Mathematical Design of Neural Network Architecture

1. Input Layer

The number of nodes in the **input layer** is equal to the number of features (variables) in the dataset.

> Number of Input Nodes = n

> Example: If the dataset has 10 features, the input layer will have 10 nodes.

2. Output Layer

The number of nodes in the **output layer** depends on the type of task:

- Regression task: Usually 1 node (predicting a single continuous value).
- Binary classification: 1 node with a sigmoid activation function (outputting probabilities).
- Multi-class classification: Equal to the number of classes, *c*, with a softmax activation.

> Number of Output Nodes = c

> Example: For a 3-class classification problem, the output layer will have 3 nodes.

3. Hidden Layer (Empirical Heuristic)

The number of nodes in the hidden layer(s) is variable and depends on factors like dataset size, feature complexity, and potential overfitting or underfitting.

Empirical Rules for Hidden Layer Nodes:

Rule of Thumb: Average of Input and Output Nodes

> `Hidden Nodes = (Input Nodes + Output Nodes) / 2`

> Example: If there are 10 input nodes and 3 output nodes, then:
>
> > `Hidden Nodes = (10 + 3) / 2 = 6.5 ≈ 7 (rounded)`

2/3 Rule: Two-thirds of Input Nodes

> `Hidden Nodes = (2/3) × Input Nodes`

> Example: For 10 input nodes:
>
> > `Hidden Nodes = (2/3) × 10 = 6.67 ≈ 7`

Twice the Number of Input Nodes

> Hidden Nodes = 2 × Input Nodes

> Example: For 10 input nodes:
>
> > Hidden Nodes = 2 × 10 = 20

Overall Formula for Network Design

For multiple hidden layers, distribute the nodes across the layers using an equal or decreasing pattern.

Key Considerations

- **Trial and Error**: Optimal nodes are often found by experimenting.
- **Avoid Overfitting**: Too many nodes can cause overfitting; too few may result in underfitting.
- **Cross-Validation**: Use cross-validation to test different architectures.