GRAPHICAL NEURAL NETWORKS AND TRANSFORMERS

INTERNSHIP REPORT IN MACHINE LEARNING

17th May 2021-17th august 2021

Submitted by:

Sibasish Padhy, a 4th-year undergraduate student of integrated 5-year MSC in Mathematics And Computing, Bit Mesra.IMH/10023/17.

Internship at Wynum.

Table of contents

Acknowledgment

I undertook a 3-month internship project at Wynum Automation and completed the internship under the guidance of Abhishek Mamdapure, Disha Patil, and Omkar Patil.

I am grateful to all the staff members at Wynum for their patience and assistance during my learning phase at their company. It was a great learning experience for me to get exposure to the latest technologies/methodologies applied in ML and AI.

Information About the company.

Wynum is an AI-based platform that brings efficiencies to operations. The company caters to the manufacturing industry-medical, precision electronics, chemicals, Fashion and food, beauty, heavy engineering-supply chain management/operations function. They are building AI-based tools to enable industry 4.0 by aggregating multiple data acquisition technologies with AI-driven automation of analysis and decision making. They help industries improve efficiency, yields, and reduce costs to focus on their growth strategies.

The tasks I undertook included:

1.Working in PyTorch.

2.Learning Python from scratch and implementing it in real-life projects and solutions.

3.Exploring the concepts of graphical neural networks and transformers.

4.Learning how to implement various types of neural networks using PyTorch and object-oriented programming language.

5.Also tried to use my newly gained knowledge base on neural networks in PyTorch in a live competition in Kaggle having a good bounty.

6.learning about the ETL concepts on datasets done before training the model and also some case study on fashion mnist dataset.

7.Explored the Wynum studios and exposing myself to the backend environment where the work is being done and the frontend environment where the code /task has to be executed.

8. I was working on a Live project where I had to perform analysis for various patents for various companies based on their claims text. Performed NLP, Tokenization of sentences using sentence transformers, and applied principal component analysis and tsne to reduce the number of dimensions to 2 dimensions without any major loss in information or data.

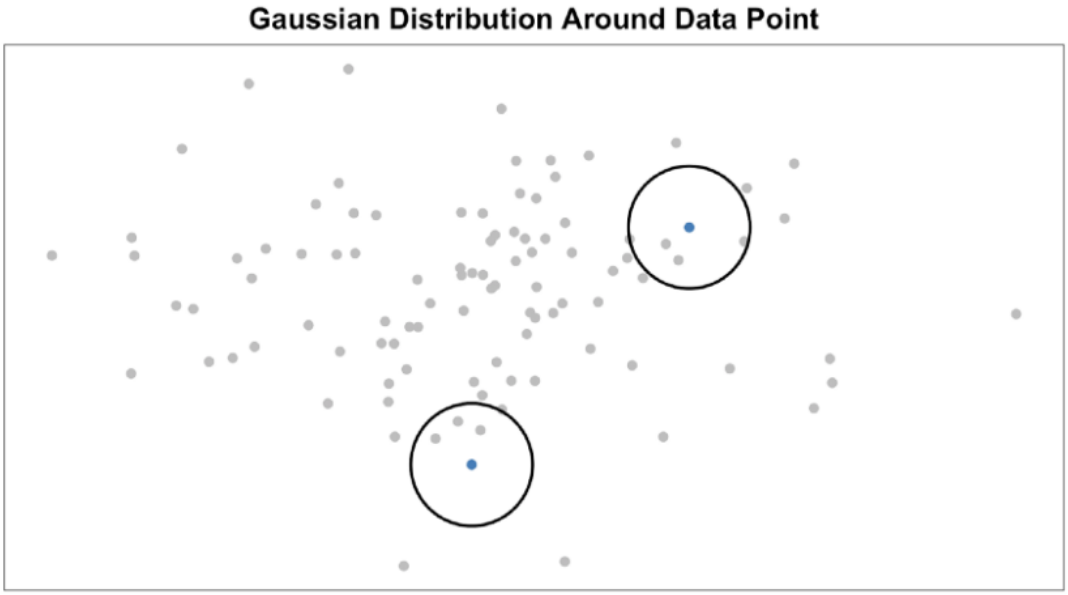
What is TSNE?

T-distributed Stochastic Neighbor Embeddings(t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data.

T-SNE differs from PCA by preserving only small pairwise distances or local similarities whereas PCA is concerned with preserving large pairwise distances to maximize variance.

How TSNE works

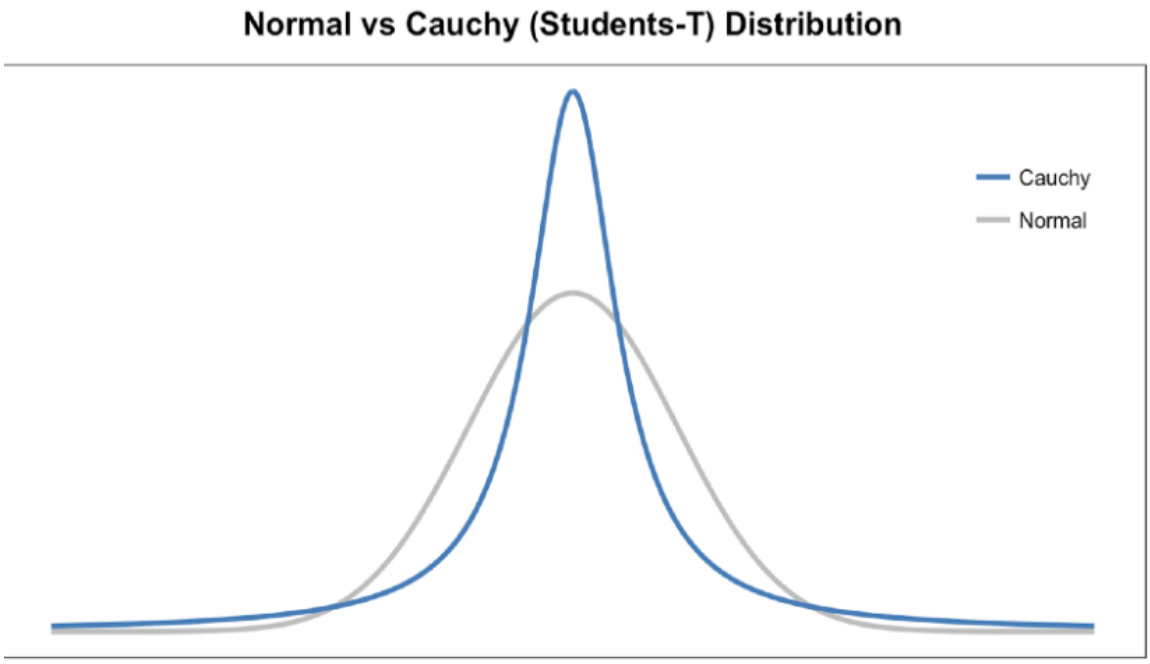
1.It measures similarities between points in high dimensional space. Think of a bunch of data points scattered in a 2d space. (as shown in the figure)



For each data point (xi) we will center a Gaussian distribution over that point. This gives us a set of probabilities (Pij) for all the points.

Those probabilities are proportional to the similarities.

2.This step is similar to step2, but instead of using a Gaussian distribution using a Student's t-distribution with one degree of freedom, which is also known as the Cauchy distribution. This gives us a second set of probabilities (Qij) in the low dimensional space.



3.We want the set of probabilities in low dimensional space(Qij) to reflect those of higher dimensional space(Pij). The map structures should be similar. So, the difference between the probability distributions of the 2-dimensional spaces is measured using Kullback-Liebler divergence (KL). It is an asymmetrical approach to compare Pij and Qij values effectively. To minimize the KL cost function we apply gradient descent.

Introduction to Principal Component Analysis.

Principal Component Analysis (PCA) is a statistical procedure that is used to reduce the dimensionality linearly. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated called principal components.

Steps involved in the PCA:

1.Standardize the dataset.

$$x_{new} = \frac{x - \mu}{\sigma}$$

Standardization formula

2.Calculate the covariance matrix for the features in the dataset.

For Population

$$Cov(x,y) = \frac{\Sigma (x_i - \bar{x}) * (y_i - \bar{y})}{N}$$

For Sample

$$Cov(x,y) = \frac{\Sigma (x_i - \bar{x}) * (y_i - \bar{y})}{(N-1)}$$

Covariance Formula

3.Calculate the eigenvalues and eigenvectors for the covariance matrix.

$$Av - \lambda v = 0 \; ; \; (A - \lambda I)v = 0$$

Here λ represents the eigenvalues and v represents the corresponding eigenvectors. The eigenvalues are found out using

$$\det(A - \lambda I) = 0$$

4.Sort the eigenvalues and their corresponding eigenvectors.

5.Pick a certain number of eigenvalues exceeding a particular threshold and form a matrix of eigenvectors.

6.Transform the original matrix.

What is BERT?

BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks. **(Source:Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework(Analytics vidhya))**

BERT relies on a Transformer (the attention mechanism that learns contextual relationships between words in a text). A basic Transformer consists of an encoder to read the text input and a decoder to produce a prediction for the task. Since BERT's goal is to generate a language representation model, it only needs the encoder part. The input to the encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network.

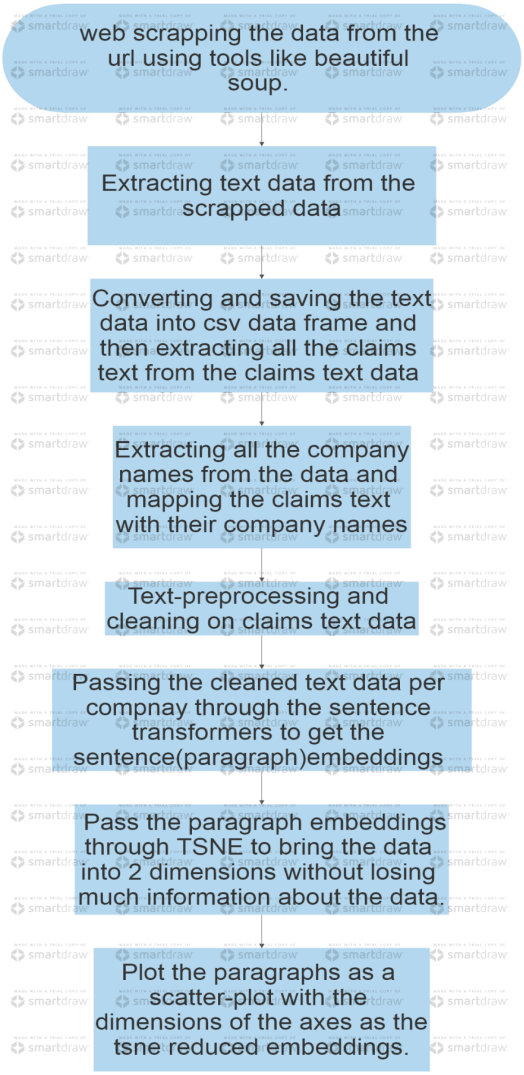
What are Sentence based Bert Transformers

This framework provides an easy method to compute dense vector representations for sentences, paragraphs, and images. The models are based on transformer networks like BERT / RoBERTa / XLM-RoBERTa etc. and achieve state-of-the-art performance in various task. Text is embedding in vector space such that similar text is close and can efficiently be found using cosine similarity.

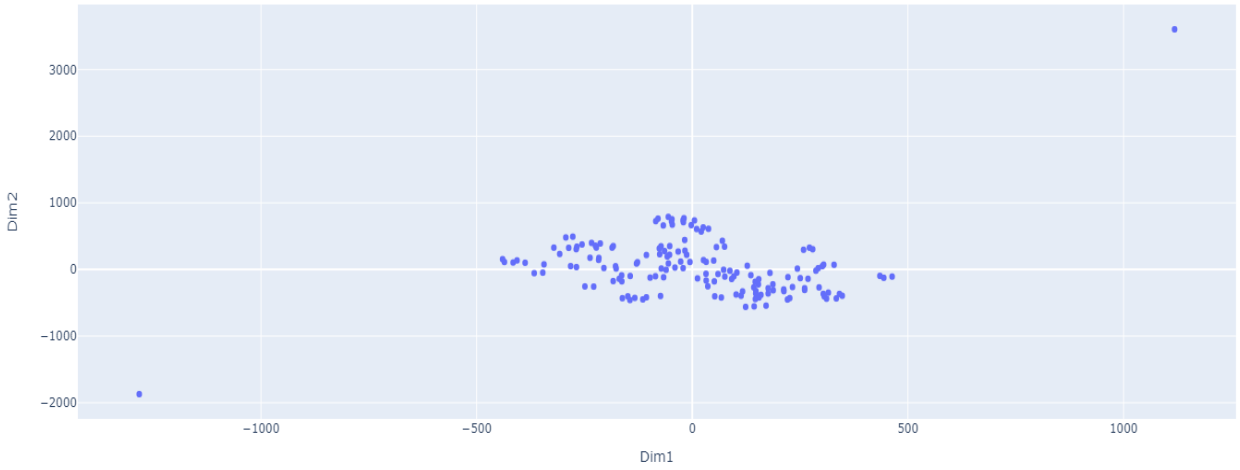
In this project we have used Sentence Bert Transformers to get paragraph based embeddings for our claims text data.

Project on Patent Analysis.

I along with my teammate Gaurav Tikhe were given the task of making some visualizations through



The graph thus obtained is as shown in the figure



Inference: The scatterplot is obtained using plotly.express . As the graph contains hover data it wasn't possible to show the labels corresponding to each data point in png. However from the graph I could infer that most of the Companies based on their claims_text follow a similar trend for their patents except for one company (Kurzorok, Nathan)who are the outliers as per my inference.

Conclusion:

The work experiences I encountered during the internship allowed me to develop skills in python, PyTorch, transformers, NLP. However, I think I have just barely scratched the surface and need to work more on my skills in PyTorch, transformers, and graphical neural networks. However, The overall experience was positive, and everything I learned will be useful in my future career in this field.