

AI / ML – Technical Session 4 (09.10.2025)

Task 1 – Explore TensorFlow Playground:

- *Work through the other datasets in TensorFlow Playground (Circle, Exclusive or, Gaussian, Spiral).*
 - *For each one, try to come up with a network architecture that allows the problem to be solved –you can add and remove hidden layers and change the number of neurons in each layer.*
 - *Experiment with the different options such as activation functions, learning rates, etc.*
 - *What is the simplest network you can make that still manages to solve the problem?*
 - *What problem might arise if you use an unnecessarily complex network?*
 - *Look for uses of unfamiliar terminology in the TensorFlow Playground user interface and research their meaning.*
-

Dataset: Circle

1. Experiment summary:

- The circle dataset needs a non-linear model. A small network (e.g. 2 hidden layers with 4–6 neurons each) with the tanh or ReLU activation function works well.

2. Simplest working network:

- Hidden Layers: 2
- Neurons: 4 per layer
- Activation: tanh
- Learning Rate: 0.03

3. What happens if network is too complex:

- Adding too many layers or neurons may cause overfitting – the model memorises the training data but fails on new data. It also takes longer to train.
-

Dataset: Exclusive OR (XOR)

1. Experiment summary:

- Classic example of linear model failure. It needs a non-linear activation function and at least one hidden layer.

2. Simplest working network:

- Hidden Layers: 1
-

- Neurons: 4
- Activation: tanh
- Learning Rate: 0.03

3. Too complex network:

- If one adds too many neurons or layers, it may still work but training takes longer, and the model may start fitting noise instead of the pattern.
-

Dataset: Gaussian

1. Experiment summary:

- Simple and can often be solved with one layer as the data points are linearly separable.

2. Simplest working network:

- Hidden Layers: 1
- Neurons: 2
- Activation: sigmoid or ReLU
- Learning Rate: 0.03

3. Too complex network:

- A deeper network adds unnecessary complexity and doesn't improve accuracy — it just increases computation.
-

Dataset: Spiral

1. Experiment summary:

- Most complex dataset as it is highly non-linear. One will need more layers and neurons to “untangle” the spiral pattern.

2. Simplest working network:

- Hidden Layers: 3
- Neurons: 8–10 per layer
- Activation: tanh
- Learning Rate: 0.01

3. Too complex network:

- A bigger model might overfit and take a long time to converge. Likely to see unstable results depending on random initialisation.
-

Unfamiliar Terminology

Term	Meaning
Activation Function	Decides how a neuron reacts to input — adds non-linearity so the network can learn complex patterns (e.g. ReLU, tanh, sigmoid)
Learning Rate	Controls how big a step the model takes when updating weights. Too high = unstable, too low = very slow
Hidden Layer	Layers between input and output — they do the actual learning
Neuron	A single unit inside a layer that takes input, applies activation, and passes output forward
Epoch	One full pass through the data while training
Overfitting	When a model learns the training data too precisely but performs poorly on new data
Underfitting	When a model is too simple to learn the data's patterns