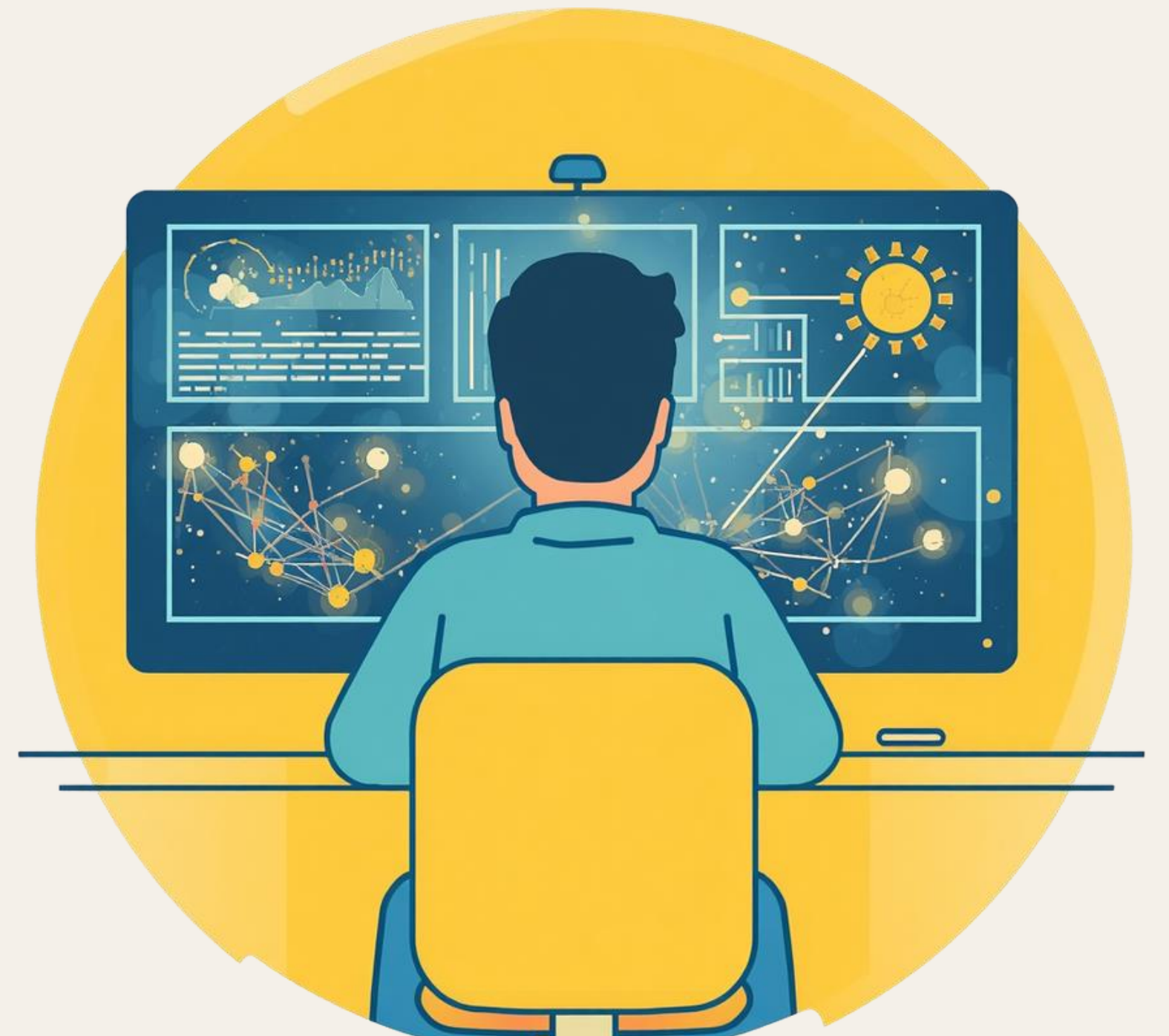


# Understanding Neural Network Architecture and Data Flow

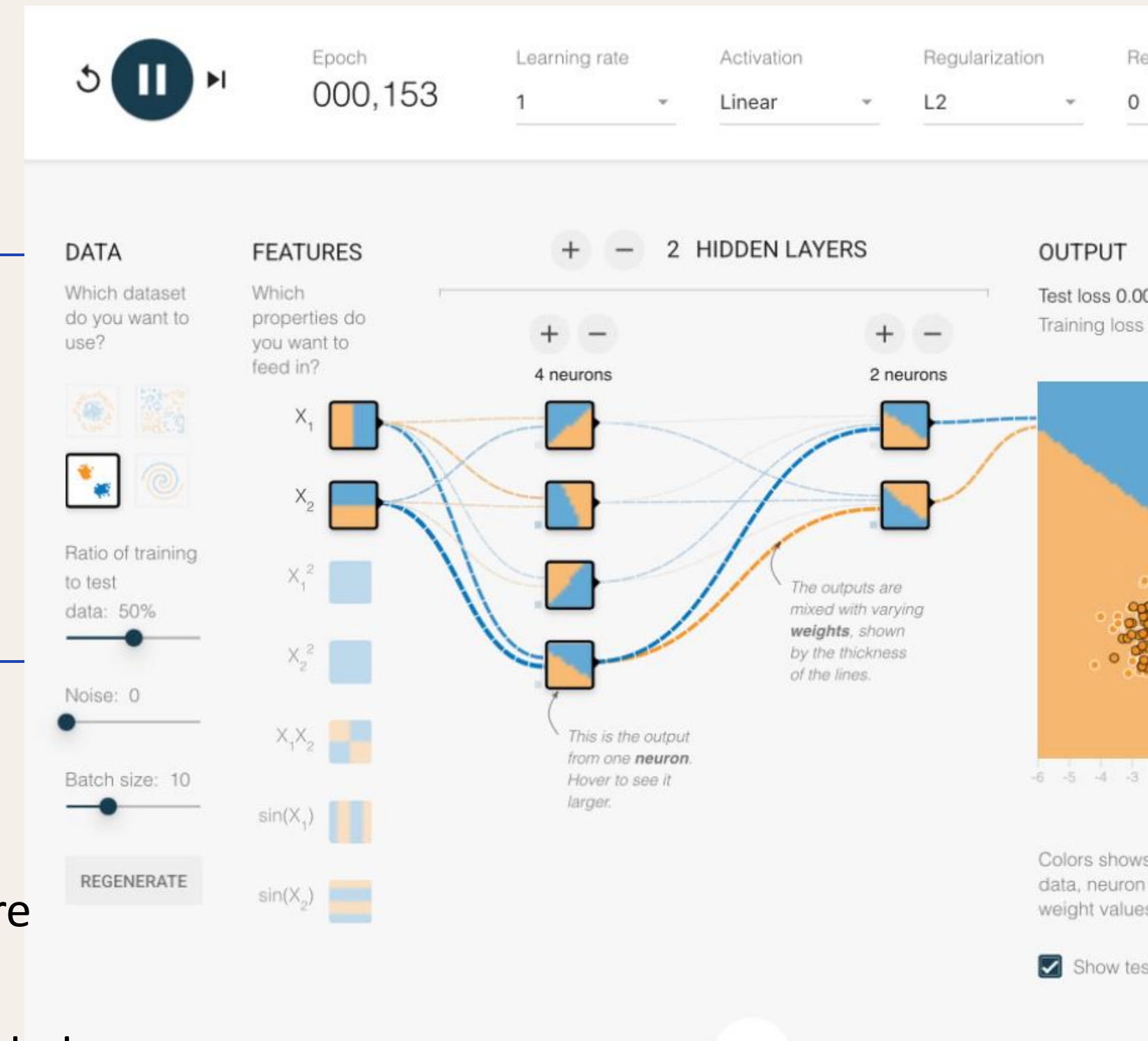
A visual overview of layers, neurons, and data propagation



# 02 - Visualizing the Network

*Input layer ( $X_1$ – $X_5$ ) connects through hidden layers to the output decision boundary*

The neural network processes raw input features  $X_1$  through  $X_5$ , each mapped to a dedicated input neuron. Data flows forward through two hidden layers — the first with 4 neurons and the second with 2 — before reaching the output layer. Weighted connections of varying thickness indicate influence strength, while the output layer generates a color-coded decision boundary classifying data points into distinct regions.



# 03 - Input Layer & Features

The input layer is the entry point for raw data into the network

The input layer serves as the gateway through which raw data enters the neural network. It consists of five neurons, each corresponding to a distinct input feature: X1, X2, X3, X4, and X5. Every data point is represented numerically, allowing the network to process it through subsequent layers.

Features may include numerical values or categorical variables that have been preprocessed into numeric form. Each input neuron holds the value of its respective feature and passes it forward via weighted connections to the first hidden layer, initiating the flow of information through the entire network.



Input

Features

# 04 - Hidden Layers

The neural network contains two hidden layers: the first with 4 neurons and the second with 2 neurons. Each neuron computes a weighted sum of its inputs plus a bias term, then applies an activation function to produce its output. In this architecture, a linear activation function is used. These layers progressively extract and transform features, building increasingly abstract representations of the input data to enable accurate classification.

*Neurons apply weights, biases, and activations to extract patterns*

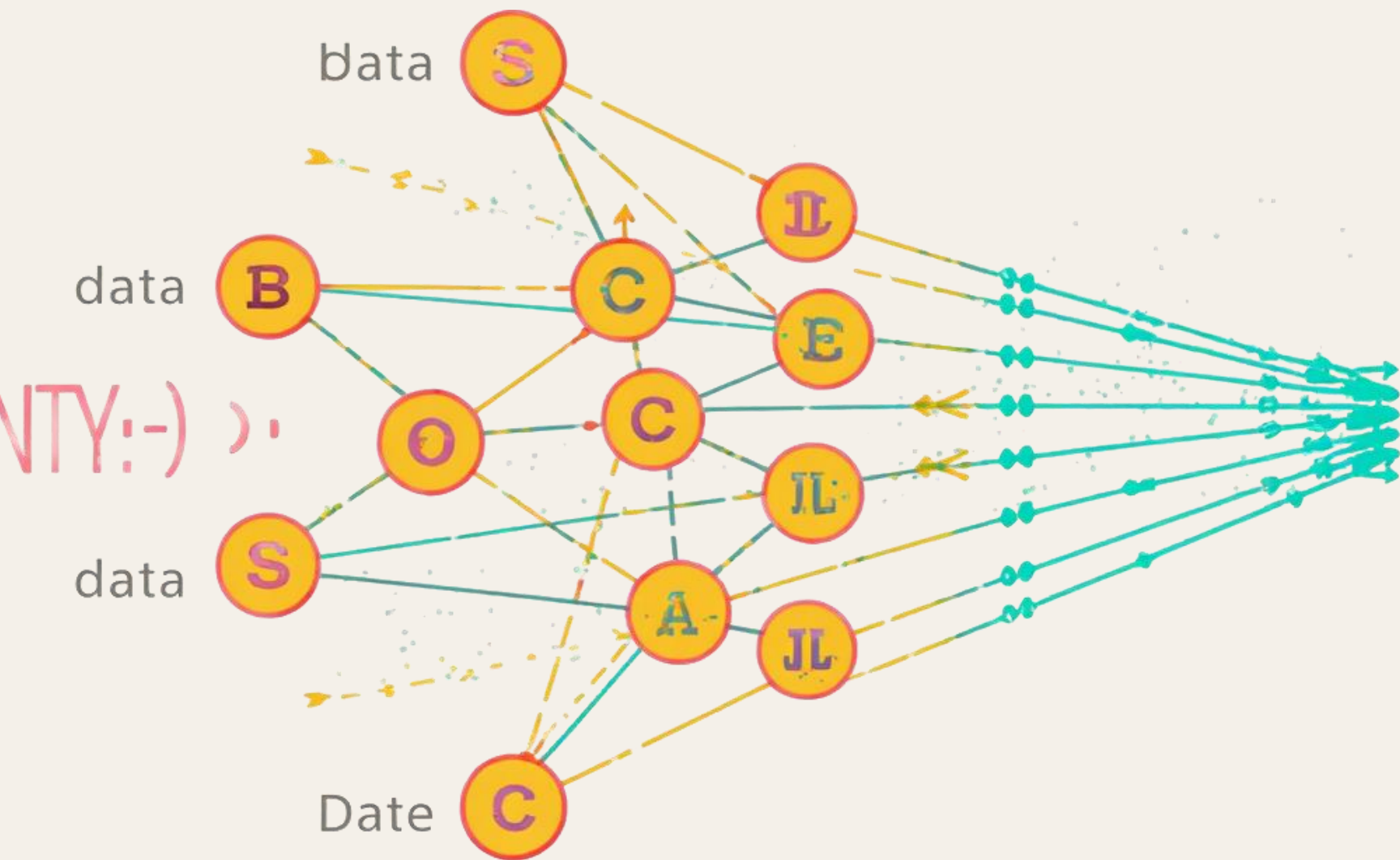
Hidden layers transform raw features into meaningful representations through weighted sums and activation functions

Neurons

Activation

# 05 - Forward Propagation

In forward propagation, data moves from the input layer through each hidden layer to the output layer. At every neuron, inputs are multiplied by their respective weights and summed together with a bias term. This weighted sum is then passed through an activation function, introducing non-linearity into the model. The resulting output of each layer becomes the input for the next layer. Weighted connections vary in thickness and color to indicate weight magnitude and influence. This sequential process repeats through all hidden layers until the output layer produces a final prediction or classification score.



*Each layer transforms data through weighted sums, biases, and activation functions*



# 06 - Output Layer & Prediction

The output layer is the final stage of the neural network, producing scores or probabilities for each class. In this architecture, the output layer generates a classification result visualized as a color-coded decision boundary — blue and orange regions represent distinct classes.

Output values are discretized to assign each data point to a class. The model's performance is measured by training loss and test loss, both recorded at 0.000, indicating a well-fitted model. The decision boundary reflects how the network separates input feature space into distinct regions based on learned weights and activations from the hidden layers.

*The output layer converts neuron activations into final class probabilities or scores*

Output

Classification

*Training loss and test loss both at 0.000 indicate strong model performance*

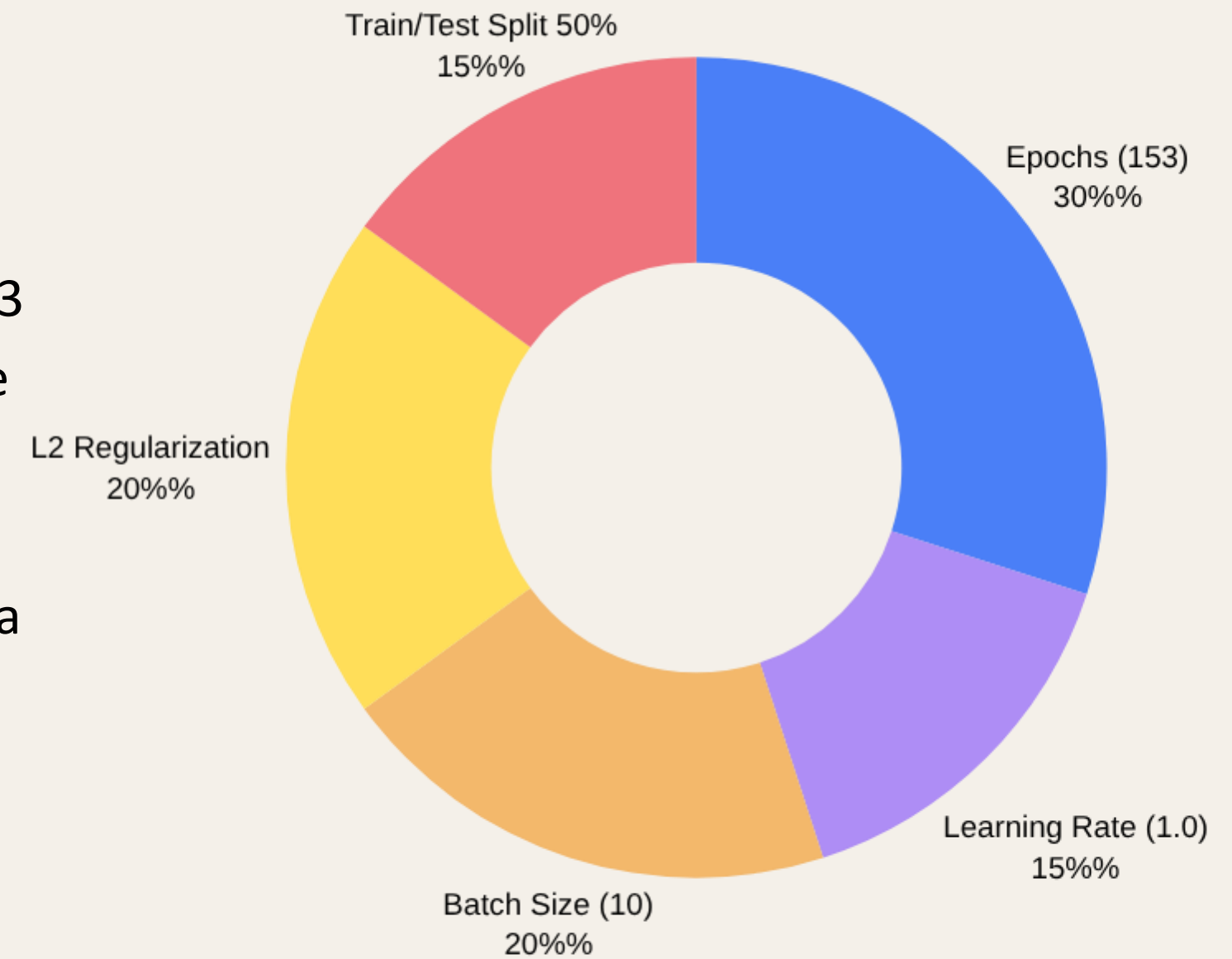
# 07 - Training Parameters and Impact

Key training parameters shape how a neural network learns. With 153 epochs, the model iterated extensively over training data to minimize loss.

A learning rate of 1 controls step size during optimization. L2 regularization prevents overfitting by penalizing large weights, while a batch size of 10 balances speed and gradient accuracy.

The 50/50 train-test split ensures unbiased evaluation. The linear activation function keeps transformations interpretable across all layers.

Together, these parameters achieved a training and test loss of 0.000, indicating a well-fitted model.



Source: AI-generated data. Replace or verify before use.

Visualizing neural networks provides deep insight into how models process and transform data. It helps practitioners understand the overall model structure — how layers connect and how information flows from input to output. By observing weight strengths, we can identify which connections are most influential in driving predictions. Visualization also clarifies decision boundaries, making classification logic transparent and interpretable. Furthermore, it serves as a powerful debugging tool, enabling teams to detect issues and iteratively improve model design and performance.



*Visualization reveals weight strengths, decision boundaries, and aids debugging neural network models.*

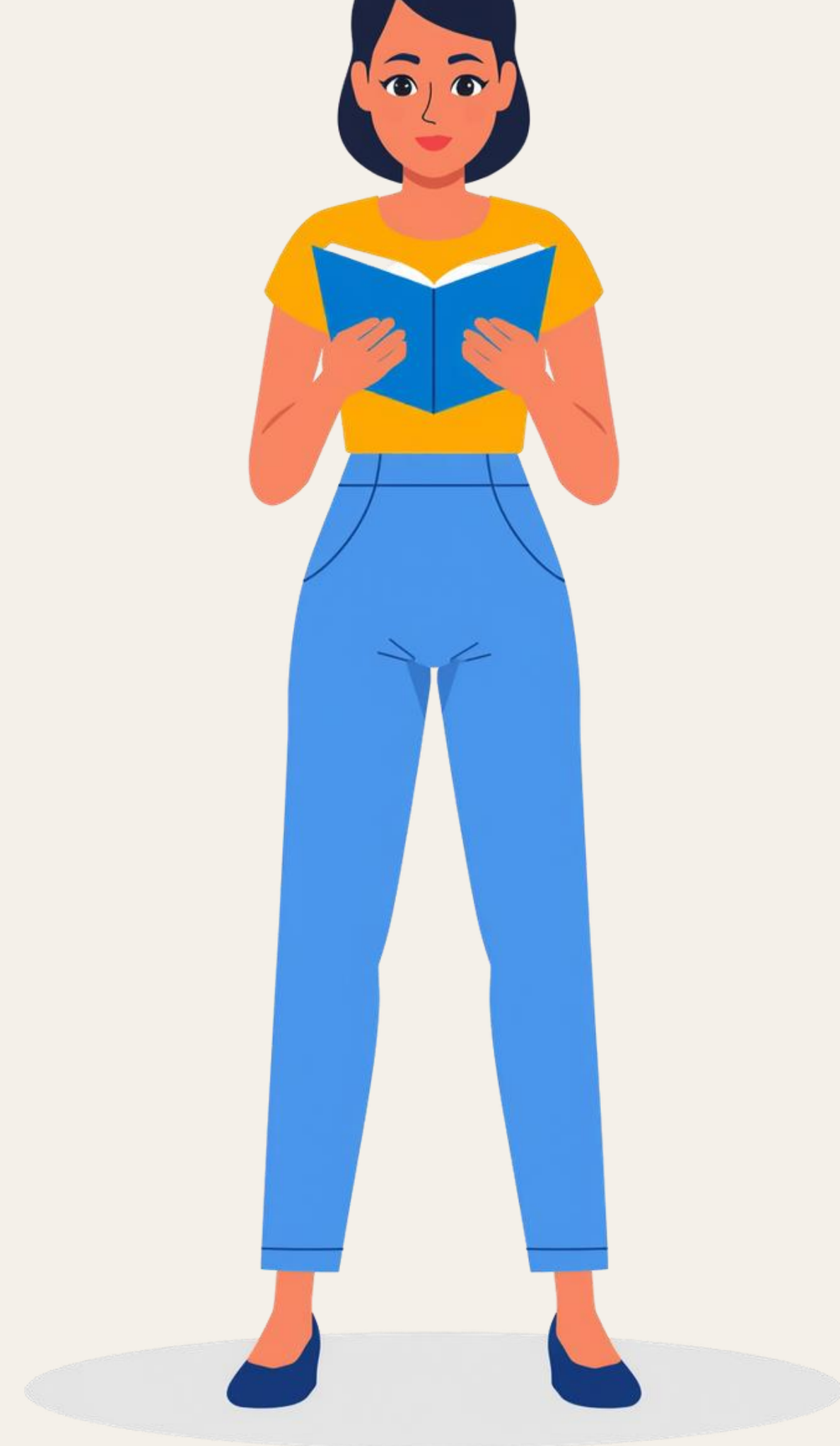
## 08 - Importance of Visualization

01 - Layered Transformation

02 - Pattern Extraction

03 - Visualization Benefits

04 - Explore Visual Tools



Neural

Networks