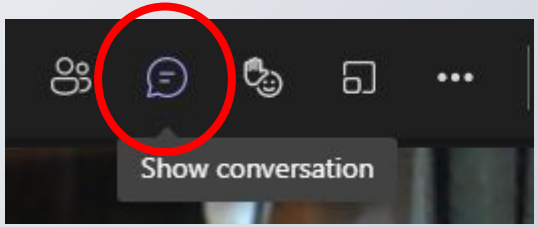
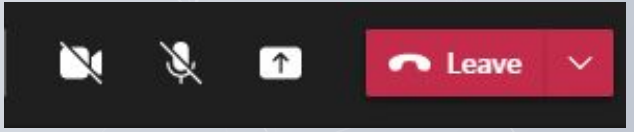
While we wait to get started ...

Open the chat



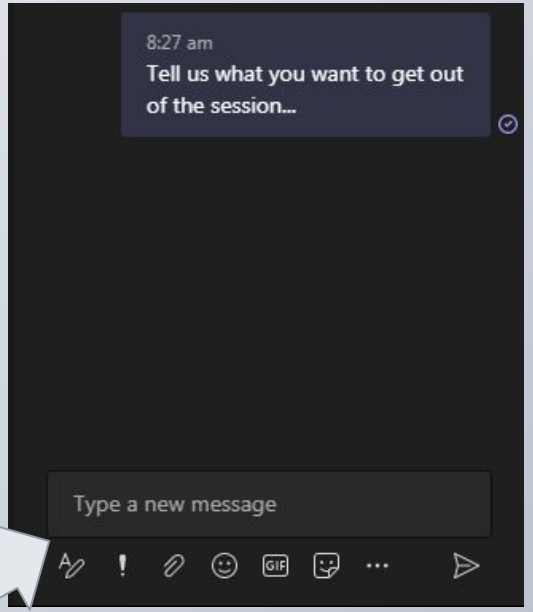


Your mic is on mute
... and camera disbaled

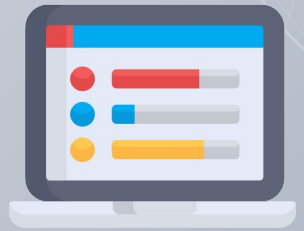
Tell us what you want to get out of the session.

NOTE:
your name will appear with your comment.

The chat won't be part of the recorded version.



Take our poll ... you will find it in the chat...



DIGITAL TECHNOLOGIES HUB

Discovering **Artificial Intelligence (AI)**

Investigating training an AI model

DIGITAL TECHNOLOGIES HUB

Acknowledgement of Country

ESA acknowledges the Eastern Kulin Nation, Traditional Custodians of the land on which our head office stands, and pays our respects to Elders past and present.

We recognise the Traditional Custodians of Country across Australia and their continuing connection and contribution to lands, waters, communities and learning

By the end of this session

Explore the **binary data** that is inputted into an AI system.

Try/observe building a Python program to:

- perform a **classification** with AI.
- perform a **regression** (line of best fit) with an AI.

Discover more about how an artificial neural network works.

Achievement standards:

Achievement Standard

By the end of Year 6, students expla
digital system components (hardwa
and how digital systems are conne
explain how digital systems use wh
representing a variety of data types

Students define problems in terms
requirements and design solutions
address the problems. They incorp
repetition and user interface desig
implement their digital solutions, i
They explain how information sys
needs and consider sustainability
creation and communication of id
collaborative digital projects using
protocols.

Achievement Standard

By the end of Year 8, students distingu
types of networks and defined purpos
image and audio data can be represe
presented in digital systems.

Students plan and manage digital pro
information. They define and decomp
functional requirements and constrai
experiences and algorithms incorpor
iterations, and test, modify and imple
They evaluate information systems a
of meeting needs, innovation and su
and evaluate data from a range of s
solutions. They use appropriate prot
and collaborating online.

Achievement Standard

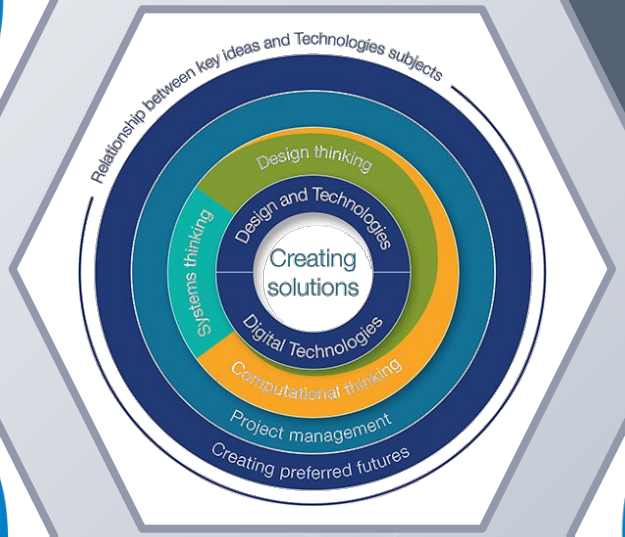
By the end of Year 10, students explain the control and management of networked digital systems and the security implications of the interaction between hardware, software and users. They explain simple data compression, and why content data are separated from presentation.

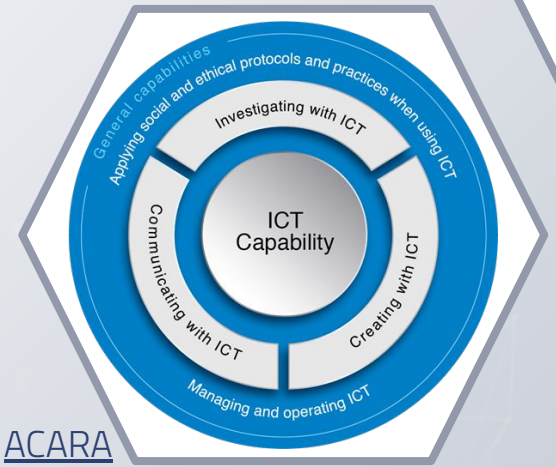
Students plan and manage digital projects using an iterative approach. They define and decompose complex problems in terms of functional and non-functional requirements. Students design and evaluate user experiences and algorithms. They design and implement modular programs, including an object-oriented program, using algorithms and data structures involving modular functions that reflect the relationships of real-world data and data entities. They take account of privacy and security requirements when selecting and validating data. Students test and predict results and implement digital solutions. They evaluate information systems and their solutions in terms of risk, sustainability and potential for innovation and enterprise. They share and collaborate online, establishing protocols for the use, transmission and maintenance of data and projects.

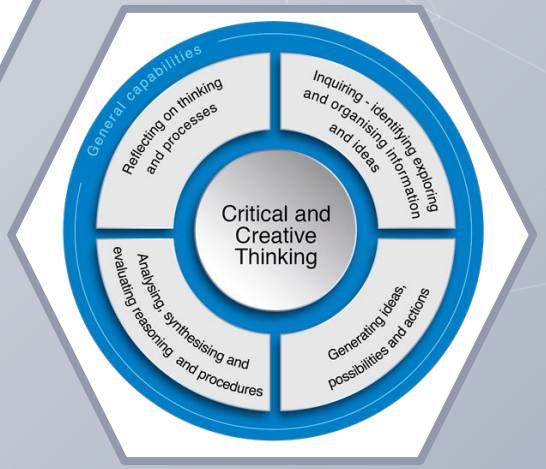
Source: ACARA

Australian CURRICULUM

AI topics

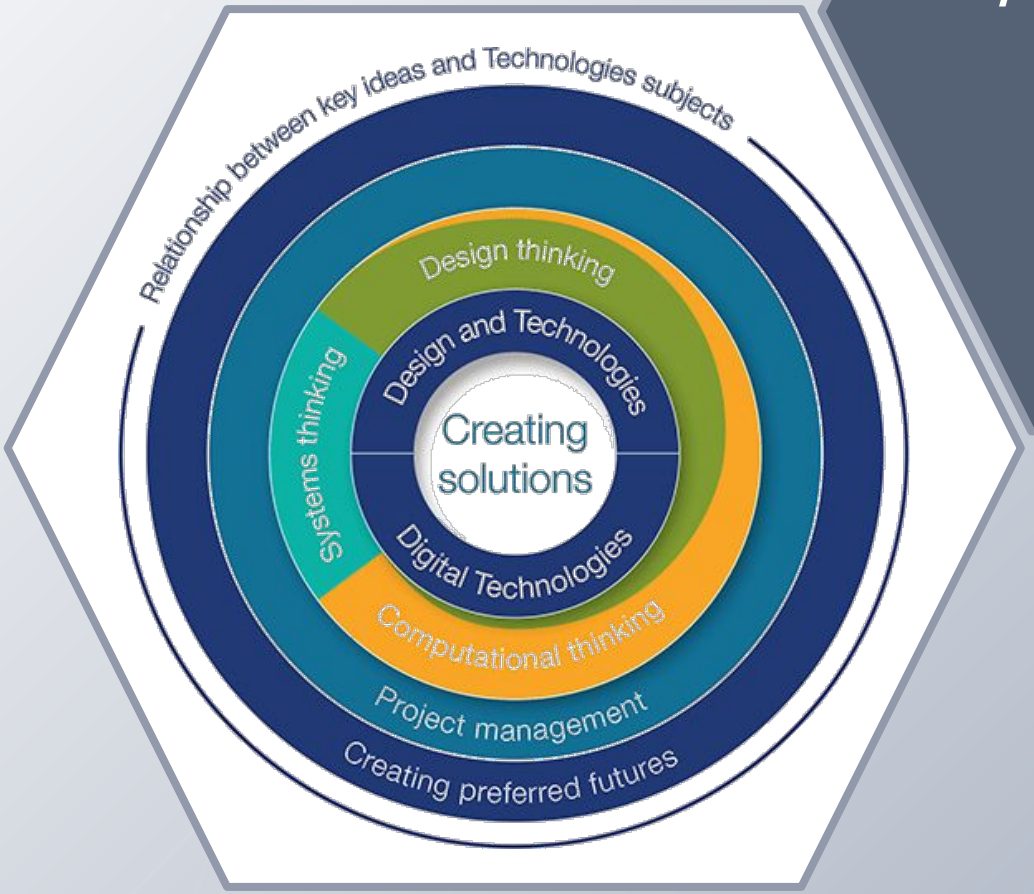




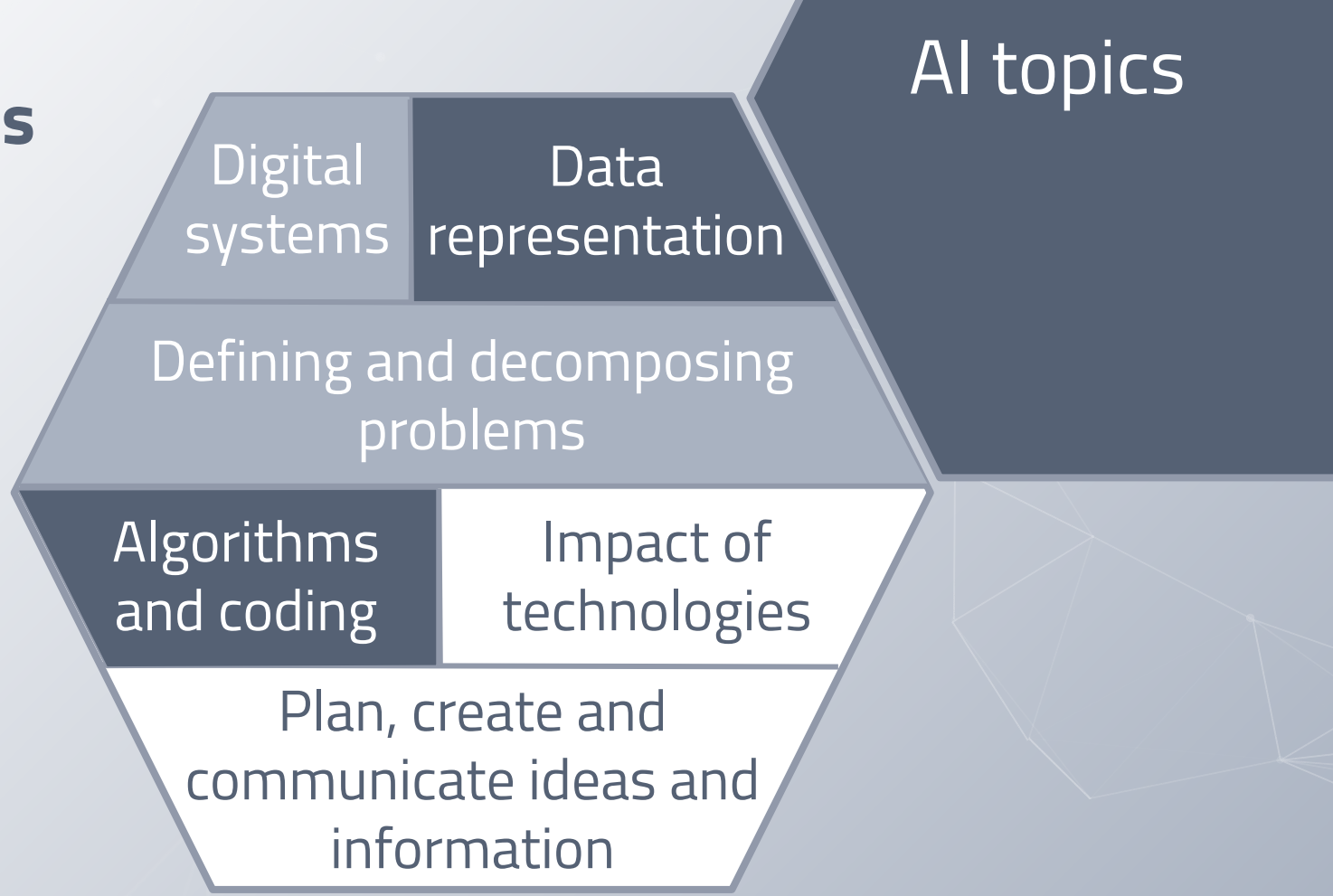


Source: ACARA

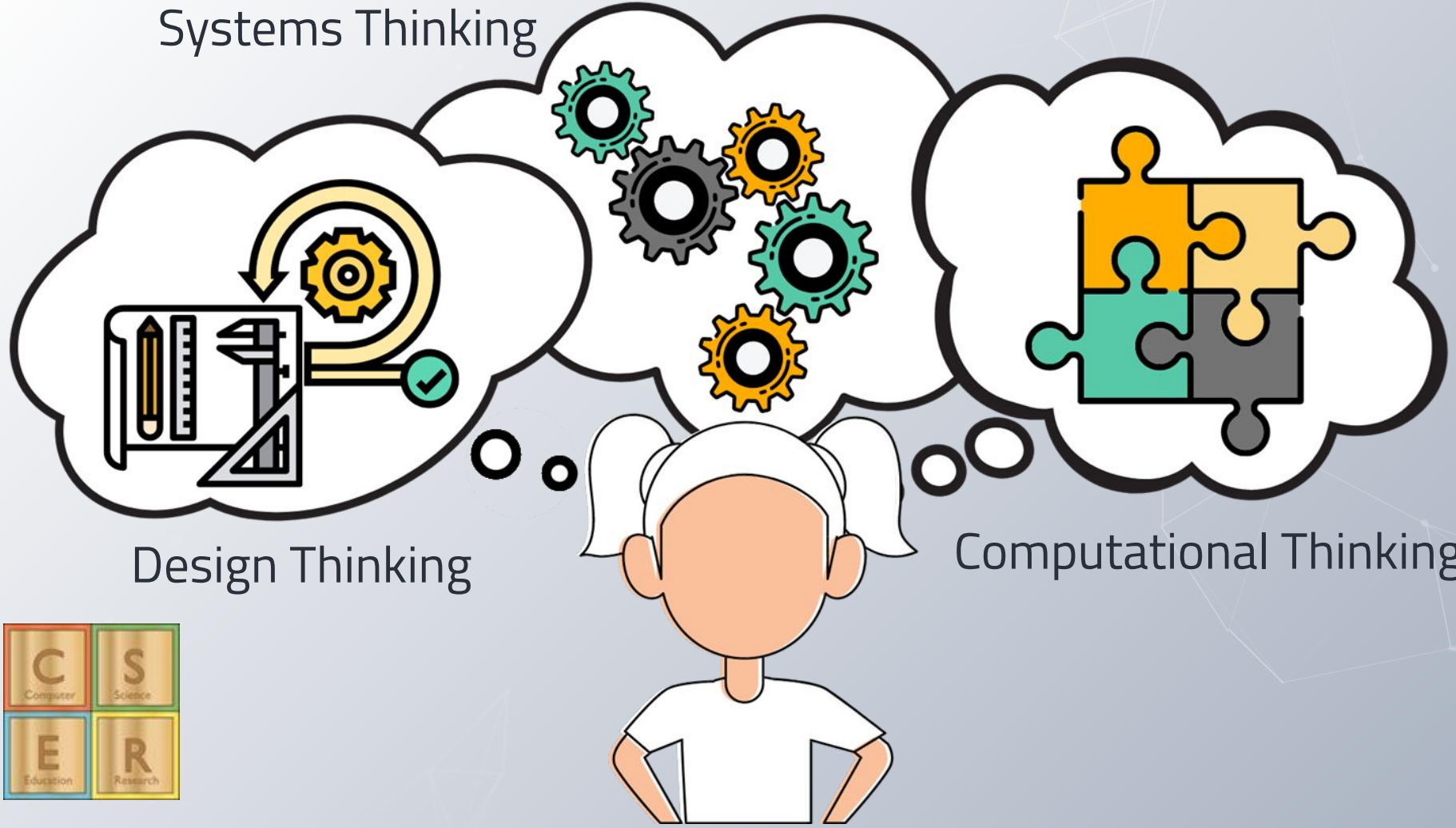
AI topics


Relationship between key ideas and Technologies subjects
Design thinking
Design and Technologies
Creating solutions
Digital Technologies
Systems thinking
Computational thinking
Project management
Creating preferred futures

Foci for this deep dive:



Systems Thinking



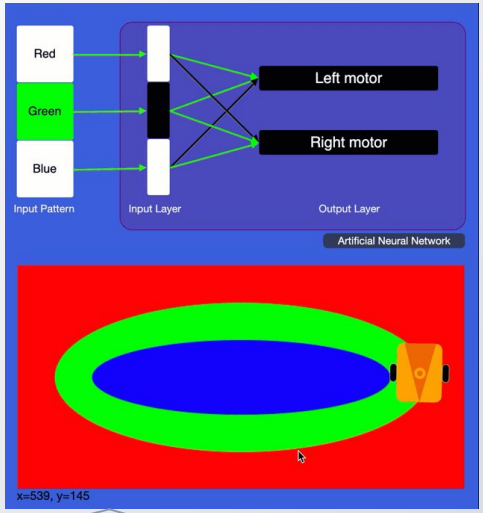
Design Thinking

Computational Thinking

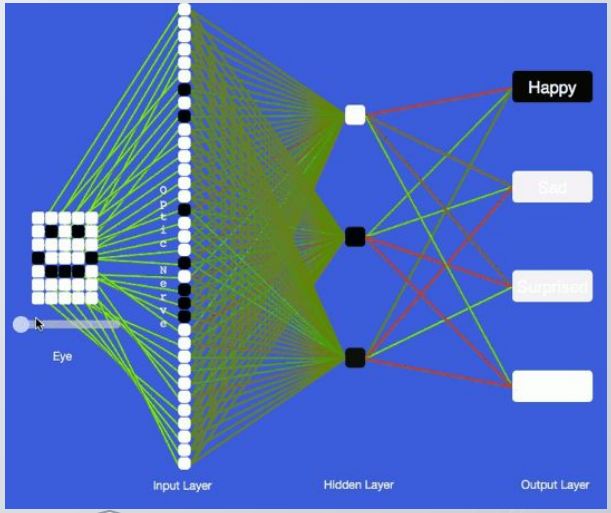
Investigating training an **Artificial Intelligence**

A closer look
"under the hood"

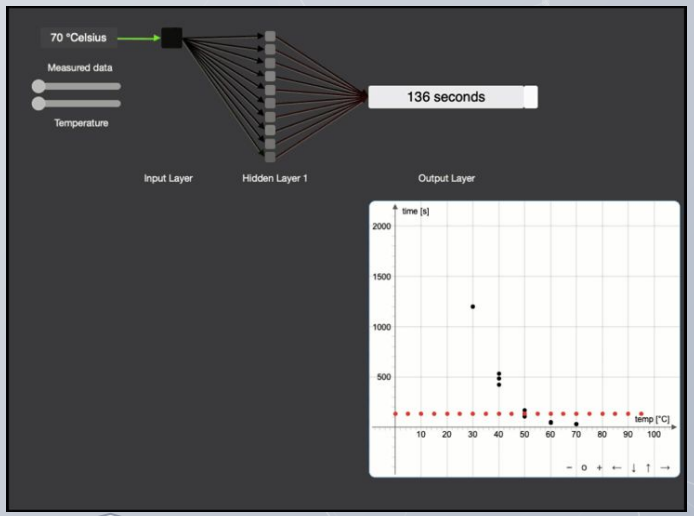
What's so special about AI?



Driving a Robot



Emoji recognition



Prediction

AI is versatile!

In each scenario, the same underlying AI is at work.

RESOURCE: AI at **My Computer Brain**

Can students code a neural network from scratch?

It's not recommended as a general classroom activity for DigiTech in 7-10.

```python
import numpy as np # helps with the math
import matplotlib.pyplot as plt # to plot error during training

# input data
inputs = np.array([[0, 1, 0],
                   [0, 1, 1],
                   [0, 0, 0],
                   [1, 0, 0],
                   [1, 1, 1],
                   [1, 0, 1]])
# output data
outputs = np.array([[0], [0], [0], [1], [1], [1]])

# create NeuralNetwork class
class NeuralNetwork:

    # intialize variables in class
    def __init__(self, inputs, outputs):
        self.inputs  = inputs
        self.outputs = outputs
        # initialize weights as .50 for simplicity
        self.weights = np.array([[.50], [.50], [.50]])
        self.error_history = []
        self.epoch_list = []

    #activation function ==> S(x) = 1/1+e^(-x)
    def sigmoid(self, x, deriv=False):
        if deriv == True:
            return x * (1 - x)
        return 1 / (1 + np.exp(-x))

```

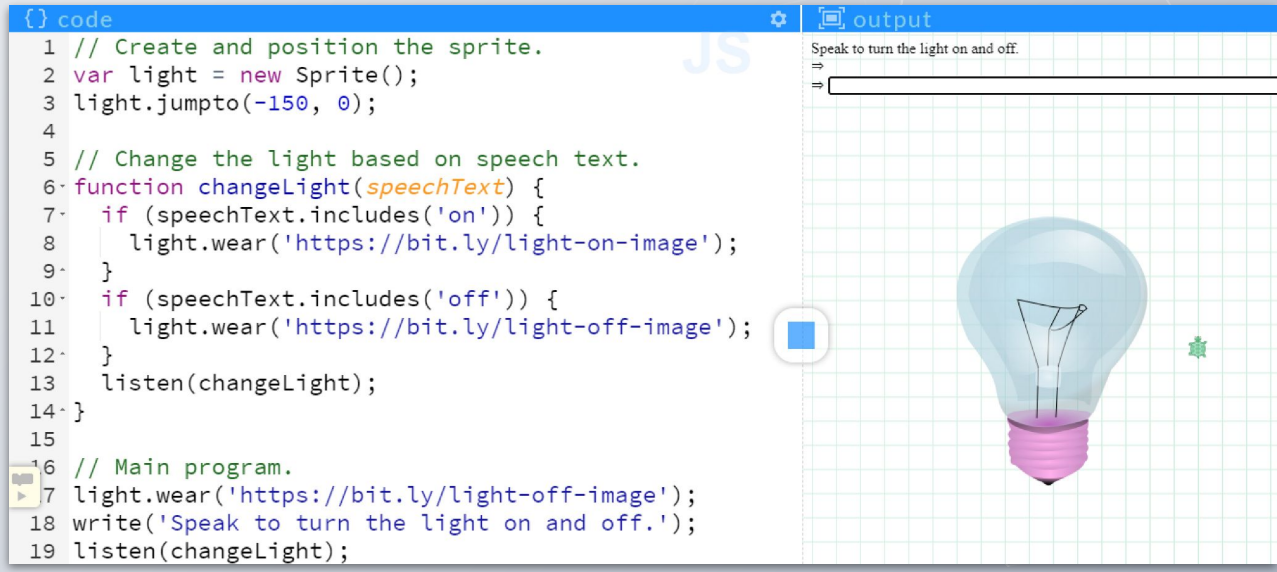
Code for a "simple" neural network to work with tiny binary arrays.

(screenshot from Simple Neural Networks in Python, Aidan Wilson)

Can students write code to investigate and use a neural network?

YES!

Utilise an existing speech recognition system in a general purpose program (**JavaScript**).



LESSON: **Home automation: General Purpose Programming** (Years 7-8)

Can students write code to investigate and use a neural network?

YES!

Utilise an existing sentiment analysis system in a general purpose program (**Python**).

```
Alice is a hero, score: 42
Queen is a hero, score: 16
Rabbit is neutral, score: 6
Duchess is a hero, score: 18
Illustration is neutral, score: 0
Hatter is a villain, score: -5
Majesty is a villain, score: -2
Alices is a hero, score: 11
Youre is neutral, score: 1
Hearts is neutral, score: 1
>
```

LESSON: **Coding a sentimental chatbot** (Years 7-10)

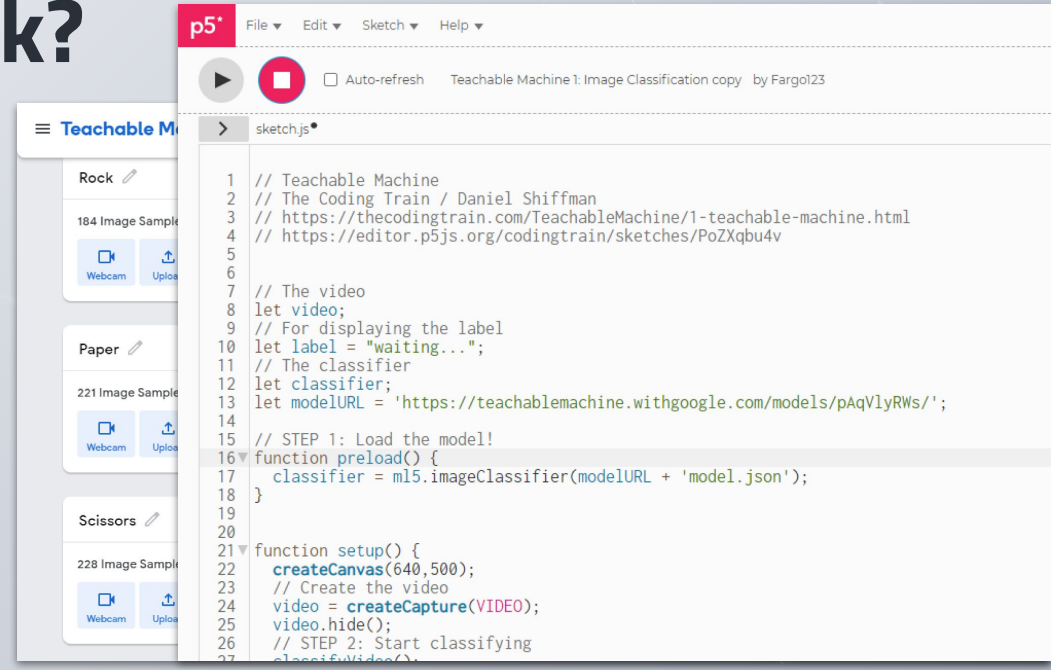
LESSON: **Book analysis with AI techniques** (Years 7-10)

Can students write code to investigate and use a neural network?

YES!

Train their own AI model with an online tool.

Bring the model into a general purpose program (**JavaScript**) to make a decision.



LESSON: Rock, Paper, Scissors, AI! (Years 7-8)

Can students write code to investigate and use a neural network?

YES!

Train *and* test an AI model with general purpose code (**Python**).

Perform **classification** and **regression**.

First, let's talk about **data**.

Years 7-8
Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)

TEXT ➡

T	=	84	01010100
E	=	69	01000101
X	=	88	01011000
T	=	84	01010100

First, let's talk about **data**.

Years 7-8

Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)


```
...
10111001
00010111
11100111
00000110
11001100
00001111
01100101
10000100
...
```

First, let's talk about **data**.

Years 7-8

Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)

R 84 G 120 B 78	R 82 G 118 B 77	R 76 G 113 B 70	R 74 G 111 B 68	R 72 G 108 B 66	R 70 G 106 B 66	R 64 G 97 B 64	R 52 G 72 B 55
R 82 G 117 B 77	R 80 G 115 B 75	R 75 G 112 B 71	R 74 G 109 B 69	R 65 G 97 B 63	R 55 G 78 B 54	R 38 G 50 B 44	R 16 G 18 B 27
R 76 G 108 B 71	R 75 G 109 B 71	R 73 G 104 B 72	R 59 G 82 B 62	R 38 G 53 B 45	R 21 G 25 B 30	R 11 G 7 B 20	R 9 G 6 B 16
R 78 G 102 B 75	R 64 G 84 B 65	R 42 G 57 B 49	R 22 G 27 B 33	R 11 G 9 B 21	R 10 G 6 B 18	R 9 G 6 B 15	R 7 G 6 B 12
R 43 G 51 B 48	R 25 G 26 B 33	R 10 G 11 B 21	R 8 G 5 B 18	R 9 G 5 B 16	R 8 G 6 B 15	R 8 G 5 B 13	R 21 G 23 B 22
R 12 G 8 B 21	R 9 G 5 B 16	R 8 G 6 B 15	R 8 G 5 B 17	R 8 G 5 B 14	R 9 G 7 B 12	R 27 G 30 B 28	R 57 G 74 B 57
R 12 G 6 B 20	R 12 G 7 B 21	R 8 G 6 B 17	R 10 G 6 B 15	R 12 G 10 B 15	R 35 G 41 B 35	R 65 G 86 B 65	R 71 G 103 B 74
R 12 G 9 B 21	R 13 G 11 B 21	R 12 G 9 B 18	R 18 G 16 B 18	R 41 G 53 B 40	R 70 G 95 B 72	R 73 G 106 B 77	R 70 G 107 B 74

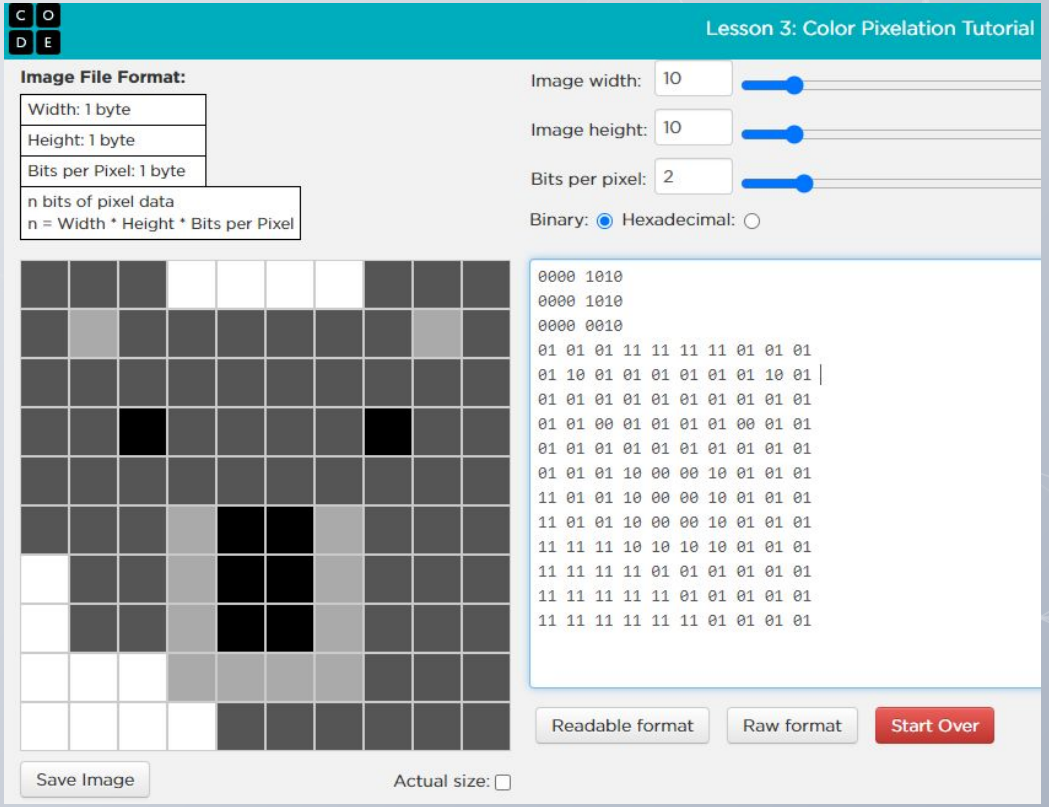
84	01010100
120	01111000
78	01001110
82	01010010
118	01110110
77	01001101
76	01001100
113	01110001
70	01000110
...	...

Working with pixel values

Colour Pixelation widget at code.org

One bit per pixel
0 = black, 1 = white

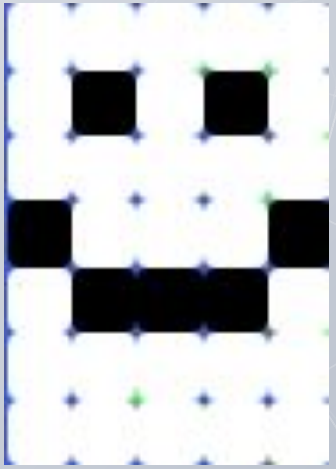
Two bits per pixel
00 = black, 01 = dark grey,
10 = light grey, 11 = white



Black and white

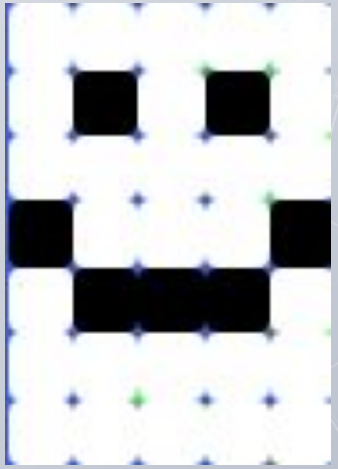
This image is a 7x5 matrix of black and white pixels

Each pixel represents a bit: **0** or **1**



Binary representation

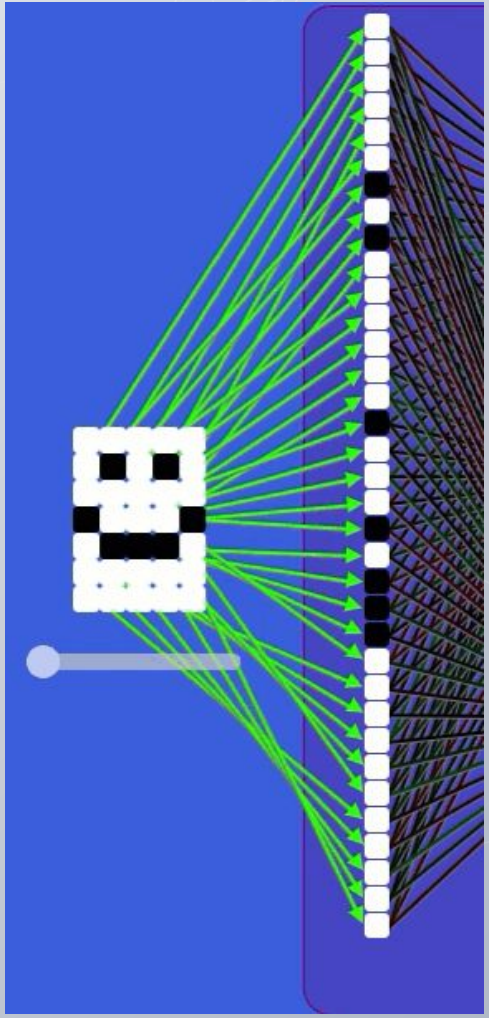
The image can be represented as binary.



```
00000
01010
00000
10001
01110
00000
00000
```

The rows of 5 bits can be connected into a 35 bit binary number

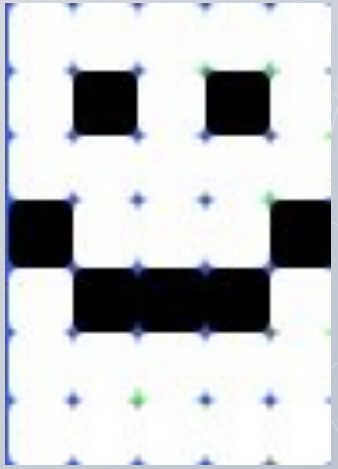
smiley = **00000 01010 00000 10001 01110 00000 00000**



Binary representation

The binary representation is preferred over a decimal representation, as binary more closely resembles the image.

By looking at the binary number, we can actually see the picture

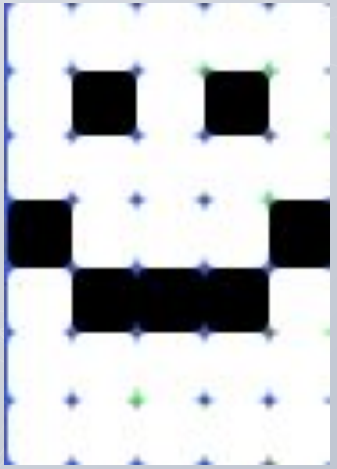


00000
01010
00000
10001
01110
00000
00000

Decimal representation

The decimal representation is fairly disconnected from the original image.

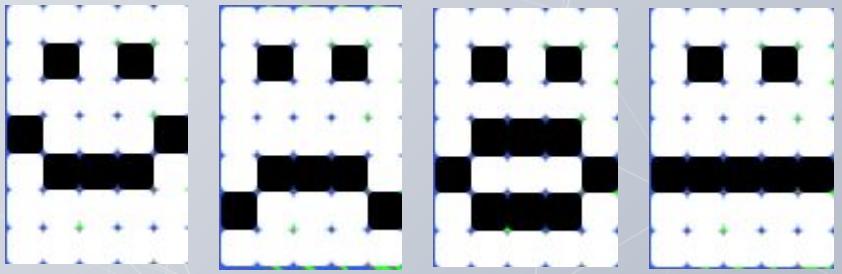
Leading zeroes are lost.



336115712

Binary Representation

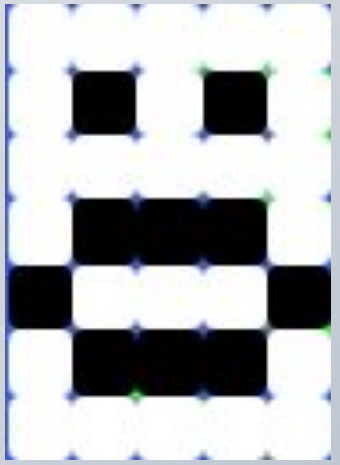
With this approach, we can represent any B/W picture as binary numbers.



happy sad surprised angry

Binary Representation activity

What is the binary representations of the **surprised** emoji?

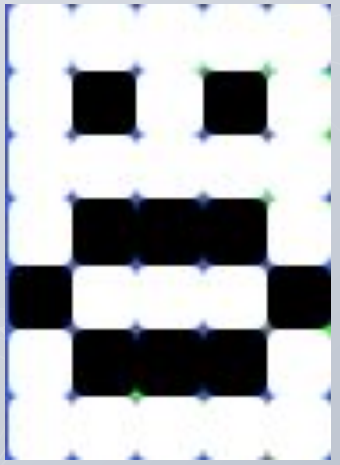


00000

?

Binary Representation activity

What is the binary representations of the **surprised** emoji?



00000
01010
00000
01110
10001
01110
00000

This approach is not limited to pixel graphics. We can also represent words in binary.

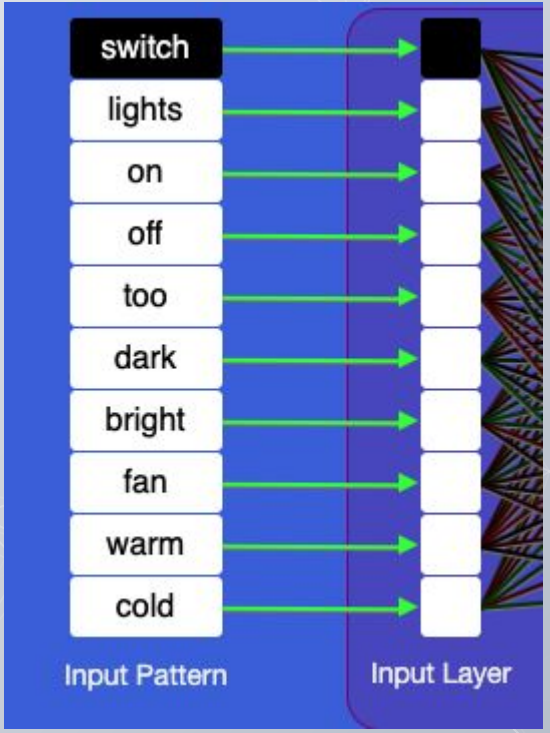
Let’s explore home automation

Home automation

Entire words can be represented in binary

In this AI ...

'Switch' is **1000000000**

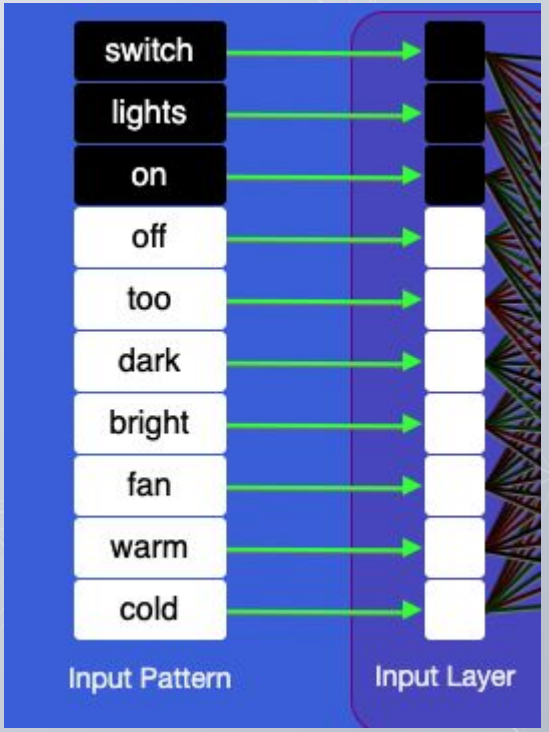


1
0
0
0
0
0
0
0
0
0
0

LESSON: **Home automation with AI** (Years 5-6)

In this AI ...

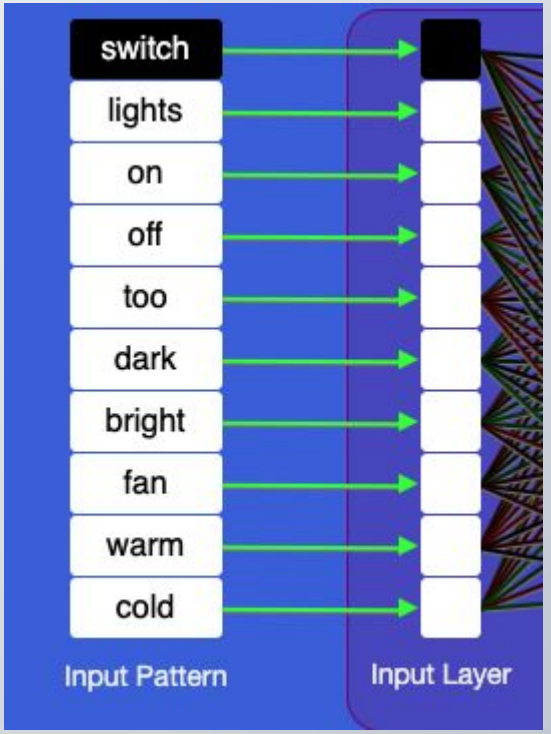
'switch lights on'
is **1110000000**

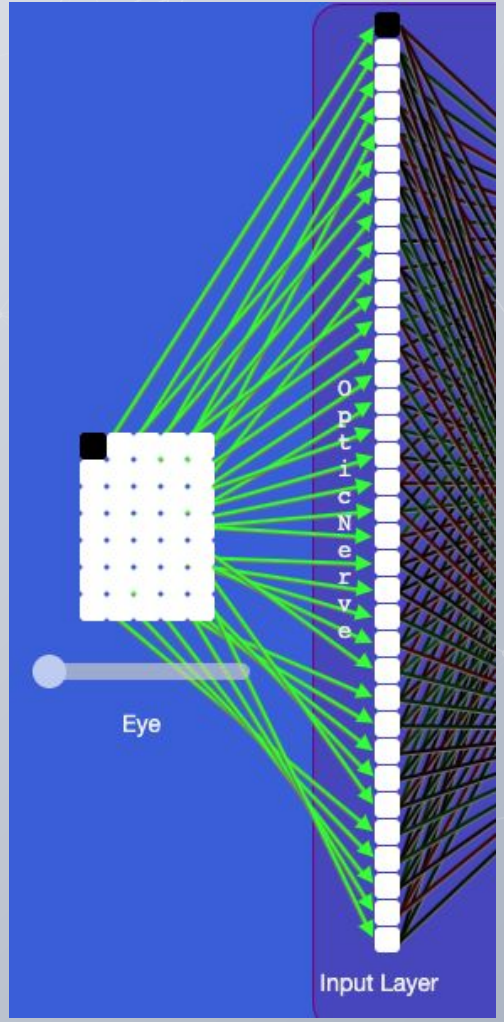


1
1
1
0
0
0
0
0
0
0
0

Ultimate equality!

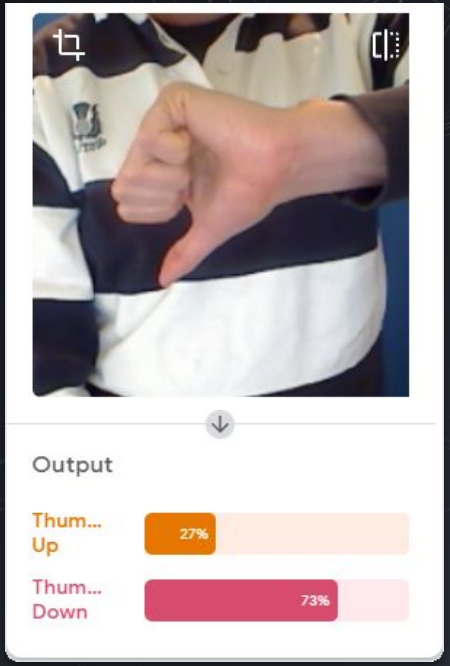
It does not matter for the AI whether a bit represents a word or a pixel





Classifier

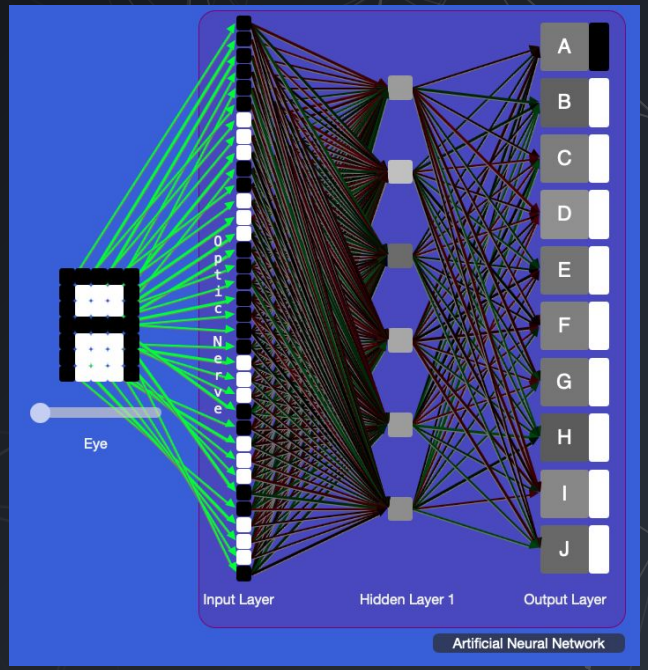
At its output, the AI tells us into which bucket (class) an object most likely belongs.



Our hands-on example

We classify the first 10 letters of the alphabet.

visual demonstration



Our hands-on example

Now lets run the simulation with **general purpose programming**.

Using:

- Python
- **replit.com** online environment
- **sci-kit** library

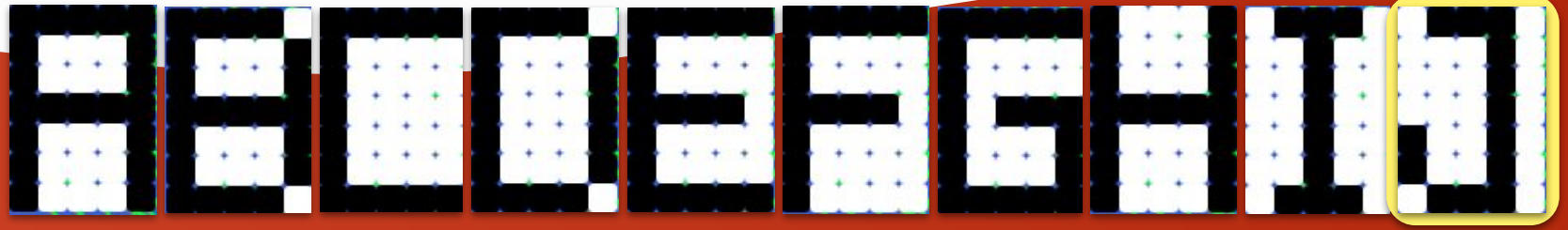
1. Include the sci-kit library

```
from sklearn.neural_network import MLPClassifier
```

- starting point
- finished program

2. Binary representation of the letters A-J

```
# The data.
letters = [
 [1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1],
 [1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0],
 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0],
 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1],
 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0],
 [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1],
 [1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1],
 [0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0],
 [0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0]]
```



3. Assign letters to classes (buckets)

```
letter_classes = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']
```

3. Configure and train the AI

```
# Set up the AI.
AIClassifier = MLPClassifier(solver = 'lbfgs', activation = 'logistic',
                             alpha = 1e-5, hidden_layer_sizes = (10),
                             random_state = 1, max_iter = 200)

# Train the AI.
AIClassifier.fit(letters, letter_classes)
```

4. Check if the classifier works

```
# Test the AI with one of the input letters.
prediction = AIClassifier.predict([[1, 1, 1, 1, 1,
                                    1, 0, 0, 0, 0,
                                    1, 0, 0, 0, 0,
                                    1, 0, 0, 0, 0,
                                    1, 0, 0, 0, 0,
                                    1, 0, 0, 0, 0,
                                    1, 1, 1, 1, 1]])
print('The letter C is recognised as', prediction)
```

```
The letter C is recognised as ['C']
```

5. Classify an unknown letter

```
# Test the AI with an unknown letter.
unknown_letter = [[1, 1, 1, 1, 1,
                   1, 0, 0, 0, 0,
                   1, 0, 0, 0, 0,
                   1, 1, 0, 1, 0,
                   1, 0, 0, 0, 0,
                   0, 0, 0, 0, 0,
                   1, 1, 1, 1, 1]]
prediction = AIClassifier.predict(unknown_letter)
print('The unknown letter is recognised as', prediction)
```

```
The unknown letter is recognised as ['E']
```

6. Also get confidence level

```
confidence = AIClassifier.predict_proba(unknown_letter)
print('Confidence level is', confidence[0][4].round(2))
```

'E' is at position **4** in the array of classes / buckets.

```
Confidence level is 0.97
```

Regression

Unlike the classification models, regression models output numeric values.

They have continuous values for both dependent and independent variables.

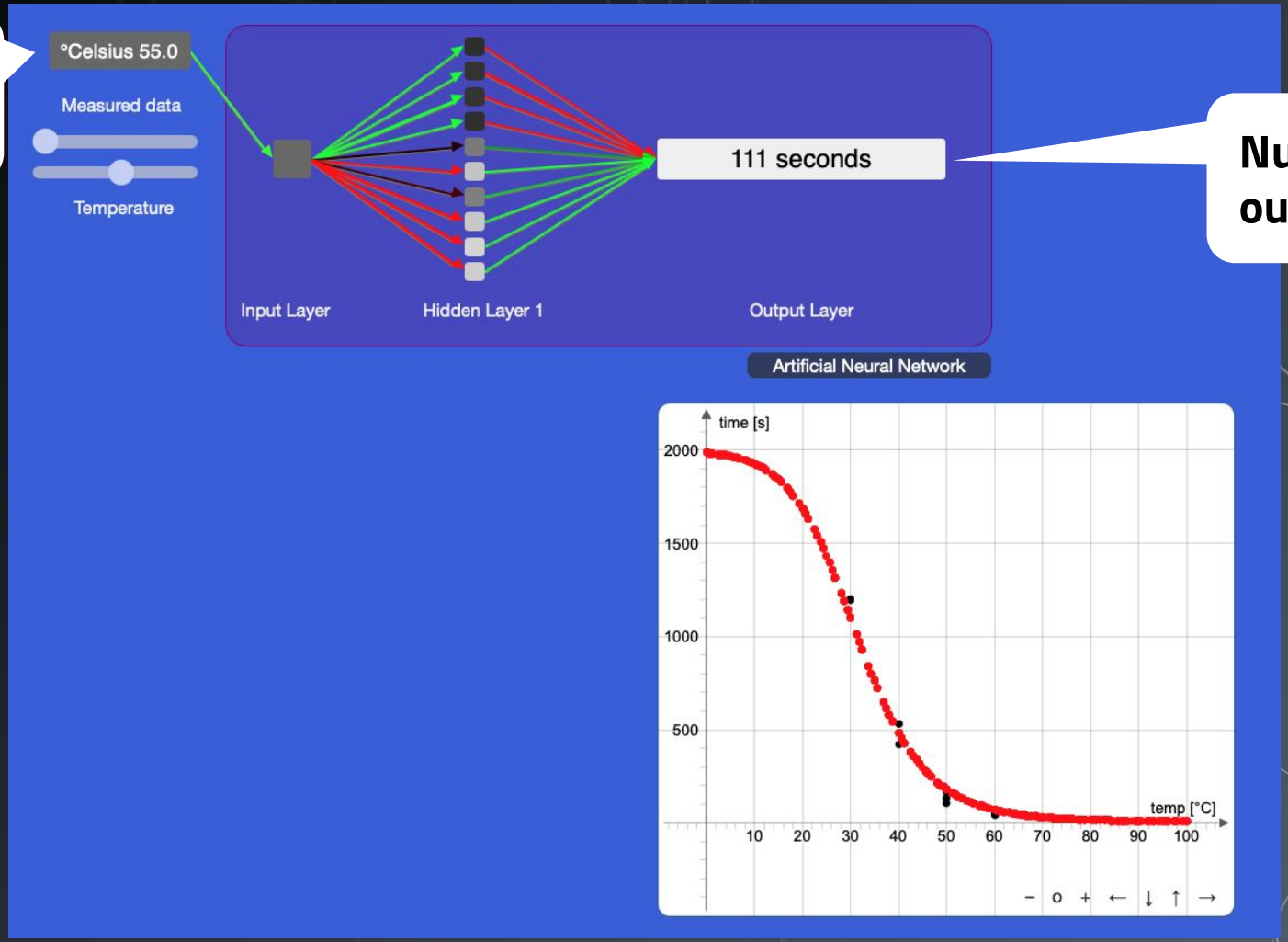
We use regression to find data points outside of a given range (**interpolation** and **extrapolation**).

Our hands-on example

We find a **curve of best fit** for a chemistry experiment.

visual demonstration

Numeric input



Numeric output

demo

Our hands-on example

Now lets run the simulation with **general purpose programming**.

Using:

- Python
- **replit.com** online environment
- **sci-kit** library

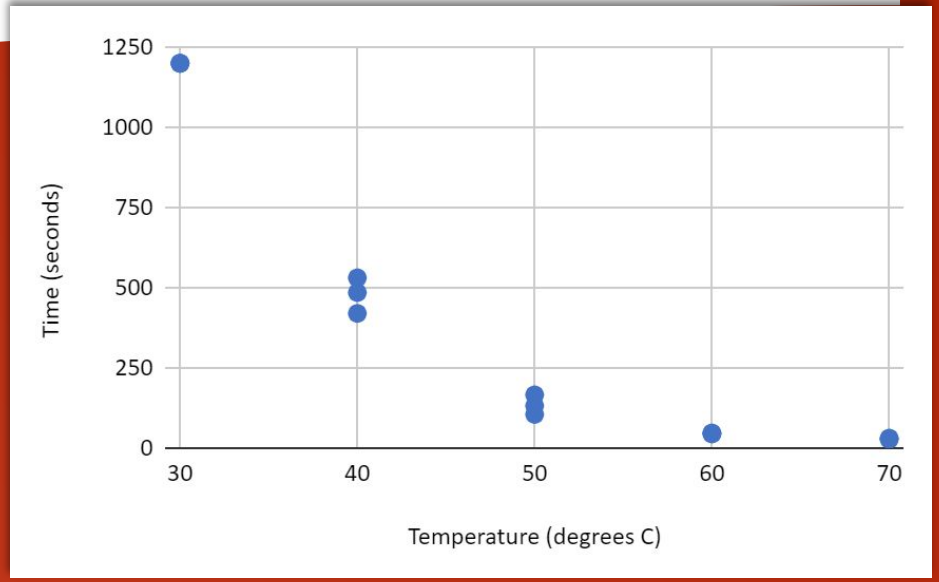
1. Include the sci-kit library + others

```
from sklearn.neural_network import MLPRegressor
import matplotlib.pyplot as plt
import numpy as np
```

- starting point
- finished program

2. Raw data from the experiment

```
# The data.
temp = [[30], [30], [30], [40], [40], [40], [50], [50], [50], [60], [60], [60], [70], [70], [70]]
time = [1200, 1200, 1200, 531, 485, 420, 166, 132, 105, 47, 44, 46, 29, 27, 31]
```



3. Normalise (scale) the data

AIs like their data in the domain and range of 0 to 1.

Can be any real number between 0 and 1.

$x \in (0,1)$

3. Normalise (scale) the data

```
# Normalise temperature by 100.
temp_normalised = [x[0] / 100 for x in temp]
temp_normalised = np.array(temp_normalised)
temp_normalised = temp_normalised.reshape(len(temp_normalised), -1)
#temp_normalised = [[0.3], [0.30], [0.30], [0.40], [0.40], [0.40], [0.50],
                    [0.50], [0.50], [0.60], [0.60], [0.60], [0.70], [0.70], [0.70]]

# Normalise time by 2000.
time_normalised = [x / 2000 for x in time]
#time_normalised = [0.60000, 0.60000, 0.60000, 0.26550, 0.24250, 0.21000, 0.08300,
                    0.06600, 0.05250, 0.02350, 0.02200, 0.02300, 0.01450, 0.01350, 0.01550]
```

4. Configure and train the AI

```
# Set up the AI.
AI_regressor = MLPRegressor(solver = 'lbfgs', activation = 'logistic',
                            alpha = 1e-4, hidden_layer_sizes = (10, 1),
                            random_state = 1, max_iter = 1000000)

# Train the AI.
AI_regressor.fit(temp_normalised, time_normalised)
```

5. Make a single prediction

```
# Make a single prediction for temp = 35 degrees  (x = 0.35).
prediction = AI_regressor.predict([[0.35]])
print('For 35 degrees, time is', prediction * 2000, 'milliseconds.')
```

Denormalise the time result.

```
For 35 degrees, time is [799.02094774] milliseconds.
```

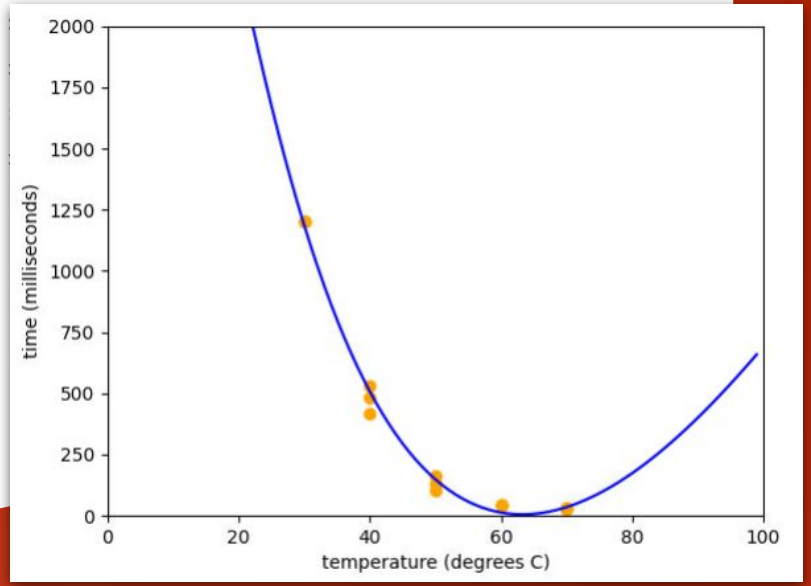
6. Plot original data and AI line of best fit

```
# Plot experiment data.
plt.scatter(temp, time, color = 'orange')

# Plot AI data. Use AI to predict y values for a series of x values.
x = np.arange(0, 1, 0.01)                          # start, stop, step
y = AI_regressor.predict(x.reshape(len(x), -1))    # put each x-value in its own array,
                                                   # suitable for predict()
plt.plot(x * 100, y * 2000, color = 'blue')        # denormalise x and y for plot

# Limit plot to an appropriate range and domain.
plt.xlim(0, 100)
plt.ylim(0, 2000)
plt.xlabel('temperature (degrees C)')
plt.ylabel('time (milliseconds)')
plt.show()
```

Denormalise the time result.

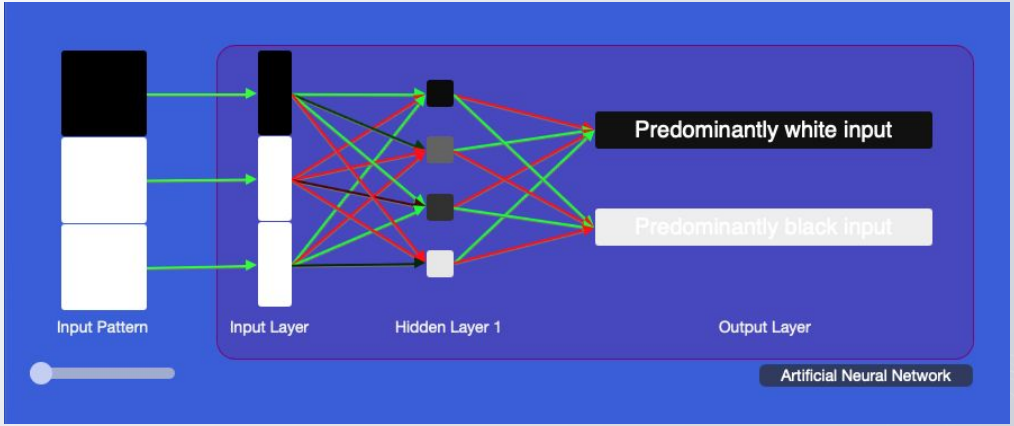
6. Plot original data and AI line of best fit

```
# Plot experiment data.
plt.scatter(temp, time, color = 'orange')

# Plot AI data. Use AI to predict y values for a series of x values.
x = np.arange(0, 1, 0.01)
y = AI_regressor.predict(x.reshape(len(x), -1))

plt.plot(x * 100, y * 2000, color = 'blue')

# Limit plot to an appropriate range and domain.
plt.xlim(0, 100)
plt.ylim(0, 2000)
plt.xlabel('temperature (degrees C)')
plt.ylabel('time (milliseconds)')
plt.show()
```



Science Alert

Inside a neural network



Hidden Layer 1

Input	Weight	Product	Sum	Output
1.00	3.43	3.43	3.43	0.9686
0.00	-1.37	0.00		
0.00	3.73	0.00		

Input	Weight	Product	Sum	Output
1.00	0.27	0.27	0.27	0.5671
0.00	-1.81	0.00		
0.00	-1.81	0.00		

Input	Weight	Product	Sum	Output
1.00	1.26	1.26	1.26	0.779
0.00	-0.34	0.00		
0.00	2.51	0.00		

Input	Weight	Product	Sum	Output
1.00	-2.40	-2.40	-2.4	0.0832
0.00	-1.43	0.00		
0.00	0.18	0.00		

Input

1
0
0

Output Layer

input fields are mainly white

Input	Weight	Product	Sum	Output
0.97	-4.23	-4.10	2.76161	0.9405658
0.57	13.61	7.72		
0.78	-2.54	-1.98		
0.08	13.46	1.12		

Input	Weight	Product	Sum	Output
0.97	4.15	4.02	-2.7663	0.0591743
0.57	-13.59	-7.71		
0.78	2.62	2.04		
0.08	-13.47	-1.12		

input fields are mainly black

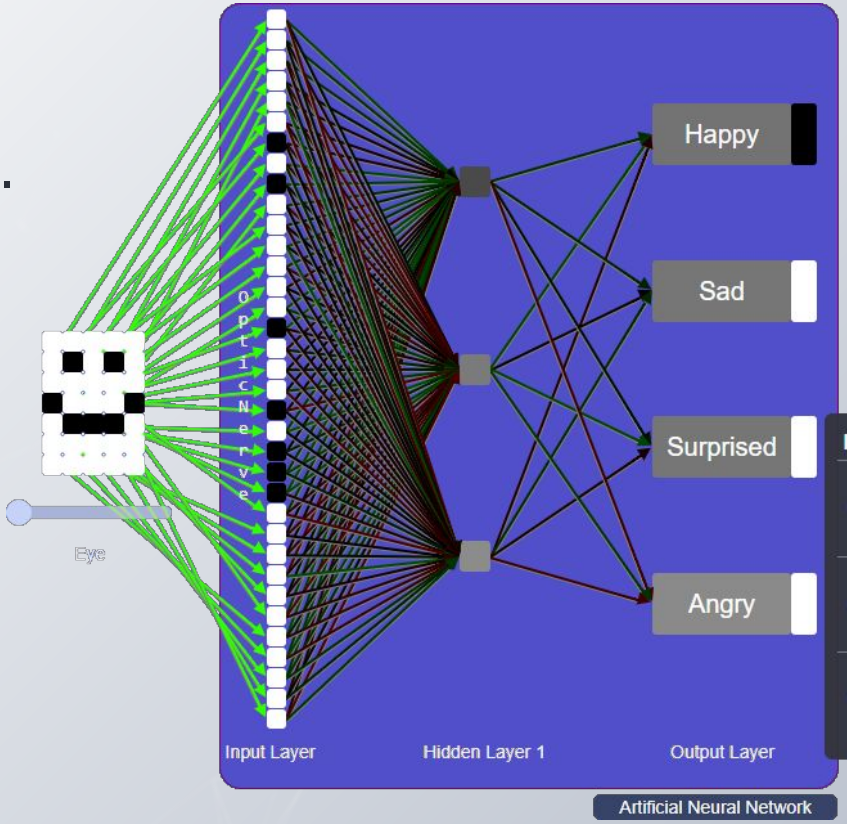
RESOURCE: My Computer Brain

LESSON COMING SOON: **Inner Workings of an AI** (Years 9-10)

EXPLAINER VIDEO: Introduction to AI and machine learning

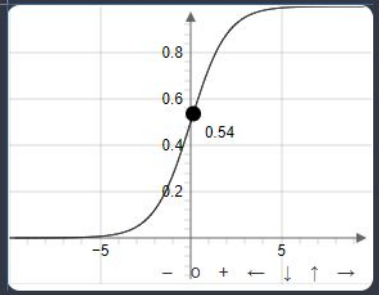
Investigating a little further

Inside the perceptron.



Input	Weight	Product	Sum
0.71	0.06	0.04	0.15
0.52	0.27	0.14	
0.45	-0.09	-0.04	

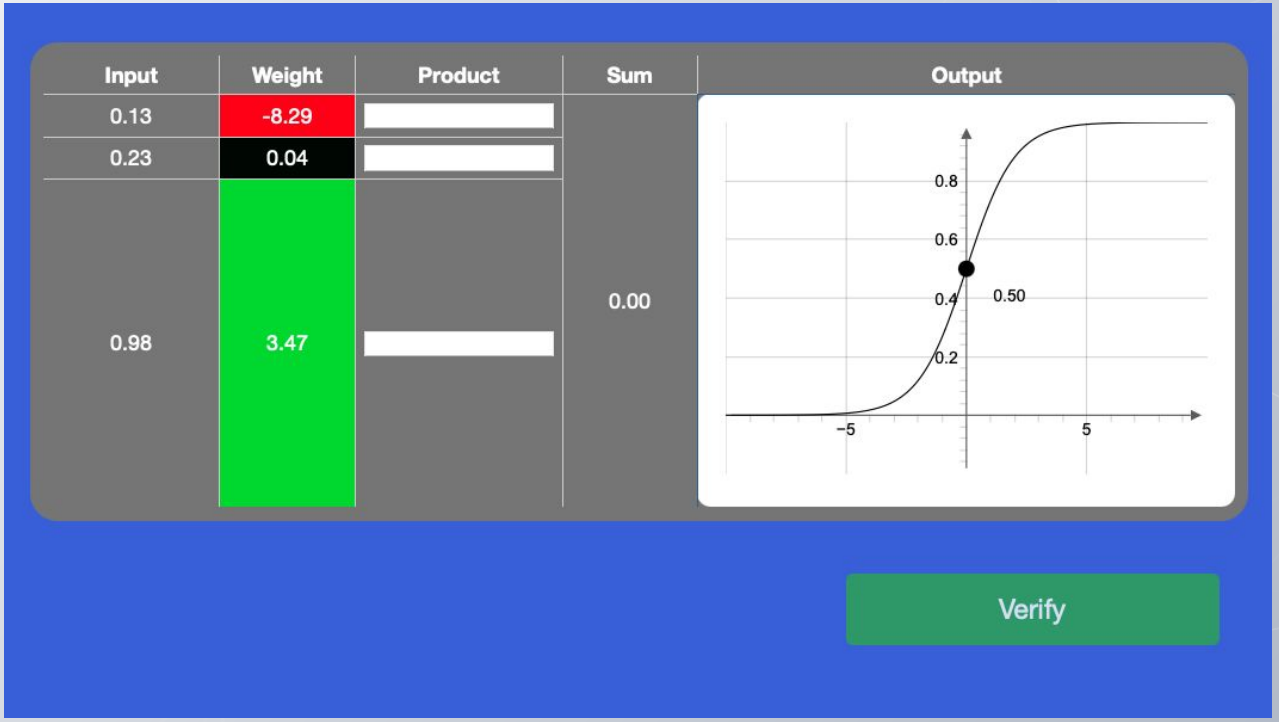
Output



Demo

Inside the Perceptron

Inside the perceptron.



Demo

Artificial Intelligence lesson plans

Humans display natural intelligence in contrast to machines that demonstrate artificial intelligence (AI).

AI has various definitions however for our purposes we are using the definition 'any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals' [1]. Read more...

The following lesson ideas cover a range of specialisations and subsets as indicated by the colour coding. Click on the coloured squares to learn more about each definition.

Machine learning | Supervised learning | Natural language processing | Computer vision | Classification | Clustering | Regres

Lesson plans

Artificial Intelligence

Access DT Hub AI lesson plans

A chance to ask questions …

Use the chat…

How can you incorporate these teaching ideas?

What do you feel more confident about?

What do you still need to know?

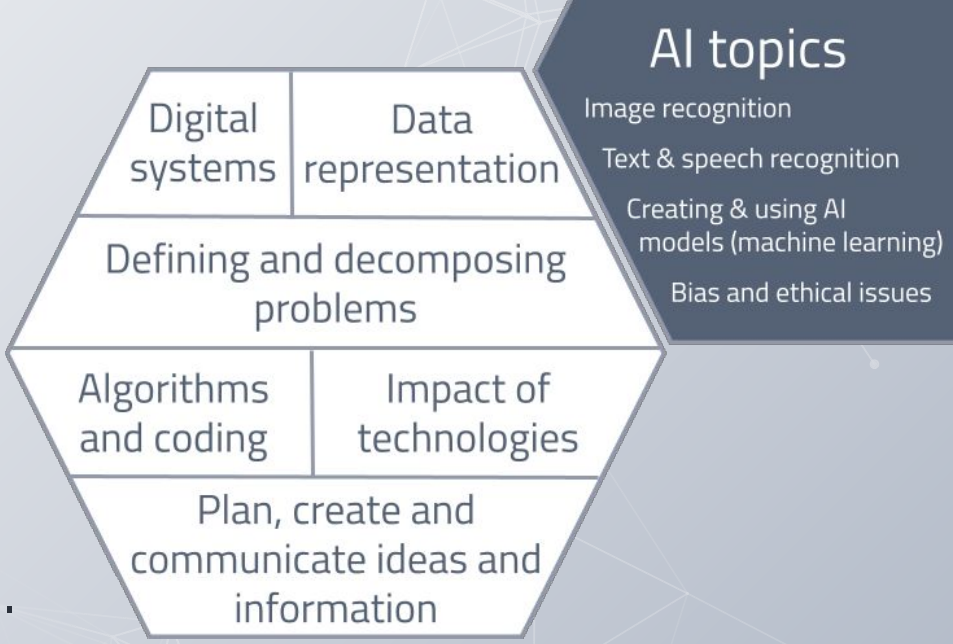
Next steps

Making a commitment to implementing AI in your classroom

Use the chat to **write your idea** of where you will include AI as part of your teaching and learning program.

Connecting and sharing with the group.

email:
digitaltechnologieshub@esa.edu.au



DIGITAL TECHNOLOGIES HUB

Other Deep Dives

Deep dive 3: Natural language processing for large text analysis	Tues 24th August 2021
Deep dive 4: AI, ethics and systems thinking	Tues 7th Sept 2021

DIGITAL TECHNOLOGIES HUB