# ANN for Clustering Customer Segments

M. Abhiram, 230150015
K. Ashmita, 230150014
B. Cherish, 230150007

## Unsupervised Learning with a Neural Network

Clustering is a type of unsupervised learning method of machine learning. Unsupervised learning finds the internal relationships between the features of the dataset, which tells us how exactly the data is structured internally. Neural Networks were traditionally used for Supervised Learning, where we are given labels for each data point and try to fit a model which predicts the labels as close to the ground truth as possible. The ANN trains on a predefined loss function by minimizing it in every iteration.
A lot of Artificial Neural Networks have been designed for clustering and unsupervised learning where there are no labels assigned to the data points. Most of these implementations such as Self-Organizing Maps, use competitive learning instead of supervised learning. In this project, we have a implemented a Neural Network which clusters based on an **internal evaluation index** which acts as a loss function, similar to supervised learning.

## Neural Network Architecture

The ANN which we implemented consists of a simple neural network architecture along with an internal clustering evaluation index, which acts as the loss function. One huge advantage of implementing this is that we do not need to fix the number of clusters we would want the data to organize into. Unlike K Means algorithm, we can just have a rough estimate or an upper bound on the number of clusters. We let the output layer of the ANN to have C neurons, where each output neuron tells us how likely it is for the point to belong to that cluster. Then we apply a softmax regressor on the output layer to get the probabilities, which gives us a **Fuzzy Clustering** of the dataset. We could also do a Hard Clustering by assigning data points to those clusters whose output neurons have the maximum probability.

## Loss Function

As we have discussed above, we need to define a loss function for any supervised learning method. For this project we have decided to go with **Min-Max-Jump K Means Loss** based on Fuzzy Clustering to be the loss function which we would like to minimize with each iteration through Batch Gradient Descent with Momentum and RMS Prop. This loss function would tell us how apart are the cluster points away from the cluster centers.

$$\mathcal{L}_s = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{1}\{c_i = k\}\|x_i - \mu_k\|^2(2 - P(x_i))$$

where N is the number of points in the data X, K is the number of clusters, $\mu_k$ is the centroid of cluster $k$, $c_i$ is the label of data point $x_i$. We first use Hard Clustering to decide which cluster data point $x_i$ belongs to, then use Soft Clustering to decide the probability of data point $x_i$ belongs to the cluster. $P(x_i)$ is the probability.

# Internal Clustering Evaluation Index

Evaluating the final clusters assigned by the ANN is another task as we would require another evaluation index. Many indices have been proposed for both internal and external evaluation. Internal evaluation tells us how good the data has been clustered, whereas external evaluation compares the formed clusters with the ground truth. Frequently used indices for internal evaluation are Silhouette Coefficient, Davies-Bouldin Index, Dunn Index, etc. For this project, we have decided to go with the **Silhouette Coefficient**. For one data point, the Silhoutte Coefficient is calculated as

$$s = \frac{a - b}{\max(a, b)}$$

where $a$ is the mean of all the distances between the sample data point to all other points present in its cluster and $b$ is the mean of all the distances between the sample data point to all the points present in the next closest cluster.
Silhouette Coefficient for a clustering is the average of Silhouette Coefficients for each and every data point in the dataset.

The Silhouette Coefficient would lie in the range of **[-1,1]** where a value closer greater 0.5 would represent an optimal clustering, between 0 and 0.5 would represent moderate clustering and a value less than 0 would represent a substandard clustering. This can be used for comparing different different clusters formed by different ANN architectures

# Dimensionality Reduction

We have also implemented an Autoencoder, which is basically a neural network which tries to deconstruct the given data into smaller chunks and **reconstructs** it back again. Here the loss function would be the error between the reconstructed data and the original dataset. This helps in dimensionality reduction as we could use only the encoded data obtained from the **bottleneck layer** instead of the original dataset, which could be hard for visualizing clusters and would also increase the **computational complexity**.

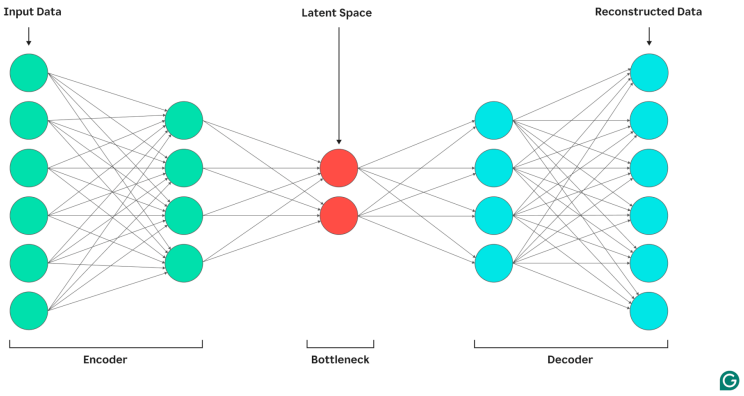


Figure 1: Autoencoder's Architecture.

## Exploratory Data Analysis

We are using the Mall Customers Segmentation dataset on Kaggle for this project. We first load the dataset into a Pandas dataframe and perform exploratory data analysis on the dataset. Overall, the distributions are fairly proportional between Men and Women. On an average, men are slightly older than women and tend to have higher incomes, while women tend to spend more than men. Based on the correlations and scatterplots, the variables in the dataset do not have very strong relationships with each other. There is a weak negative association between Age and Spending Score of -0.33 which suggests that as Age increases Spending Score decreases which can also be observed in the scatterplot. A dense cluster in this scatter plot between Annual Income and Spending score suggests that A significant portion of customers fall within a specific range of both Annual Income and Spending Score.

## Implementation

We first standardize all the numerical features of the dataset using scikit-learn. We design 3 types of autoencoders, where the first one's architecture is quite simple $4 \rightarrow 2 \rightarrow 4$ neurons, the second one's is $4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4$ neurons, whereas the last one's is a more complex one with $4 \rightarrow 4 \rightarrow 3 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 4$ neurons, with tanh as our activation function. As we could guess, the loss function of the first autoencoder is very high, and the last one's loss is low. We could fairly conclude that there is **underfitting** in the first one and a heavy **overfitting** in the last one, on the feature 'Gender'. We could also see that the data is following some pattern in the 2nd plot whereas it is completely random in the 1st plot. So, **we go with the second architecture** as we proceed forward as it looks to be neither underfitting nor overfitting. We take a rough guess at the upper bound on the number of clusters that could be formed to be 7. We could change it accordingly by looking at how well the clusters are formed. So now the number of output neurons in our ANN would be equal to 7.
We have decided to go with 4 different ANN architectures, with our motivation coming from the research paper Clustering with Neural Network and Index. After training all the 4 different ANNs and finding the Silhouette Coefficient of the Clustering, we find that the **Silhouette Coefficient is the highest in the 1st Neural Network** with a score of 0.46, which is a moderately good clustering based on the definition.

## Analyzing the Clusters

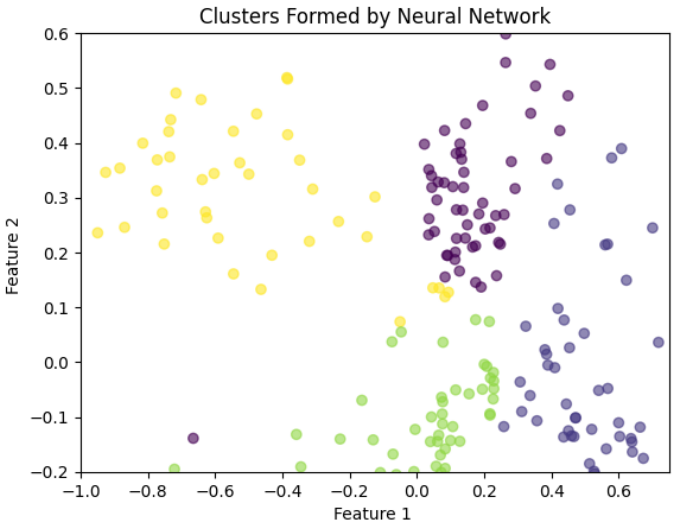


Figure 2: Clustering Results. Here we can see that the ANN has divided the dataset into 4 clusters

**Cluster 1 (Purple)**
**Dominant Gender: Male**
**Mean Age**: **55**, ( Variance: 85.4 )
**Mean Income**: **49.4**, ( Variance: 268 )
**Mean Spending Score**:**42.2**,(Variance:256)

**Cluster 2 (Lilac)**
**Dominant Gender: Male**
**Mean Age**: **27** ( Variance: 44.8 )
**Mean Income**: **32.4**, ( Variance: 171 )
**Mean Spending Score**:**59.5**,(Variance: 527)

**Cluster 3 (Green)**
**Dominant Gender: Male**
**Mean Age**: **30.6**, ( Variance: 33.9 )
**Mean Income**: **78.6**, ( Variance: 273 )
**Mean Spending Score**:**71.8**,(Variance:330)

**Cluster 4 (Yellow)**
**Dominant Gender: Female**
**Mean Age**: **41.4**, ( Variance: 116 )
**Mean Income**: **85.5**, ( Variance: 286 )
**Mean Spending Score**:**21**,(Variance:195)

- Cluster No. 1 mostly consists of older men who make decent amount of money, with a decent spending score as well. Marketing more **vintage** and **antique** products could well be beneficial in this cluster.
- Cluster No. 2 mostly consists of younger men who earn a bit less but still are willing to purchase products from the mall. This cluster could be well maximized by marketing products with the **highest value to price ratio**, because it looks like the customers are willing to spend, but are constrained due to their income..
- Cluster No. 3 is the most important cluster which consists of younger men with a very high income and also the highest spending score. Marketing **trendy** and **fashionable** products could be the way to go in this cluster.
- Cluster No. 4 mostly consists of middle aged women who make the highest amount money, buy also the least spending score out of all. This has a potential to be a very important cluster, but with a spending score so less, the companies should start marketing more **expensive** products as that could be the only to make the most out of this cluster as the spending score can't go any lower.

## Contribution of Each Member

We think that all three members of the group have contributed equally to this wonderful project. It took a lot of persistent discussions and brainstorming to come up with the implementation strategy. We collectively worked on data preprocessing, EDA, designing autoencoders and ANNs with different architectures, and later, visualizing the formed clusters, interpreting them, and writing this well-documented report. **Coming together as a group and collaborating with each other, was what made us set the bar higher**.

## References Used


- Clustering with Neural Network and Index, Gangli Liu, Tsinghua University Link
- Mall Customer Segmentation Data ,Vijay Choudhary, Kaggle Link
- Matplotlib 3.10.1 documentation, Link
- seaborn: statistical data visualization Link
- PyTorch documentation Link