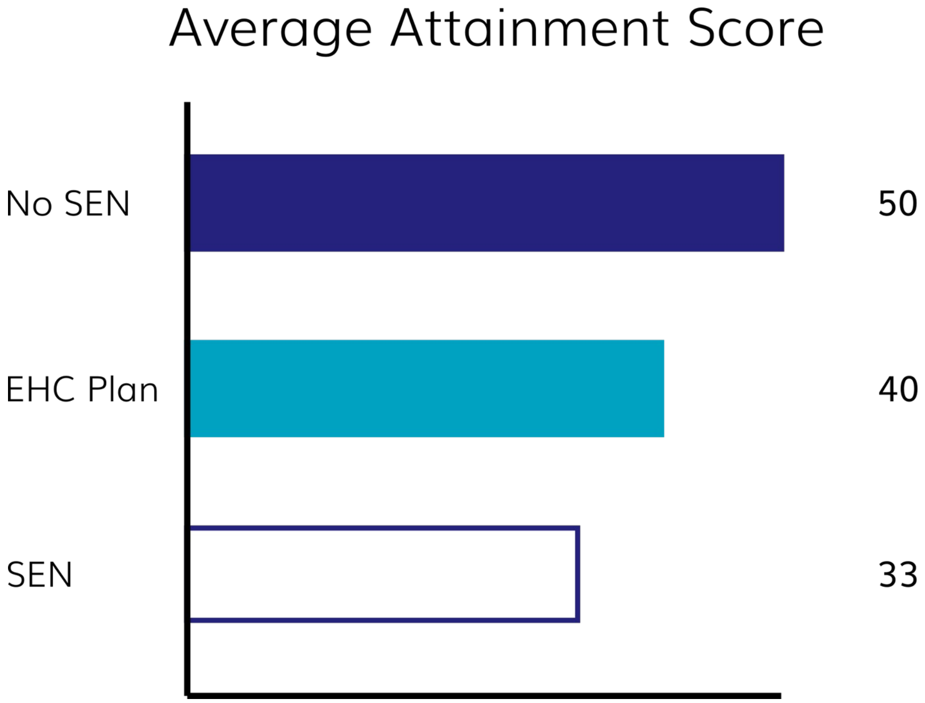
Education of Neurodiverse Students

William Jones
Ian Potter

EMBECOSM®



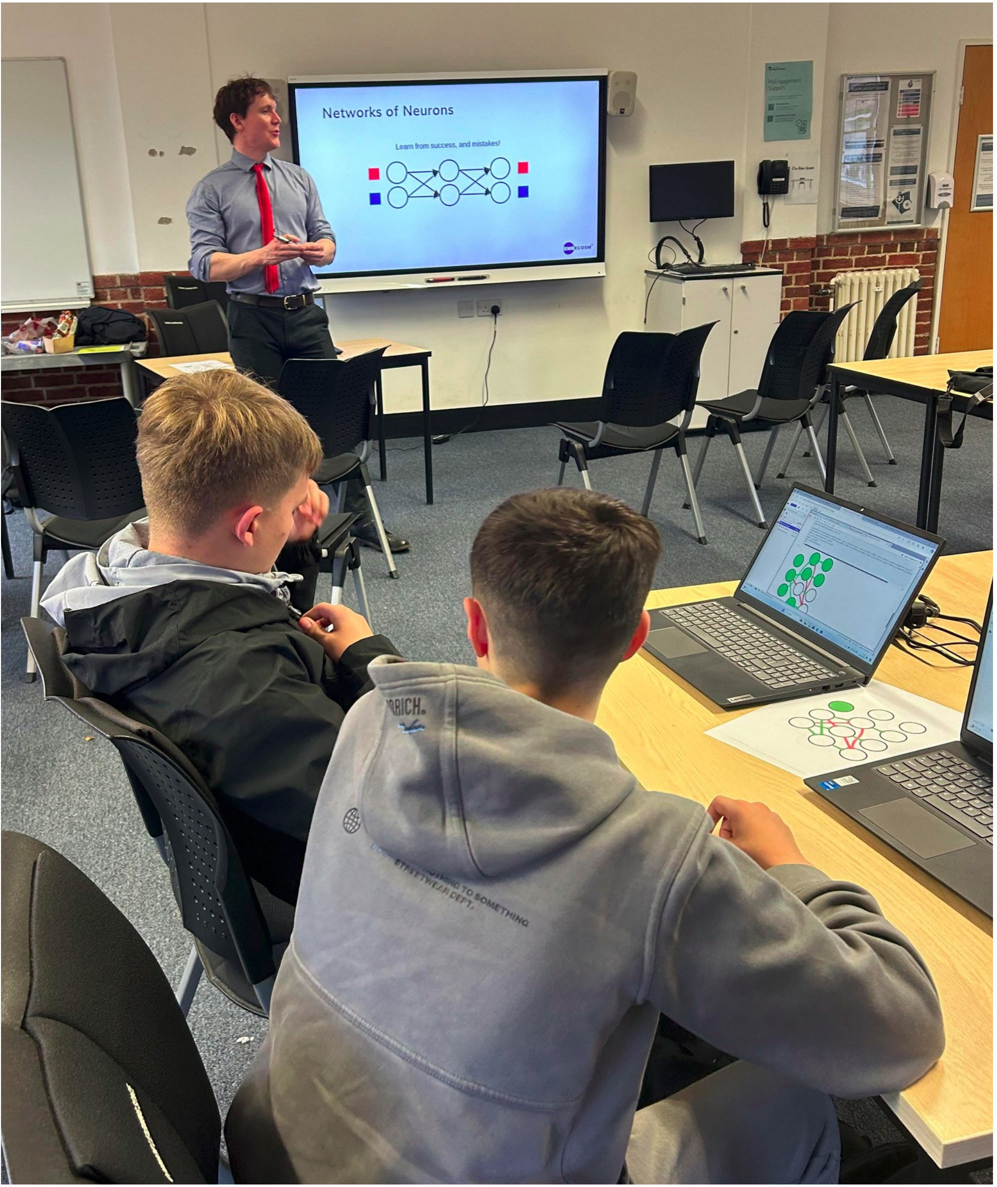
Source, Department for Education 2024

“Neurodivergent people are more likely to be in precarious employment or out of work, have less opportunity for advancement, can be misunderstood, and can be anxious about the transition to work”

(Doyle 2020)

“Creating new opportunities for learning for employment with neurodiverse young people, schools, businesses enables us to better understand approaches to young peoples wellbeing, how the curriculum can be a planned around missing skills in young peoples learning… how young people come to understand the unwritten social rules of the workplace”

(Doyle and Mcdowall 2022)

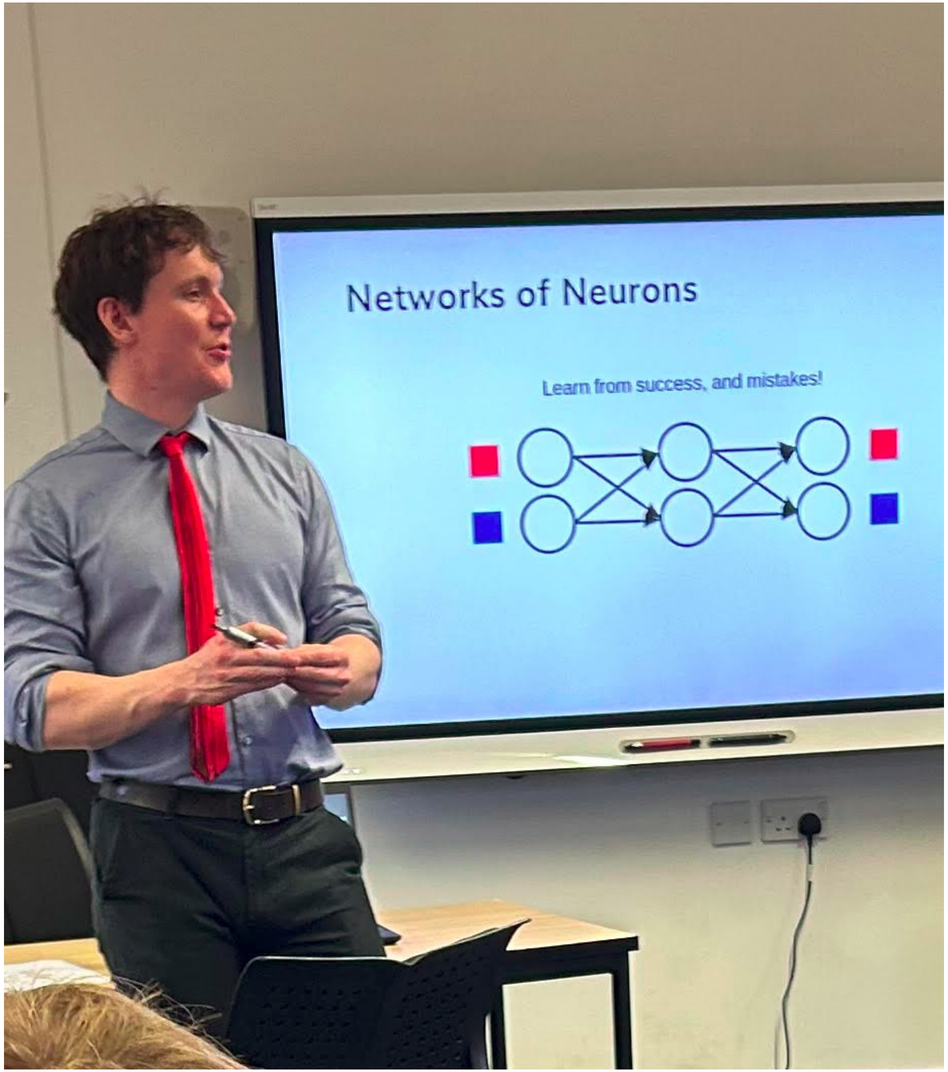


Collaboration with Redbridge school & Southampton University

Half day workshops

Artificial Intelligence

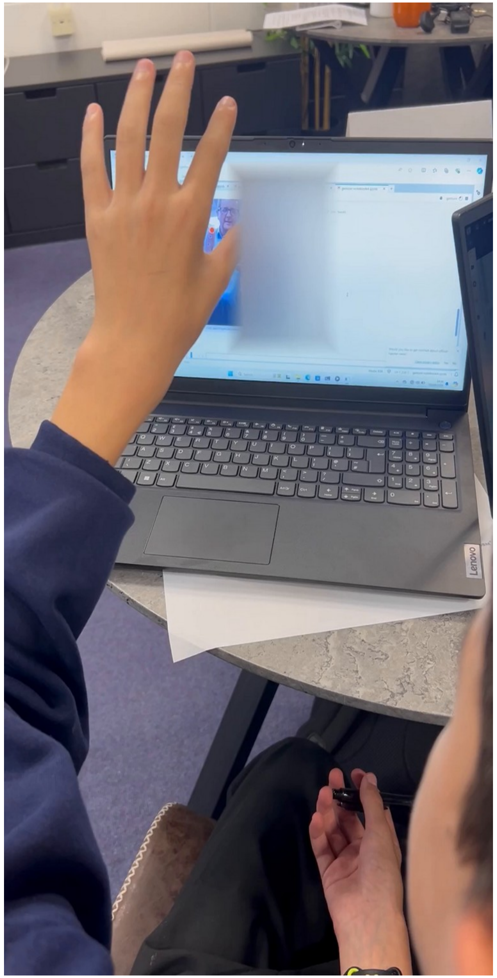
EMBECOSM®



Real Coding

Understand from first principles

Open Source!



HCI and Computer Vision

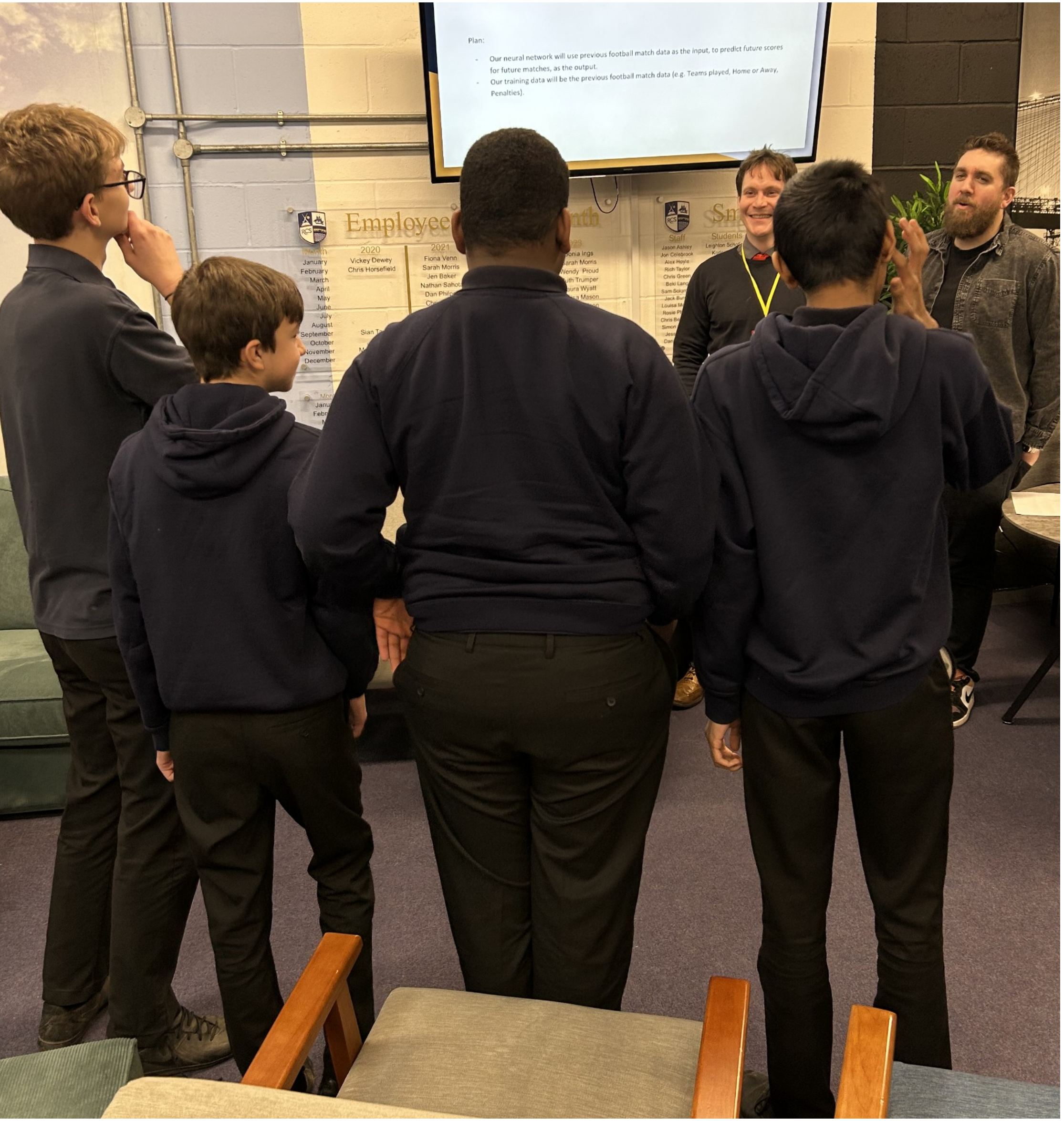
Build up a skeleton

Let the students innovate!

Neural Nets from scratch

Work through a simple example

Let the students innovate!



Plan:

- Our neural network will use previous football match data as the input, to predict future scores for future matches, as the output.
- Our training data will be the previous football match data (e.g. Teams played, Home or Away, Penalties).

Student Project

Football score prediction

Group collaborative effort

Impact on the students

*“The kids had a fantastic time, I couldn't believe how animated O*** was!”*

Richard Taylor, Teacher at Redbridge

“I got involved in programming…is fun and a great way to engage in helping change the world. Learnt how to make complex programme”

"This program was fun and informative"

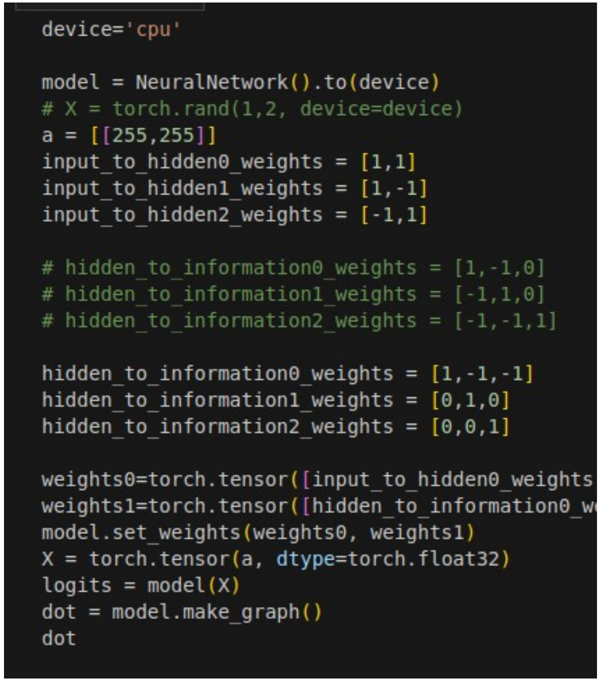
"I think the coding has worked for me because it has helped me to learn more about it"

EMBECOSM®

Feedback

“It would have been even better if we could discuss what each procedure does and its purpose”

“More examples of tasks related to coding”



```
device='cpu'

model = NeuralNetwork().to(device)
# X = torch.rand(1,2, device=device)
a = [[255,255]]
input_to_hidden0_weights = [1,1]
input_to_hidden1_weights = [1,-1]
input_to_hidden2_weights = [-1,1]

# hidden_to_information0_weights = [1,-1,0]
# hidden_to_information1_weights = [-1,1,0]
# hidden_to_information2_weights = [-1,-1,1]

hidden_to_information0_weights = [1,-1,-1]
hidden_to_information1_weights = [0,1,0]
hidden_to_information2_weights = [0,0,1]

weights0=torch.tensor([input_to_hidden0_weights
weights1=torch.tensor([hidden_to_information0_we
model.set_weights(weights0, weights1)
X = torch.tensor(a, dtype=torch.float32)
logits = model(X)
dot = model.make_graph()
dot
```

open source initiative®

Any questions?