KNOWLEDGE REPRESENTATION: THE LINK BETWEEN BRAIN AND MACHINE!

By Yaiza Arnáiz Alcácer and Pablo Marcos

01.

INTRODUCTION

Knowledge representation for machines and humans

LITERATURE

- In **connectionist models**, knowledge is embedded in networks of relationships between different elements, allowing basic stimuli to give rise to large networks.

- In **symbolic models**, knowledge is represented as a series of declarative sentences, usually based on logic, describing the attributes of a series of "symbols", mental models modifiable by rules that hold cognizable properties.

LEGENYEL'S EXPERIMENT

- Humans are good at recognizing symbols, but, ¿what about chimeras?
- By showing datasets of 6 elements grouped pairwise, Legenyel showed humans are worse at finding chimeras than normal groups
- To improve human accuracy, Legenyel added a haptic stimuli: breaking appart the images
- This improved learning significantly

02.

OUR EXPERIMENT

Using ML to classify chimeras

THREE NEURAL NETWORKS

Naive Network

This neural network is trained only with figures: squares, triangles, circles

Random Network

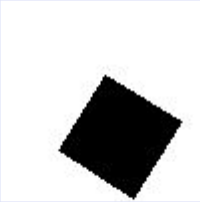
This neural network is trained with figures and with random pixels

Chimaera Network

This neural network is trained with the figures and chimaeras

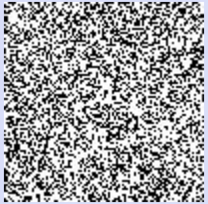
THE DATASET

SQUARES

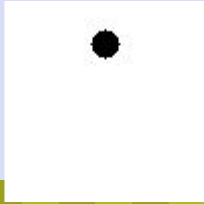


TRIANGLES

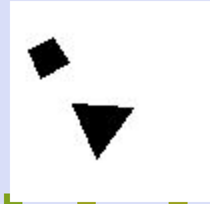
RANDOM



CIRCLES



CHIMAERAS



THIS IS OUR NETWORK

```
#NEURAL NETWORK
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(in_channels=1, out_channels=6, kernel_size=13, padding="same")
        self.conv2 = nn.Conv2d(in_channels=6, out_channels=6, kernel_size=11, padding="same")
        self.conv3 = nn.Conv2d(in_channels=6, out_channels=6, kernel_size=9, padding="same")
        self.conv4 = nn.Conv2d(in_channels=6, out_channels=6, kernel_size=7, padding="same")
        self.conv5 = nn.Conv2d(in_channels=6, out_channels=6, kernel_size=5, padding="same")
        self.fc1 = nn.Linear(25*25*6, 84)
        self.fc2 = nn.Linear(84, 3)
        # loss: 0.03


    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = nn.MaxPool2d(2, 2)(x)
        x = F.relu(self.conv2(x))
        x = nn.MaxPool2d(2, 2)(x)
        x = F.relu(self.conv3(x))
        x = F.relu(self.conv4(x))
        x = F.relu(self.conv5(x))
        x = torch.flatten(x, 1)  # flatten all dimensions except batch
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

This is changed when we add the new set of shapes for training and testing.

THE NAIVE NETWORK

- Trained with raw shapes: Circles, Triangles, Squares
- Efficient at recognizing raw shapes, but fails with chimeras
- How does it learn? Most likely, it decides by grouping pixels; chimeras have multi-type pixels, which messes things up
- How can we train it, just as legenyel trained humans?

03.

A FOURTH CLASS

RANDOM PIXELS

- By adding random pixels, the neural network disocciates learning from specific shapes -> it learns that other types of pixels exist

- The performance is good, but not classify the chimeras

03.

A NEW IDEA

DIRECTLY CLASSIFYING CHIMERAS

CLASSIFYING CHIMERAS

- If we know chimeras exist, ¿why not directly train the algorithm with said chimeras?

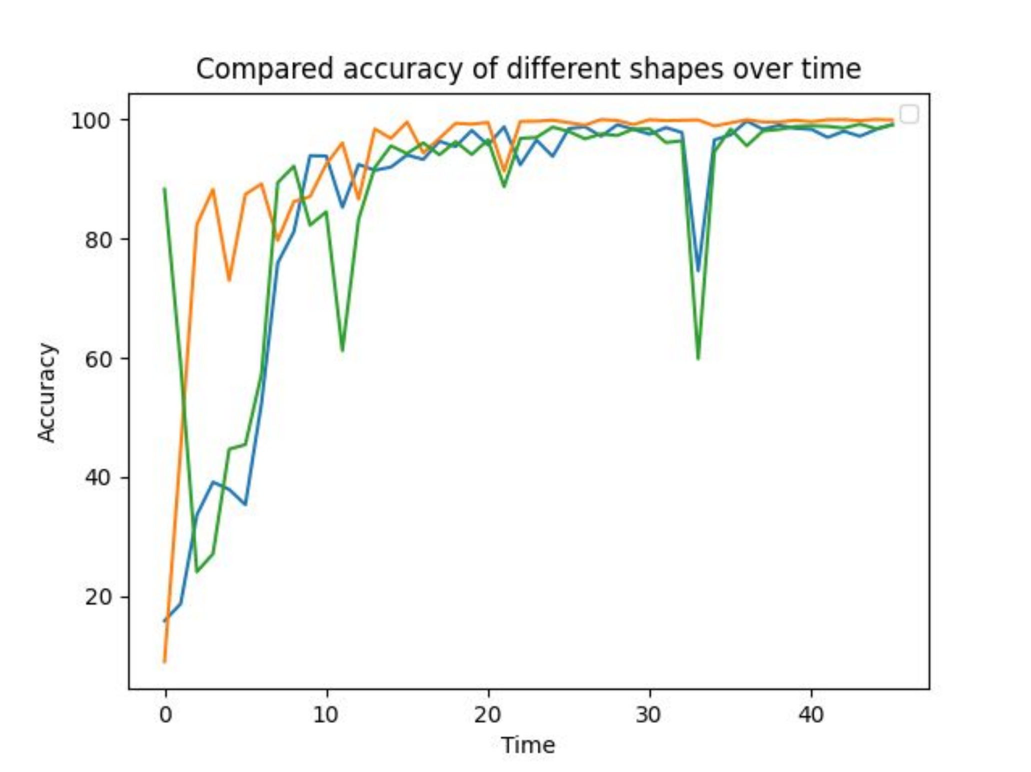
- The accuracy increases **massively**, as was to be expected

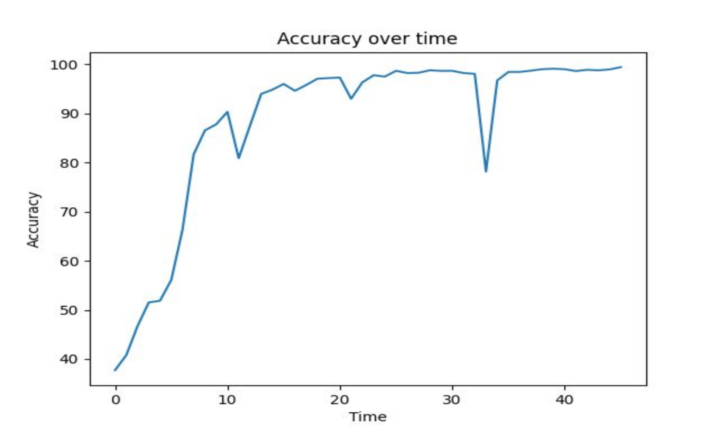
- Now, our neural network can classify chimeras as such!

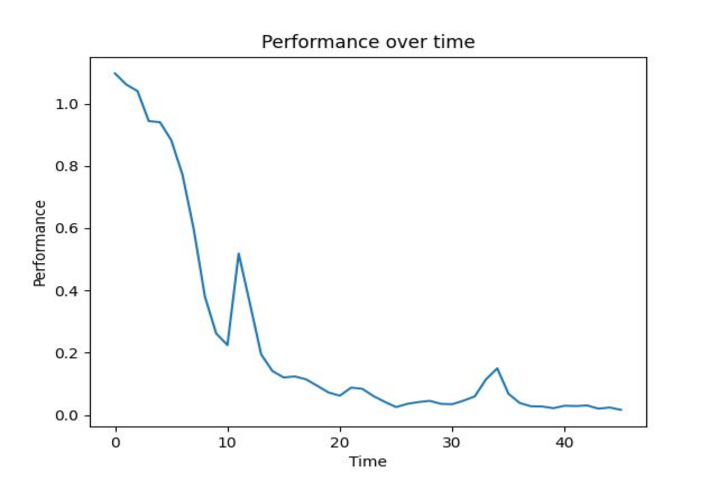
03.

RESULTS

NAIVE NETWORK RESULTS



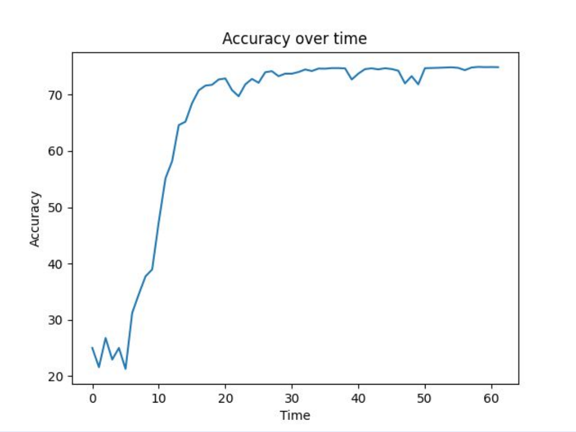


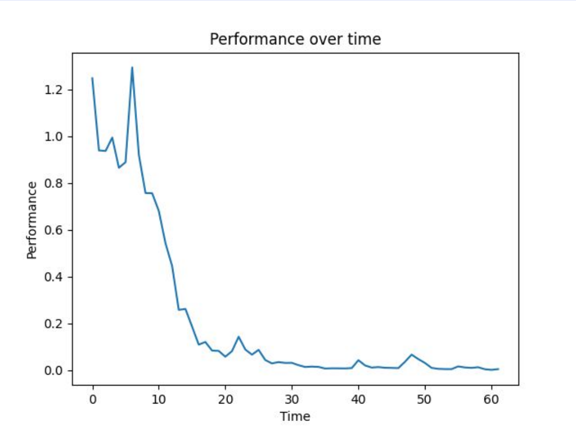


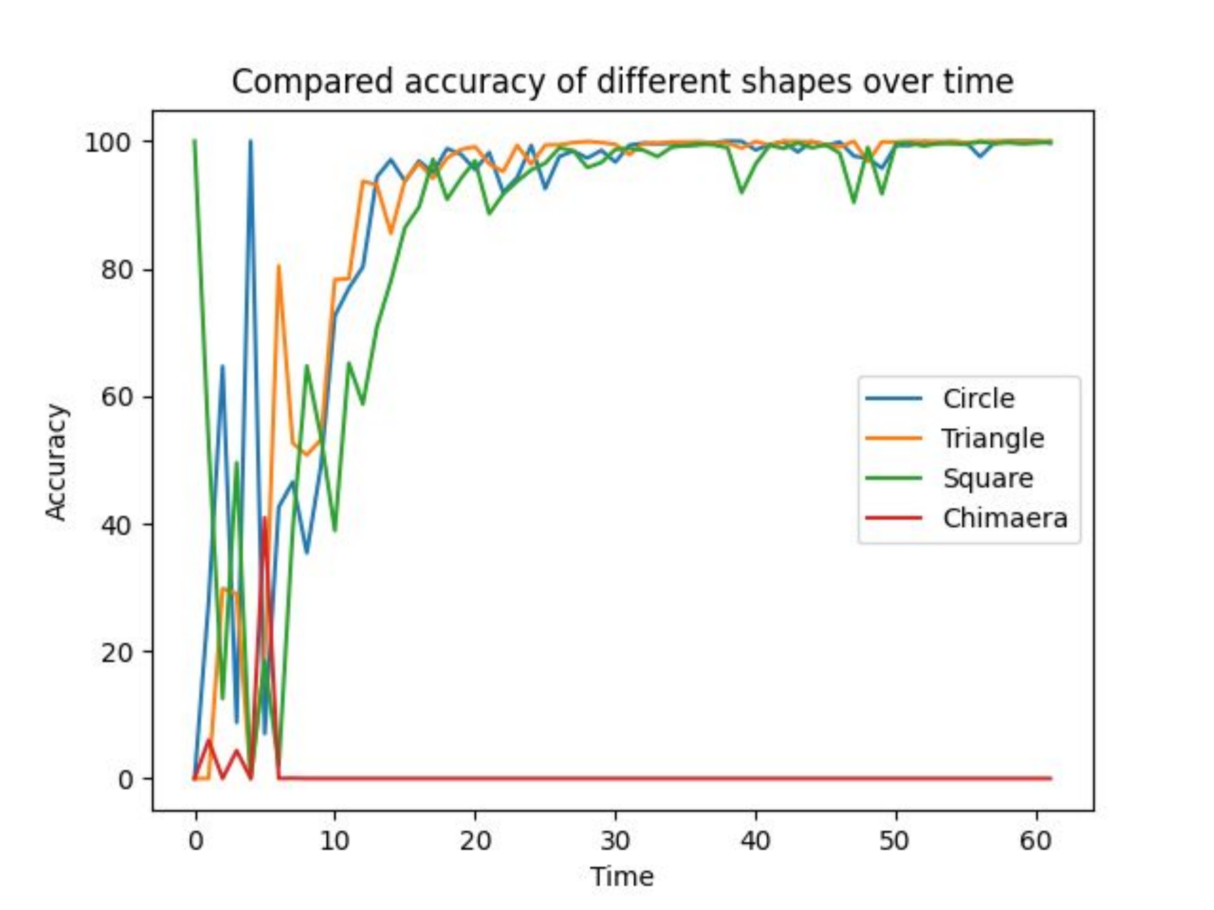
PREDICTING CHIMAERAS

shape_1	shape_2	chimaera	choice	square_output	triangle_output	circle_output
13068	444	13509	0	16.170805	-19.070637	-2.3680634
8688	504	9192	0	9.532941	-6.8284335	-6.1816993
9003	9120	15288	2	12.716791	-42.56076	21.096523
396	5379	5775	1	10.962057	-62.382343	50.11208
8970	3387	11238	1	17.44374	-25.415987	0.29061002
1452	2127	3579	1	3.9442225	-36.620987	23.64105
4551	3162	7713	2	13.0789995	-11.913027	-5.12081
6279	1776	7602	0	11.756158	-9.254048	-6.8481803
1281	3027	4308	1	3.1535027	-80.64171	54.30713
6360	339	6699	1	10.875695	-7.8255324	-7.486672

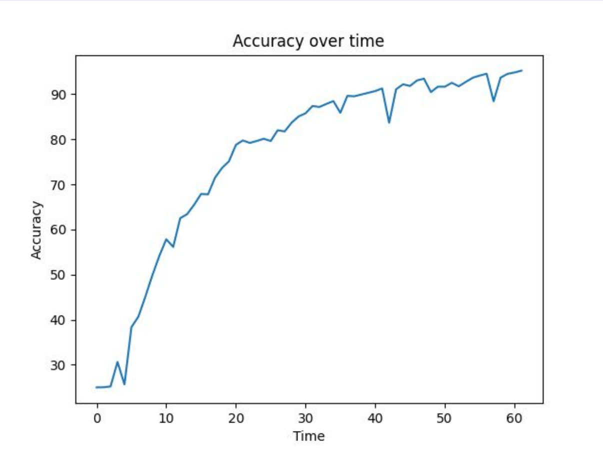
RANDOM NETWORK RESULTS

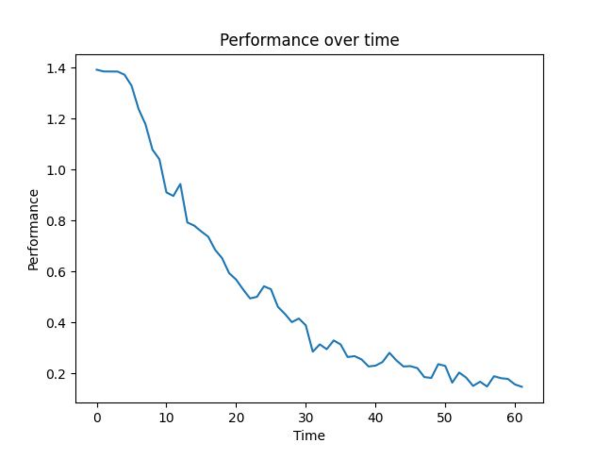


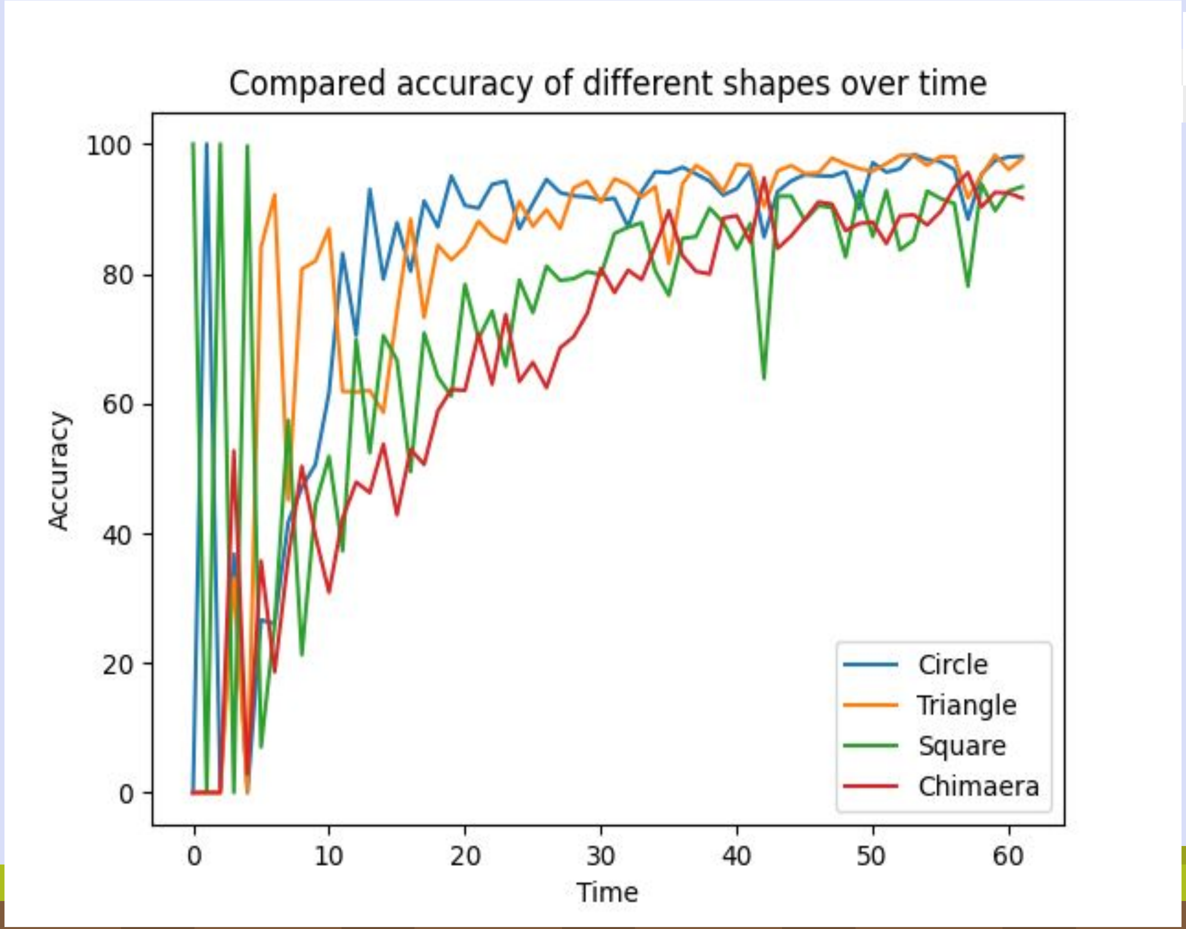




CHIMAERA NETWORK RESULTS







04.

CONCLUSIONS

What we learnt

CONCLUSSIONS

MACHINE LEARNING

Is complicated

KNOWLEDGE REPRESENTATION

Helps create interpretable machine learning, and make fairer decissions

UNDERSTANDING

Is also useful in a practical sense since it helps us improve!

THANK YOU!