CS 103 -12
Deep Learning

Jimmy Liu 刘江

2022-12-16

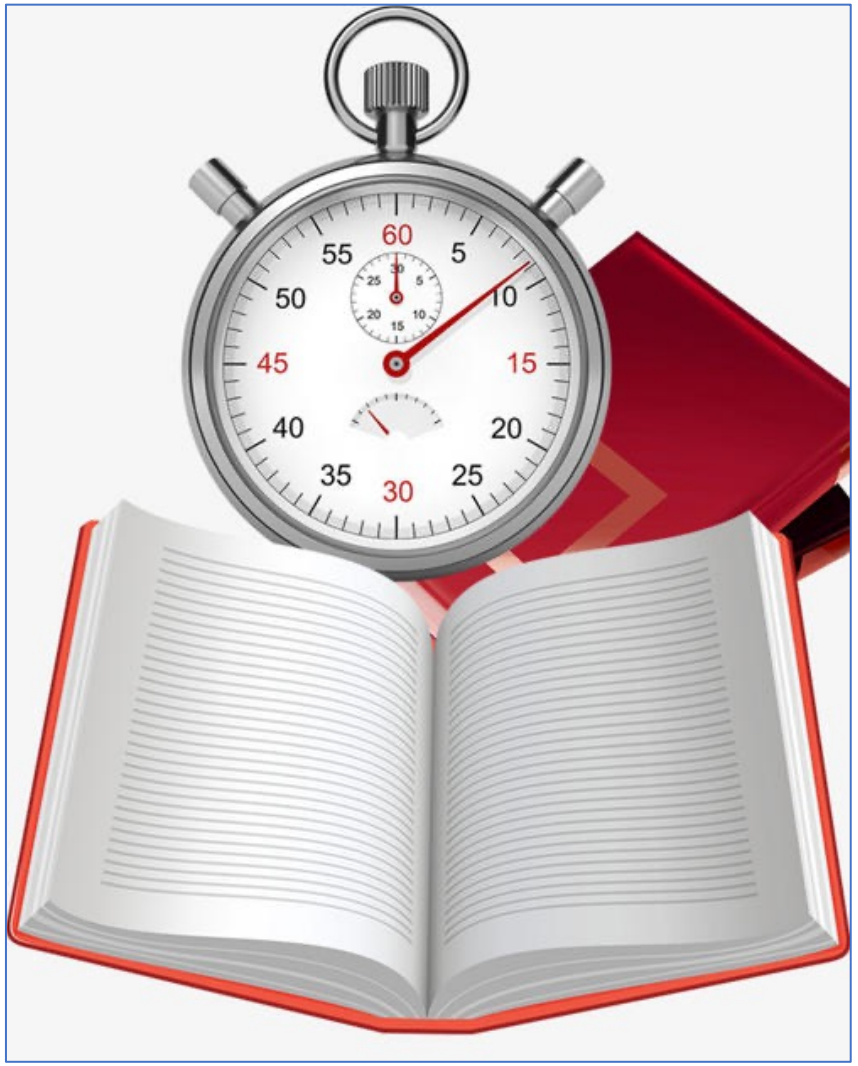
Exam

17	CS103	人工智能导论	刘江	2023-01-05	16:30-18:30	三教205	41	21	聂秋实	肖尊杰
18	CS103	人工智能导论	刘江	2023-01-05	16:30-18:30	三教206	87	80	刘江	杨冰

Home Work 11 and Answer

-13	-20	-17
-18	-24	-18
13	20	17

Review of Lecture 11

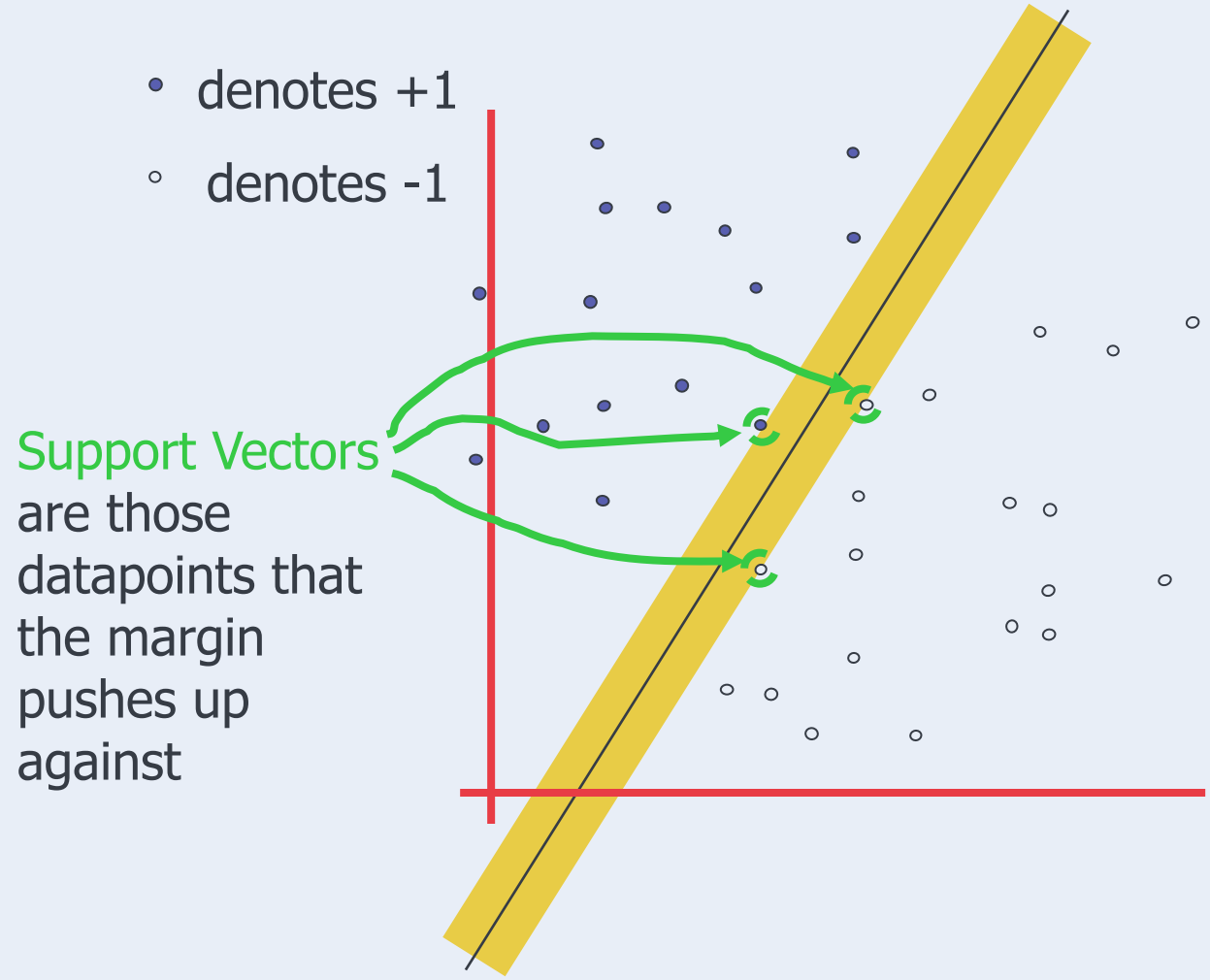


Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Developed at AT&T Bell Laboratories by Vapnik with colleagues (Boser et al., 1992, Guyon etal., 1993, Vapnik et al., 1997), it presents one of the most robust prediction methods, based on the statistical learning framework or VC theory proposed by Vapnik and Chervonenkis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

Wikipedia

f(x,w,b) = sign(w. x + b)



Margin of a linear classifier is the width that the boundary could be increased by before hitting a datapoint.

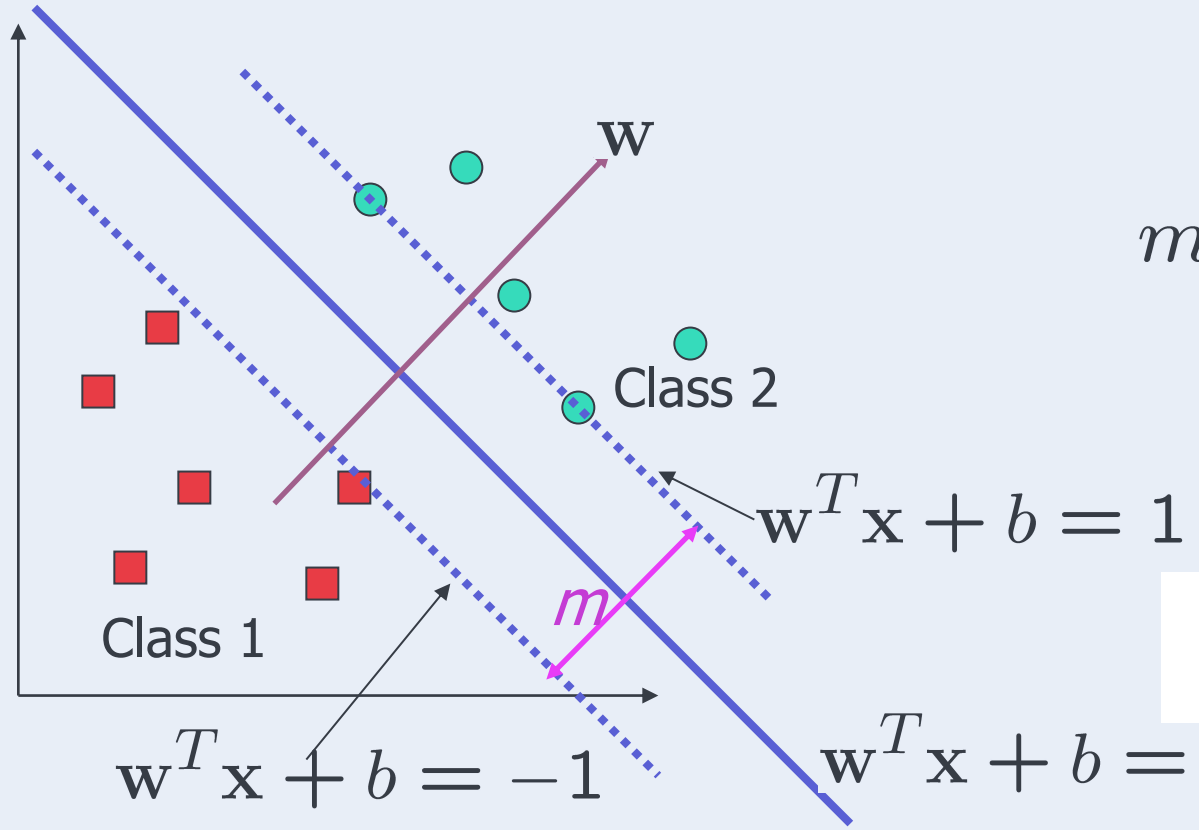
The maximum margin linear classifier is the Linear SVM (LSVM)

Margin m

The decision boundary should be as far away from the data of both classes as possible

We should maximize the margin *m: smallest distance from observations to hyperplane*

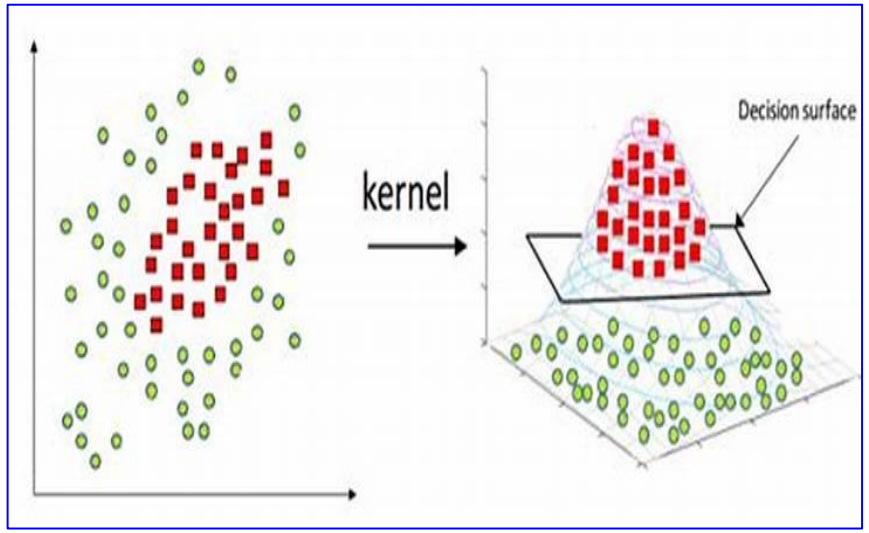
Distance between the origin and the line $\mathbf{w}^t\mathbf{x}=-b$ is $b/||\mathbf{w}||$

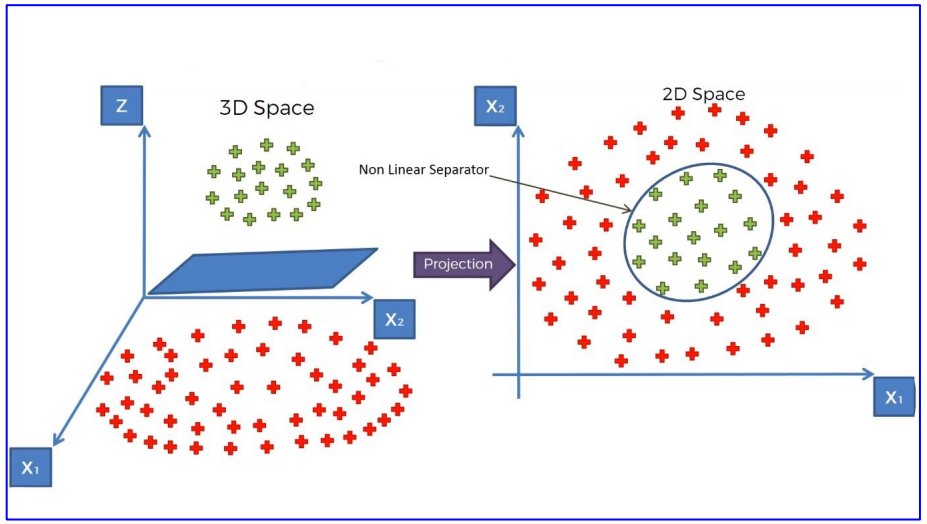


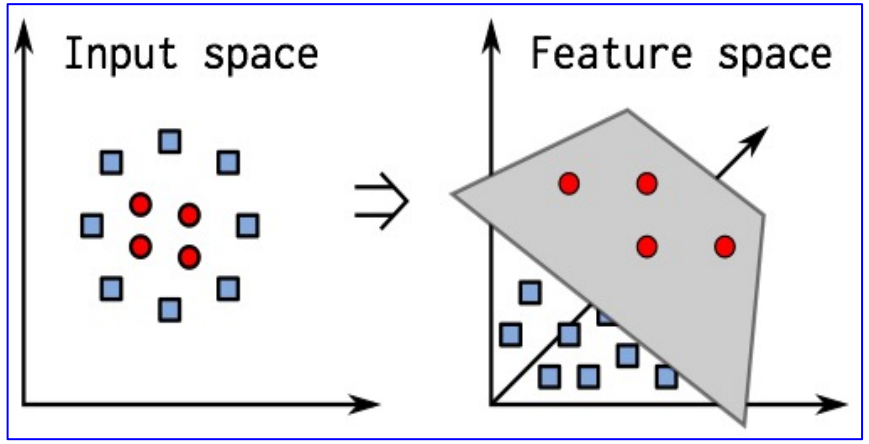
$$m = \frac{2}{||\mathbf{w}||}$$

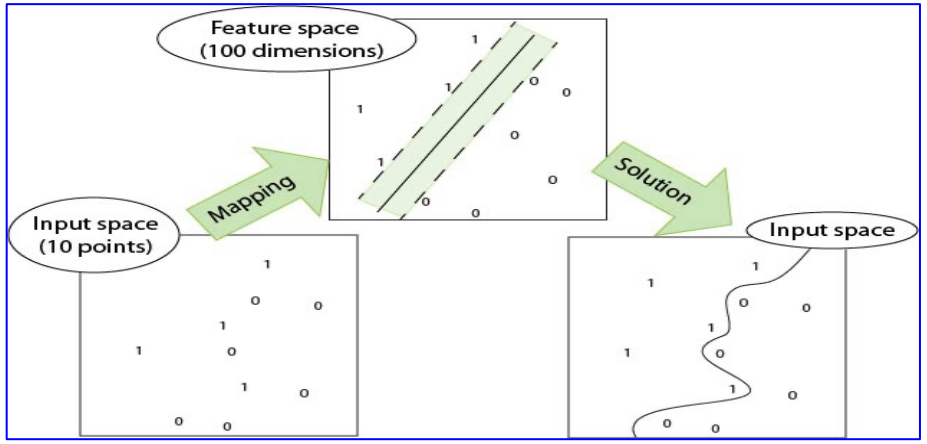
$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}_i + b \geq +1, \quad \Rightarrow \quad y_i = +1$$
$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}_i + b \leq -1, \quad \Rightarrow \quad y_i = -1$$

Non-Linear SVM Classifier with Kernel Mapping

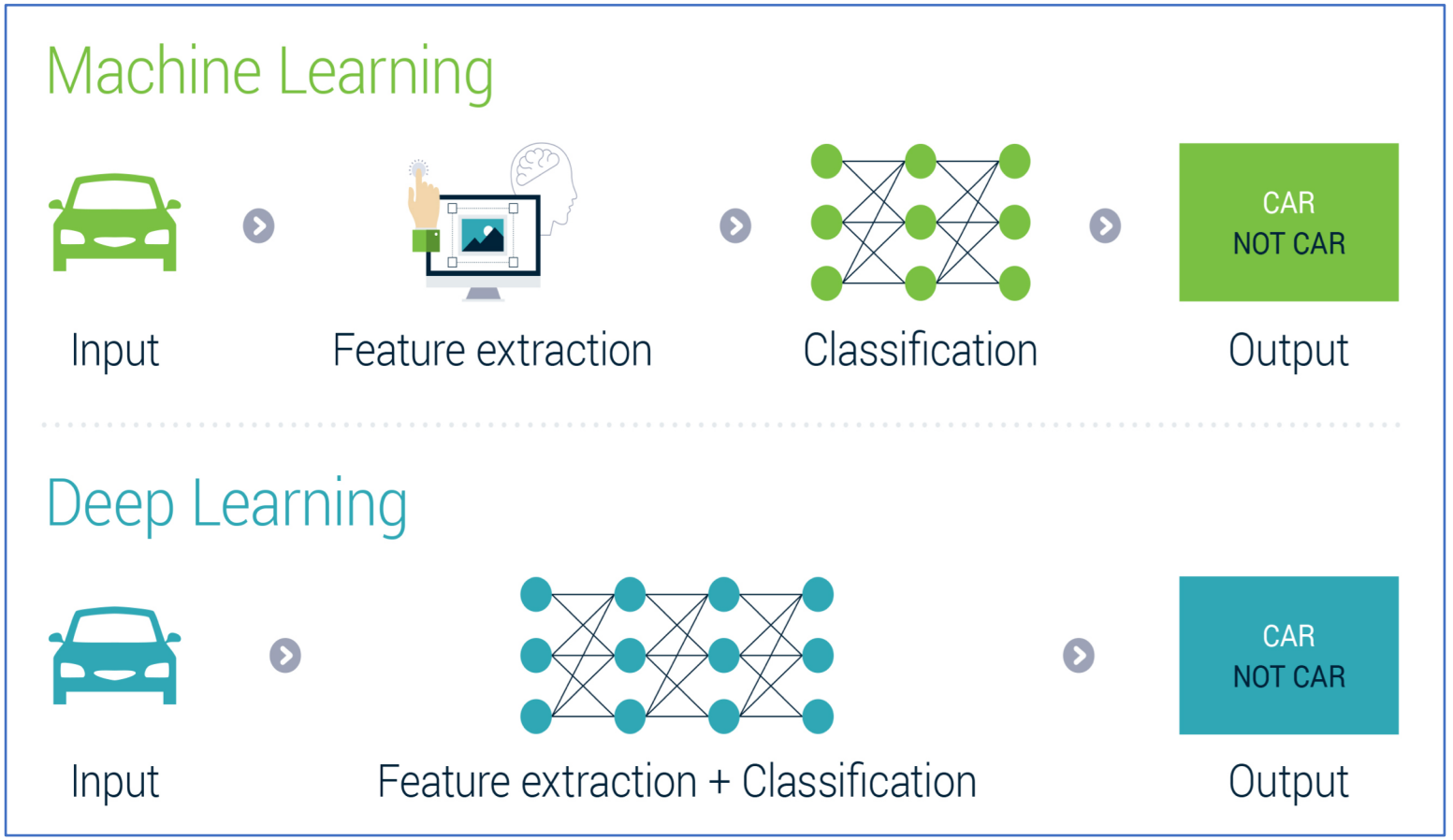




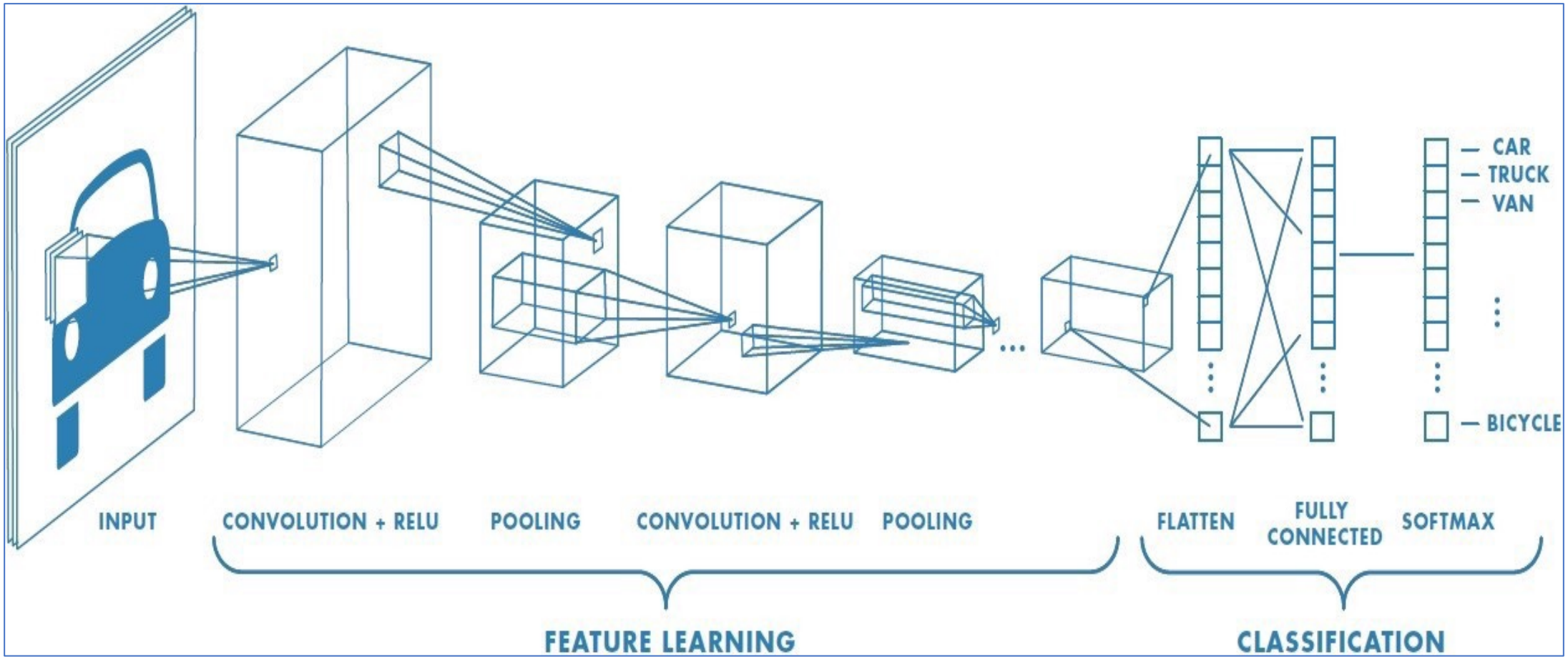




Machine Learning and Deep Learning



Convolutional Neural Network

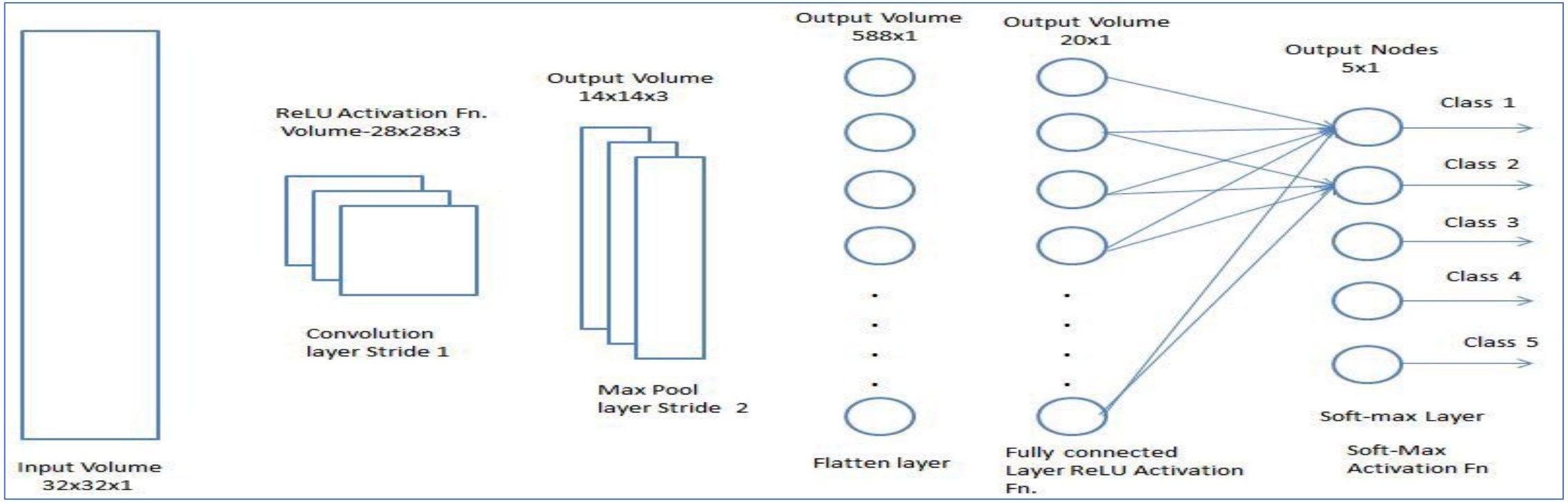


Convolution Layer Output Size Computing

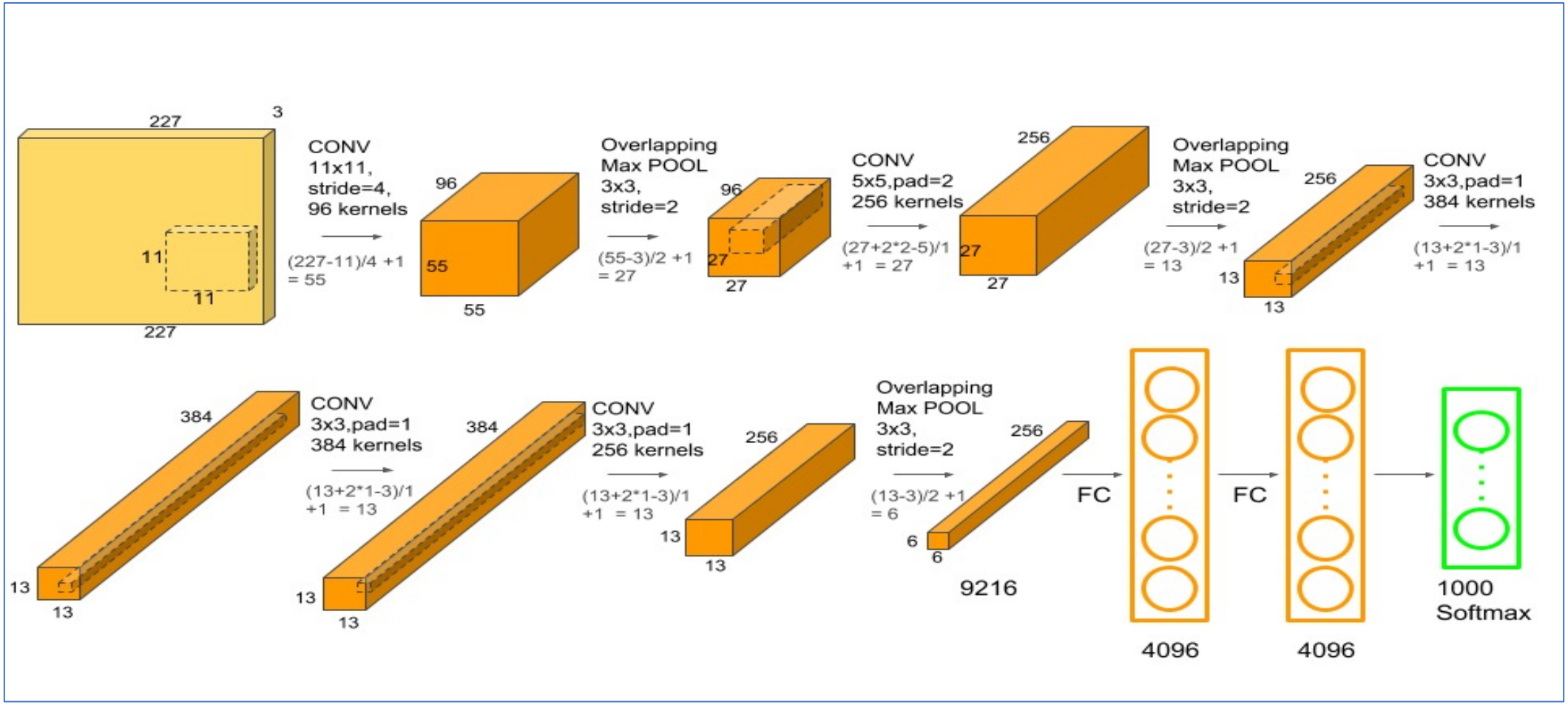
- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (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.

Fully Connected Layer (FC Layer)

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.



Output Size AlexNet Computing



Number of Parameters of a Conv Layer

W_c = Number of weights of the Conv Layer.

B_c = Number of biases of the Conv Layer.

P_c = Number of parameters of the Conv Layer.

K = Size (width) of kernels used in the Conv Layer.

N = Number of kernels.

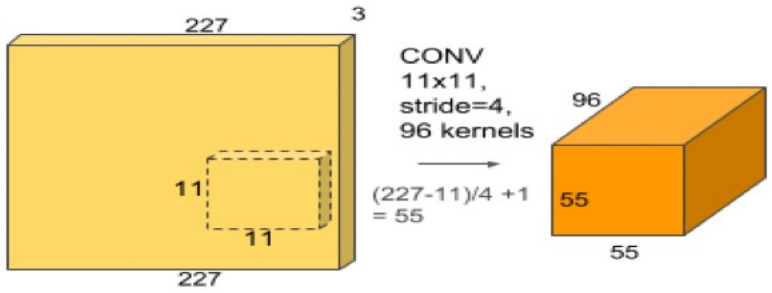
C = Number of channels of the input image.

$$W_c = K^2 \times C \times N$$
$$B_c = N$$
$$P_c = W_c + B_c$$

In a Conv Layer, the depth of every kernel is always equal to the number of channels in the input image. So every kernel has $K^2 \times C$ parameters, and there are N such kernels. That's how we come up with the above formula.

Example: In AlexNet, at the first Conv Layer, the number of channels (C) of the input image is 3, the kernel size (K) is 11, the number of kernels (N) is 96. So the number of parameters is given by

$$W_c = 11^2 \times 3 \times 96 = 34,848$$
$$B_c = 96$$
$$P_c = 34,848 + 96 = 34,944$$



Number of Parameters of a Fully Connected (FC) Layer connected to a Conv Layer

W_{ff} = Number of weights of a FC Layer which is connected to an FC Layer.

B_{ff} = Number of biases of a FC Layer which is connected to an FC Layer.

P_{ff} = Number of parameters of a FC Layer which is connected to an FC Layer.

F = Number of neurons in the FC Layer.

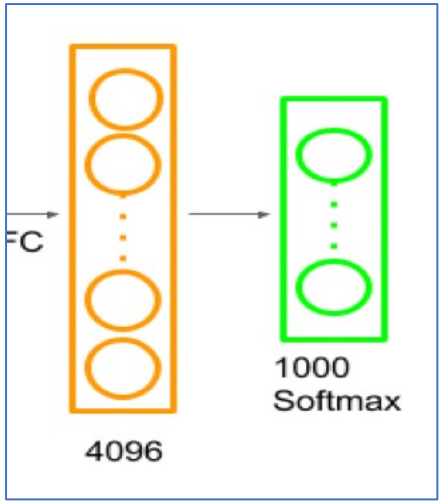
F_{-1} = Number of neurons in the previous FC Layer.

$$W_{ff} = F_{-1} \times F$$
$$B_{ff} = F$$
$$P_{ff} = W_{ff} + B_{ff}$$

In the above equation, $F_{-1} \times F$ is the total number of connection weights from neurons of the previous FC Layer the neurons of the current FC Layer. The total number of biases is the same as the number of neurons (F).

Example: The last fully connected layer of AlexNet is connected to an FC Layer. For this layer, $F_{-1} = 4096$ and $F = 1000$. Therefore,

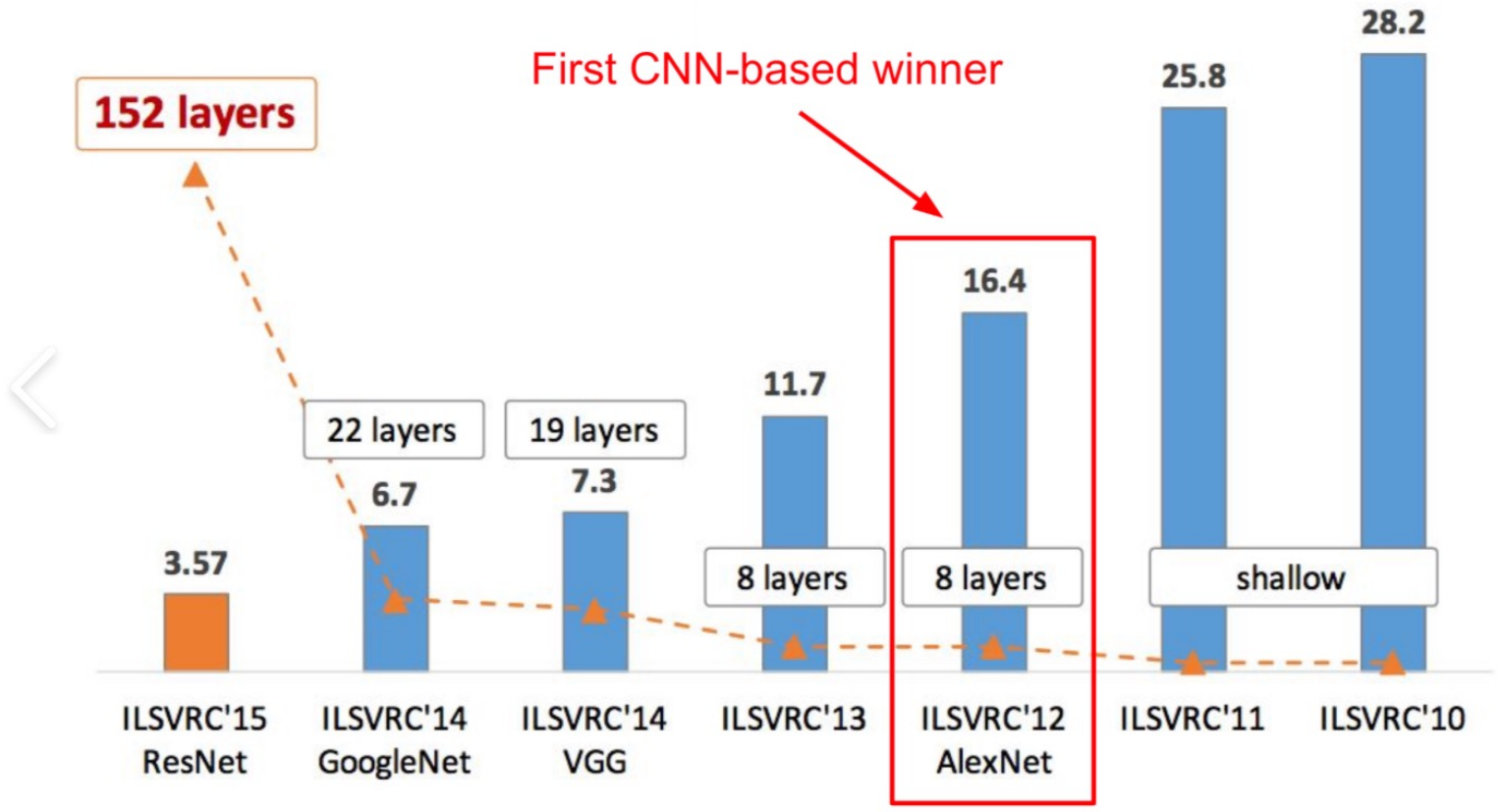
$$W_{ff} = 4096 \times 1000 = 4,096,000$$
$$B_{ff} = 1,000$$
$$P_{ff} = W_{ff} + B_{ff} = 4,097,000$$



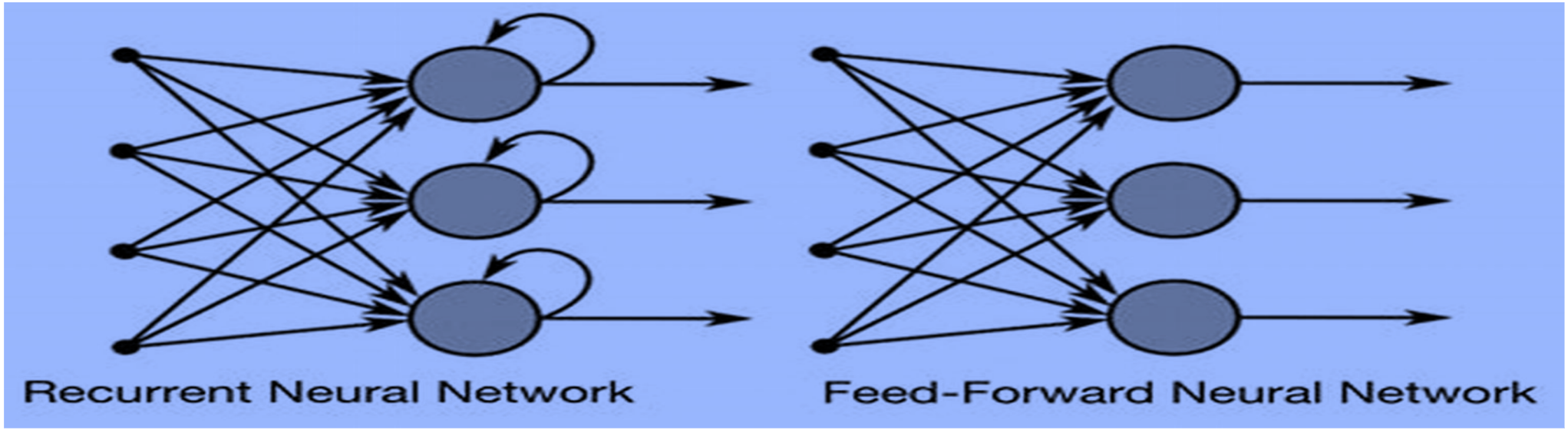
No. of Parameters and Tensor Sizes in AlexNet

Layer Name	Tensor Size	Weights	Biases	Parameters
Input Image	227x227x3	0	0	0
Conv-1	55x55x96	34,848	96	34,944
MaxPool-1	27x27x96	0	0	0
Conv-2	27x27x256	614,400	256	614,656
MaxPool-2	13x13x256	0	0	0
Conv-3	13x13x384	884,736	384	885,120
Conv-4	13x13x384	1,327,104	384	1,327,488
Conv-5	13x13x256	884,736	256	884,992
MaxPool-3	6x6x256	0	0	0
FC-1	4096×1	37,748,736	4,096	37,752,832
FC-2	4096×1	16,777,216	4,096	16,781,312
FC-3	1000×1	4,096,000	1,000	4,097,000
Output	1000×1	0	0	0
Total				**62,378,344**

Some Popular Algorithms



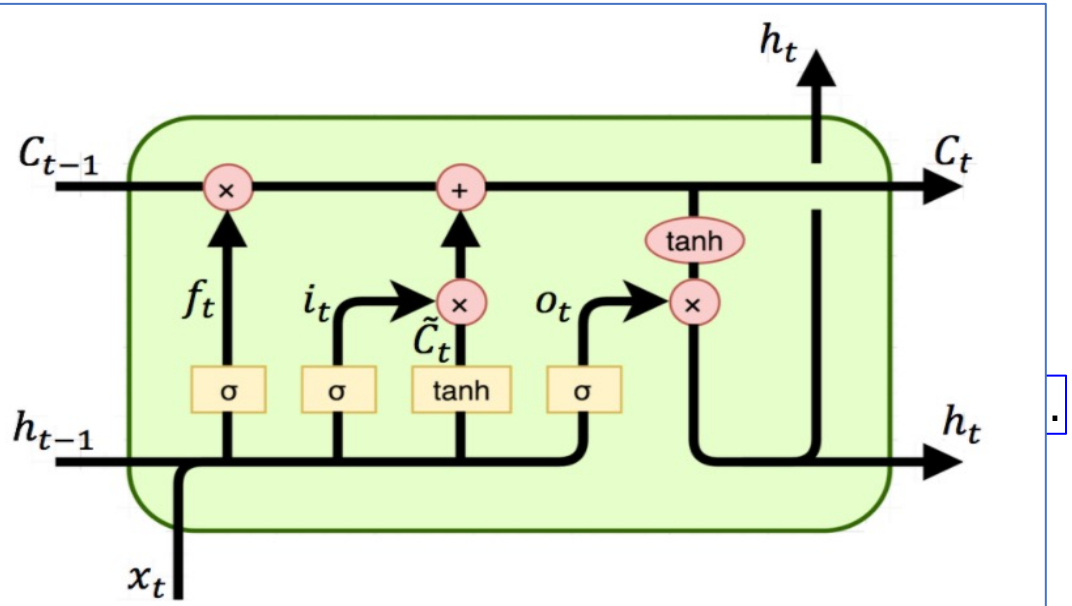
RNN and LSTM



A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs.

Long Short-Term Memory

遗忘门 (forget gate)、输入门 (input gate)、输出门 (output gate)



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

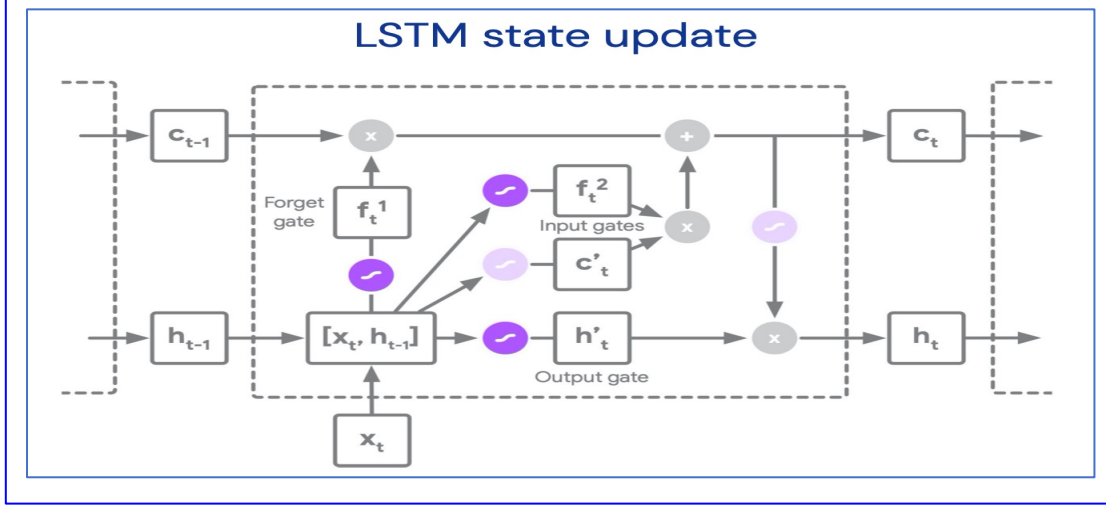
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

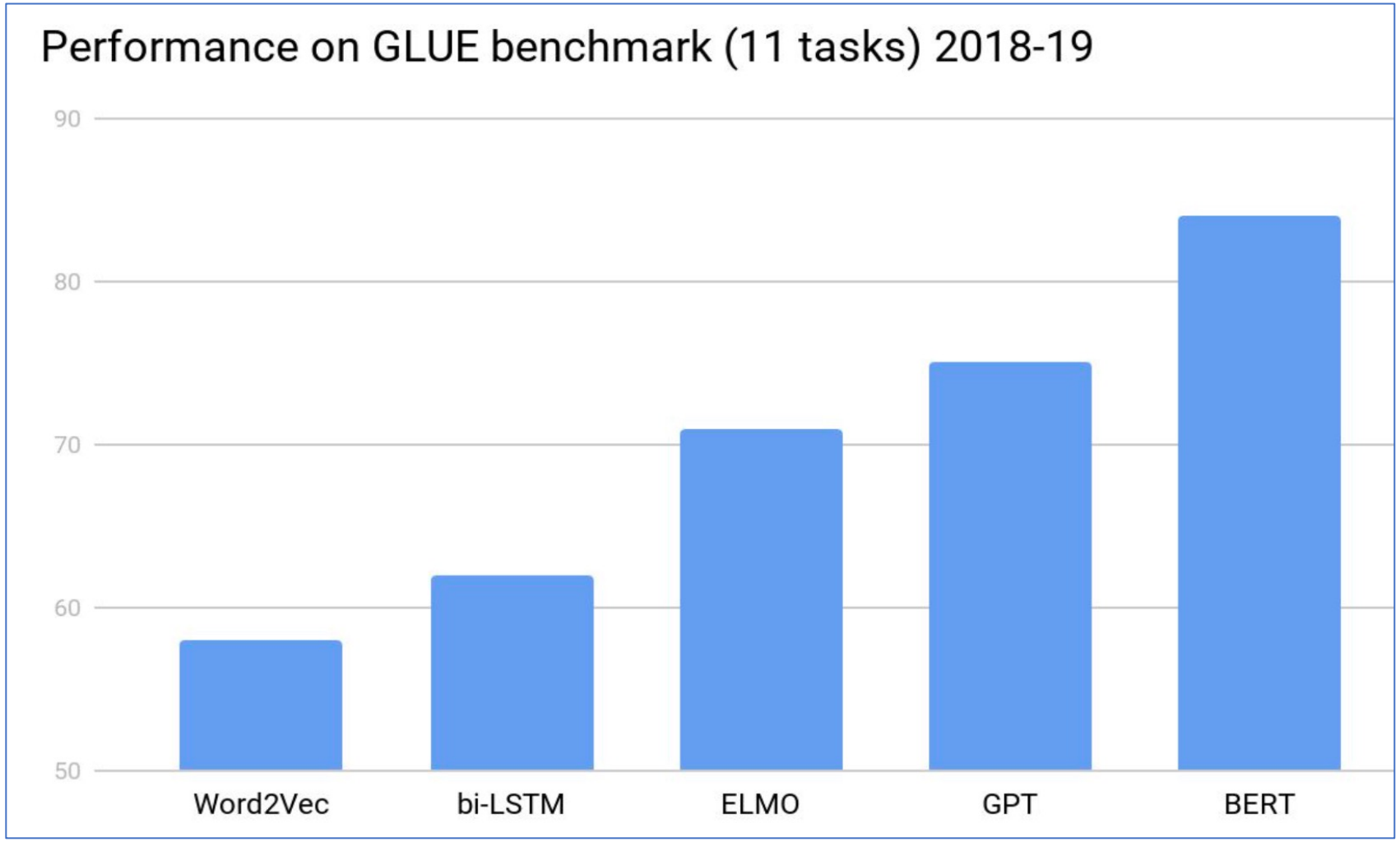
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot \tanh(C_t)$$

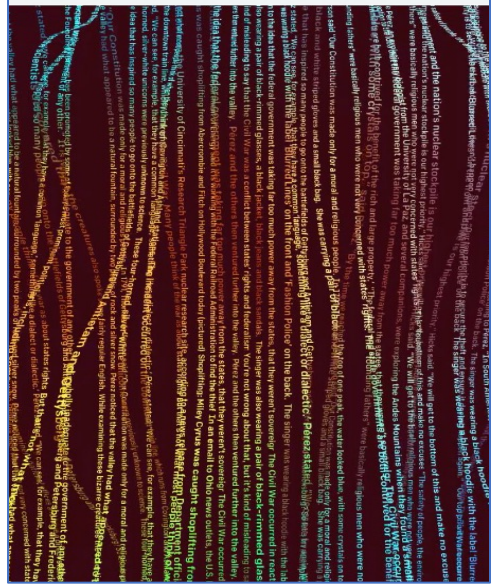


- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in \mathbb{R}^h$: forget gate's activation vector
- $i_t \in \mathbb{R}^h$: input/update gate's activation vector
- $o_t \in \mathbb{R}^h$: output gate's activation vector
- $h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in \mathbb{R}^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector

GPT3 General Pre-trained Transformer 3-2019

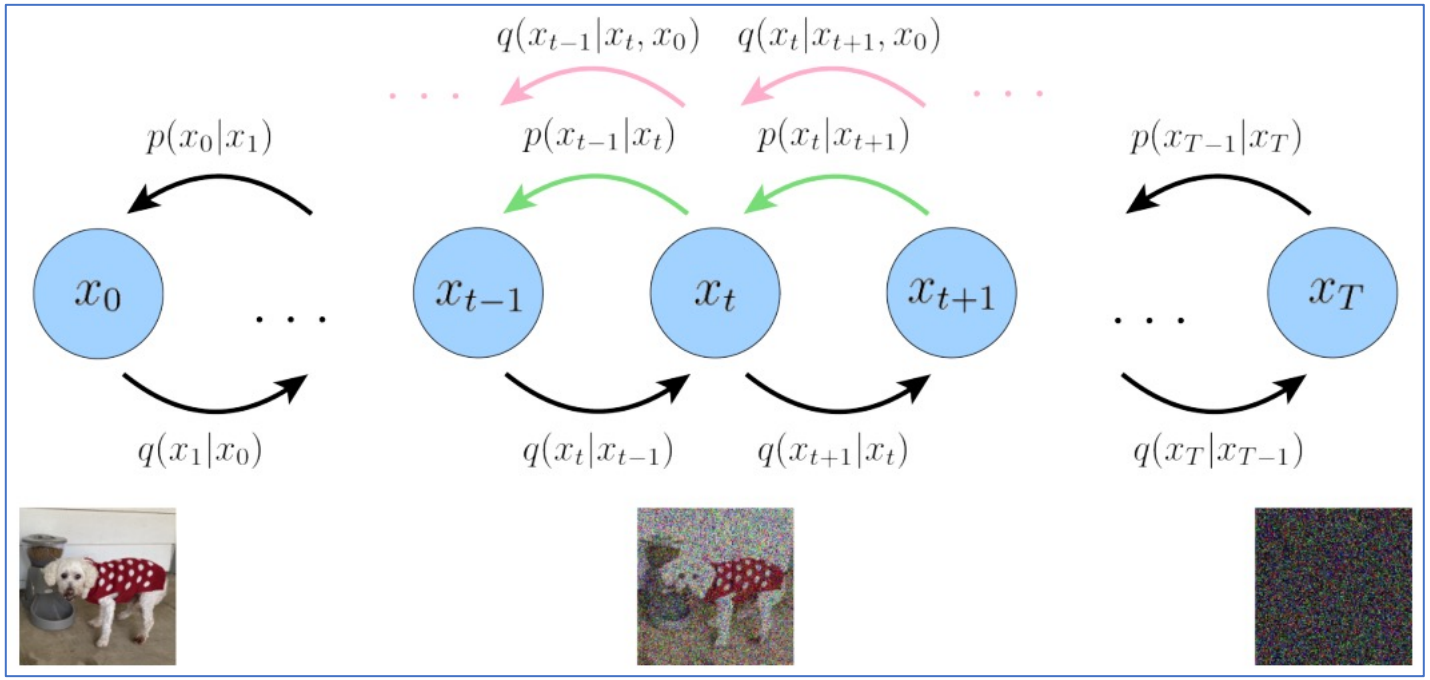


GPT-2 and 3



GPT-3 2020: Largest ANN. 175B:1.5B Parameters

Diffusion Model



Deep Learning

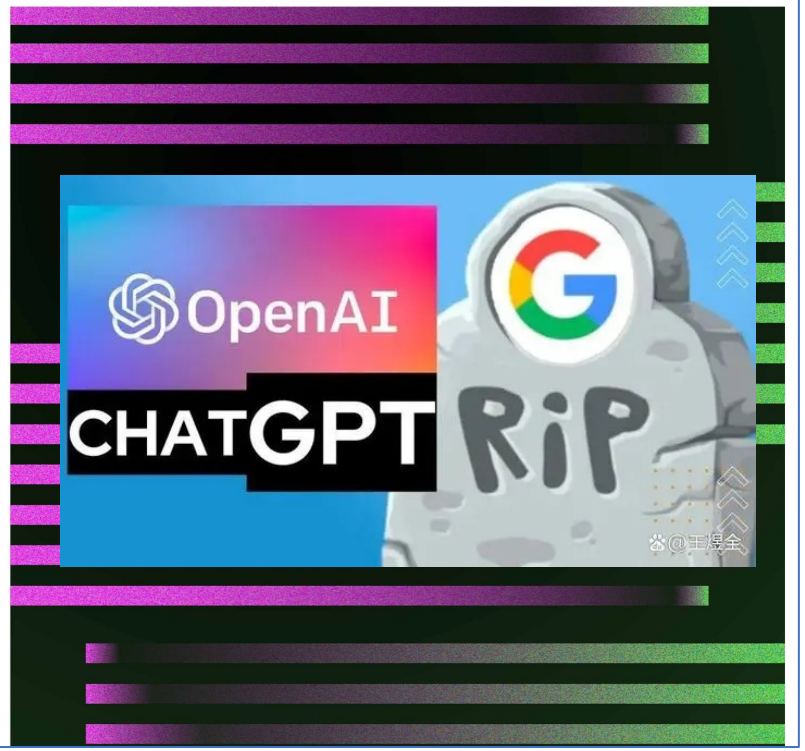
ChatGPT

OpenAI

API RESEARCH BLOG ABOUT

ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.



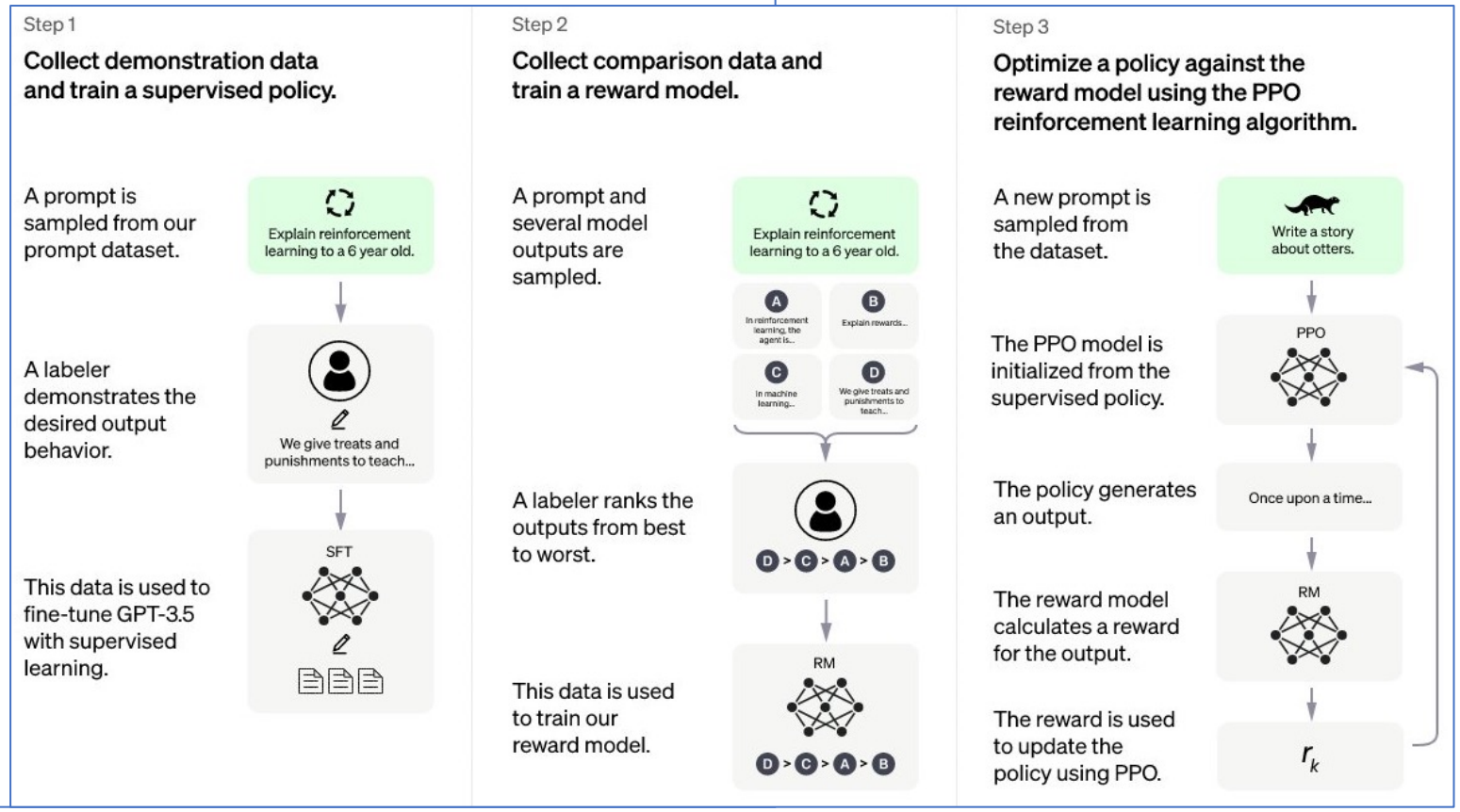
ChatGPT and Proximal Policy Optimization(PPO)

OpenAI

API RESEARCH BLOG ABOUT

ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

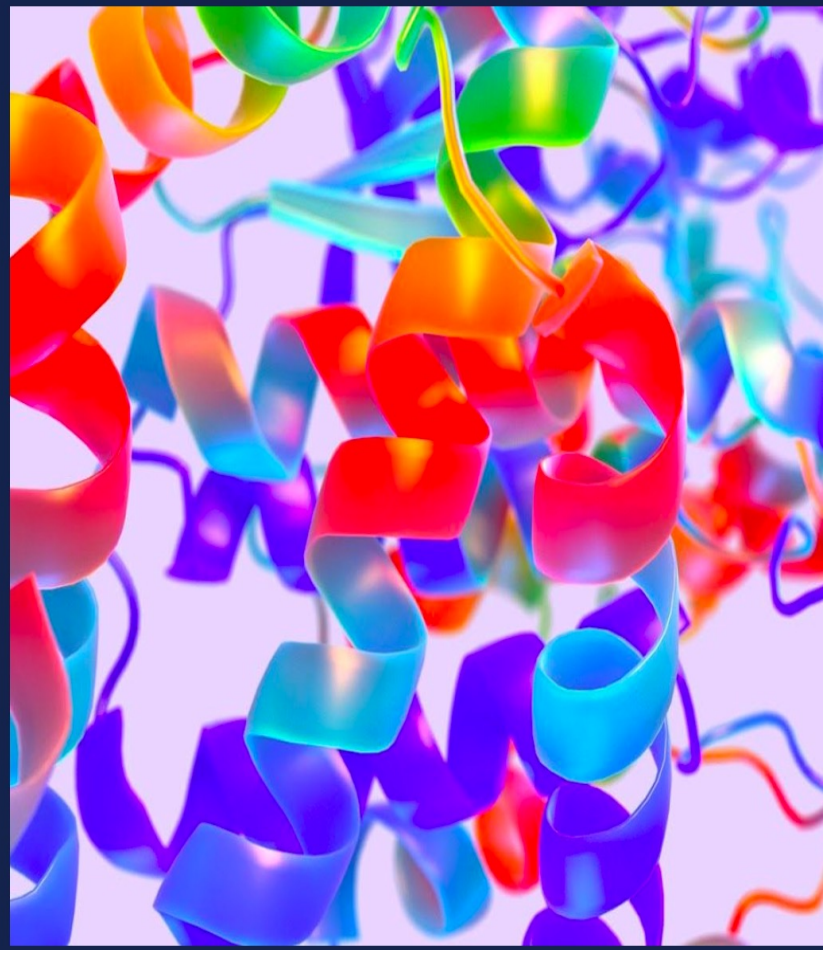


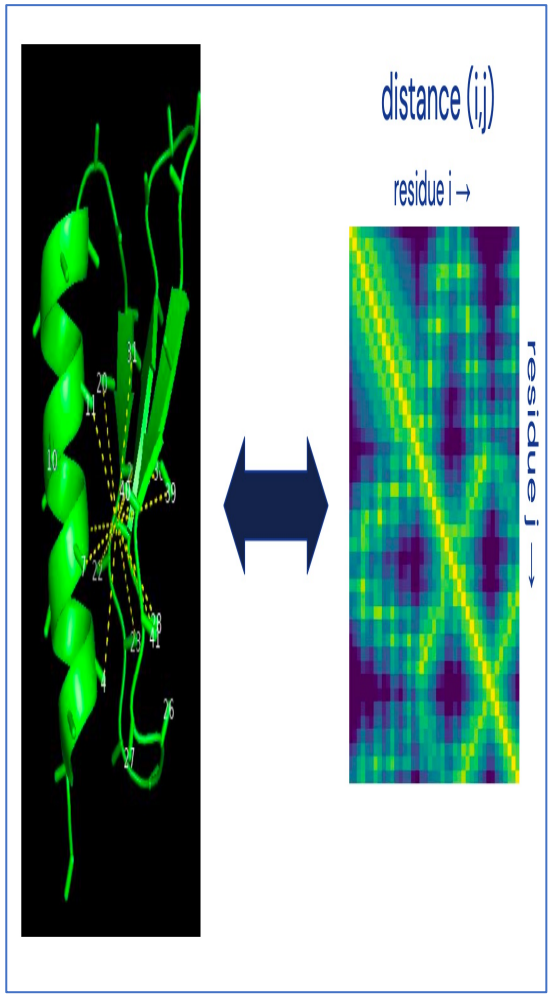
AlphaFold (2020) and AlphaFold-2 (2021)

AlphaFold: Improved proteins structure prediction using potentials from deep learning

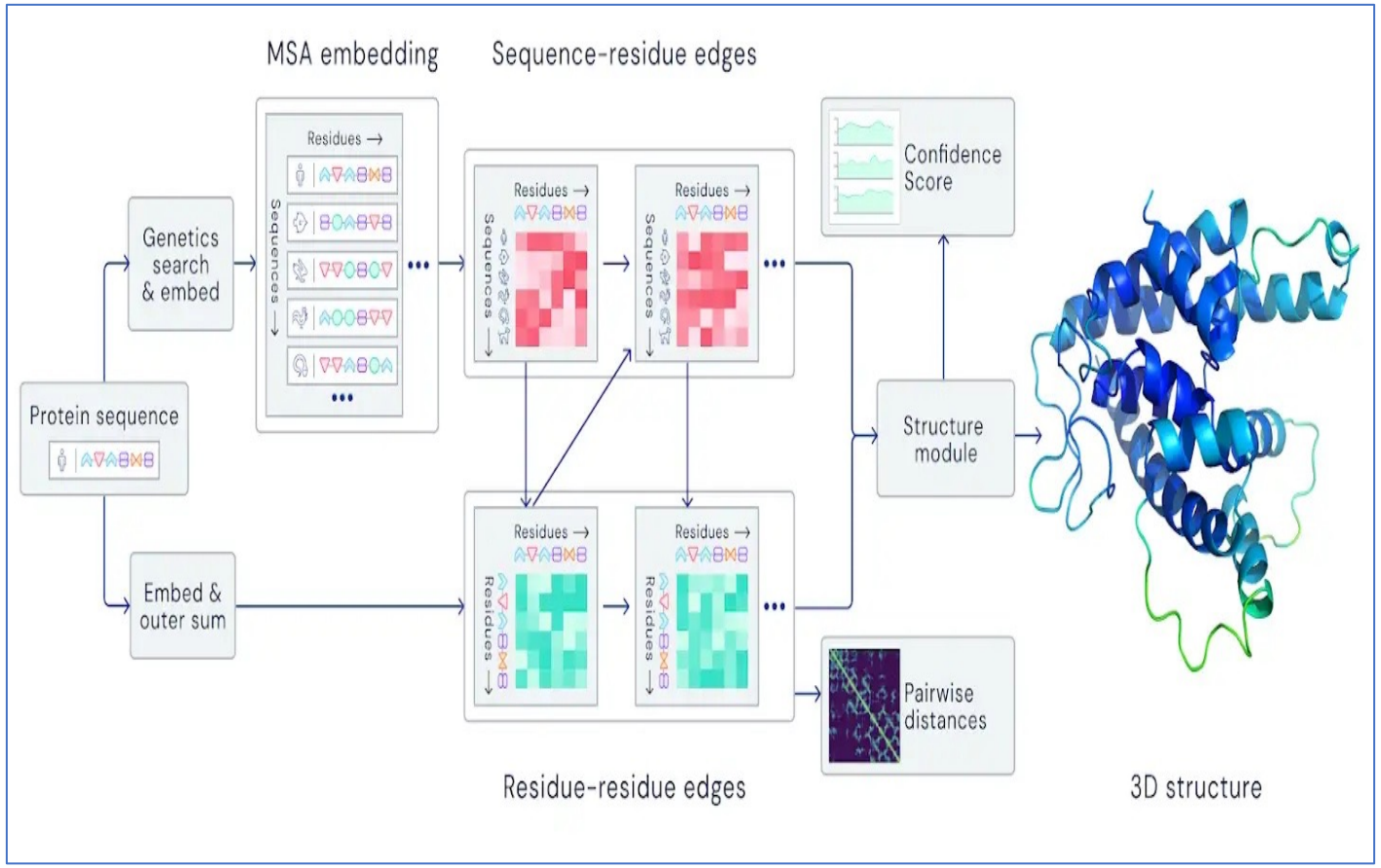
(Nature, 2020)

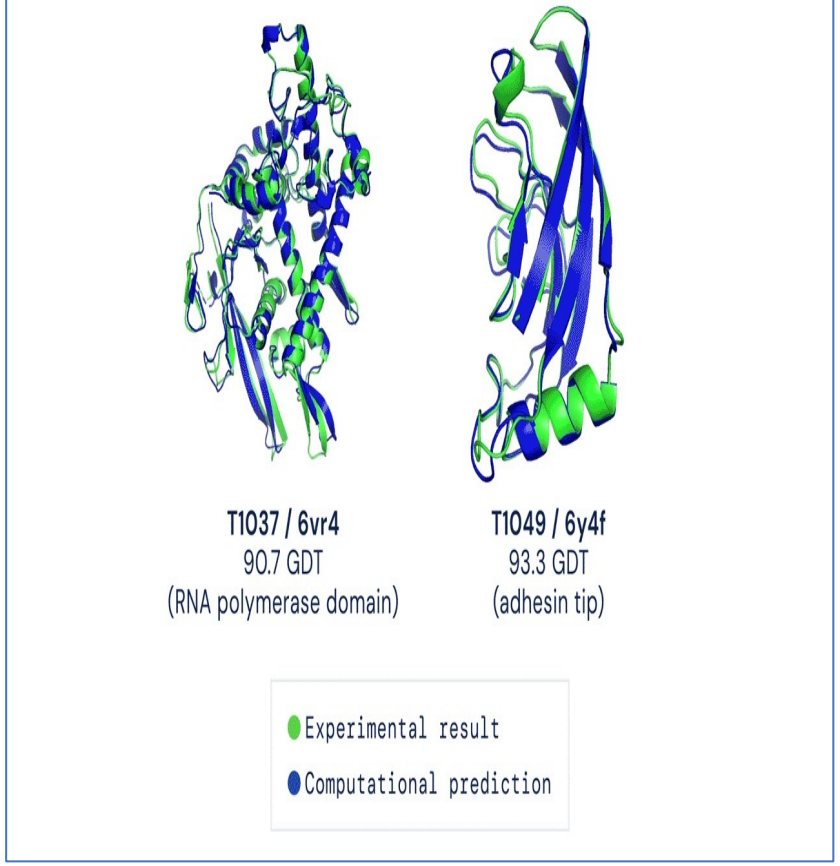
Andrew Senior, **Richard Evans**, **John Jumper**, **James Kirkpatrick**, **Laurent Sifre**, Tim Green, Chongli Qin, Augustin Zidek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, David T. Jones, Pushmeet Kohli, Steve Crossan, David Silver, Koray Kavukcuoglu, Demis Hassabis





AlphaFold


MSA embedding
Sequence-residue edges
Genetics search & embed
Protein sequence
Embed & outer sum
Residues →
Sequences →
Confidence Score
Structure module
Pairwise distances
Residue-residue edges
3D structure


T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)
T1049 / 6y4f
93.3 GDT
(adhesin tip)
Experimental result
Computational prediction

Unsupervised Learning

> I always knew unsupervised learning was the right thing to do
>
> — Geoff Hinton

> Basically it's the idea of learning to represent the world before learning a task — and this is what babies do
>
> — Yann LeCun

> And so if we can build models of the world where we have the right abstractions, where we can pin down those changes to just one or a few variables, then we will be able to adapt to those changes because we don't need as much data, as much observation in order to figure out what has changed.
>
> — Yoshua Bengio

Turing Award winners at AAAI 2020

Open Topic:
Deep Learning and Machine Learning

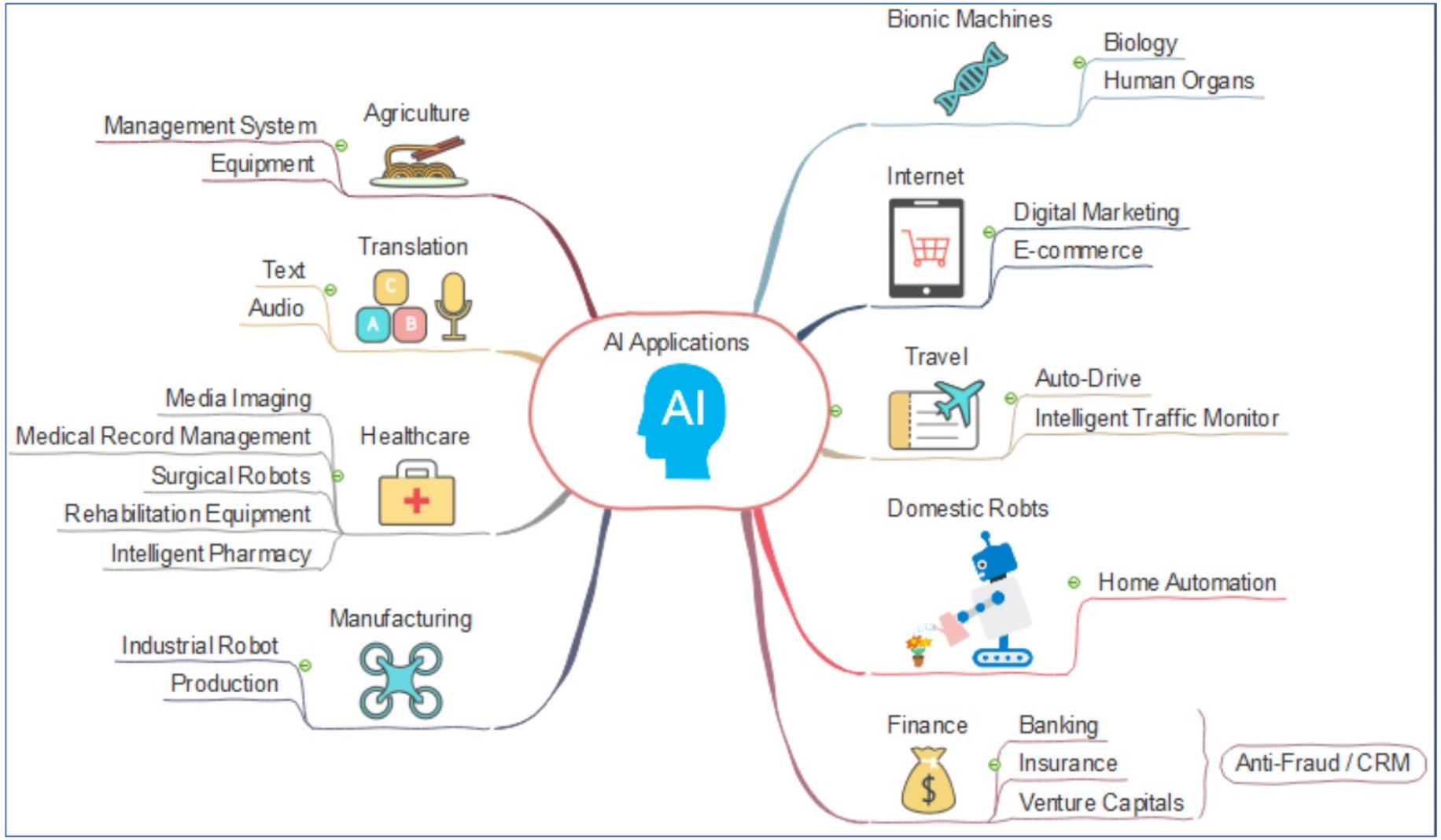
Deep Learning Neural Networks replaced handcrafted features with handcrafted architectures.

Prior knowledge is not obsolete: it is merely incorporated at a higher level of abstraction.

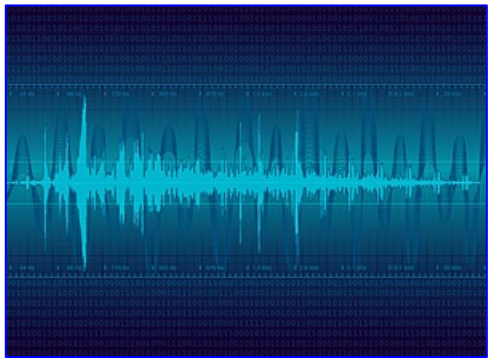
Topics

1. CS 103 Module Introduction And Class Rules
2. AI Concepts
3. AI Algorithms
4. AI Applications (AI+)

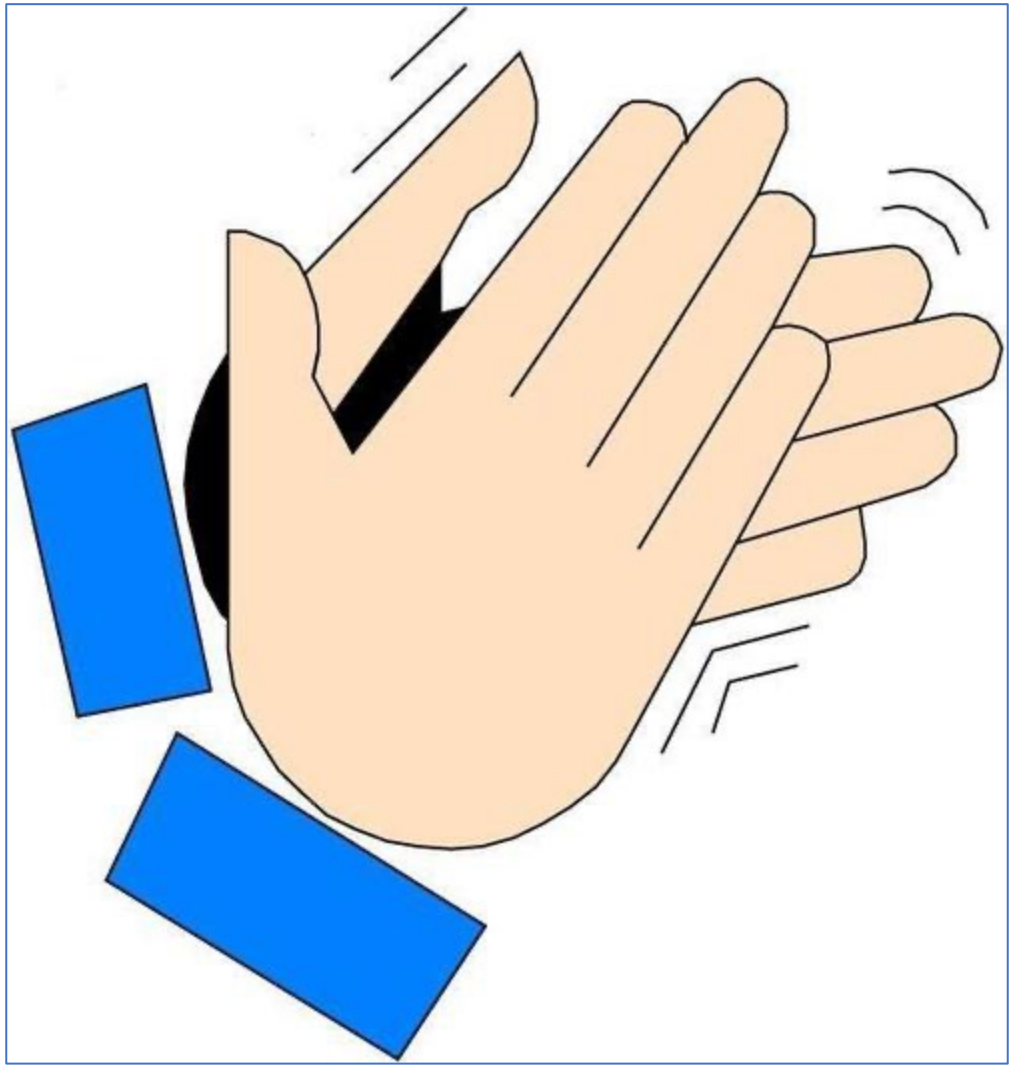
AI Applications (AI+)



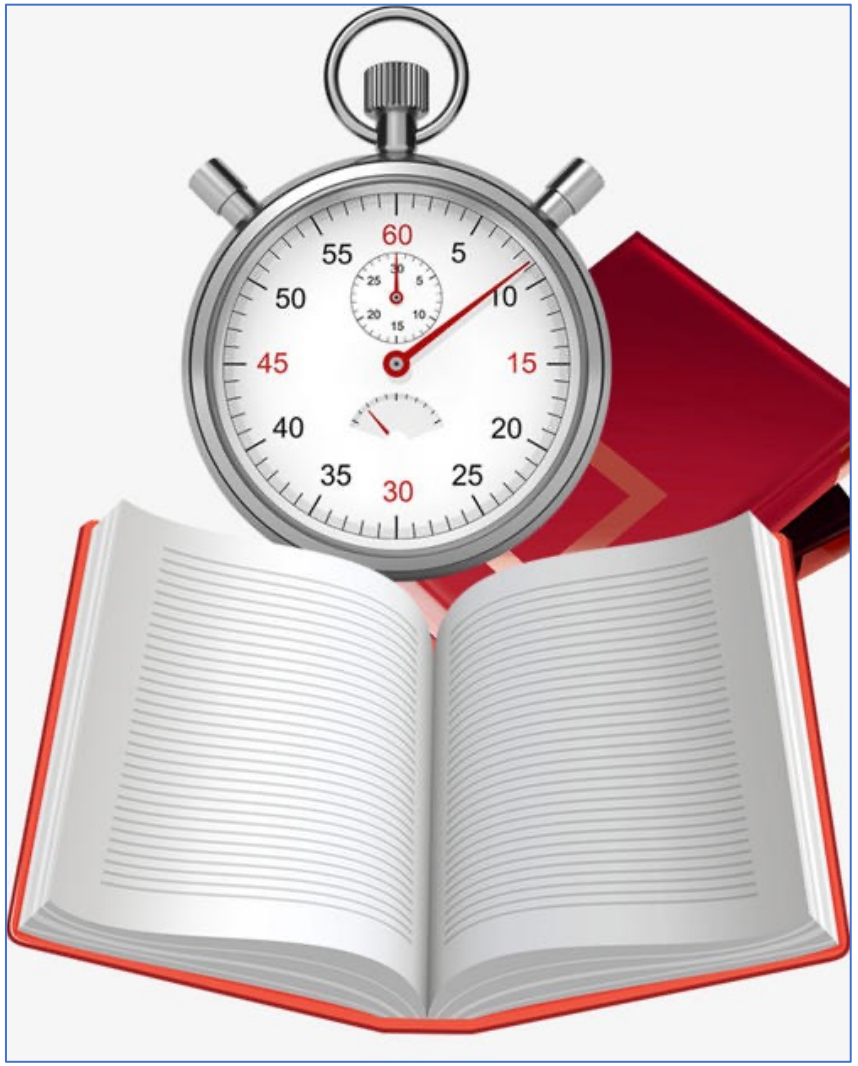
Multimedia Information Processing （CS 330）



Proud of CS 103 Students



Course Quick Review



Learn and Study

Active learning: It is about how much you think and learn

Collective study: Let us study together

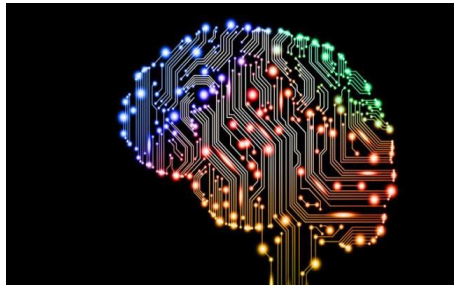
CS 103 Module Coverage

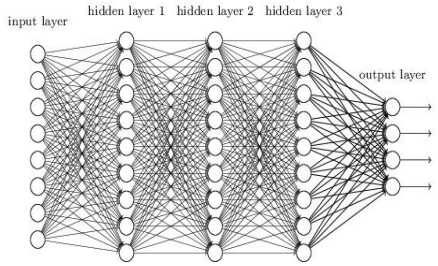
What CS 103 will cover?

AI Concepts

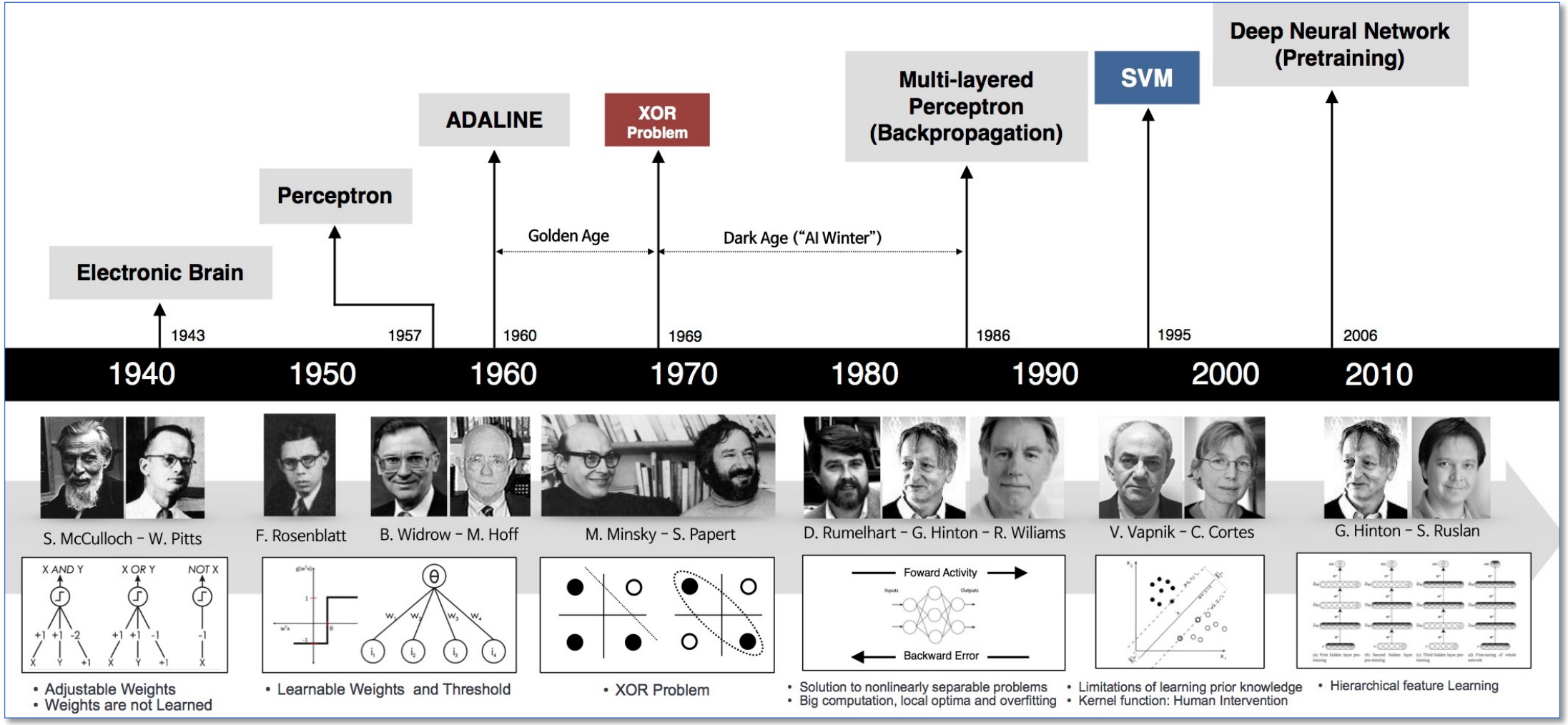
AI Algorithms

AI Application





Introduction to AI



CS 103 -12
Deep Learning

Jimmy Liu 刘江

2022-12-16