Lecture 13
AI

Mechatronics
ETM 4931
Dr. Farbod Khoshnoud

Artificial intelligence vs machine learning

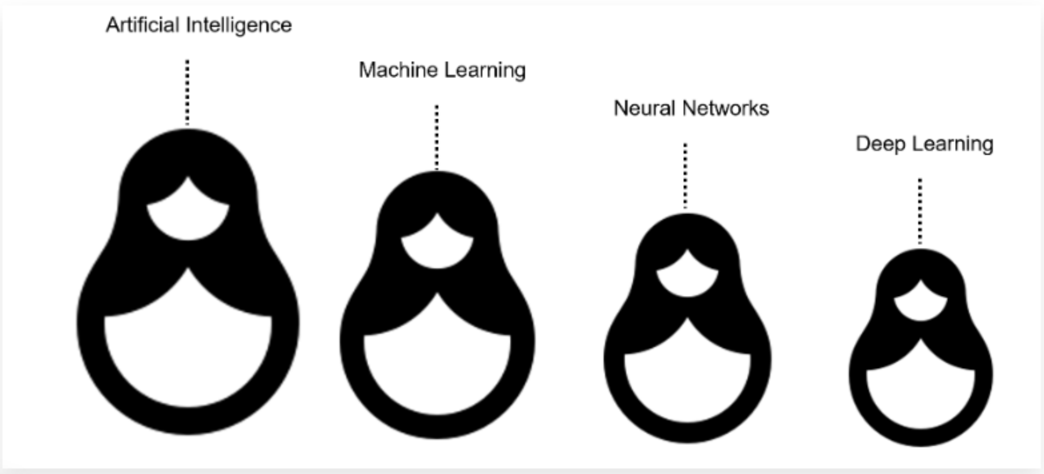
Artificial intelligence refers to the general ability of computers to emulate human thought and perform tasks in real-world environments, while machine learning refers to the technologies and algorithms that enable systems to identify patterns, make decisions, and improve themselves through experience and data.

https://ai.engineering.columbia.edu/ai-vs-machine-learning/#:~:text=Put%20in%20context%2C%20artificial%20intelligence,and%20improve%20themselves%20through%20experience

What is AI?
https://www.youtube.com/watch?v=NbEbs6I3eLw

AI and Machine Learning

- How do artificial intelligence, machine learning, neural networks, and deep learning relate?



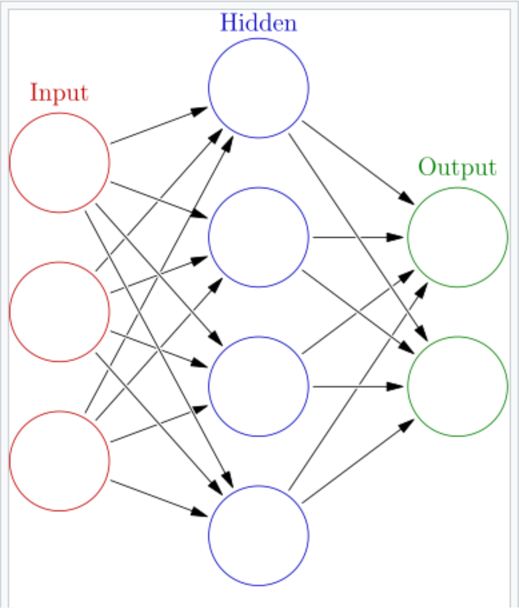
https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

What's the Difference Between AI, Machine Learning, and Deep Learning?
https://www.youtube.com/watch?v=J4Qsr93L1qs&t=29s

ANNs

Artificial Neural Networks (ANNs)—mimic the human brain through a set of algorithms (Similar to linear regression).

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

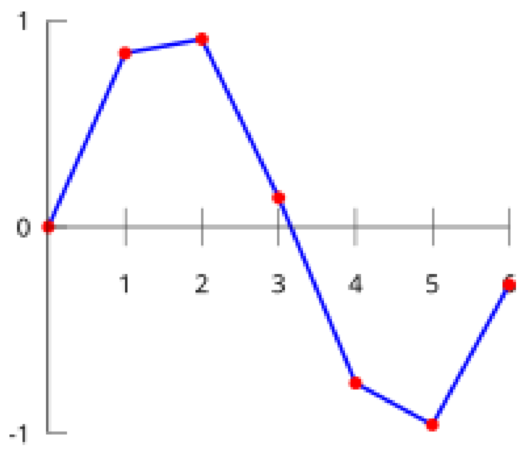


https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

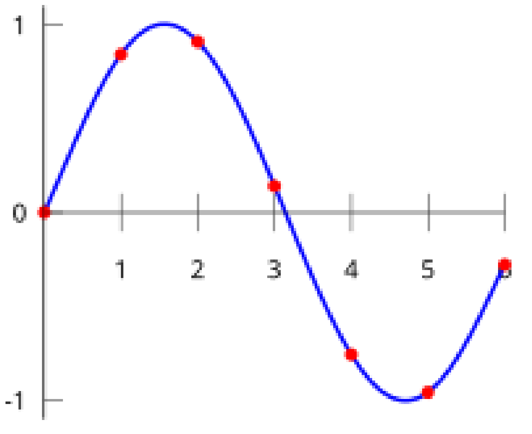
Interpolation

The goal in interpolation is to find a polynomial curve that passes through a set of points. For example, the following data points are given:

$$(x_0, y_0), (x_1, y_1), \ldots (x_n, y_n)$$

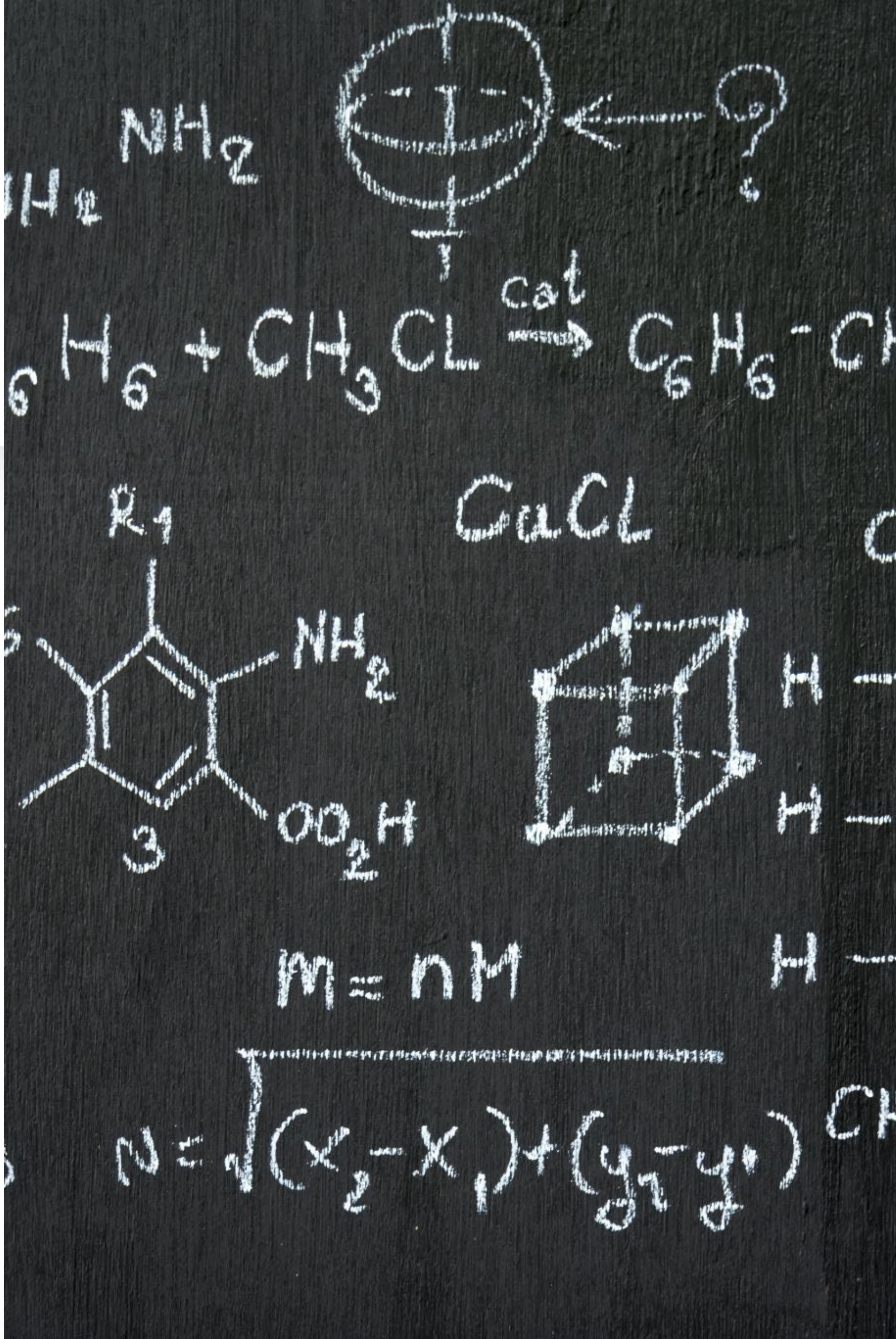


Plot of a set of data with linear interpolation (Wikipedia)



Plot of the data with polynomial interpolation applied (Wikipedia)

Interpolation methods



- Fit a polynomial of the form
- Lagrange polynomial interpolation
- Cubic spline interpolation

Interpolation - Polynomial Fit

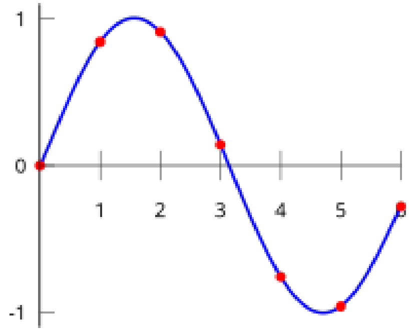
- The goal in interpolation is to find a polynomial curve that passes through a set of points. For example, the following data points are given for (n number of points):

'n' number of equations and 'n' unknowns

$$y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + \dots a_n x_1^n$$
$$y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + \dots a_n x_2^n$$
$$\vdots \quad \vdots$$
$$y_n = a_0 + a_1 x_n + a_2 x_n^2 + \dots a_n x_n^n$$

We can write the equations as a matrix equation $Ax = b$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$



Interpolation - Lagrange polynomial interpolation

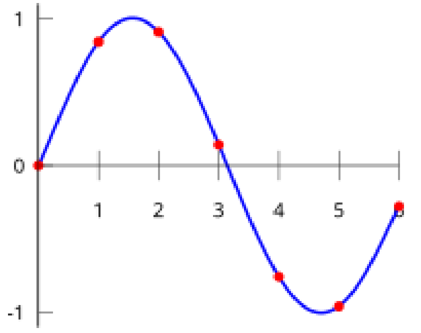
Given the points $(x_1, y_1), \ldots (x_n, y_n)$, the Lagrange polynomial interpolation is

$$y(x) = L_1(x)y_1 + L_2(x)y_2 + \ldots L_n(x)y_n = \sum_{i=1}^{n} L_i(x)y_i$$

where

$$L_i(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{i-1})\ \ (x-x_{i+1})\ldots(x-x_n)}{(x_i-x_1)(x_i-x_2)\ldots(x_i-x_{i-1})\ \ (x_i-x_{i+1})\ldots(x_i-x_n)}$$

$$= \prod_{j=1, j\neq i}^{j=n} \frac{(x-x_j)}{(x_i-x_j)}$$



Lagrange polynomial with 2 points (x_1, y_1) and (x_2, y_2).

$$y(x) = L_1(x)y_1 + L_2(x)y_2$$
$$= \frac{x-x_2}{x_1-x_2}y_1 + \frac{x-x_1}{x_2-x_1}y_2$$

Interpolation - Cubic spline

Instead of doing this problem in general, let's focus on the problem of interpolating 5 points (x1; y1), (x2; y2), (x3; y3), (x4; y4) and (x5; y5).

We will use four cubic polynomial segments. Each segment i (i = 1; 2; 3; 4) has an equation

$$y_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$$

Interpolation - Cubic spline

$$y_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$$

The four cubic polynomial segments (i = 1; 2; 3; 4) in a matrix form

[A][a,b,c,d constants]=[Vector of y data points]

It is easier to work with re-parametrized x values, where each segment will start at x = 0 and run to x = 1.

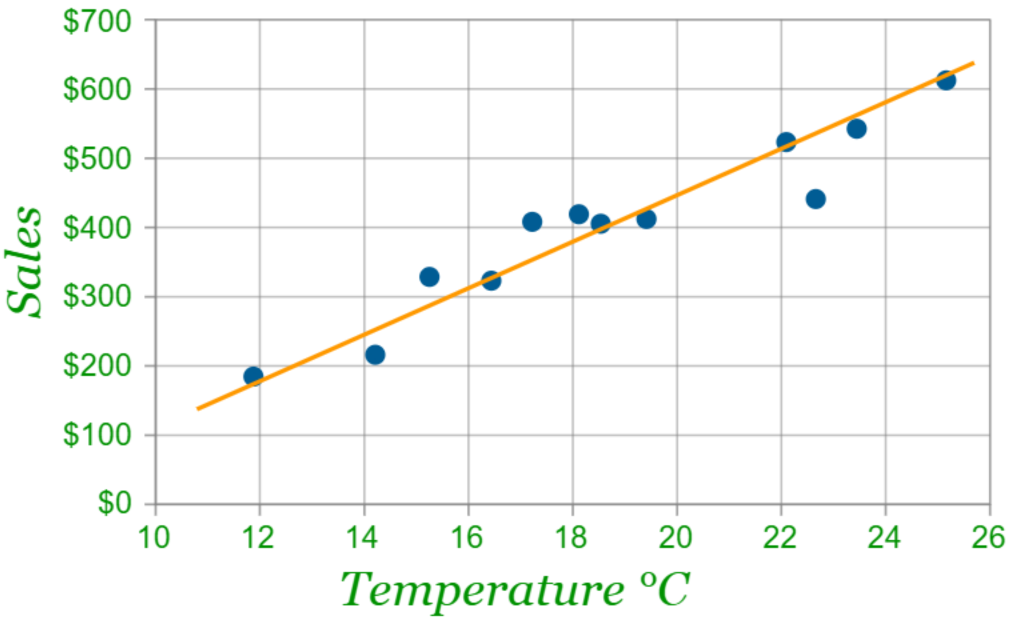
$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & -2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 2 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6
\end{bmatrix}
\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \\ a_3 \\ b_3 \\ c_3 \\ d_3 \\ a_4 \\ b_4 \\ c_4 \\ d_4 \end{bmatrix}
=
\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ 0 \\ 0 \\ \bar{y}_2 \\ \bar{y}_3 \\ 0 \\ 0 \\ \bar{y}_3 \\ \bar{y}_4 \\ 0 \\ 0 \\ \bar{y}_4 \\ \bar{y}_5 \\ 0 \\ 0 \end{bmatrix}$$

Linear Regression

Reminder: Typical equation of a line: y = mx + b

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3)\ldots$, (x_n, y_n), find the equation of the line

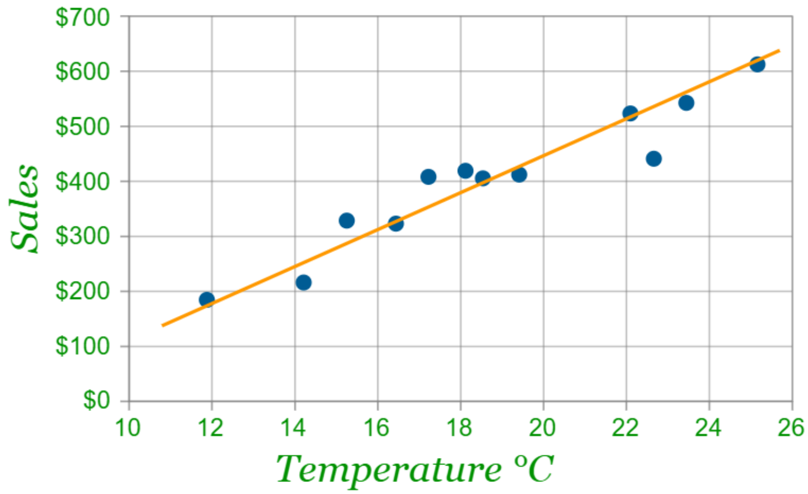
$$y = a_0 + a_1 x$$



https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

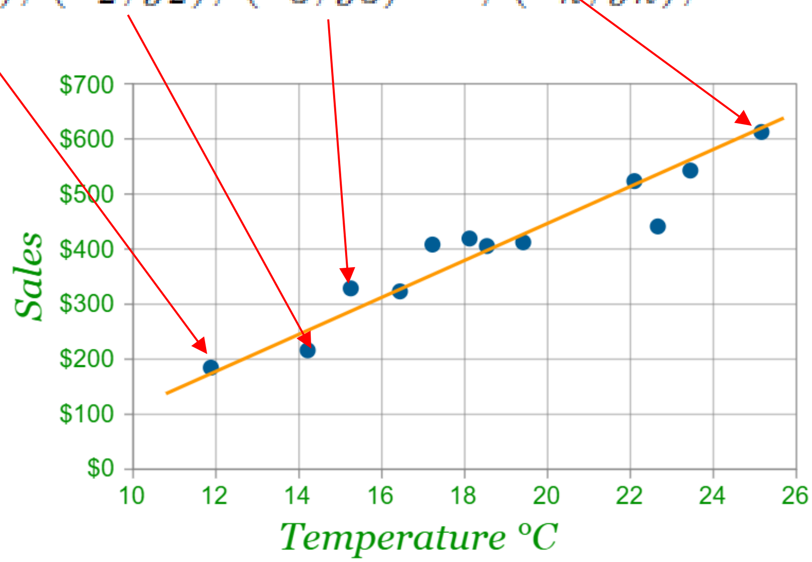
Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots$, (x_n, y_n), find the equation of the line



https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

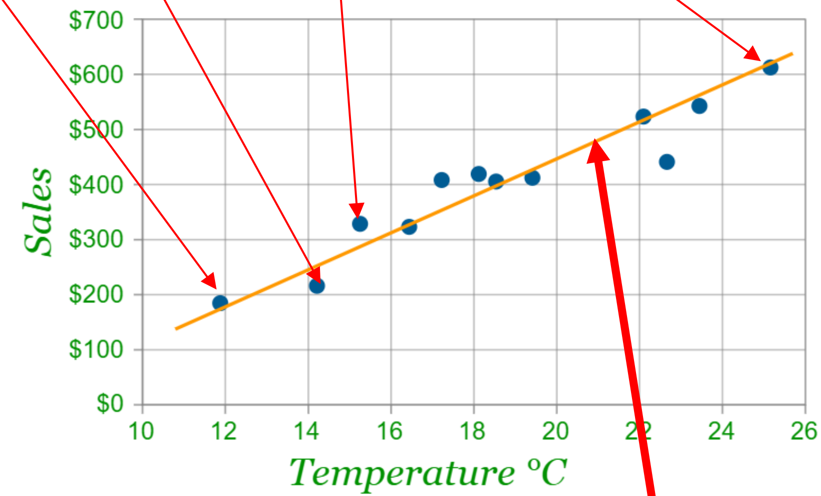
Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots$, (x_n, y_n), find the equation of the line



https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3)\ldots$, (x_n, y_n), find the equation of the line

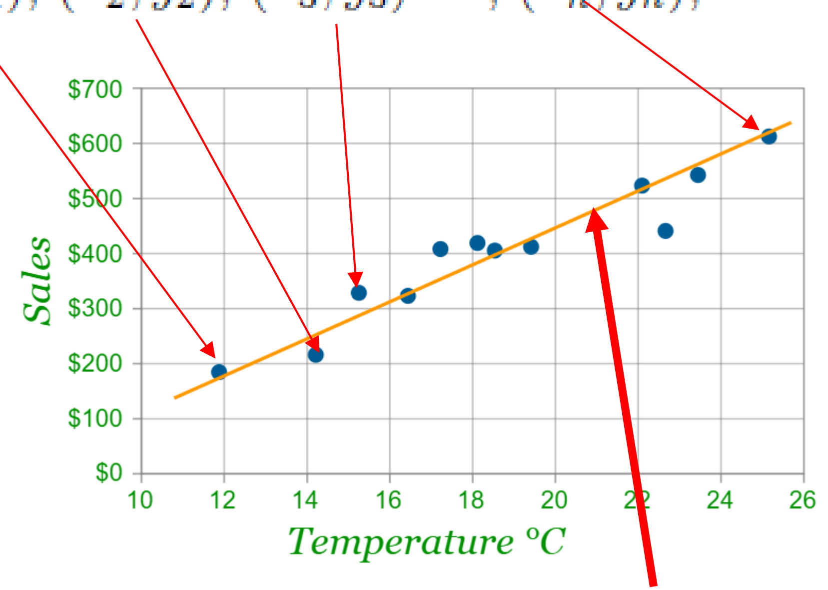


We can only find a line that "best fits" the points: $\hat{y}_i = a_0 + a_1 x$

https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3)\ldots$, (x_n, y_n), find the equation of the line



We can only find a line that “best fits" the points: $\hat{y}_i = a_0 + a_1 x$

The error in prediction for each point is: $E_i = (y_i - \hat{y}_i)$

https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots,$ (x_n, y_n), find the equation of the line

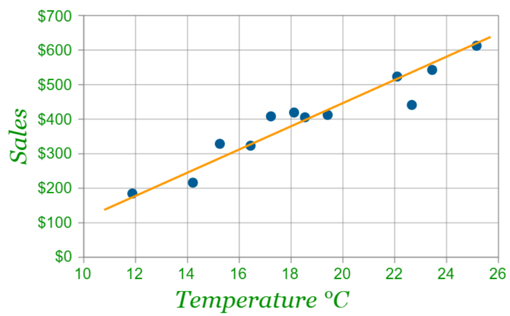
$$E_i = (y_i - \hat{y}_i)$$

We can only find a line that “best fits" the points: $\hat{y}_i = a_0 + a_1 x$

https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots,$ (x_n, y_n), find the equation of the line



We can only find a line that “best fits" the points.

$$\hat{y}_i = a_0 + a_1 x$$

The error in prediction for each point is:

$$E_i = (y_i - \hat{y}_i)$$

To find the total error we can’t just add the errors because positive errors cancel negative errors. Therefore, the total error is given a squared form as:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

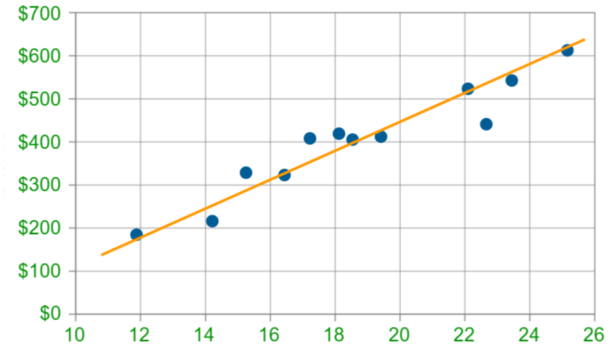
https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3)\ldots$, (x_n, y_n), find the equation of the line

$$\hat{y}_i = a_0 + a_1 x$$

The total error:



$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

https://www.mathsisfun.com/data/least-squares-regression.html

Linear Regression

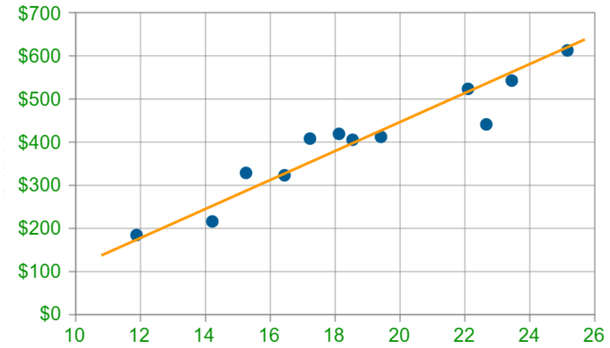
Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots,$ (x_n, y_n), find the equation of the line

$$\hat{y}_i = a_0 + a_1 x$$

The total error:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$



Since we want this error to be minimum, we differentiate the error first with respect to a_0 and then with respect to a_1 and set equal to 0.

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i)(-1) = 0$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i)(-x_i) = 0$$

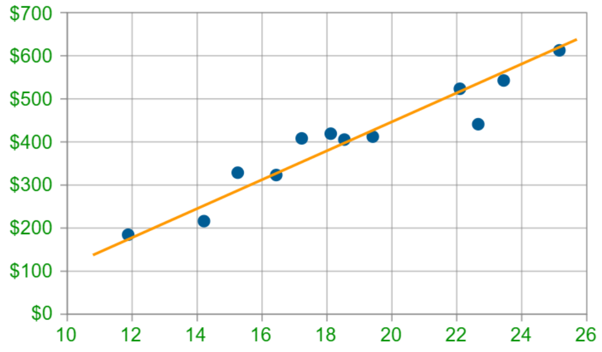
Linear Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3)\ldots$, (x_n, y_n), find the equation of the line

$$\hat{y}_i = a_0 + a_1 x$$

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i)(-1) = 0$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i)(-x_i) = 0$$



Rewriting the equation as a system of linear equations, we get

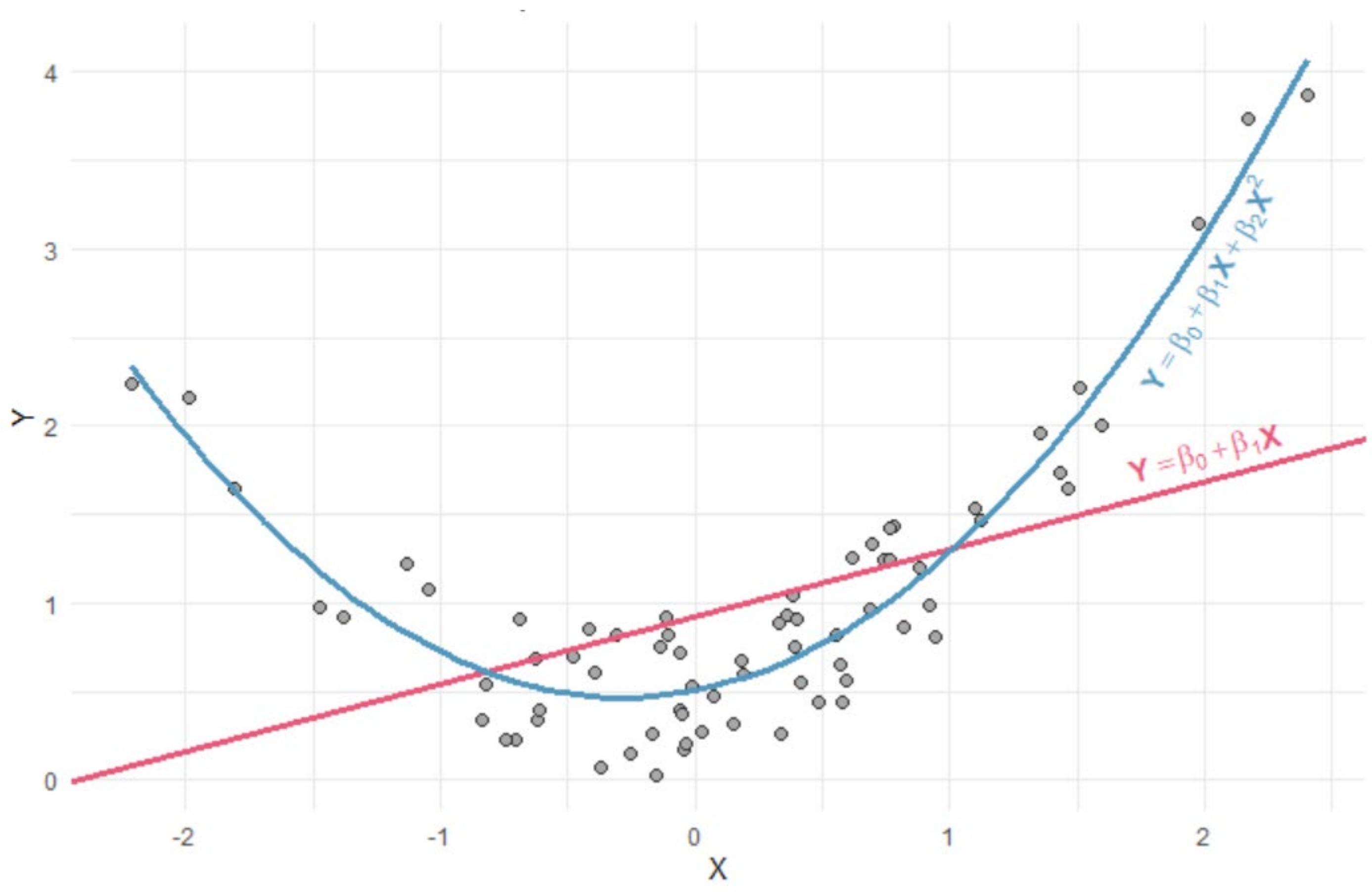
$$(\sum_{i=1}^{n} 1) a_0 + (\sum_{i=1}^{n} x_i) a_1 = \sum_{i=1}^{n} y_i$$

$$(\sum_{i=1}^{n} x_i) a_0 + (\sum_{i=1}^{n} x_i^2) a_1 = \sum_{i=1}^{n} x_i y_i$$

Rearranging as a matrix equation, and since $\sum_{i=1}^{n} 1 = n$,

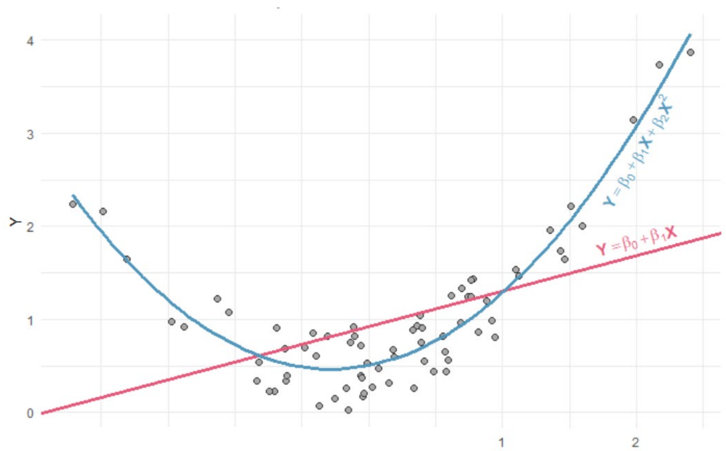
$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}$$

Quadratic Regression



Quadratic Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots$, (x_n, y_n)



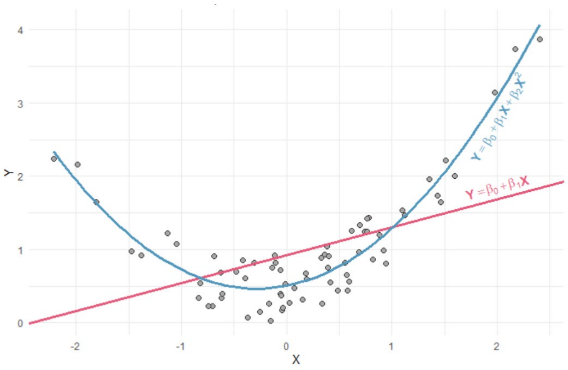
$$y = a_0 + a_1 x + a_2 x^2$$

The procedure is the same as linear regression:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

Quadratic Regression



Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots$, (x_n, y_n)

$$y = a_0 + a_1 x + a_2 x^2$$

The procedure is the same as linear regression:

$$E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-1) = 0$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-x_i) = 0$$

$$\frac{\partial E}{\partial a_2} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-x_i^2) = 0$$

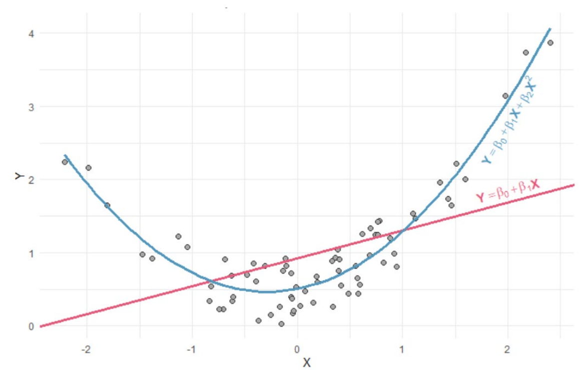
Quadratic Regression

Given points (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots,$ (x_n, y_n)

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-1) = 0$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-x_i) = 0$$

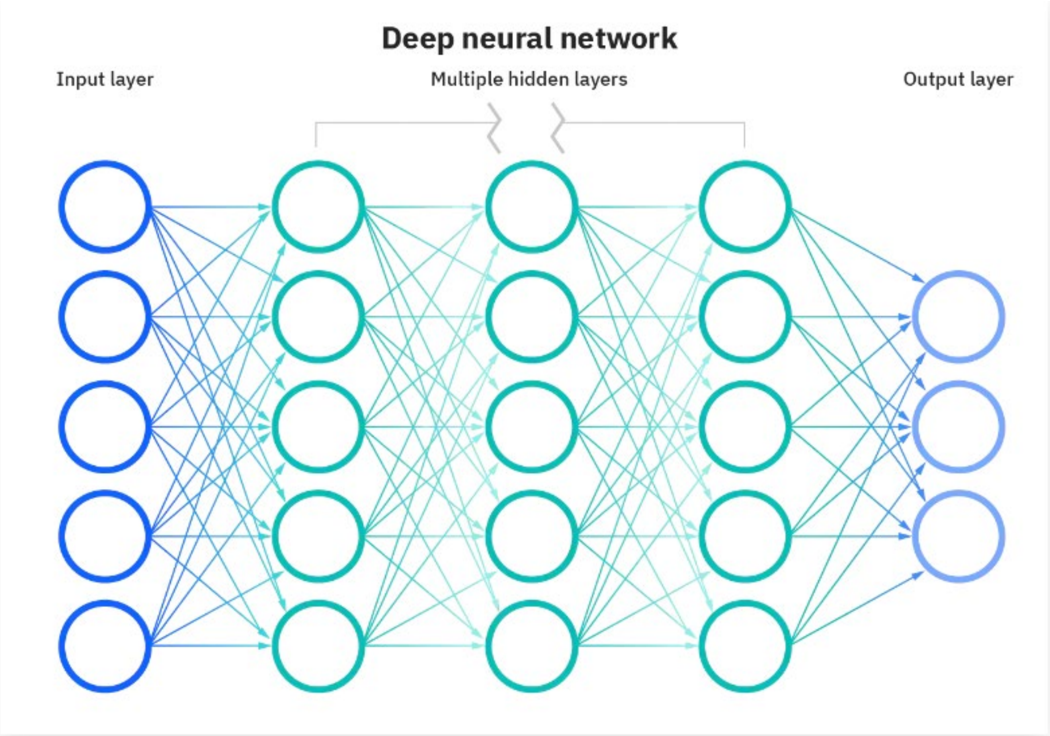
$$\frac{\partial E}{\partial a_2} = \sum_{i=1}^{n} (2)(y_i - a_0 - a_1 x_i - a_2 x_i^2)(-x_i^2) = 0$$



Rearranging in the form of a matrix form:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix}$$

Deep Learning the depth of layers in a neural network



https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

Deep Learning

- I were to show you a series of images of different types of fast food, “pizza,” “burger,” or “taco.” The human expert on these images would determine the characteristics which distinguish each picture as the specific fast food type.
- "Deep" machine learning can leverage labeled datasets, also known as supervised learning, to inform its algorithm, but it doesn’t necessarily require a labeled dataset. It can ingest unstructured data in its raw form (e.g. text, images), and it can automatically determine the set of features which distinguish "pizza", "burger", and "taco" from one another.

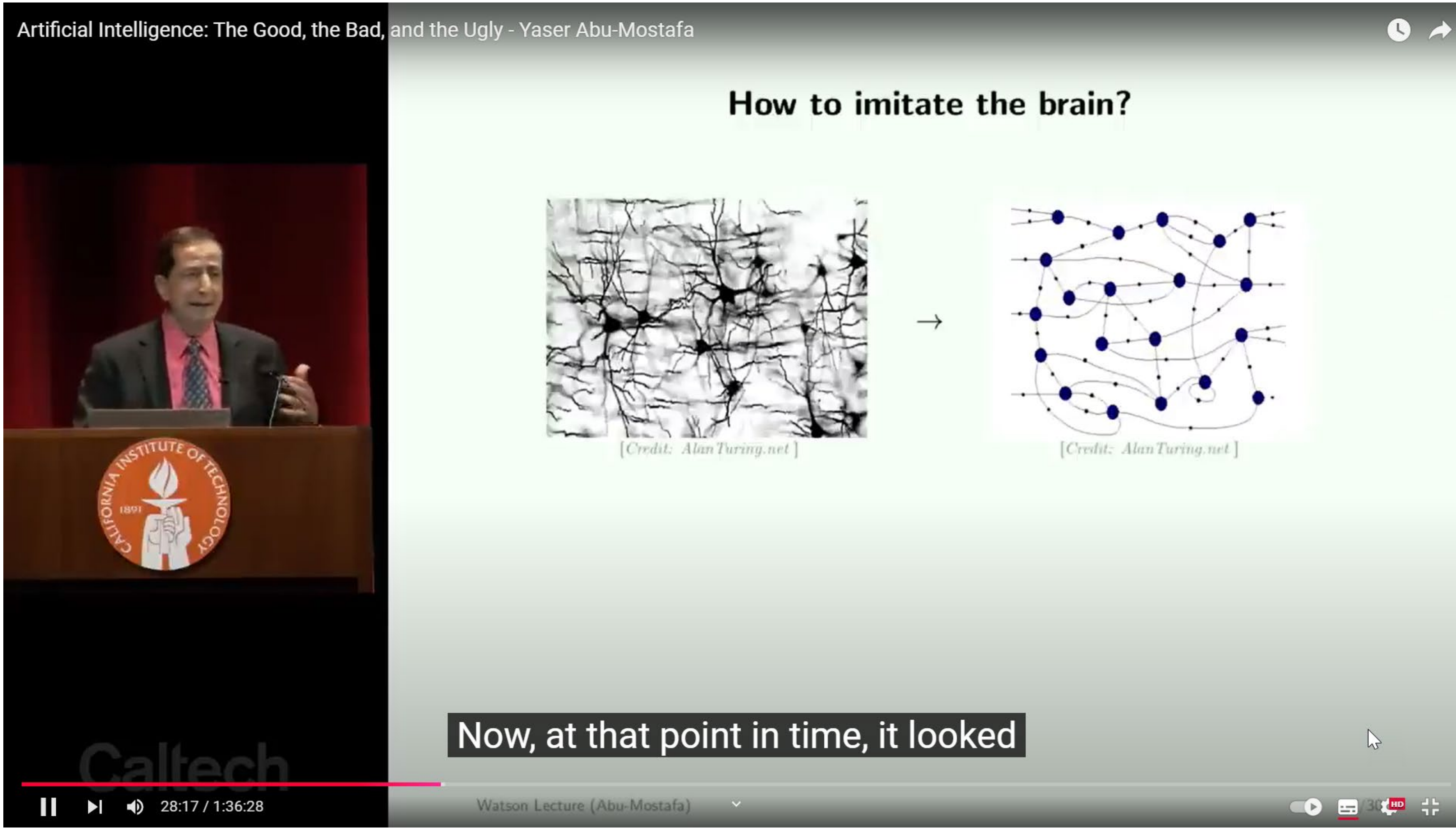
https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

Brute-force approach to Artificial Intelligence

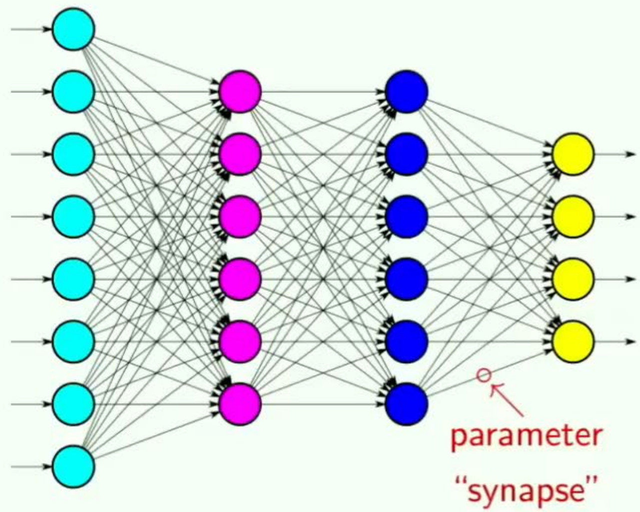
The '**Lookup Table**':

1. Store gigantic amount of information in a computer.
2. Look up the relevant information when someone asks.

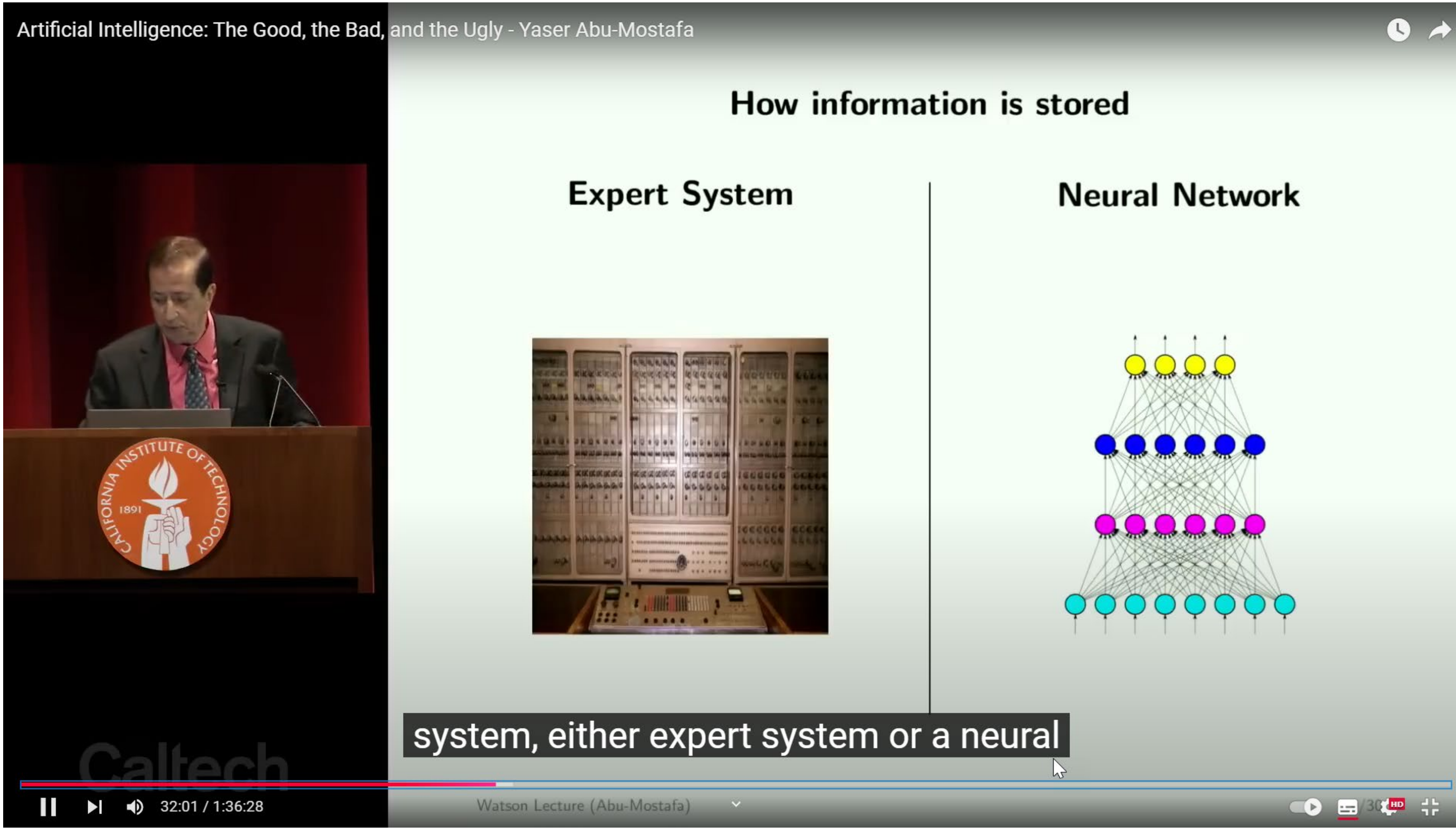
https://www.youtube.com/watch?v=-a61zsRRONc&t=1396s

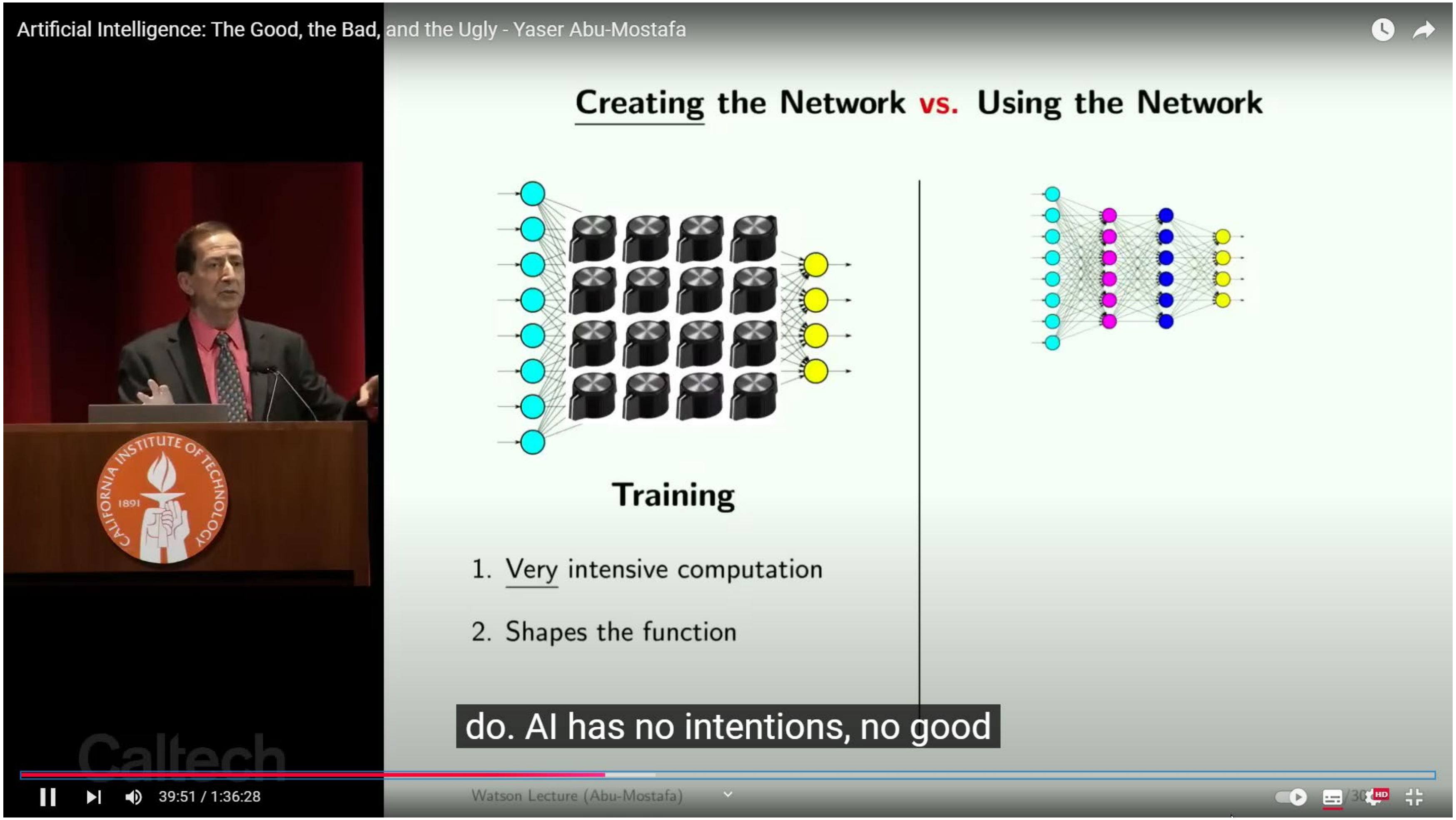

Artificial Intelligence: The Good, the Bad, and the Ugly - Yaser Abu-Mostafa
How to imitate the brain?
[Credit: AlanTuring.net]
[Credit: AlanTuring.net]
Now, at that point in time, it looked
Caltech
28:17 / 1:36:28
Watson Lecture (Abu-Mostafa)

The Neural Network

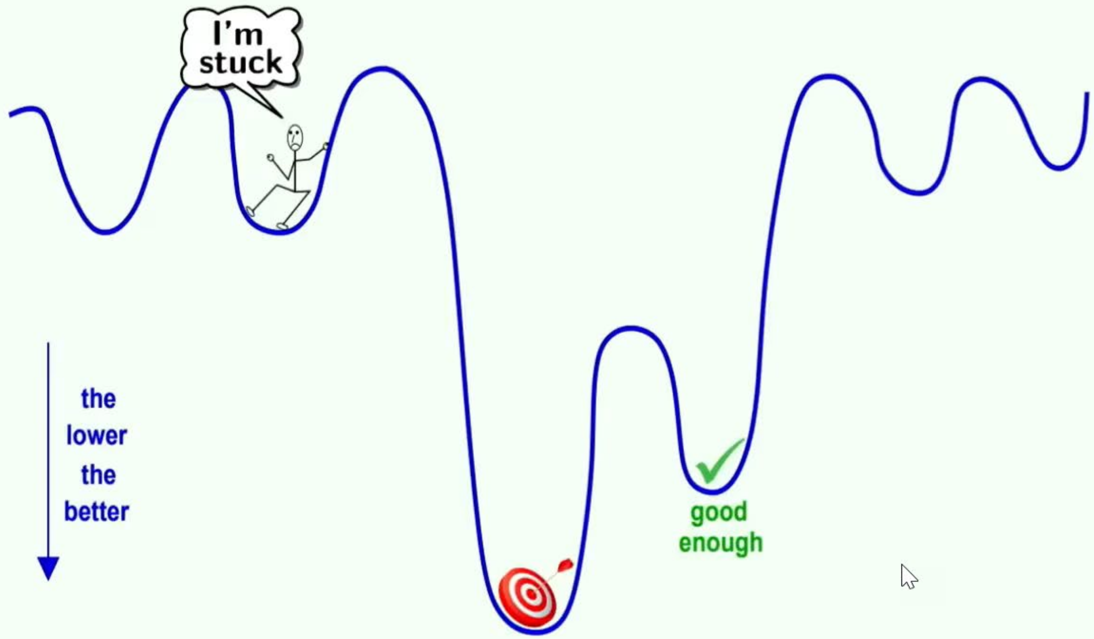


[Credit: *Hi Clip Art*]

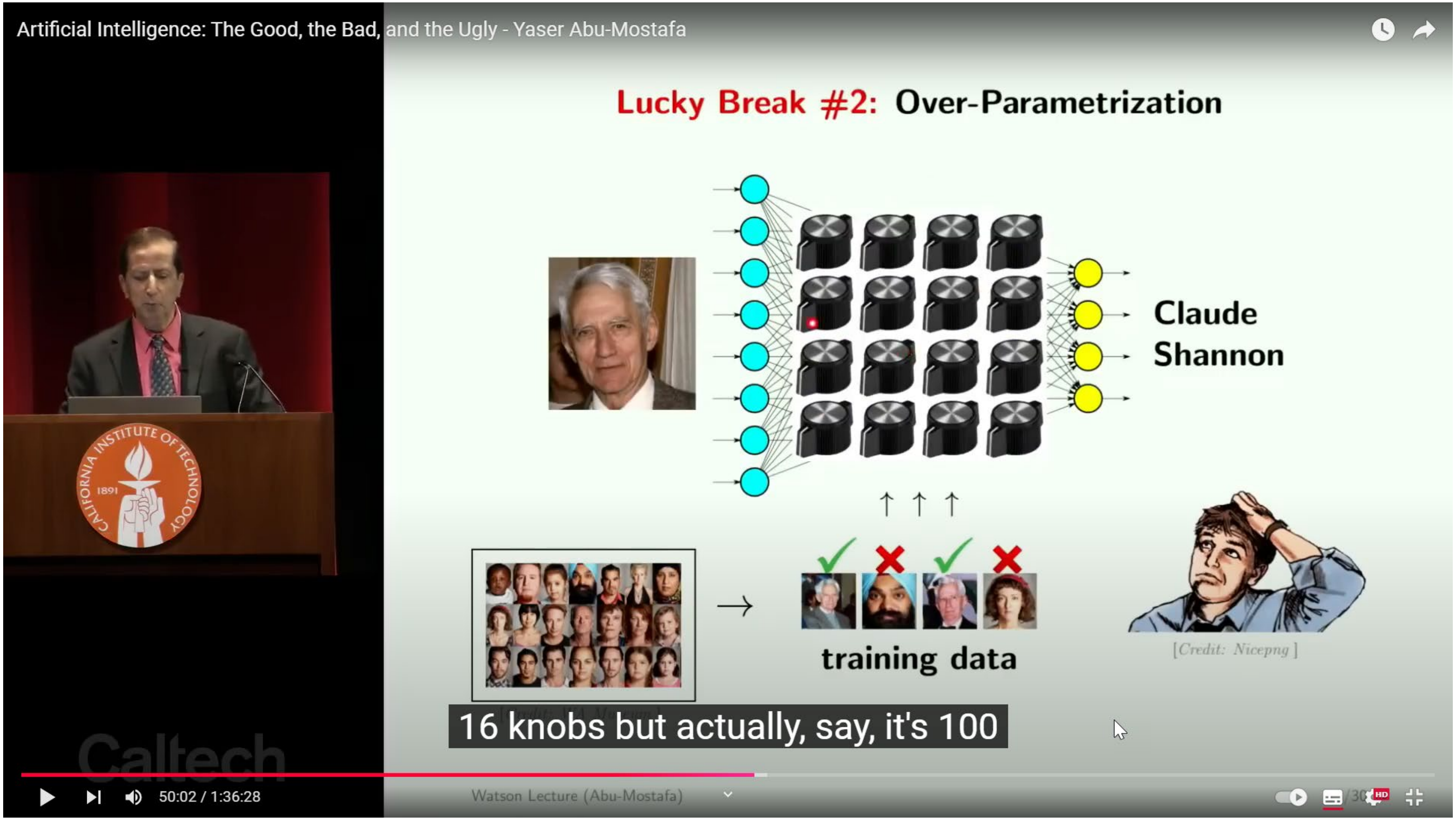

Artificial Intelligence: The Good, the Bad, and the Ugly - Yaser Abu-Mostafa
How information is stored
Expert System
Neural Network
CALIFORNIA INSTITUTE OF TECHNOLOGY
1891
system, either expert system or a neural
Caltech
32:01 / 1:36:28
Watson Lecture (Abu-Mostafa)


Artificial Intelligence: The Good, the Bad, and the Ugly - Yaser Abu-Mostafa
Creating the Network vs. Using the Network
Training
1. Very intensive computation
2. Shapes the function
do. AI has no intentions, no good
CALIFORNIA INSTITUTE OF TECHNOLOGY
1891
Caltech
39:51 / 1:36:28
Watson Lecture (Abu-Mostafa)

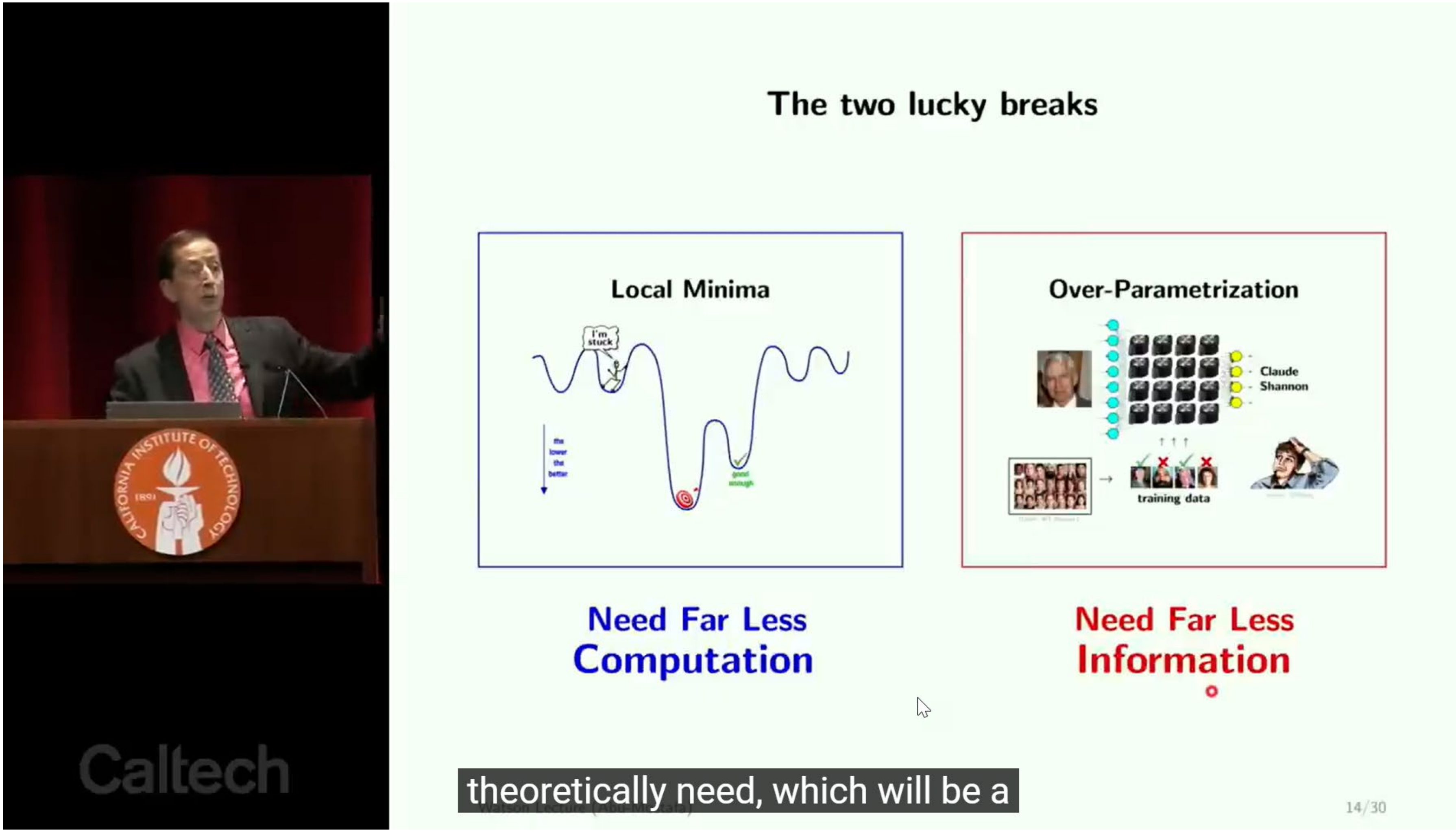
Lucky Break #1: Local Minima



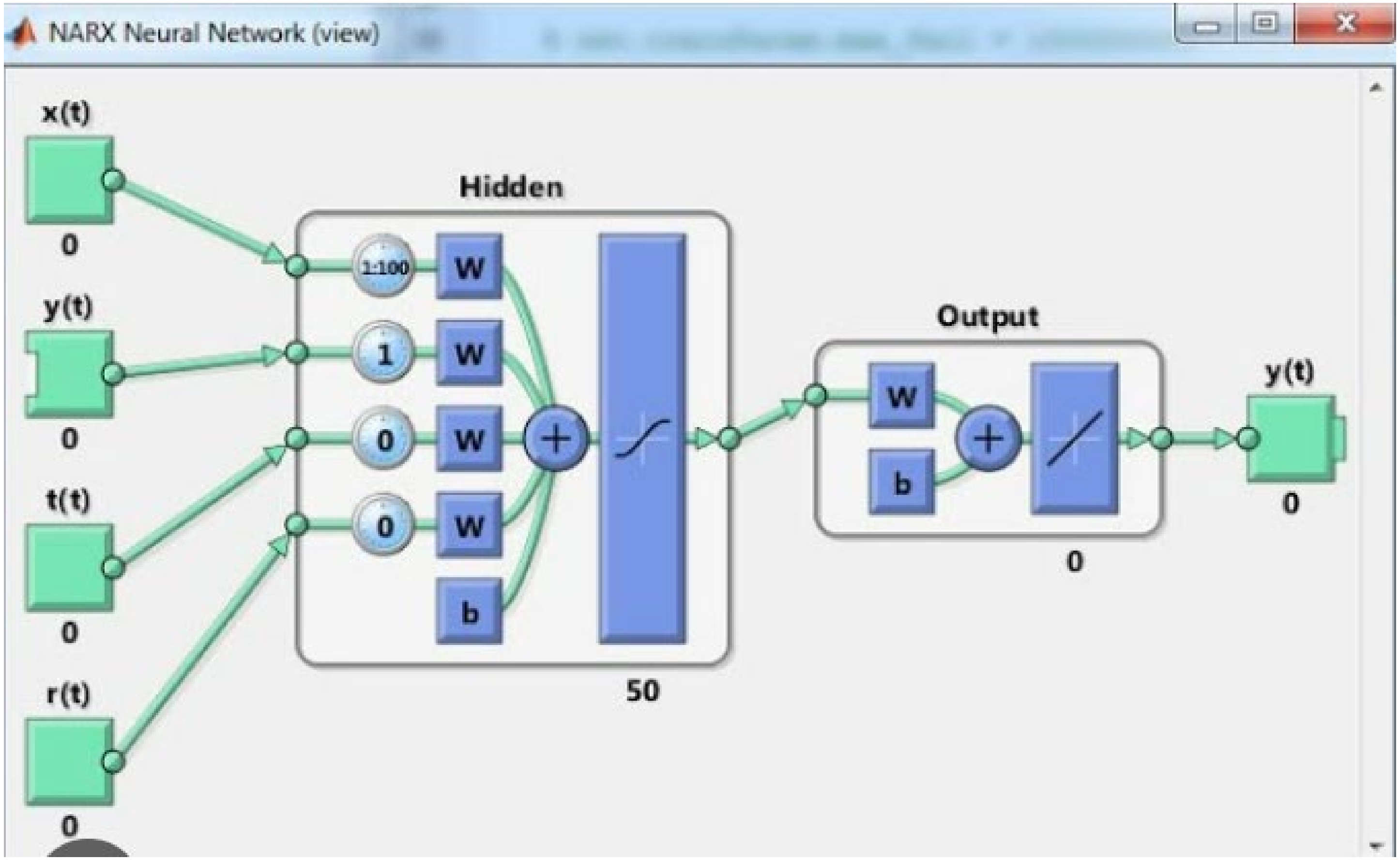
Now, there are so many of these shallow



https://www.youtube.com/watch?v=-a61zsRRONc&t=1396s


The two lucky breaks
Local Minima
I'm stuck
the lower the better
good enough
Over-Parametrization
Claude Shannon
training data
Need Far Less Computation
Need Far Less Information
14/30
CALIFORNIA INSTITUTE OF TECHNOLOGY
1891
Caltech
theoretically need, which will be a

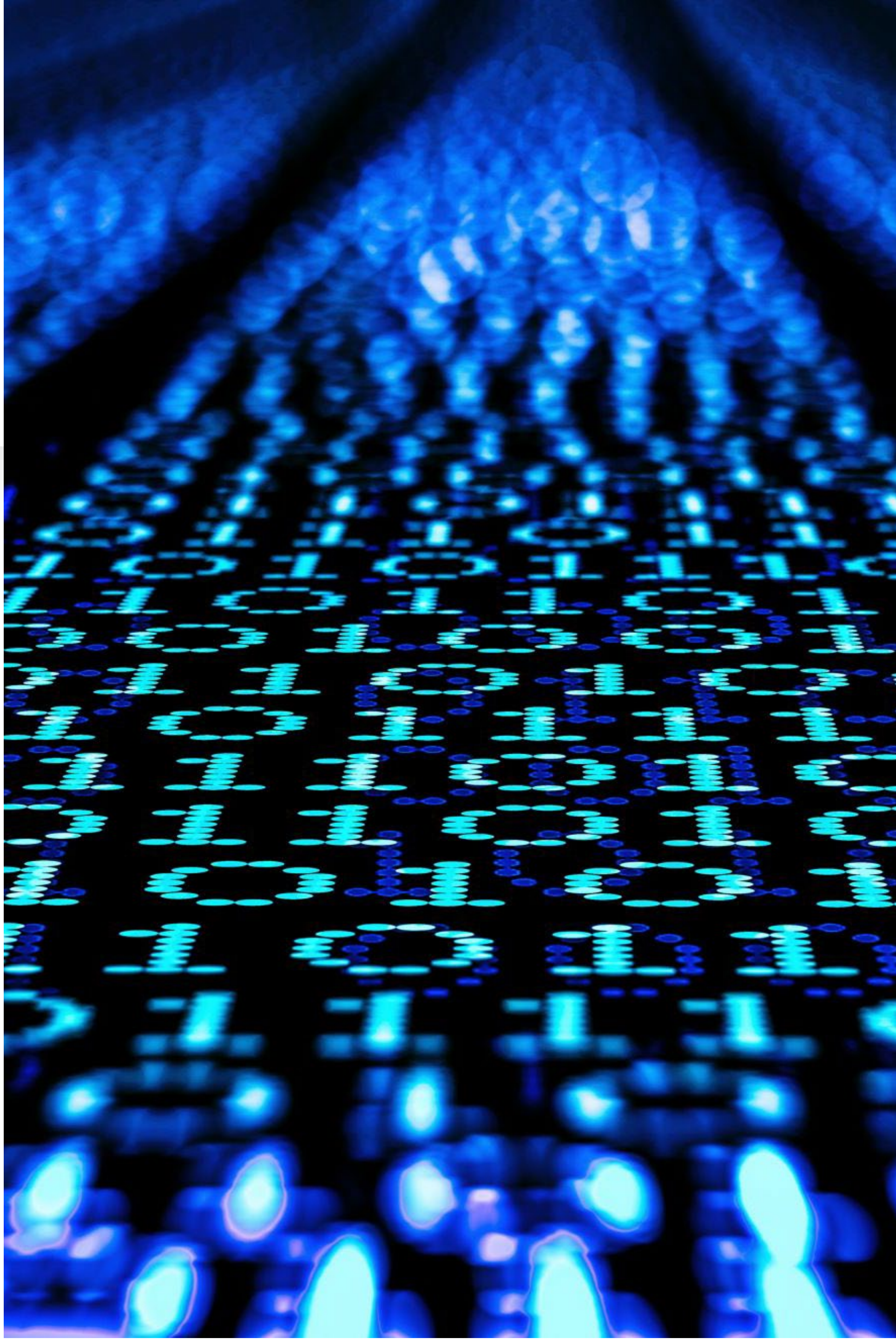
MATLAB ANN



https://www.mathworks.com/videos/getting-started-with-neural-networks-using-matlab-1591081815576.html

Introduction to Fuzzy Logic

- Fuzzy Logic was introduced in 1960s by Prof. Lotfi A. Zadeh.
- Fuzzy Logic has been applied to many fields, from control theory to artificial intelligence.
- Fuzzy Logic allows using approximate values in control as well as incomplete or ambiguous data (fuzzy data) rather than crisp data (binary yes/no choices).



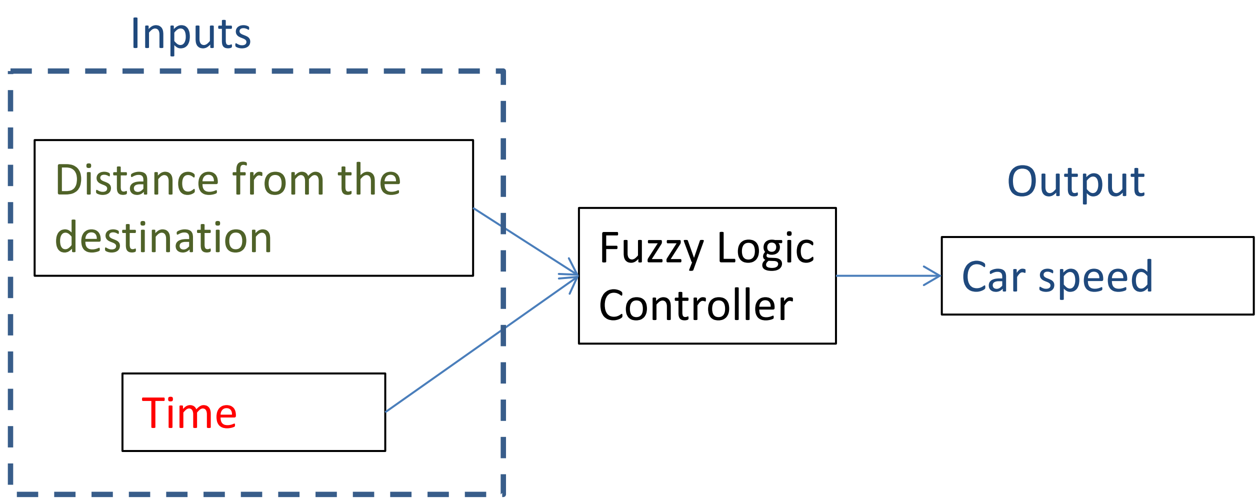
Introduction to Fuzzy Logic

- Fuzzy Logic uses linguistic terms rather than numerical values such as small, large, warm, cold, hot, far, near, etc.
- Fuzzy sets use rules to define the control systems. The rules are similar to human decisions.
- Fuzzy Logic is particularly advantageous for systems where we don’t have the mathematical model of the system to be used for control. Also, if the system is complex and human expert decision can help in implementing the control system.

Fuzzy Logic - Example

Control the speed of a car using fuzzy logic.

Problem: Control the speed of the car to arrive at a destination on time:

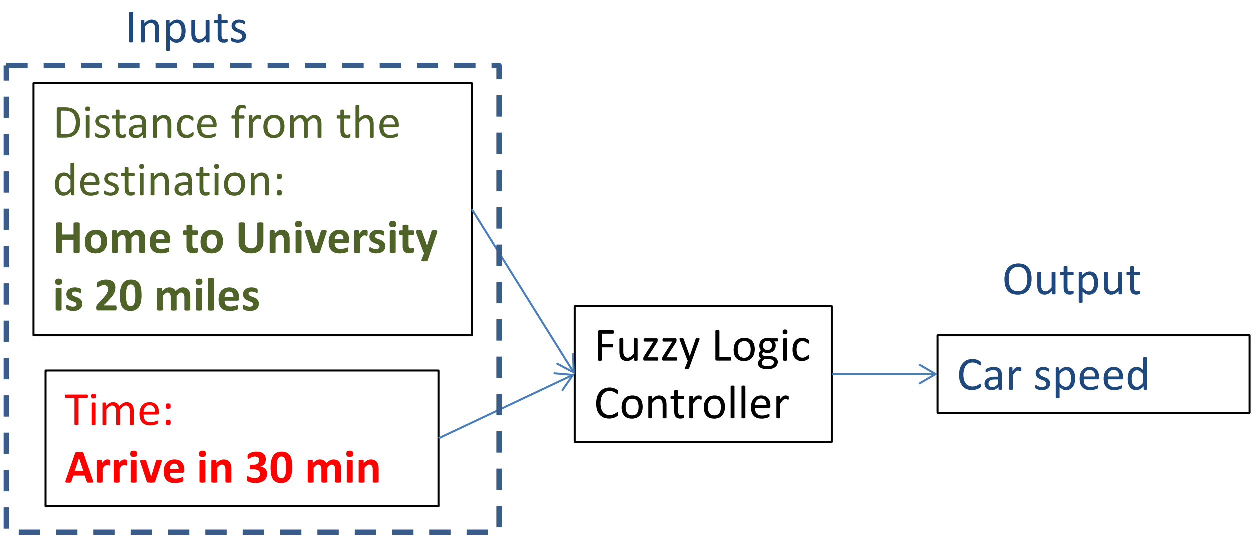


Fuzzy Control rules:

1) If the distance is far, and time is short, then increase speed.
2) If the distance is near, and time is long, then reduce speed.
3) If the distance is far, and time is long, then keep the speed constant.
4) If the distance is near, and time is fine, then keep the speed constant.

Fuzzy Logic - Example

Problem: Arrive from Home to University in 30 Minutes :



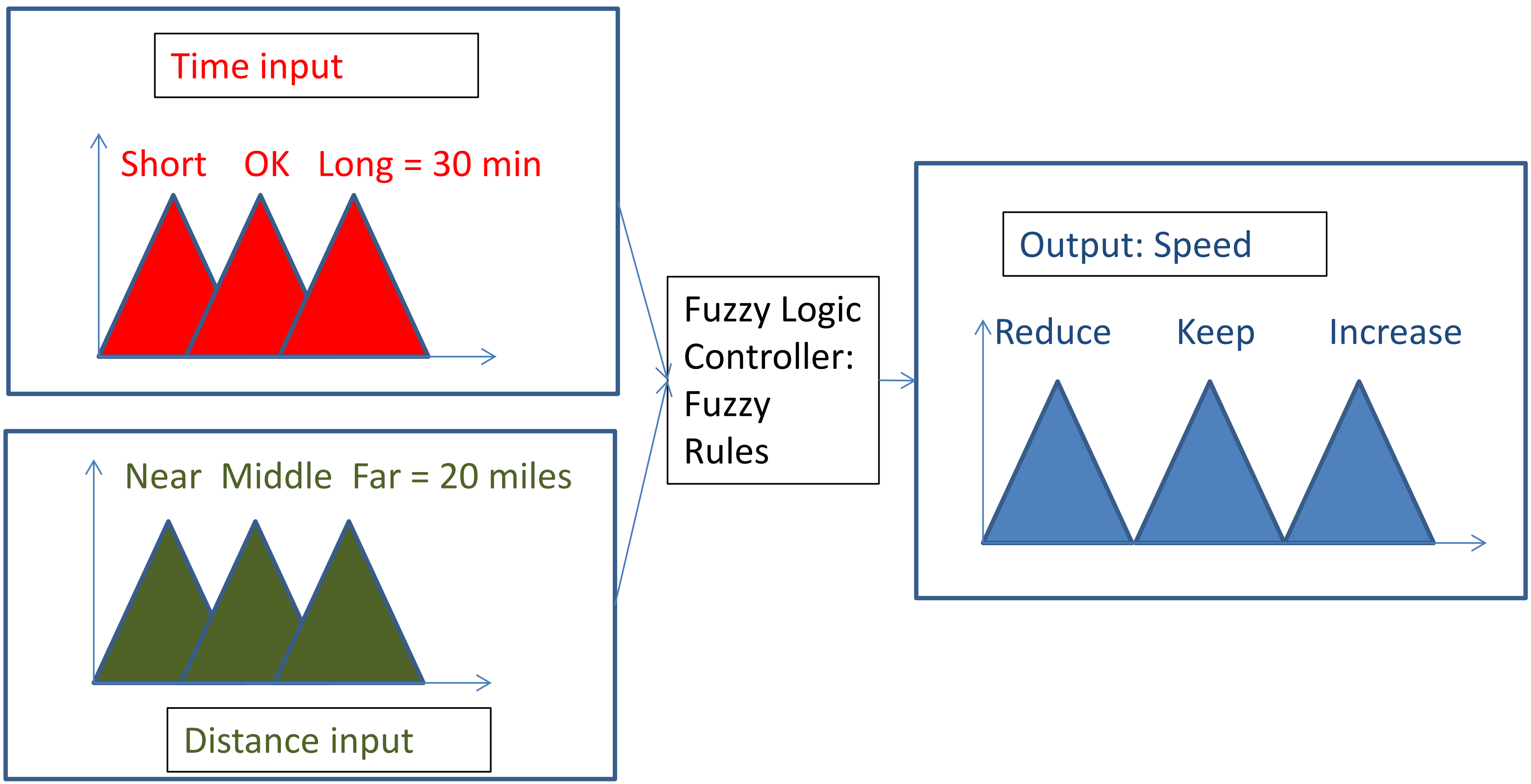
Fuzzy Control rules:

1) If the distance is far, and time is short, then increase speed.
2) If the distance is near, and time is long, then reduce speed.
3) If the distance is far, and time is long, then keep the speed constant.
4) If the distance is near, and time is ok, then keep the speed constant.

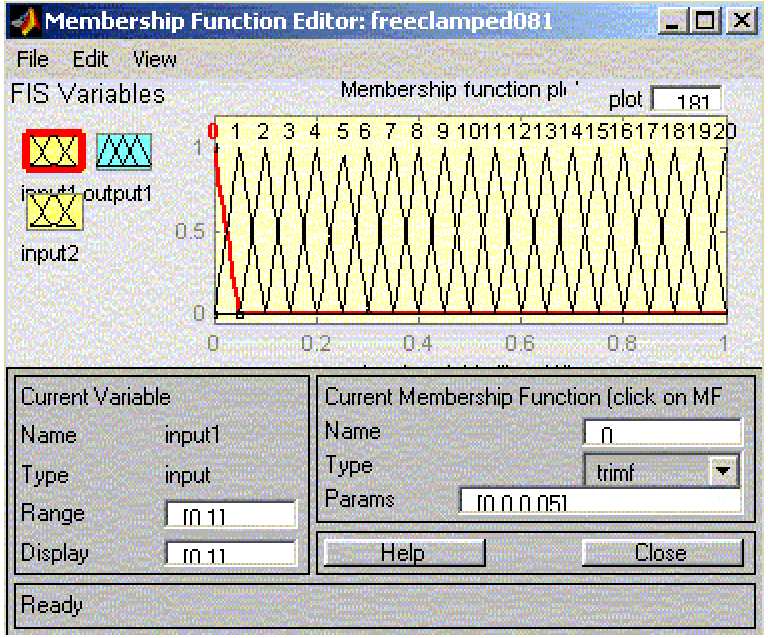
Fuzzy Sets

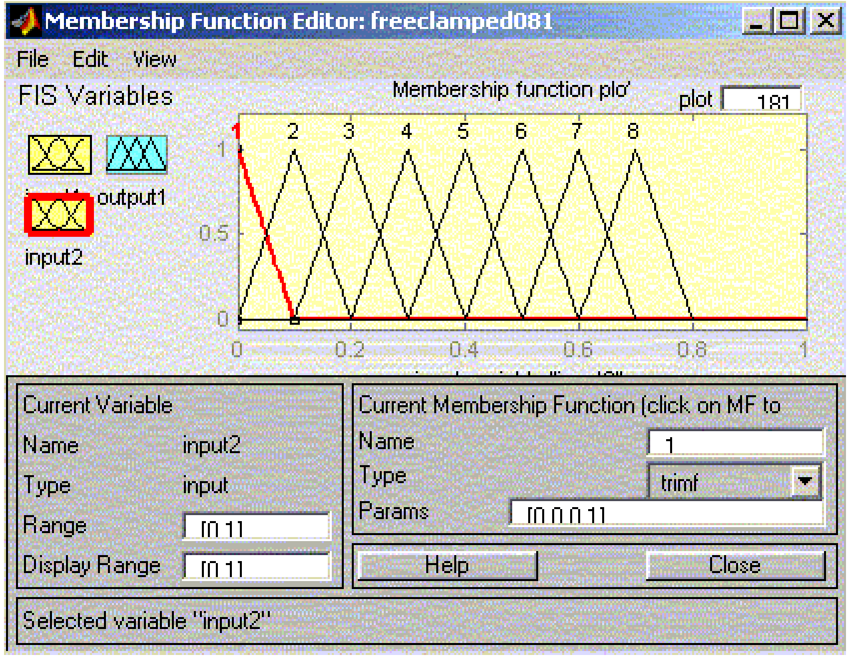
Fuzzy Control rules:

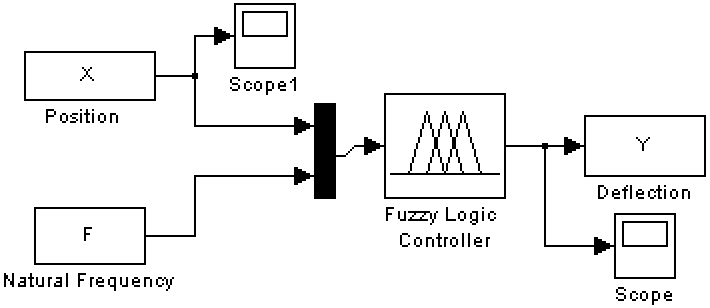
1) If the distance is far, and time is short, then increase speed.
2) If the distance is near, and time is long, then reduce speed.
3) If the distance is far, and time is long, then keep the speed constant.
4) If the distance is near, and time is fine, then keep the speed constant.



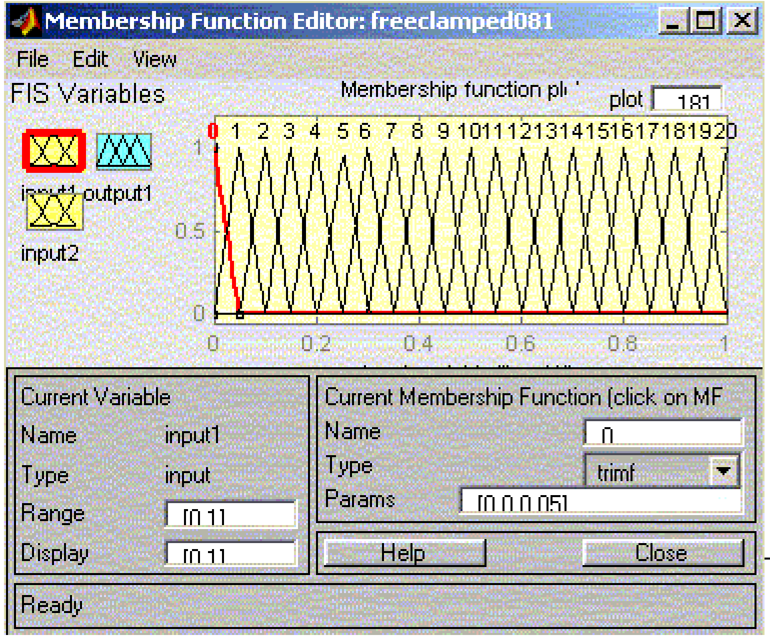
SIMULINK fuzzy controller in obtaining deflections (output) from the inputs

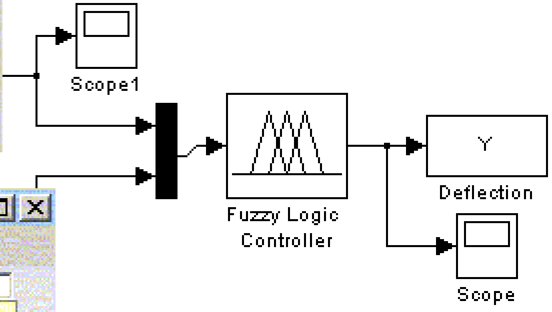


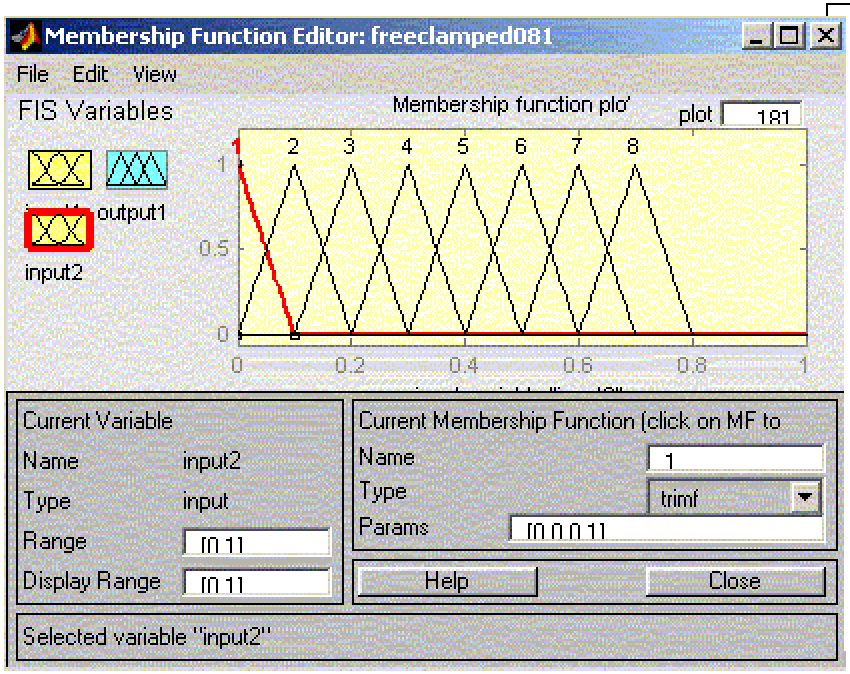


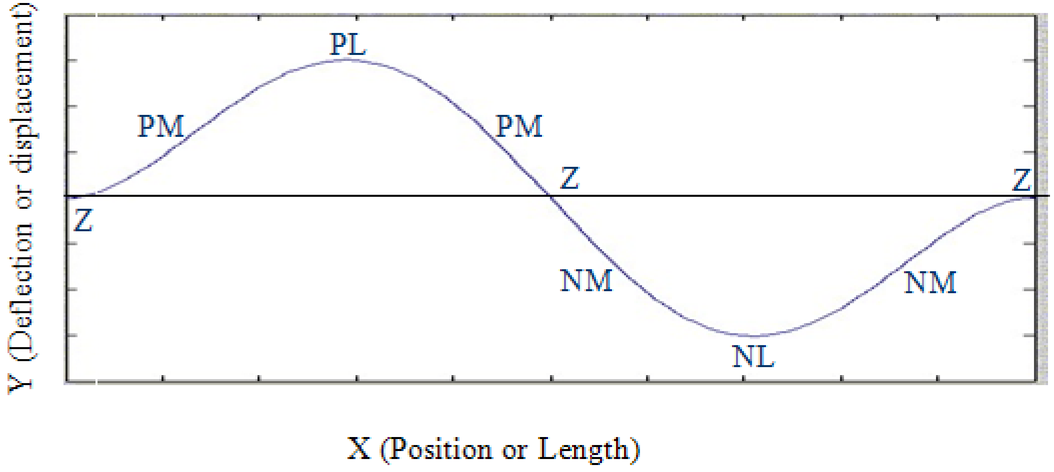


SIMULINK fuzzy controller in obtaining deflections (output) from the inputs

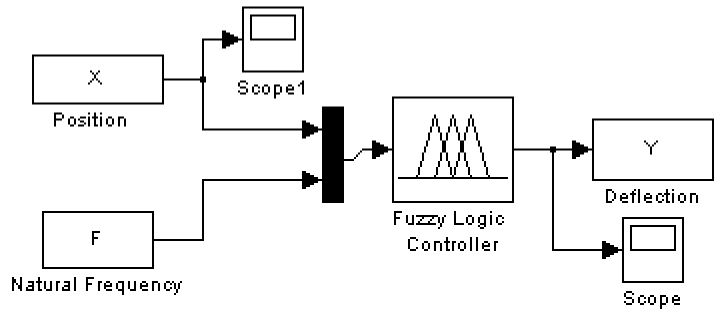


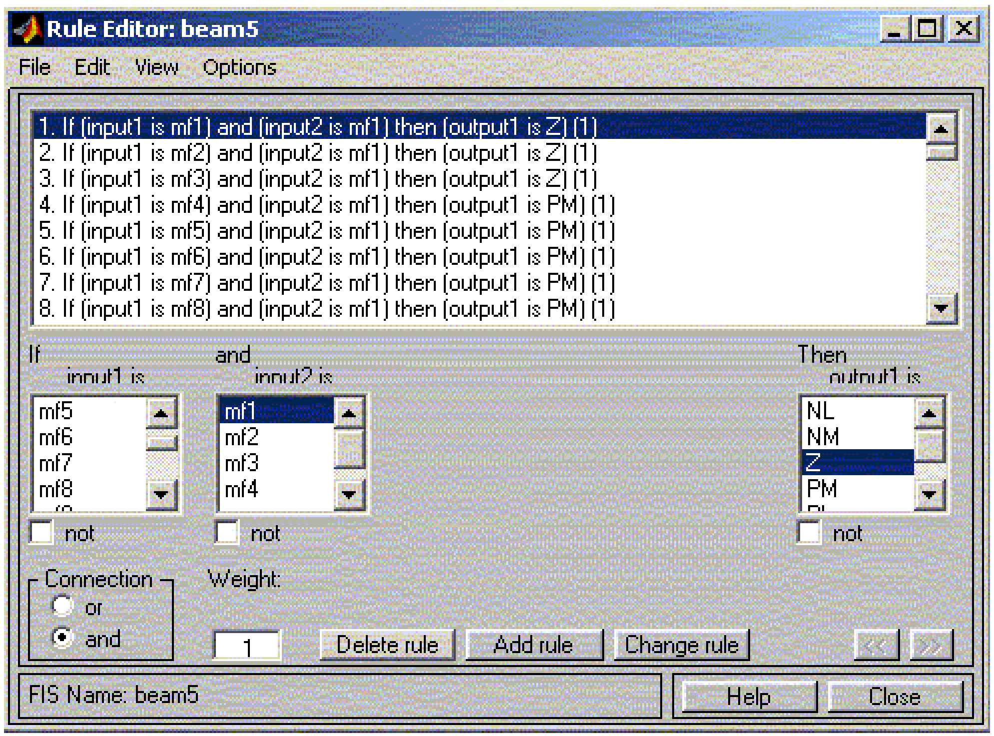


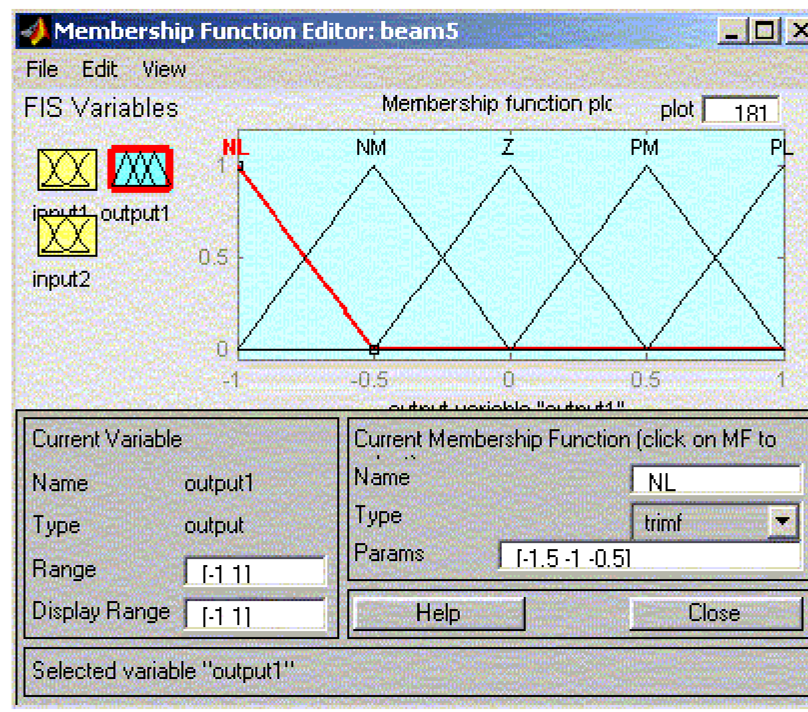




SIMULINK fuzzy controller in obtaining deflections (output) from the inputs







An AI Generated Image