ANN Model Architecture for Customer Churn Prediction

This document specifies the architecture of an Artificial Neural Network (ANN) model optimized for predicting customer churn.

Model Architecture Overview

The ANN model is structured as follows:

1. Input Layer:
The input layer consists of features derived from a preprocessed dataset, which excludes the target variables ('Churn_Yes'). The number of features in the input layer corresponds to the number of input variables in the dataset.

2. Hidden Layers:
The model has three hidden layers with ReLU activation functions. Dropout regularization is applied to prevent overfitting.

3. Output Layer:
The output layer is a single node with a sigmoid activation function for binary classification, predicting customer churn (Yes or No).

Input Layer

The input layer has a number of nodes equal to the number of features in the dataset. This is dynamically determined based on the input dataset, where each feature corresponds to one node. For example, if there are 10 features in the dataset, the input layer will contain 10 nodes.

Hidden Layers
The model has three hidden layers with the following configuration:
1. First Hidden Layer: 64 units with ReLU activation function, followed by a dropout layer with a 20% dropout rate.
2. Second Hidden Layer: 32 units with ReLU activation function, followed by a dropout layer with a 20% dropout rate.
3. Third Hidden Layer: 16 units with ReLU activation function, followed by a dropout layer with a 20% dropout rate.

Output Layer
The output layer has a single unit with a sigmoid activation function. This layer outputs a value between 0 and 1, which is interpreted as the probability of customer churn ('Churn_Yes'). A threshold of 0.5 is used to classify whether a customer will churn or not.

Model Optimization
The model is optimized using the Adam optimizer with a learning rate of 0.001, and the loss function used is binary crossentropy. This setup is well-suited for binary classification tasks such as customer churn prediction.
Regularization and Generalization

To prevent overfitting, dropout regularization is applied after each hidden layer with a rate of 0.2. This helps the model generalize better to unseen data by randomly deactivating neurons during training.

Training Details

The model is trained for up to 50 epochs with early stopping based on validation loss to avoid overtraining. The training data is split into 80% training and 20% validation sets. The batch size used during training is 32.