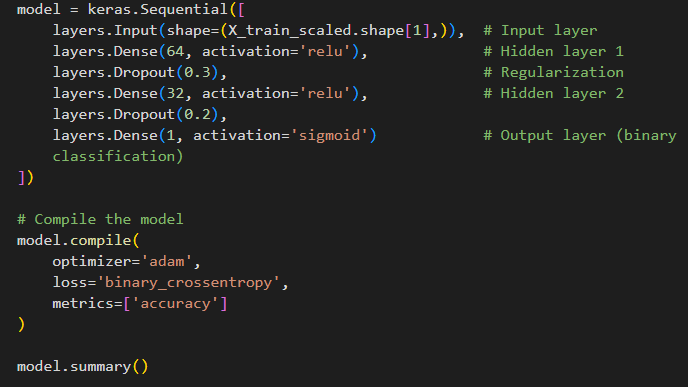
# The Architecture of the Artifical Neural Network (ANN) Model

The telecom churn prediction Artificial Neural Network (ANN) has a well-organized architecture that is optimized with respect to predictive performance and generalization. The network starts with an input layer that receives the entire collection of scaled customer attributes and all necessary information like the tenure, monthly charges, the type of contract, and the internet service are reflected in the learning process.



*Figure 1. ANN model Architecture*

The initial hidden layer is comprised of sixty-four neurons that are enabled using the ReLU function. This fairly broad layer encompasses a large spectrum of nonlinear interactions between features that tend to be the basis of complex behavioral patterns in churn scenarios. Immediately following this layer is a dropout rate of thirty percent that aims to prevent overfitting by randomly turning off a percentage of the neurons in the training process, which pushes the model to learn more generalizable patterns.

The second concealed layer reduces to thirty-two neurons, again with the ReLU activation function. Such dimensionality reduction obliges the network to extract high-level abstractions in the first layer into more condensed and informative representations. An additional dropout rate of twenty percent is used here further strengthening the regularization approach and justifying the model to generalize well on the unseen data.

Lastly, the output layer has one neuron whose activation is a sigmoid. The configuration is appropriate in binary classification tasks, since the probability value between zero and one is the output of the configuration, and it is the probability that a customer will churn. This probability is converted into a certain churn or non-churn prediction by using a threshold.

It was trained using the Adam optimizer with adaptive learning rates to ensure efficient convergence and binary cross-entropy as the loss function to show the binary character of the prediction task. The most important measure of evaluation was accuracy to track the predictive performance directly during training and validation.

During the course of developing the churn prediction model with an Artificial Neural Network, several challenges were encountered. One of the first issues involved setting up the coding environment and managing package installations. Incompatibility between library versions and runtime errors often disrupted progress. The use of a virtual environment and a clear requirements file provided a reliable solution. Understanding the wide range of hyperparameters in neural networks proved to be a significant difficulty. Choosing the number of layers, units per layer, activation functions, learning rate, and batch size required experimentation, and without prior experience, it was often unclear which configurations would lead to better performance. At times, the model failed to improve simply because of an unsuitable parameter choice. This challenge was managed by starting with simple architectures, gradually testing different settings, and relying on established practices from the literature to guide parameter selection.

From a modeling perspective, the imbalance in churn versus non-churn cases affected predictive performance. This was addressed by applying class weighting and oversampling techniques. Overfitting and underfitting also emerged at different stages, requiring regularization methods, dropout layers, and early stopping mechanisms to maintain model generalizability. Interpreting results presented an additional challenge, particularly in explaining outputs to a non-technical audience. To address this, supporting tools such as confusion matrices and feature importance techniques were introduced to provide clearer insights. Finally, reproducibility proved important for continuity. Setting random seeds, saving models, and maintaining concise documentation were adopted as best practices. Collectively, these strategies reduced barriers and ensured the project could progress efficiently despite initial challenges.