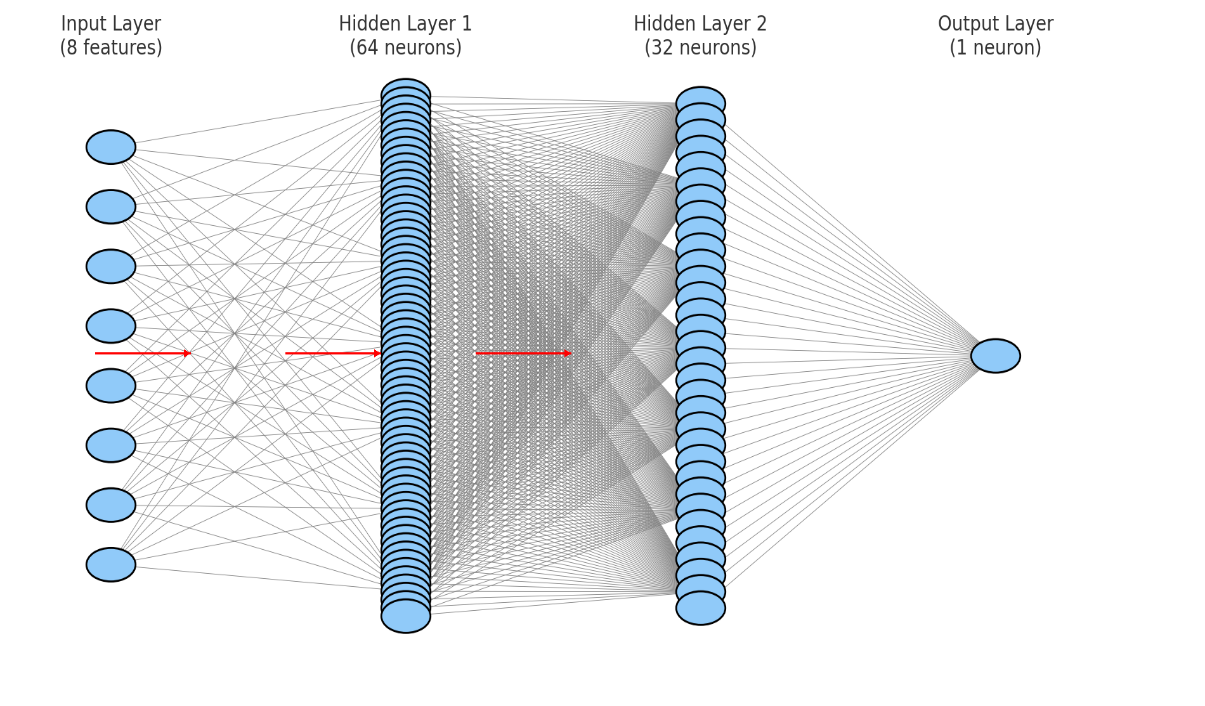
**MODEL ARCHITECTURE AND DESIGN**

* Designed and implemented a deep feedforward neural network.
* The network is specifically tailored for a regression task.
* The goal is to predict sales values.
* The model uses a combination of categorical and numerical features as input.
* It is built to learn complex relationships between the input features and the target variable (Sales).
* The learning is achieved through multiple layers of nonlinear transformations

**Model Architecture Visualization**



**Feedforward Neural Network**

**Input Layer**

The input layer accepts **8 features**, which were pre-processed using appropriate techniques such as **label encoding** for categorical variables and **standard scaling** for numerical features. These features represent crucial business information, including:

* Shipping Mode
* Segment
* City
* State
* Region
* Category
* Sub-Category
* Shipping Time

Each of these input nodes represents one feature, forming the first interface with the data.

**Hidden Layers**

The model comprises two fully connected hidden layers:

1. **Hidden Layer 1**:
   * **Number of neurons**: 64
   * **Activation Function**: ReLU (Rectified Linear Unit)  
     This layer introduces non-linearity and helps the model capture complex patterns in high-dimensional feature interactions.
2. **Hidden Layer 2**:
   * **Number of neurons**: 32
   * **Activation Function**: ReLU  
     This layer refines the representation learned in the first hidden layer and reduces dimensionality while preserving essential learned features.

Each neuron in these layers is connected to every neuron in the previous layer (dense connections), and biases are added to each neuron to help with learning non-zero-centered data distributions.

**Output Layer**

* **Number of neurons**: 1
* **Activation Function**: Linear  
  This layer outputs a single continuous value representing the predicted sales for the input instance.

**Model Summary**

* **Architecture**: Feedforward Neural Network (Multilayer Perceptron)
* **Total Layers**: 4 (including input and output)
* **Loss Function**: Mean Squared Error (MSE)
* **Optimizer**: Adam (adaptive learning rate optimization)
* **Evaluation Metric**: Mean Absolute Error (MAE), RMSE (Root Mean Squared Error), and R² Score.

**Model Training**

The model was trained using batch gradient descent with backpropagation. During training:

* We used a batch size of 32.
* Learning rate and other hyperparameters were tuned for better convergence.
* Dropout or early stopping mechanisms were optionally experimented with to avoid overfitting (if needed).
* The model was trained over multiple epochs until validation loss plateaued.

**Training Process, Loss Curves & Evaluation Metrics**

The dataset was split with **80% for training** and **20% for testing**. During training, **loss curves** (based on Mean Squared Error) were used to track learning progress and detect overfitting

After training, the model was evaluated using:

* **MAE** (Mean Absolute Error): Average prediction error.
* **RMSE** (Root Mean Squared Error): Penalizes larger errors more.
* **R² Score**: Measures how well input features explain the variance in sales.

These metrics offered a well-rounded view of the model’s predictive performance.