Defined Architecture of ANN Model

Project: Customer Churn Prediction

Model Type: Artificial Neural Network (Multi-Layer Perceptron)

Optimization Strategy: SMOTE & Threshold Tuning

1. Executive Summary of Architecture

To address the predictive modeling task, I designed a deep feed-forward neural network with a "funnel" architecture. The model is specifically optimized to handle the class imbalance inherent in churn data. By integrating **SMOTE (Synthetic Minority Over-sampling Technique)** for training and implementing a **decision threshold of 0.3**, the architecture prioritizes **Recall (Sensitivity)** to maximize the identification of at-risk customers.

2. Detailed Layer Specifications

The model consists of an input layer, three hidden layers, and one output layer, totaling **3,077 parameters**.

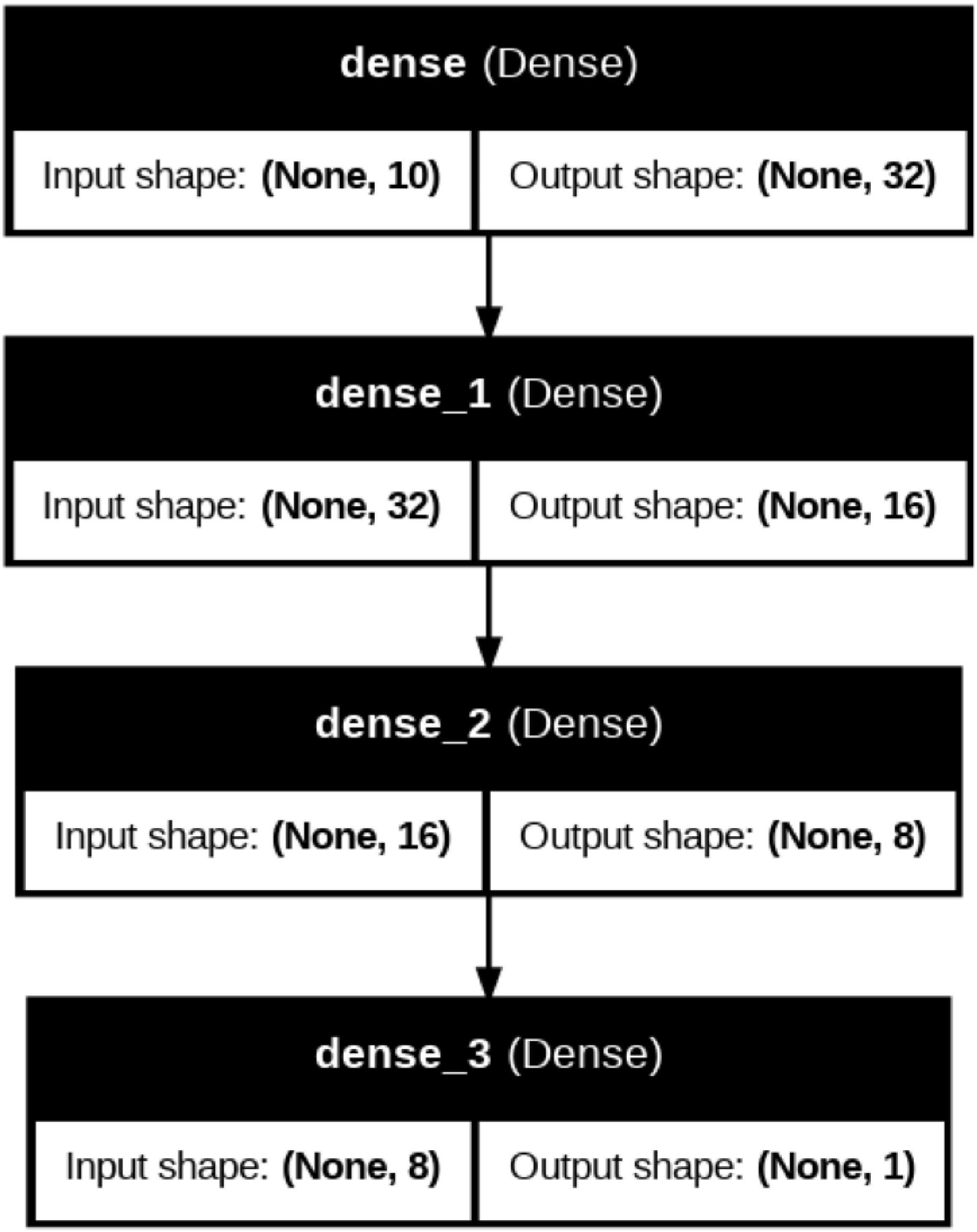
- **Input Layer (Implicit):**
  - **Input Features:** 10 (Derived from preprocessed customer attributes including Tenure, Monthly Charges, and Contract Type).
  - **Preprocessing:** One-Hot Encoding for categorical variables and Min-Max scaling for numerical variables.
- **Hidden Layer 1 (Feature Extraction):**
  - **Neurons:** 32
  - **Activation:** ReLU (Rectified Linear Unit)
  - **Function:** Acts as the primary feature extractor, expanding the input space to capture non-linear interactions between customer demographics and service patterns.
- **Hidden Layer 2 (Dimensionality Reduction):**
  - **Neurons:** 16
  - **Activation:** ReLU
  - **Function:** Compresses the feature information, forcing the model to focus on the most significant signals contributing to churn.

- **Hidden Layer 3 (Refinement):**
    - **Neurons:** 8
    - **Activation:** ReLU
    - **Function:** A final refinement layer added to stabilize the learning process on the SMOTE-augmented dataset before classification.
- **Output Layer (Classification):**
    - **Neurons:** 1
    - **Activation:** Sigmoid
    - **Function:** Outputs a probability score ($P \in [0, 1]$) indicating the likelihood of customer churn.

3. Optimization & Training Strategy

- **Handling Imbalance (SMOTE):**
    - The original dataset was imbalanced (4132 Non-Churn vs. 1502 Churn).
    - **Strategy:** Applied SMOTE to synthesize minority class examples during training, resulting in a perfectly balanced training set (4132 vs. 4132). This prevents the model from biasing towards the majority class.
- **Hyperparameters:**
    - **Optimizer:** Adam (Adaptive Moment Estimation) for efficient gradient descent.
    - **Loss Function:** Binary Cross-Entropy (Log Loss).
    - **Batch Size:** 32.
    - **Epochs:** 50 (with Early Stopping capabilities).
- **Decision Threshold Optimization:**
    - **Standard Threshold:** 0.5
    - **Optimized Threshold: 0.3**
    - **Rationale:** In customer churn retention, a "False Negative" (missing a churner) is more costly than a "False Positive" (investigating a loyal customer). By lowering the classification threshold to 0.3, the architecture is tuned to be highly sensitive.
    - **Result:** This architecture achieved a **Recall of 81%** for the Churn class, successfully identifying 298 out of 367 actual churners in the test set.

4. Architecture Diagram



dense (Dense)
Input shape: (None, 10)
Output shape: (None, 32)
dense_1 (Dense)
Input shape: (None, 32)
Output shape: (None, 16)
dense_2 (Dense)
Input shape: (None, 16)
Output shape: (None, 8)
dense_3 (Dense)
Input shape: (None, 8)
Output shape: (None, 1)