# Educational Escape Room Design for Machine Learning: Week 8 - Neural Networks

Targeted at teams (2–4 students), using **non-linear puzzle structure**, and integrating materials from **lectures, tutorials, and assigned readings**.

**🧩 Escape Room Title: “NeuroLock: The AI Gatekeeper”**

**🎯 Learning Objectives**

* Understand architecture and components of neural networks
* Interpret model training metrics
* Apply activation functions and hyperparameter tuning
* Connect theoretical concepts to applied debugging of models

**🧠 Narrative**

You are agents in an AI research lab. The lab’s autonomous security system, “NeuroLock,” has malfunctioned and locked out all staff. It refuses entry unless a neural model passes all internal validation checks. Your mission: analyse the broken neural net logs, correct faults, and restore access before the lockdown completes.

You have 60 minutes.

**⏳ Timing & Mode**

* 60 minutes
* **Collaborative**
* **Non-linear structure** with interconnected puzzles

**📘 Learning Clues from Course Materials**

* Activation function needed for Puzzle 1 is discussed in **Week 8 lecture slides**
* Correct learning rate is mentioned in the **suggested reading**: [LeCun et al. 2015]
* Tutorial worksheet includes a faulty diagram used in Puzzle 4

**🧩 Puzzle 1: Activation Malfunction**

**Task:** A part of the model uses an incorrect activation function. Use lecture materials to match output pattern to correct function (ReLU, sigmoid, tanh).

**Input-output mapping provided.**  
Students compare it to lecture slide graphs to identify mismatch.

**Clue:** Correct function name's first letter (e.g., R from ReLU) is clue #1.

**🧩 Puzzle 2: The Learning Rate Trap**

**Task:** Logs show exploding loss values. Use extract from [LeCun et al.] to find a recommended learning rate range.

“We found that rates above **0.01** often destabilise shallow networks…”

**Clue:** Learning rate used was 0.1. Input this into a script to simulate loss divergence and unlock a clue embedded in the generated plot title.

**🧩 Puzzle 3: Lost Layers**

**Task:** The number of layers used in the saved model architecture is incorrect. Students receive part of the .json model definition and must count layers by function type (Dense, Dropout, etc.).

{

"layers": [

{"type": "Dense", ...},

{"type": "Dropout", ...},

{"type": "Dense", ...}

]

}

**Clue:** Total number of **trainable layers × 2** = clue #3.

**🧩 Puzzle 4: The Vanishing Gradient Diagram**

**Task:** Students are given a backpropagation diagram taken from the Week 8 tutorial but altered subtly (missing derivative at one node). They must spot the error and justify the fix.

**Clue:** The index of the node with the missing gradient (e.g., Layer 2) = clue #4.

**🔐 Final Puzzle: Entering the Network**

Combine four clues (e.g., “R”, 1, 6, 2) into the unlock code:

First letter of correct activation + #layers × 2 + index of error

Final check: the AI asks them to interpret the **training log**:

Epoch 10/20

Loss: 0.02 | Accuracy: 95% | Val\_loss: 0.55 | Val\_accuracy: 60%

**Reflection task:** “Why might the model be overfitting? Suggest one fix.”  
→ Acceptable answers: regularisation, dropout, early stopping, etc.

**✅ Educational Breakdown**

| **Puzzle** | **ML Focus** | **Educational Skill** |
| --- | --- | --- |
| Activation | Function impact | Pattern matching and debugging |
| Learning rate | Hyperparameters | Reading-to-practice transfer |
| Layers | Network structure | Model architecture inspection |
| Backprop | Gradient flow | Critical analysis of training |