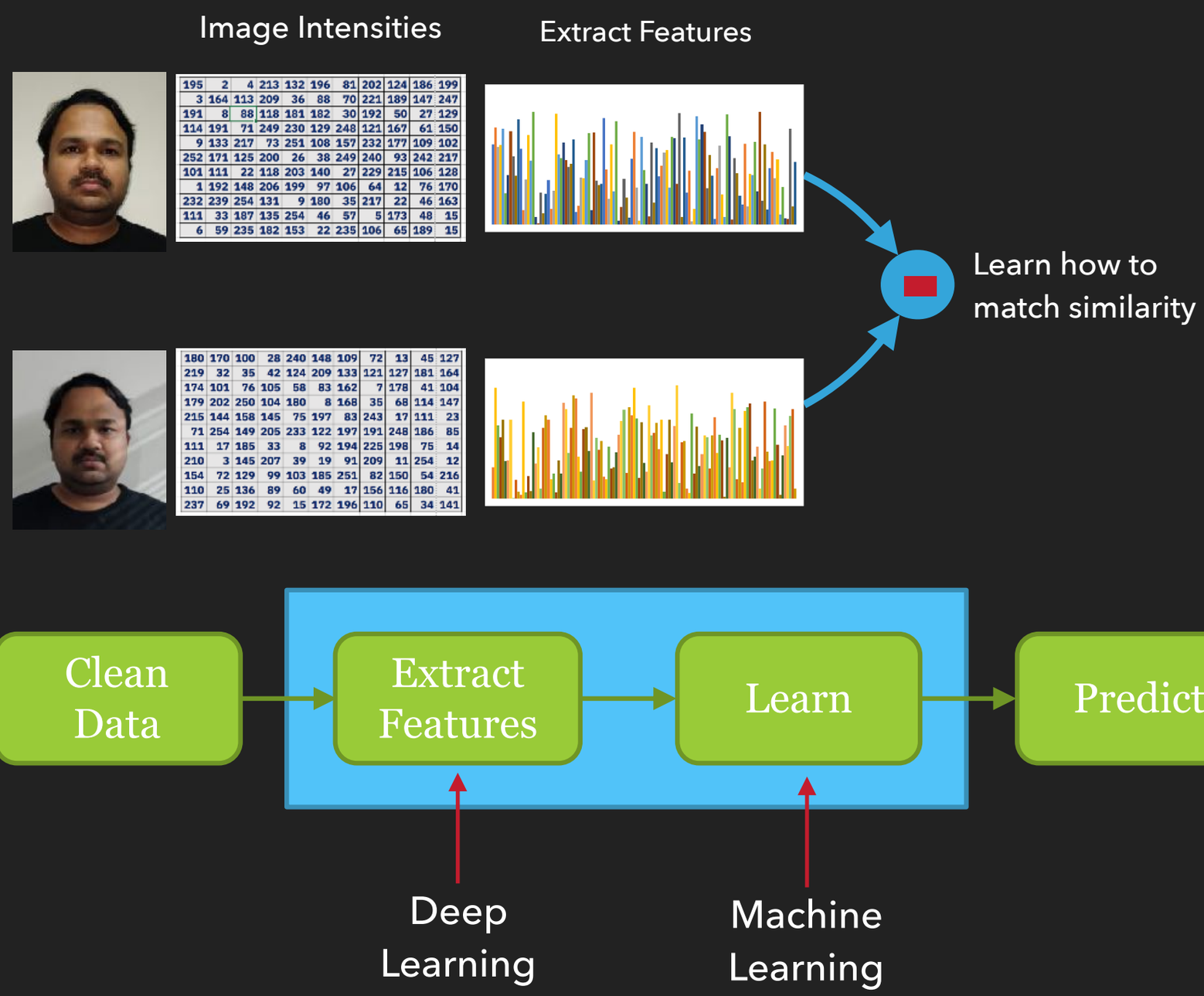
RECAP: REMAINING STRUCTURE OF THIS COURSE

- Week 1: Introduction to Neural Networks, Loss Functions and Optimization , Intro to Image Classification
- Week 2: Convolutional Neural Networks, Training Neural Networks
- Week 3: Different CNN Architectures
- Week 4: Advanced training strategies and interesting applications
- Week 5: Generative Models , Visualizing, and Understanding
- Week 6: Face recognition, Face Detection, and generating Face images

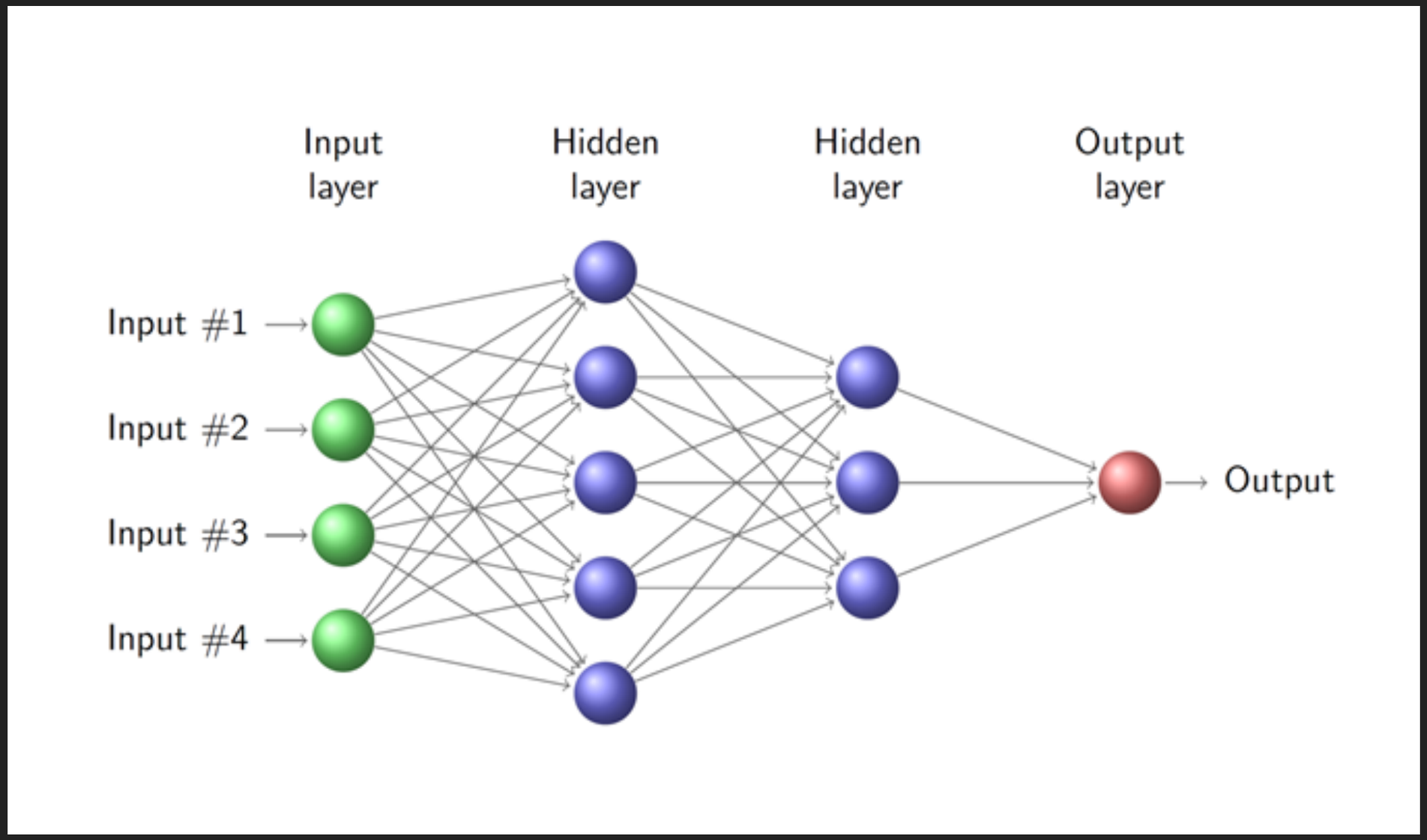
And we will have lots of fun…

Don’t forget that I am a wannabe StandUp Comedian

RECAP: DEEP LEARNING VS. MACHINE LEARNING

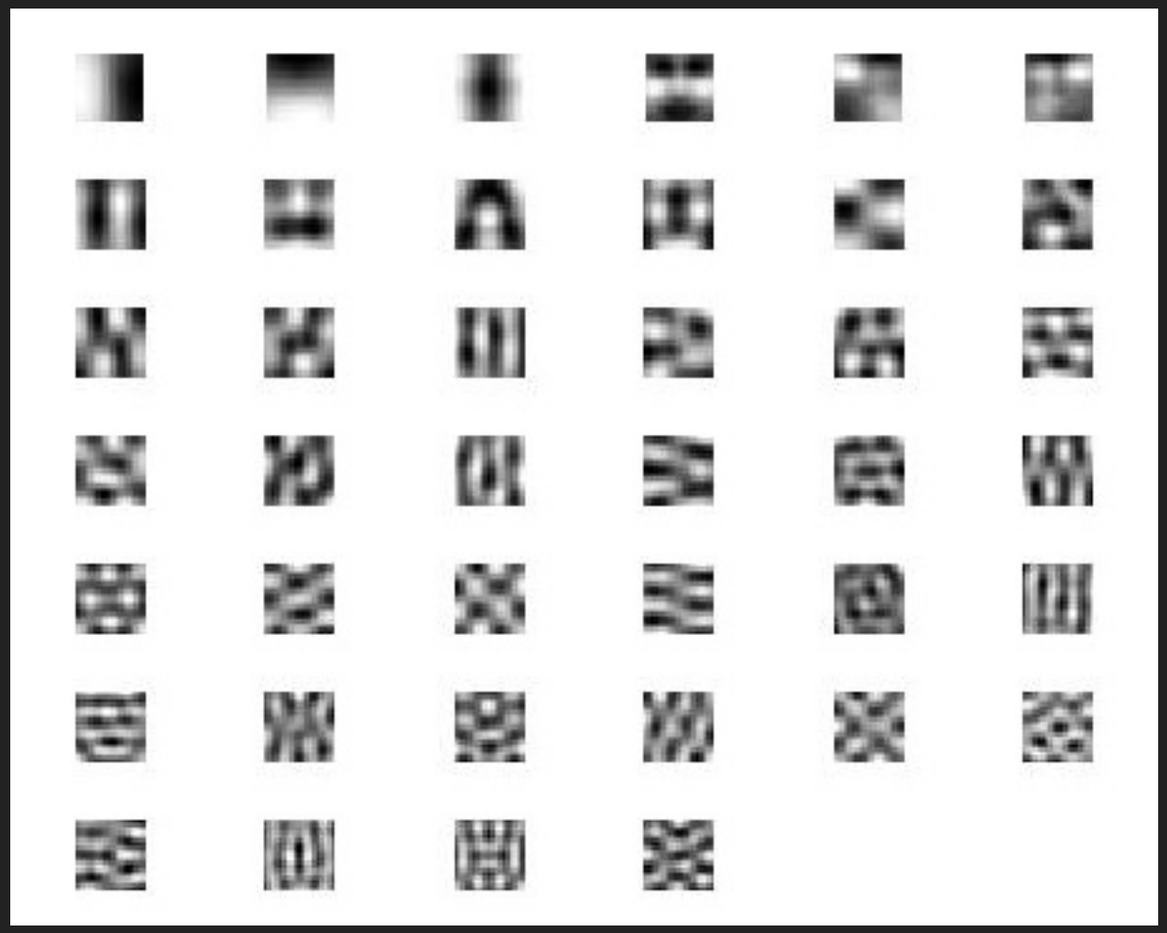


RECAP: WHAT IS THE INPUT HERE?



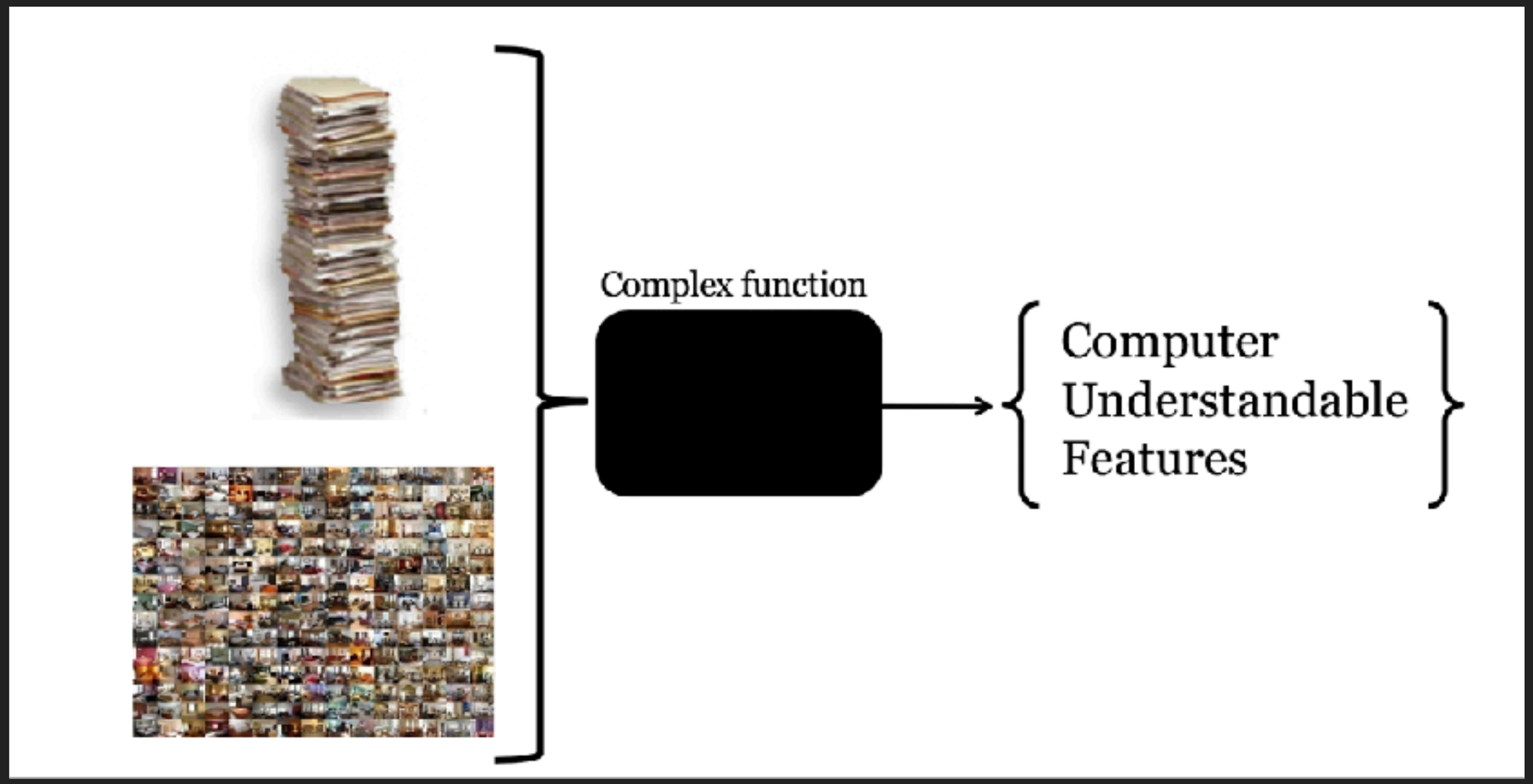
Assumption is that the input here is sufficient.. What if it is not?

RECAP: FEATURES: AND A FAMILY OF FILTERS . . .



- Each of these filters is a mathematical function that has to be manually defined
- Which filter works is a hit and trial method

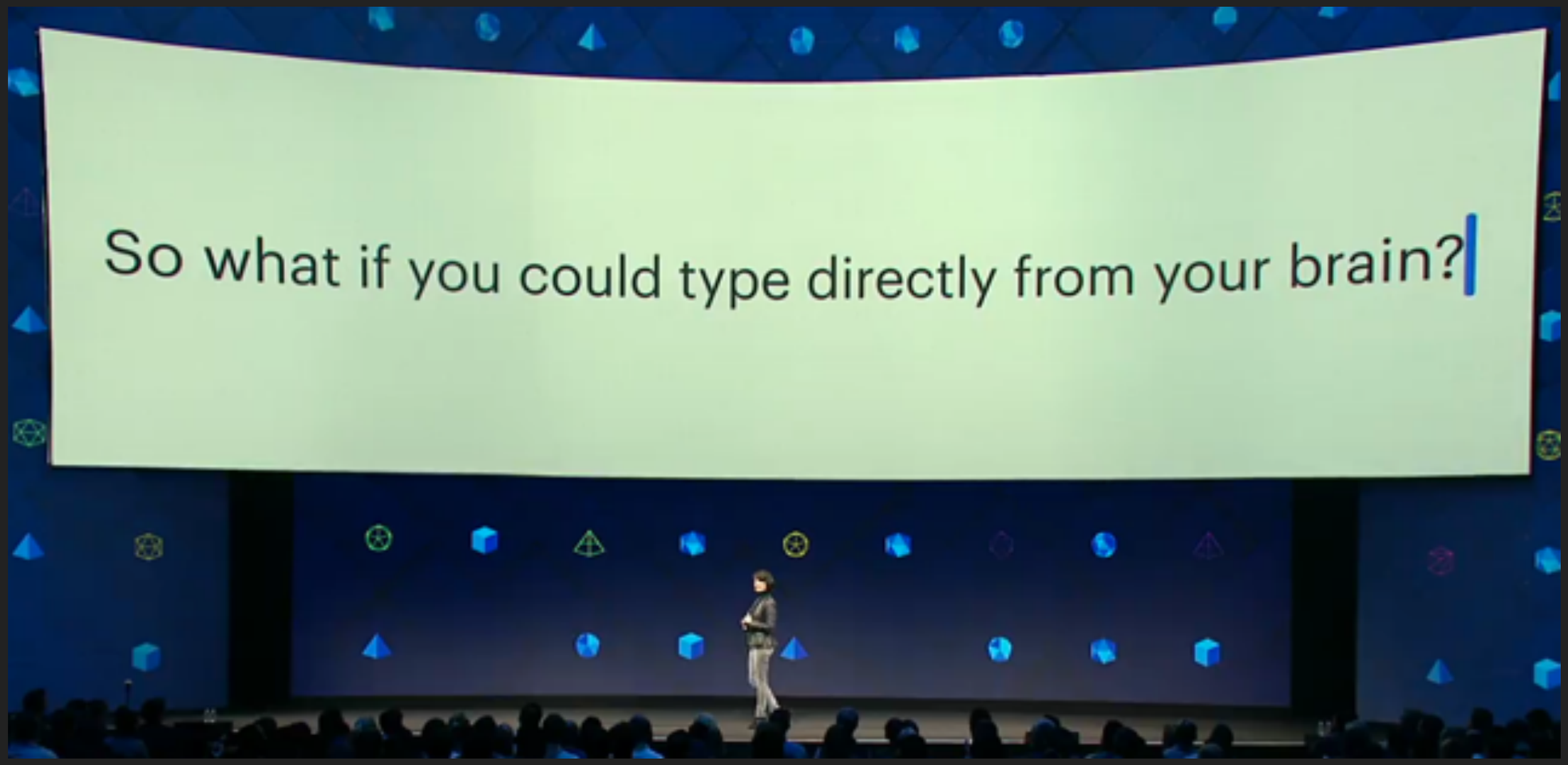
RECAP: UNSUPERVISED FEATURE LEARNING



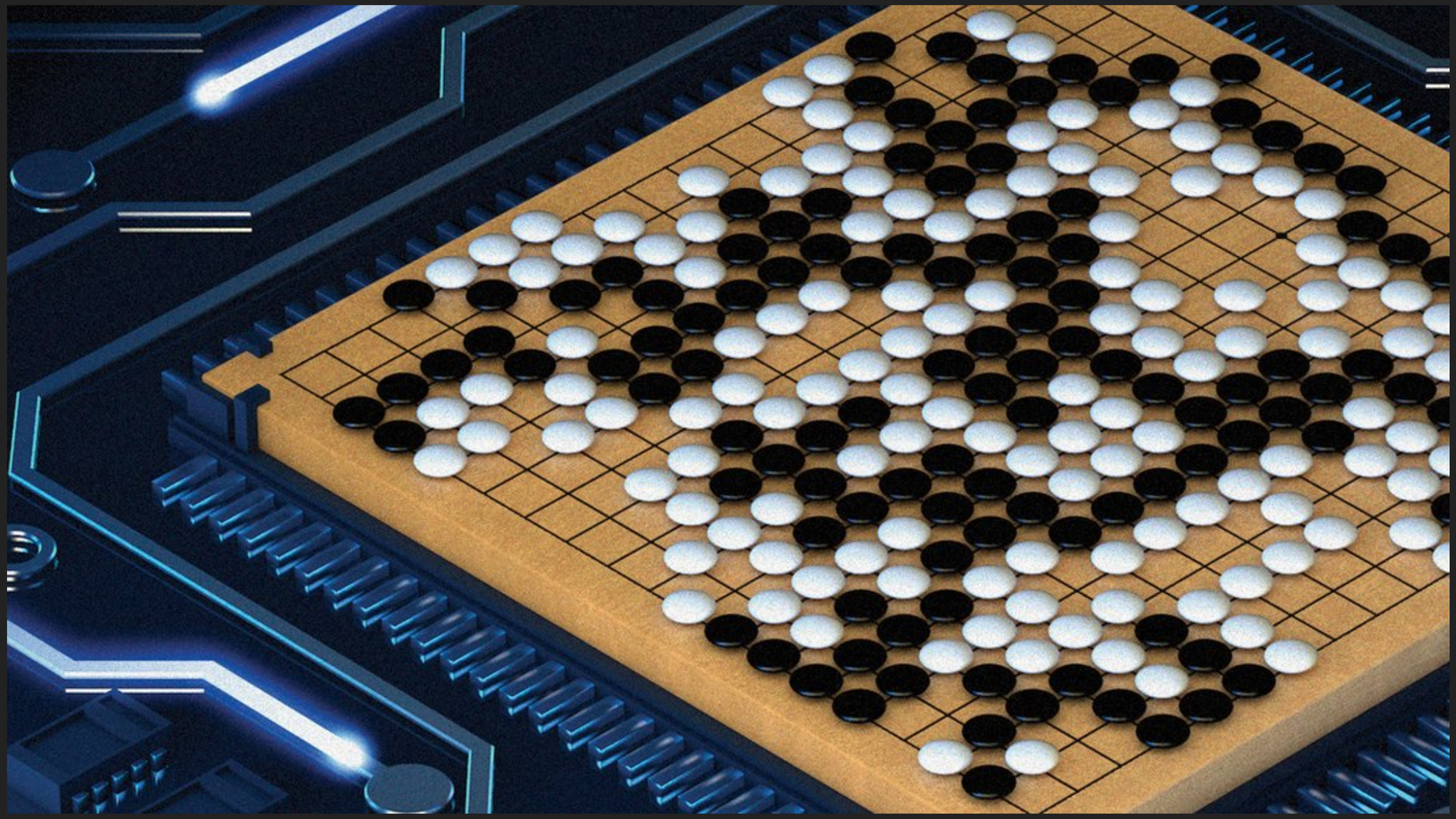
Deep Learning is one technique to perform Unsupervised Feature Learning

TOP 10 BREAKTHROUGHS

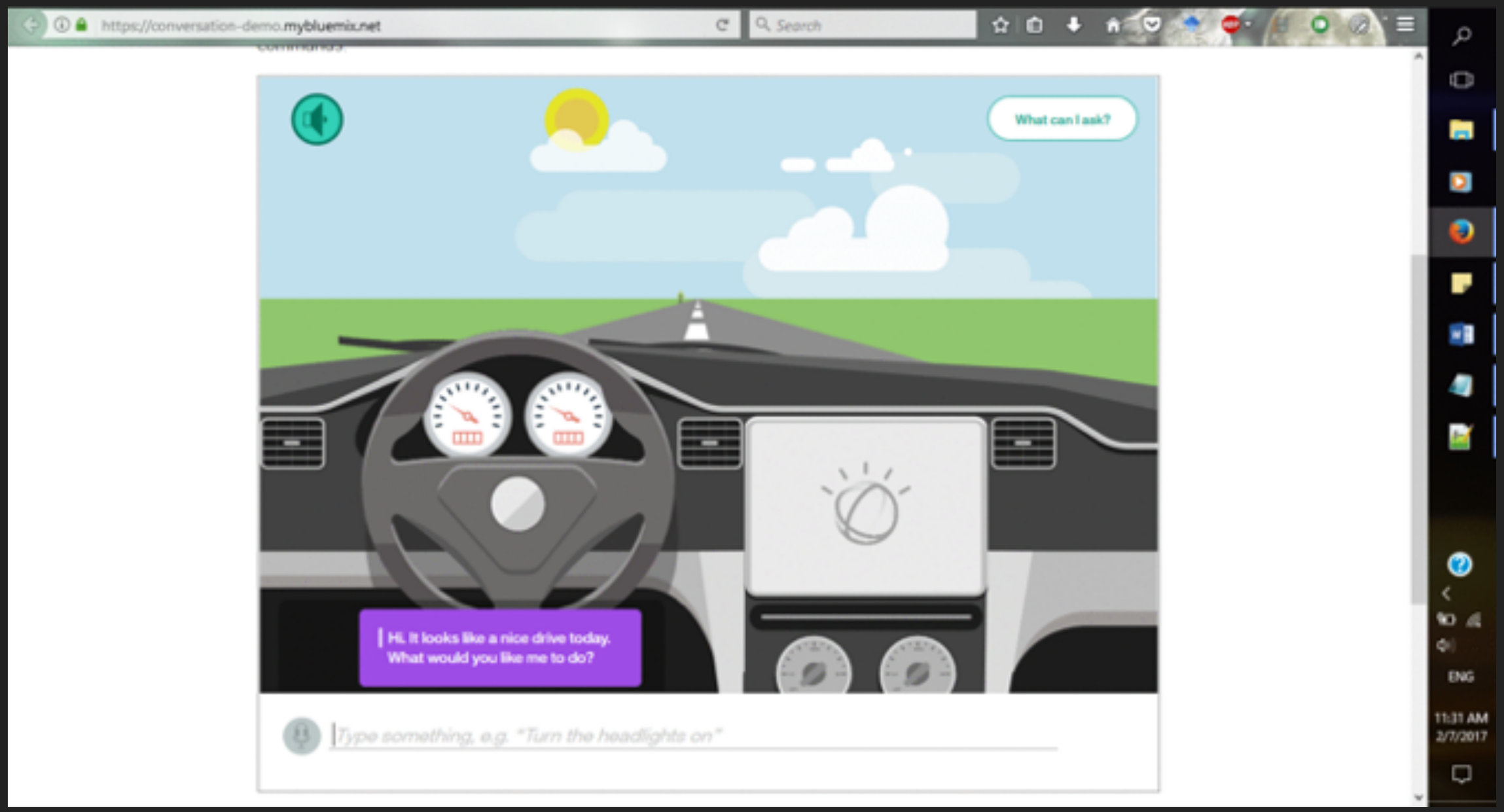
1. NEURAL MAPPING OF THOUGHTS



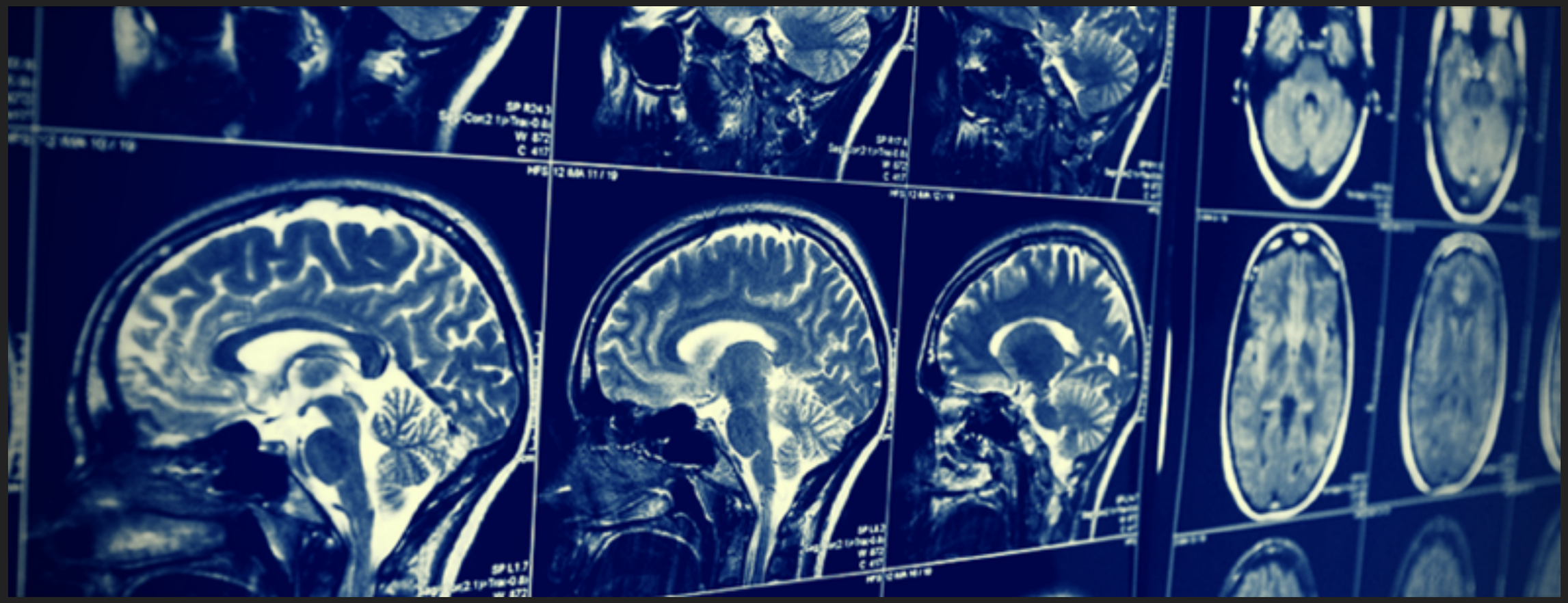
2. AI DEFEATS WORLD'S TOP GO PLAYERS



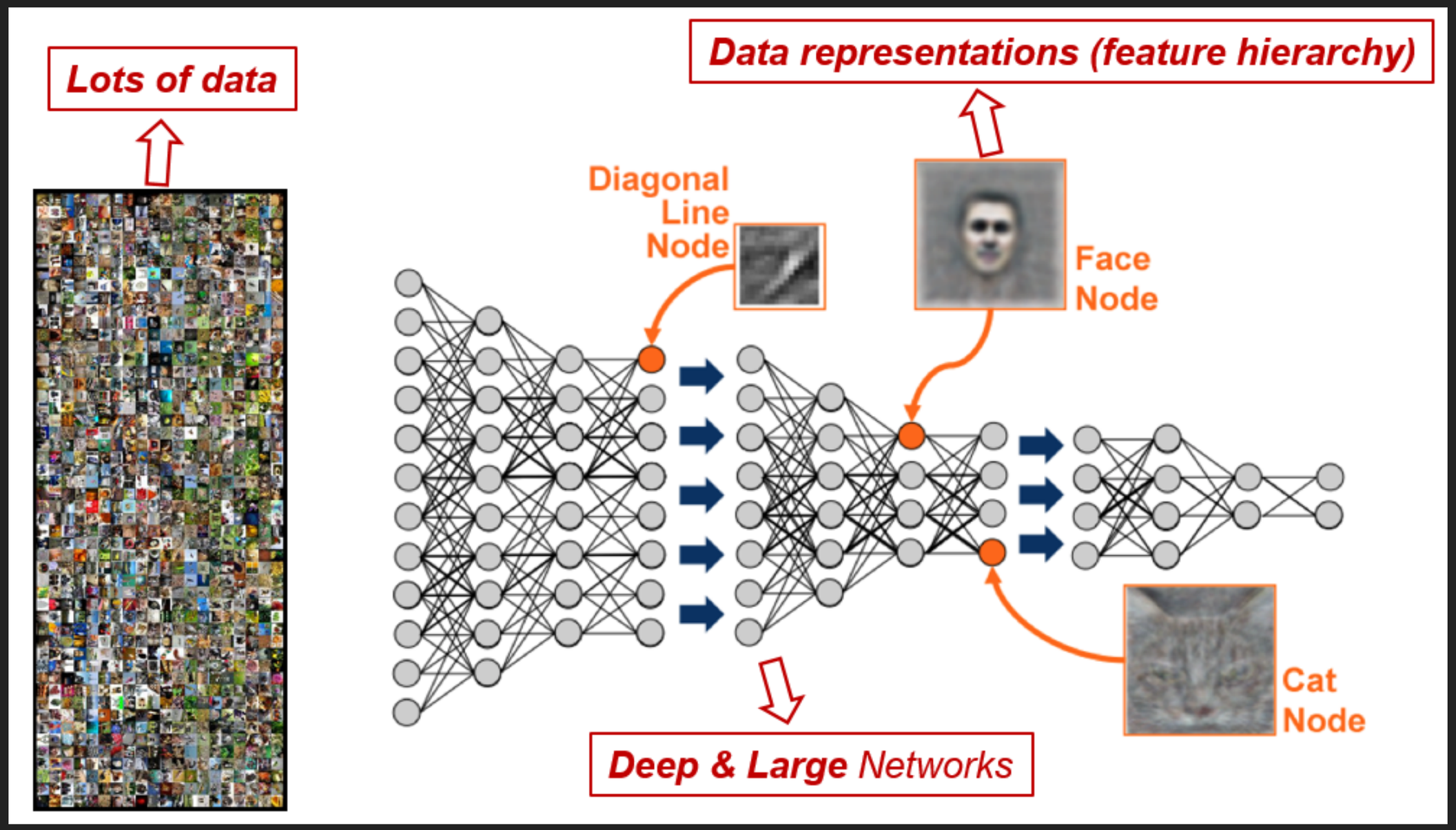
3. HUMAN LIKE CHATBOTS



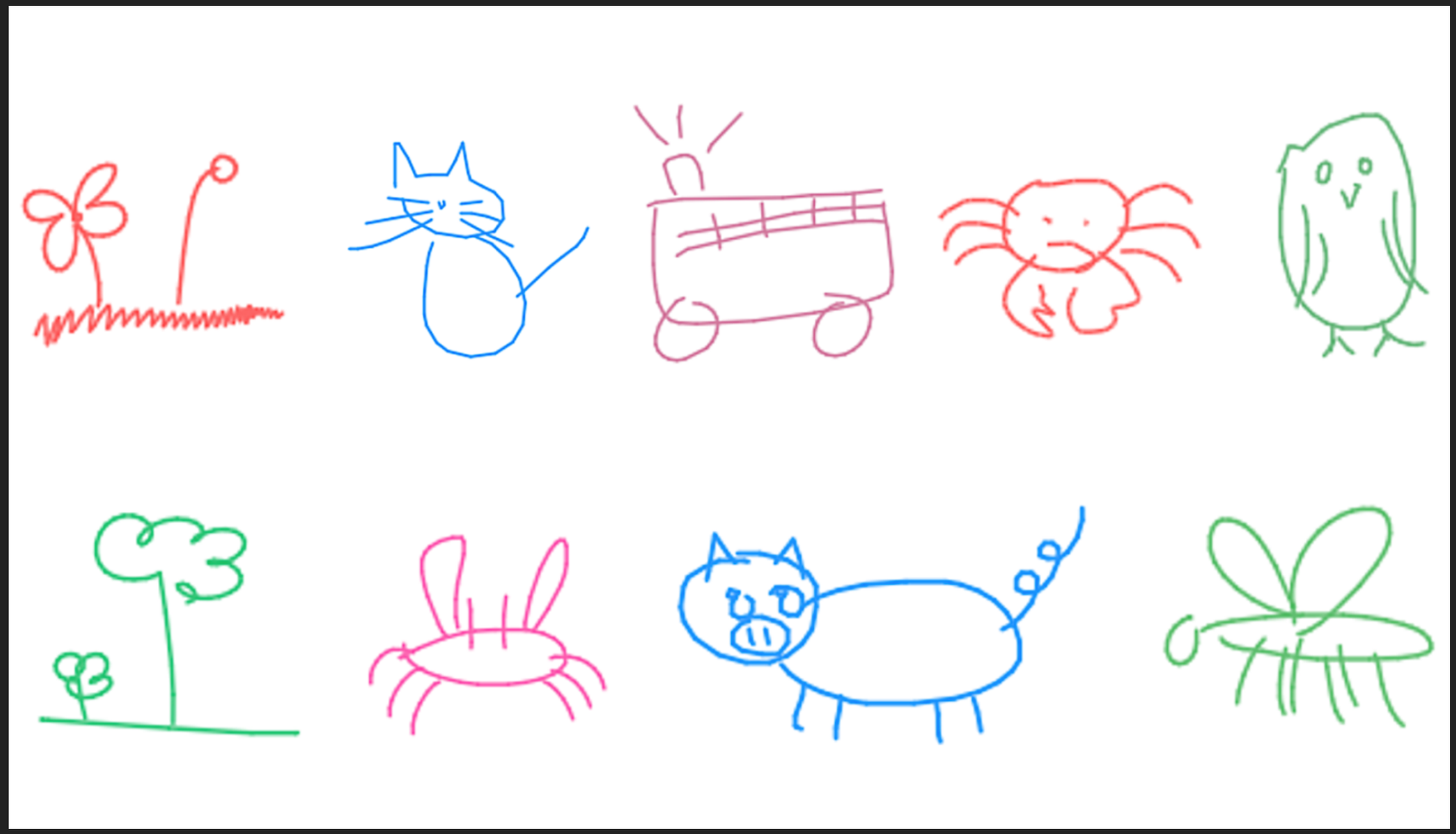
4. DETECTING BRAIN CANCER



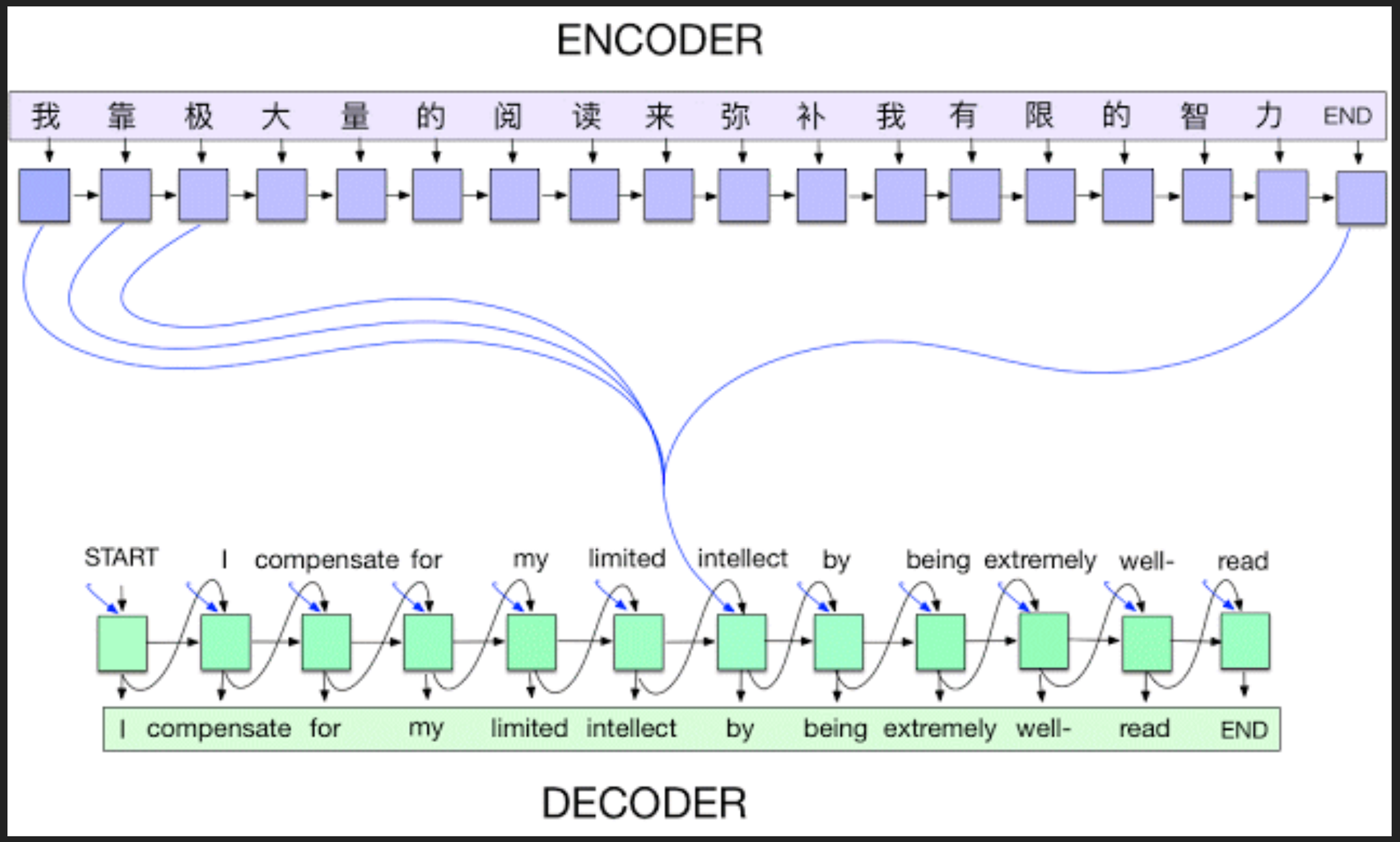
5. HUMAN FACE RECOGNITION



6. MACHINES LEARNING TO DRAW

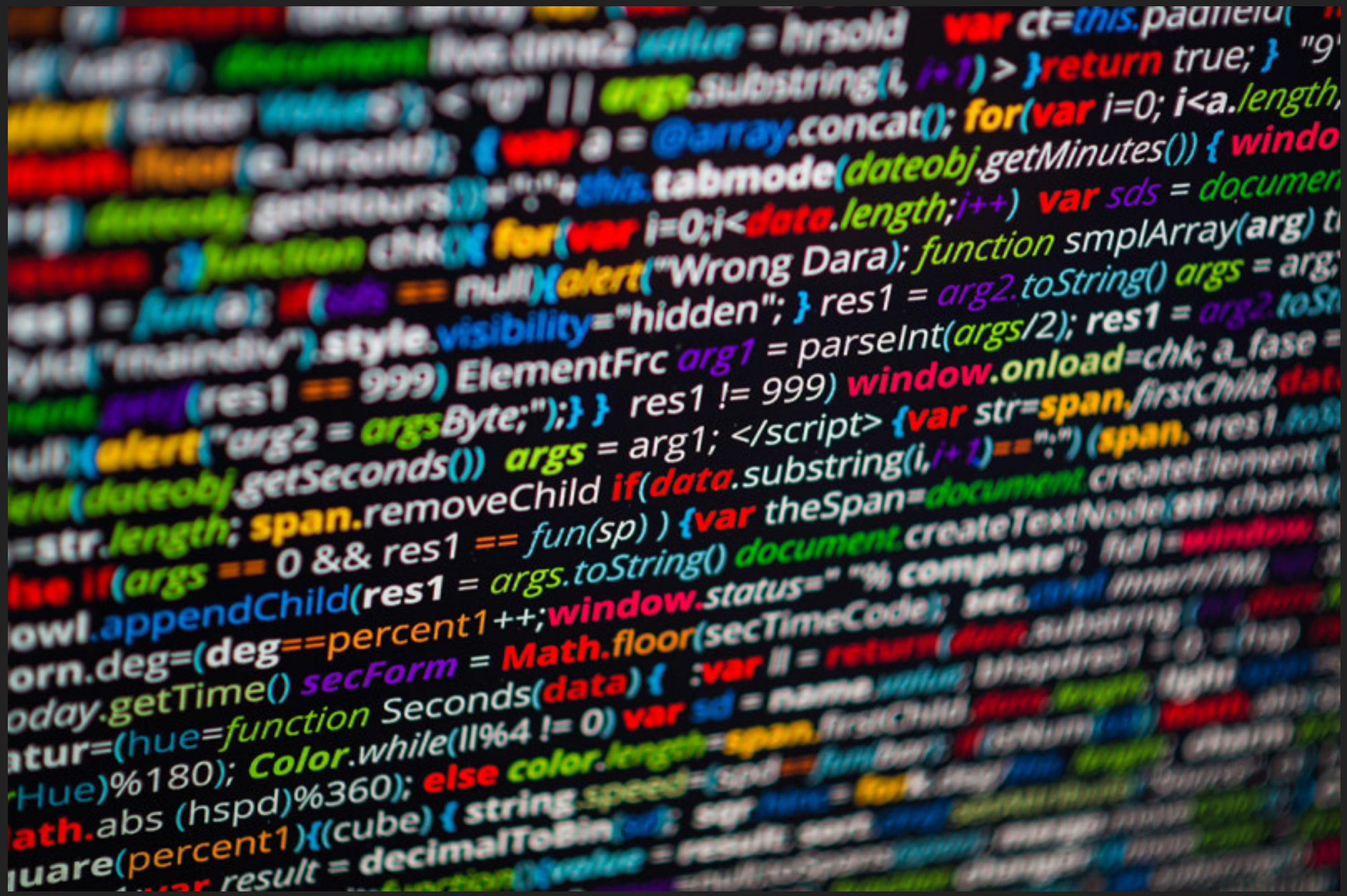


7. REAL-TIME LANGUAGE TRANSLATION



8. SELF DRIVING CARS

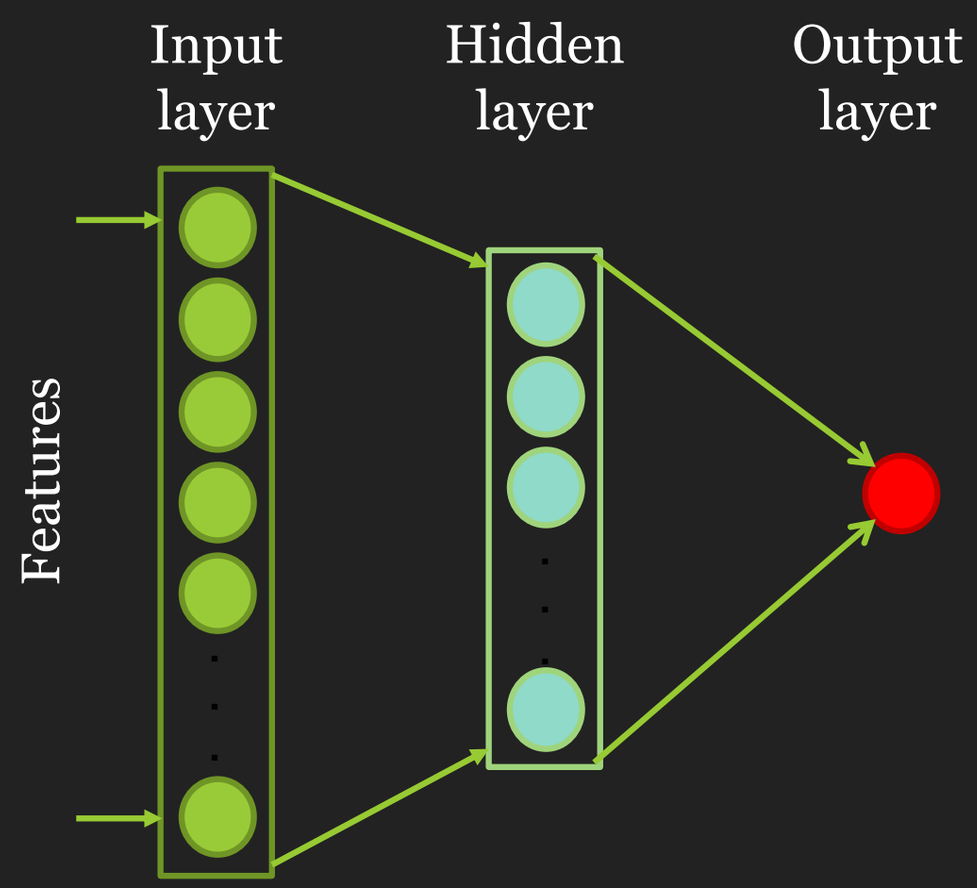
9. LEARNING TO WRITE PROGRAMS

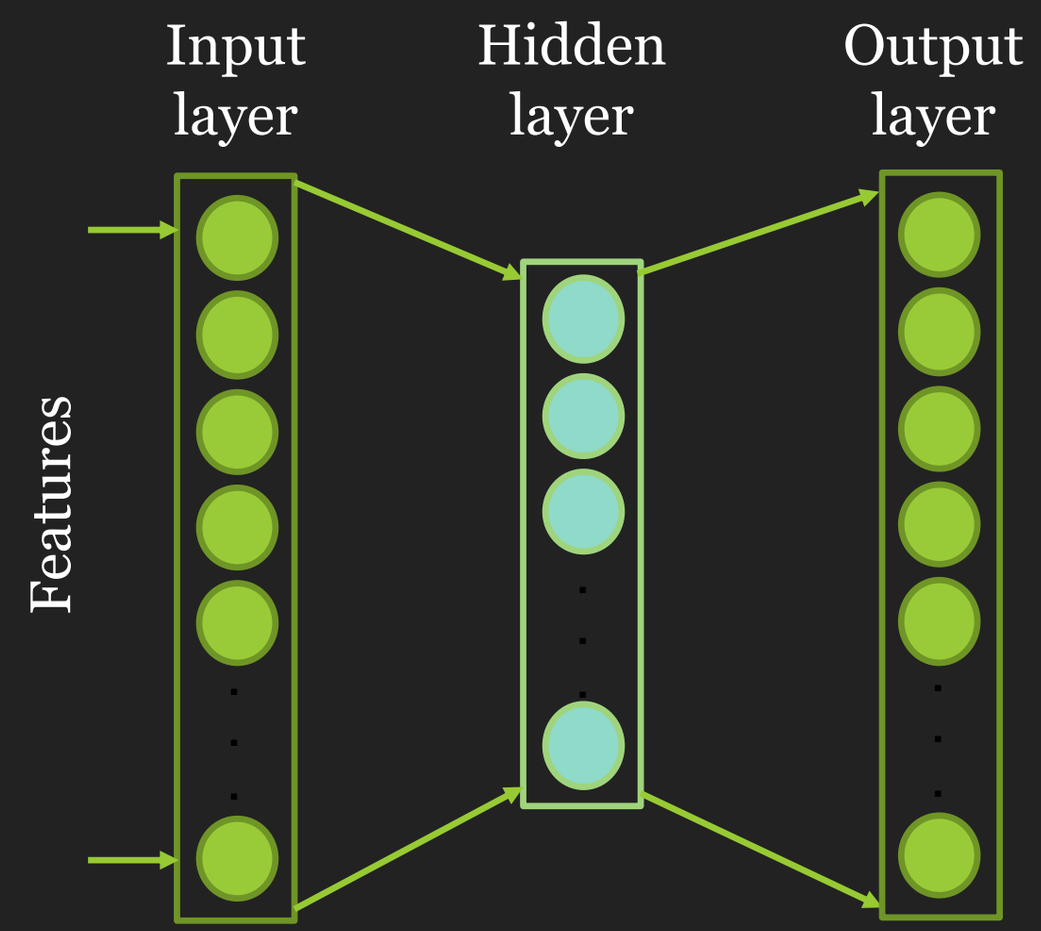


10. GENERATE ART

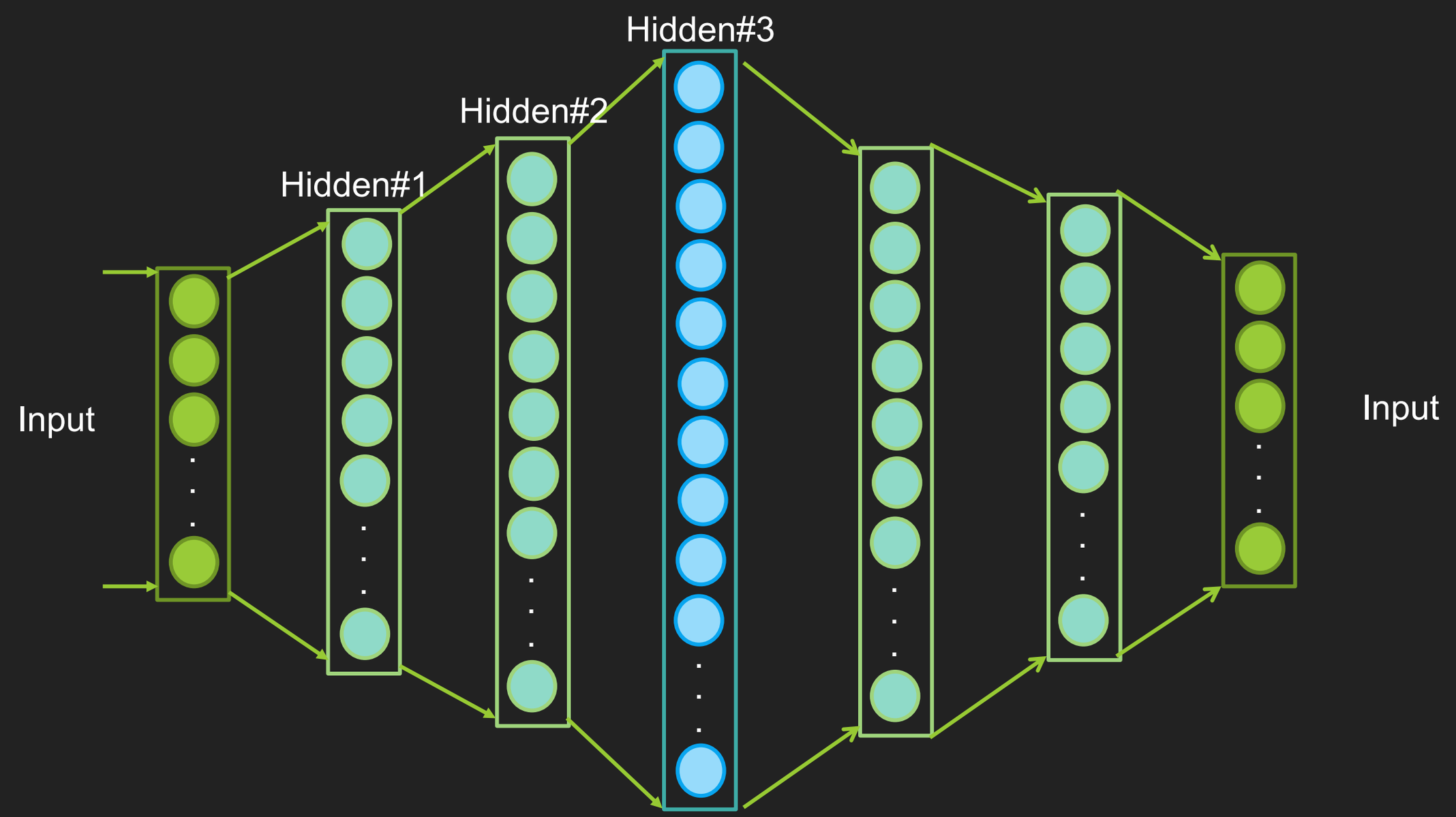
ALL POSSIBLE BECAUSE OF DEEP LEARNING!

WHAT'S THE TRICK?





STACKED AUTO ENCODER



SAMPLE OUTPUT: MNIST

After 1 epoch

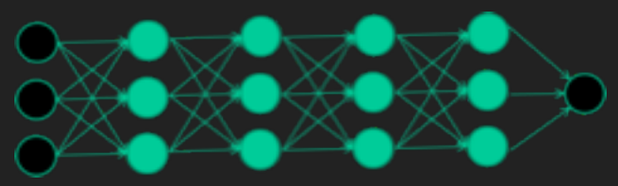
After 9 epochs

Original

SUMMARY: WHAT IS DEEP LEARNING

1. Deep Learning is one method for Unsupervised Feature learning
2. Deep Learning is a neural network with **several layers of layers** between the input and output

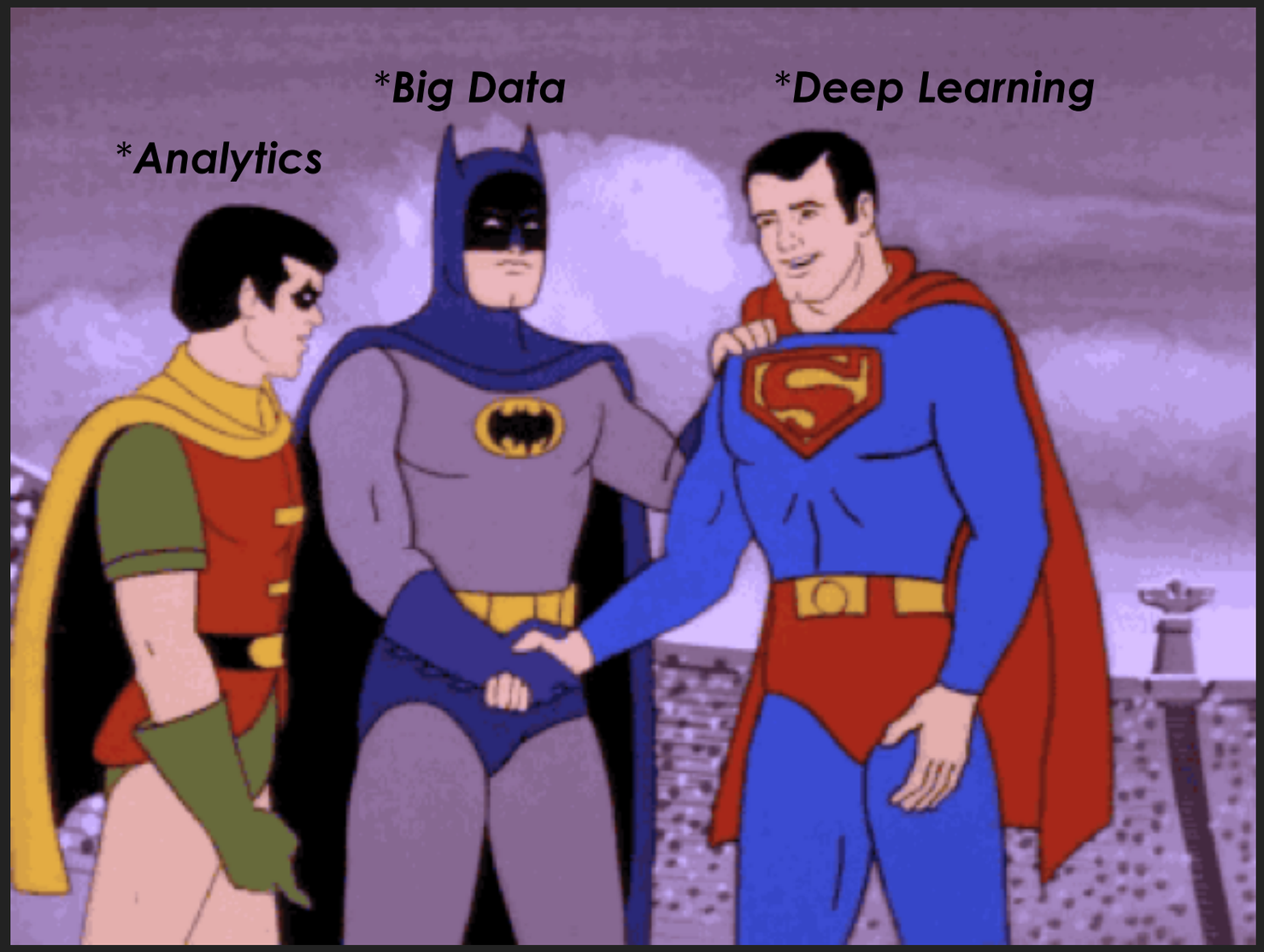
Neural Network



Deep Neural Network

3. Raw data as input: image pixels, one-hot encoding of words etc.
4. Several layers of transformation, leading to several feature representation.

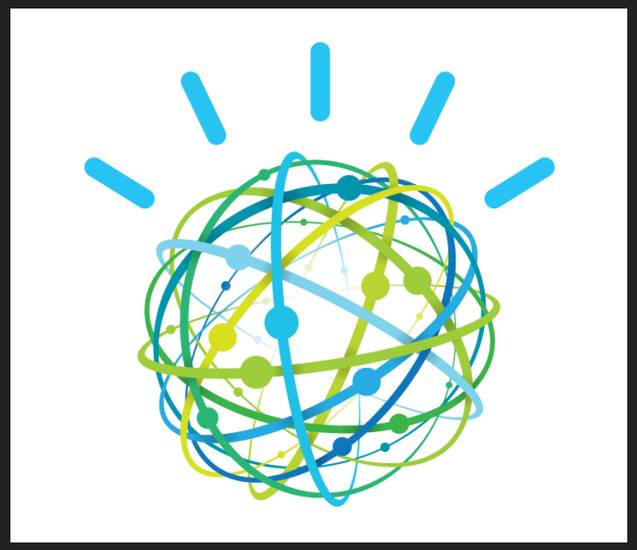
BIG DATA & DEEP LEARNING



HOW IS DEEP LEARNING POSSIBLE?

Growth of Big Data

Improved Algorithms



Fast GPU Processing

Thanks to gamers !

VARIANTS OF DEEP LEARNING

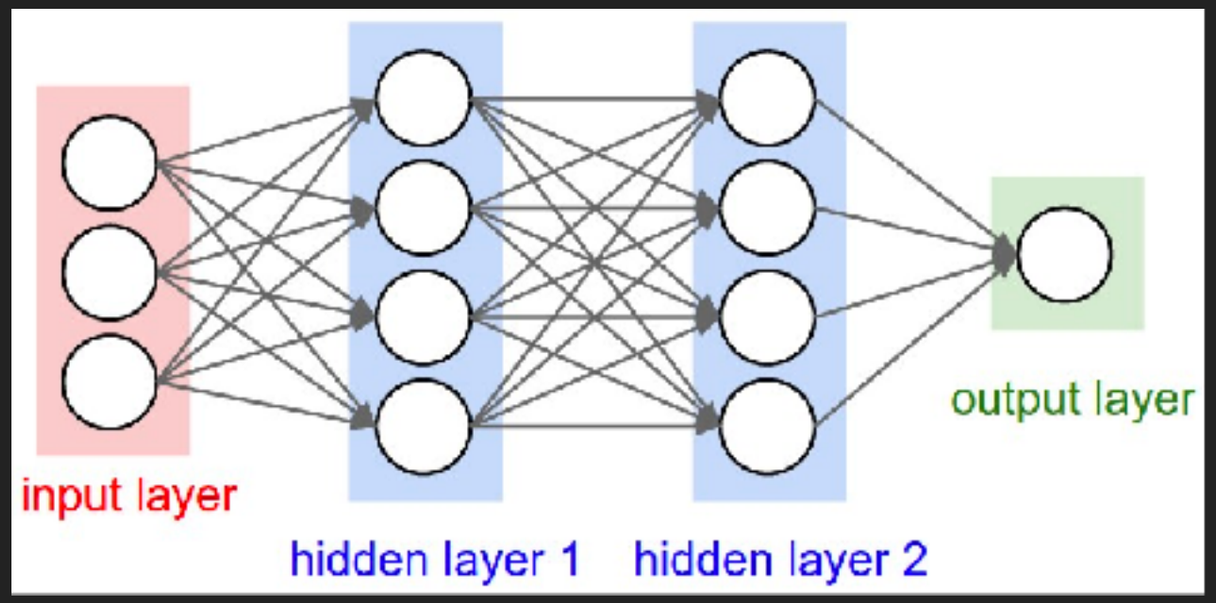
- Input data is a vector Stacked
  - Autoencoder
  - Restricted Boltzmann Machines
  - Deep Belief Networks

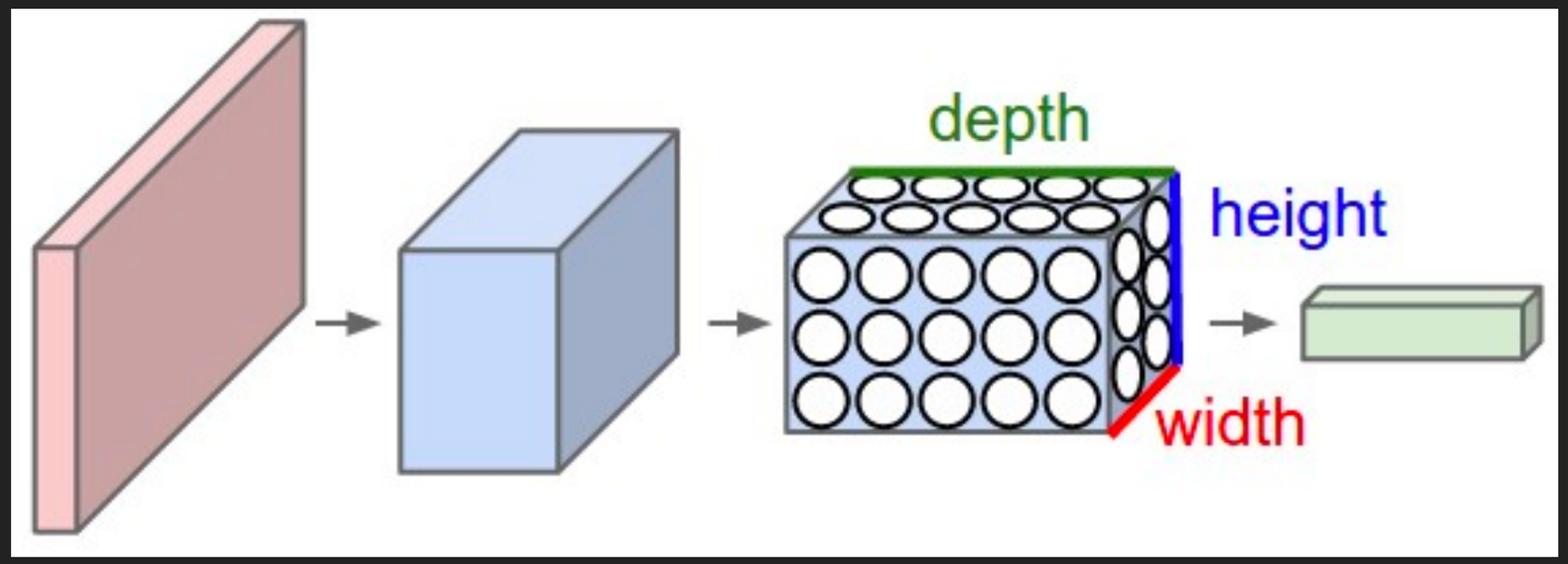
- Input data is multi-dimensional
  - Convolutional Neural Network
  - Example: Image analysis, Object recognition

- Input data is time-series
  - Recurrent Neural Network
  - Example: Language Modelling, Speech recognition

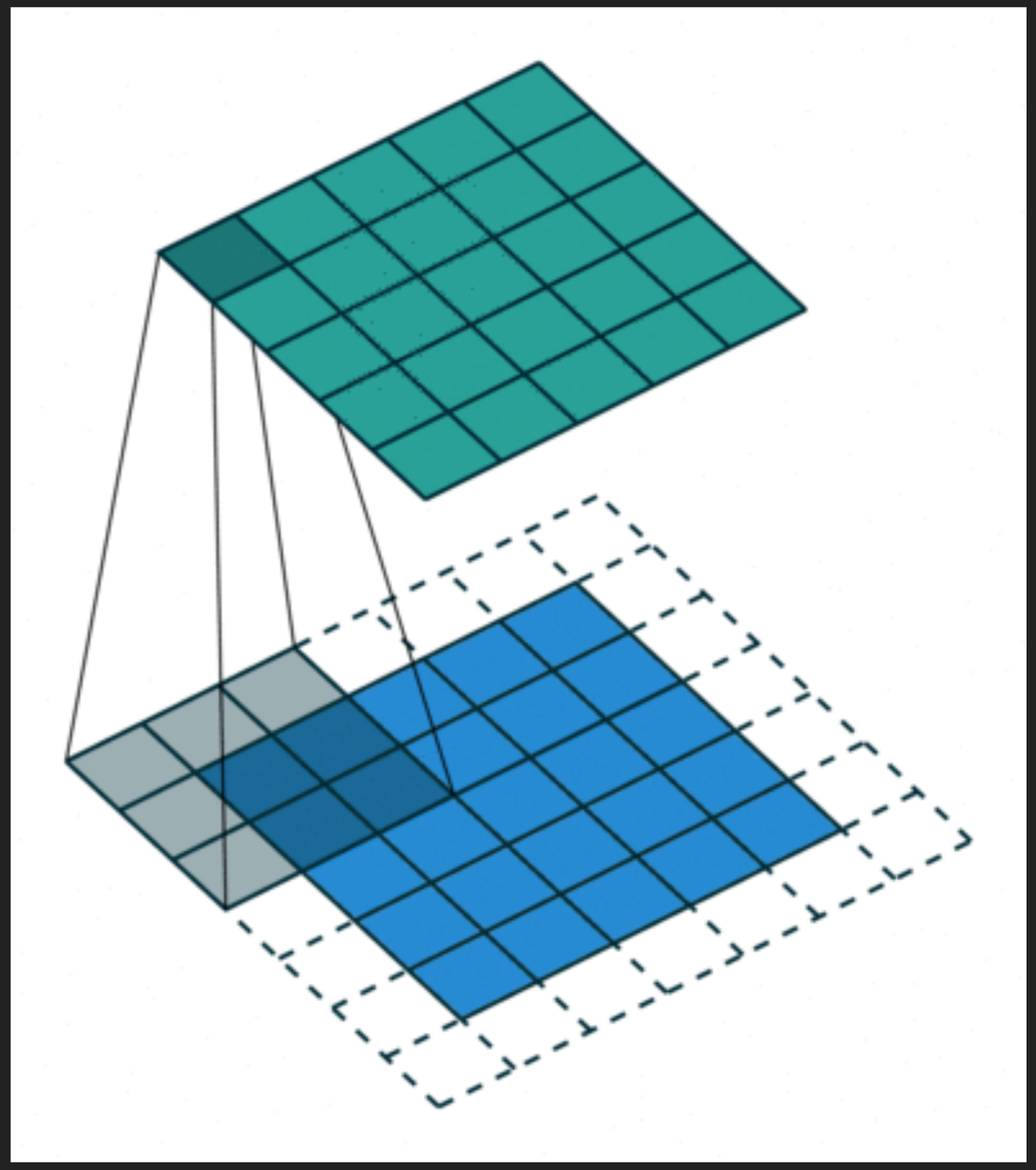
CONVOLUTIONAL NEURAL NETWORK

GOING MULTIDIMENSIONAL





CONVOLUTION OPERATOR



- Local Connectivity
- Shared weights (Parameter sharing)
- Spatial arrangement (Stride)
- Importance of Padding

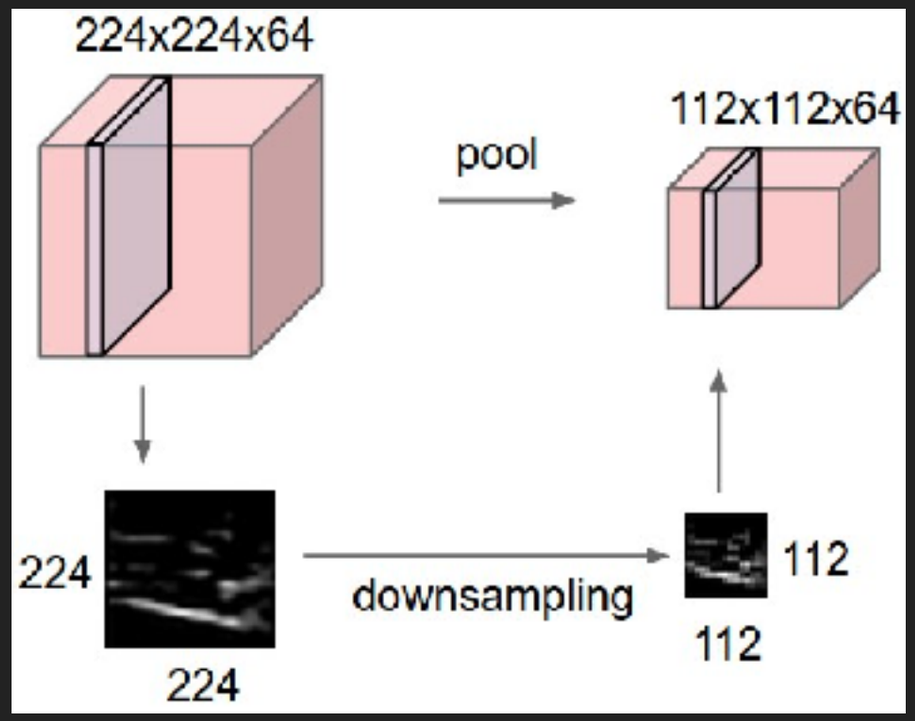
SUMMARY: CONVOLUTIONAL LAYER

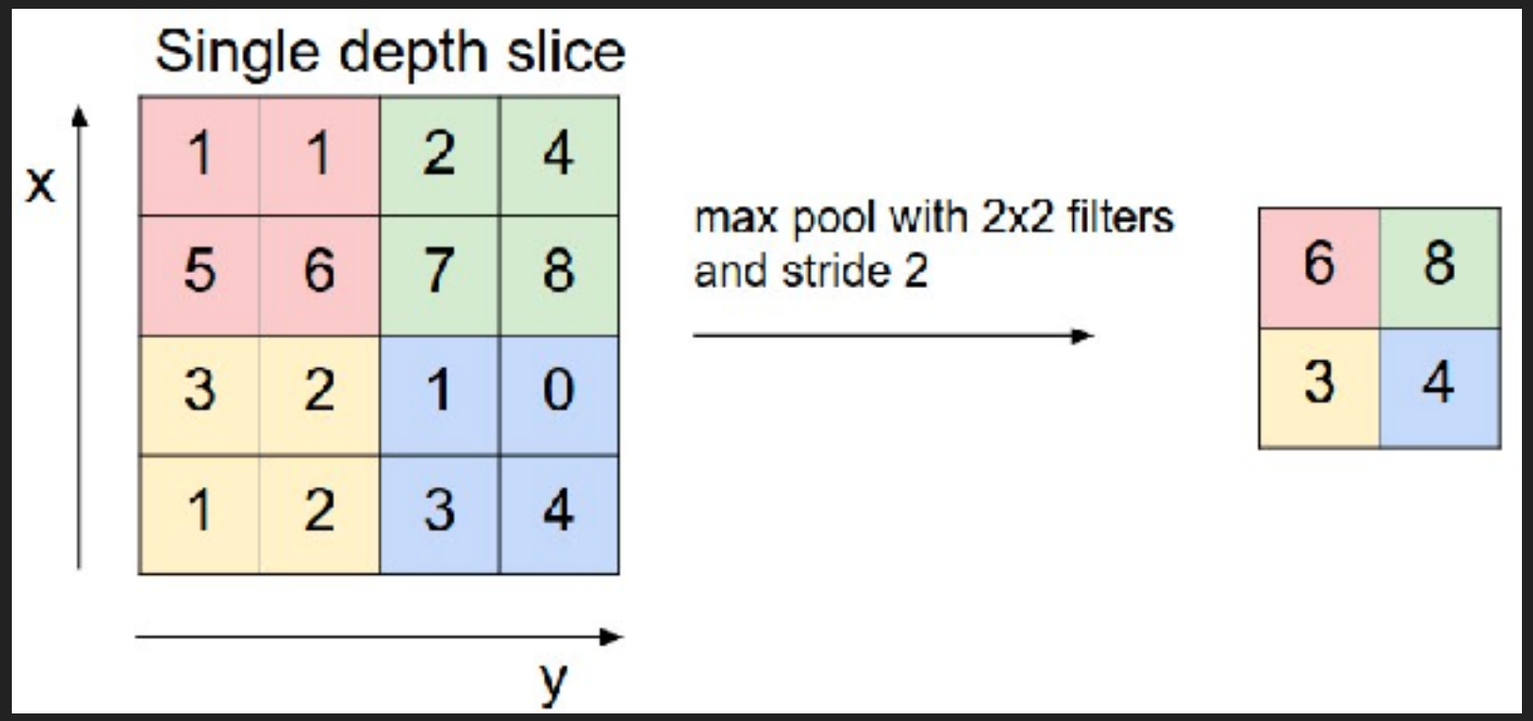
Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d-th depth slice (of size $W_2 \times H_2$) is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

A common setting of the hyperparameters is $F = 3, S = 1, P = 1$. However, there are common conventions and rules of thumb that motivate these hyperparameters. See the ConvNet architectures section below.

POOLING LAYER





SUMMARY: POOLING

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires two hyperparameters:
  - their spatial extent F,
  - the stride S,
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$
- Introduces zero parameters since it computes a fixed function of the input
- For Pooling layers, it is not common to pad the input using zero-padding.

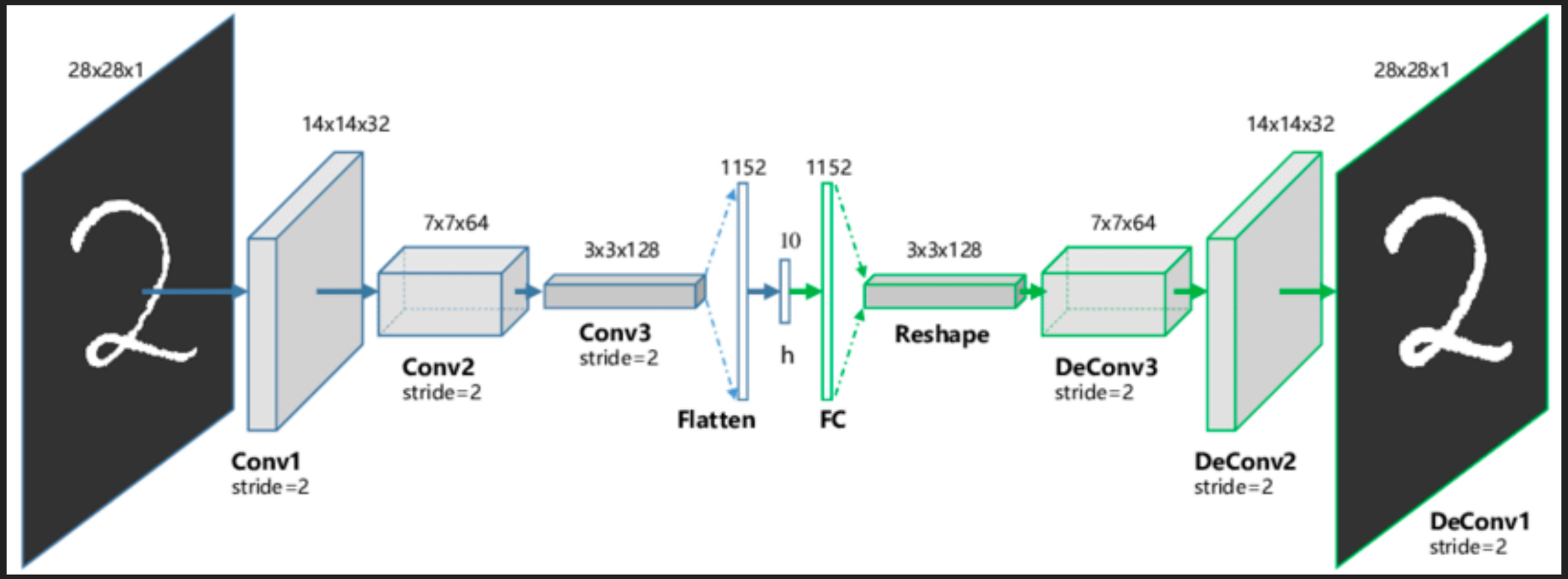
It is worth noting that there are only two commonly seen variations of the max pooling layer found in practice: A pooling layer with $F = 3, S = 2$ (also called overlapping pooling), and more commonly $F = 2, S = 2$. Pooling sizes with larger receptive fields are too destructive.

WHAT IS A CONVOLUTIONAL NEURAL NETWORK

HANDS ON

CONVOLUTIONAL AUTOENCODER

CONVOLUTIONAL AUTOENCODER



Reference Links:

1. https://www.datacamp.com/community/tutorials/autoencoder-keras-tutorial
2. https://github.com/snatch59/keras-autoencoders/blob/master/convolutional_autoencoder.py

HANDS ON STEPS: KERAS 2.X WITH PYTHON 3.6.X

- Prepare the data
  - Data should be 4D numpy
  - num_images * channels * width * height –> channels first
  - num_images * width * height * channels –> channels last
- Define the static computational graph (model object)
  - Could be sequential or functional
- Define the training process
  - Includes loss function and optimiser
  - Compile the model
- Train and test the model
  - Train is defined as “fit” and test is defined as “predict”
  - Provide the actual data to the compiled model in this step

ASSIGNMENT

CONVOLUTIONAL AUTOENCODER

ASSIGNMENT (DEADLINE: 14TH APRIL)

- Part 1: Implement Convolution2D, Pooling2D, Fully Connected Layer
  - Implement Conv2D, Pool2D, Fully connected layers from scratch without using Keras
  - Take a trained Conv. Autoencoder (using Keras, the one discussed in the class) & Save the weights as Numpy
  - Make a forward pass with your implemented layers and reproduce the actual output
  - Compare the output of your implementation with Keras implementation
- Part 2: Implement only decoder
  - Write a decoder to convert random noise into image (MNIST images)
  - Use Keras for this. For any random vector, I provide, the decoder should output a MNIST like image