**Assignment 06**

**1. What are Vanilla autoencoders ?**

**Ans:** The Vanilla autoencoder, as proposed by Hinton in his 2006 paper Reducing the Dimensionality of Data with Neural Networks, consists of one hidden layer only. The number of neurons in the hidden layer are less than the number of neurons in the input (or output) layer.

This results in producing a bottleneck effect in the flow of information in the network. The hidden layer in between is also called the "bottleneck layer." Learning in the autoencoder consists of developing a compact representation of the input signal at the hidden layer so that the output layer can faithfully reproduce the original input.

In the following diagram, you can see the architecture of Vanilla autoencoder:

A picture containing screenshot, rectangle, diagram, design

Description automatically generated

**2. What are Sparse autoencoders ?**

**Ans:** A Sparse Autoencoder is a type of autoencoder that employs sparsity to achieve an information bottleneck. Specifically the loss function is constructed so that activations are penalized within a layer. The sparsity constraint can be imposed with L1 regularization or a KL divergence between expected average neuron activation to an ideal distribution P. Or A sparse autoencoder is one of a range of types of autoencoder artificial neural networks that work on the principle of unsupervised machine learning. Autoencoders are a type of deep network that can be used for dimensionality reduction – and to reconstruct a model through backpropagation.

Autoencoders seek to use items like feature selection and feature extraction to promote more efficient data coding. Autoencoders often use a technique called backpropagation to change weighted inputs, in order to achieve dimensionality reduction, which in a sense scales down the input for corresponding results. A sparse autoencoder is one that has small numbers of simultaneously active neural nodes.

A picture containing screenshot, circle, diagram, design

Description automatically generated

**3. What are Denoising autoencoders ?**

**Ans:** A **Denoising Autoencoder** is a modification on the [autoencoder](https://paperswithcode.com/method/autoencoder) to prevent the network learning the identity function. Specifically, if the autoencoder is too big, then it can just learn the data, so the output equals the input, and does not perform any useful representation learning or dimensionality reduction. Denoising autoencoders solve this problem by corrupting the input data on purpose, adding noise or masking some of the input values.

To achieve this equilibrium of matching target outputs to inputs, denoising autoencoders accomplish this goal in a specific way – the program takes in a corrupted version of some model, and tries to reconstruct a clean model through the use of denoising techniques. Engineers may apply noise in a particular amount as a percentage of the model and try to force the hidden layer to work from the corrupted version to produce a clean version. Denoising autoencoders can also be stacked on each other to provide iterative learning toward this key goal.

**4. What are Convolutional autoencoders ?**

**Ans:** Convolutional Autoencoder is a variant of Convolutional Neural Networks that are used as the tools for unsupervised learning of convolution filters. They are generally applied in the task of image reconstruction to minimize reconstruction errors by learning the optimal filters.

The block diagram of a Convolutional Autoencoder is given in the below figure.

A picture containing diagram, screenshot, font, design

Description automatically generated

**5. What are Stacked autoencoders ?**

**Ans:** Some datasets have a complex relationship within the features. Thus, using only one Autoencoder is not sufficient. A single Autoencoder might be unable to reduce the dimensionality of the input features. Therefore for such use cases, we use stacked autoencoders. The stacked autoencoders are, as the name suggests, multiple encoders stacked on top of one another. A stacked autoencoder with three encoders stacked on top of each other is shown in the following figure.

A picture containing screenshot, diagram, line, parallel

Description automatically generated

According to the architecture shown in the figure above, the input data is first given to autoencoder

1. The output of the autoencoder 1 and the input of the autoencoder 1 is then given as an input to autoencoder
2. Similarly, the output of autoencoder 2 and the input of autoencoder 2 are given as input to autoencoder
3. Thus, the length of the input vector for autoencoder 3 is double than the input to the input of autoencoder 2.

This technique also helps to solve the problem of insufficient data to some extent.

**6. Explain Extractive summarization ?**

**Ans:** The NLP text summarization can be divided into two main categories.

* Extractive Approach
* Abstractive Approach

It simply takes out the important sentences or phrases from the original text and joins them to form a summary. Now, the question that comes is, exactly on what basis are those sentences termed as important? So, basically, a ranking algorithm is used, which assigns scores to each of the sentences in the text based on their relevance to the overall meaning of the document. The most relevant sentences are then chosen to be included in the summary.There are various ways through which the ranking of sentences can be performed.  
TF-IDF (term frequency-inverse document frequency)  
Graph-based methods such as TextRank  
Machine learning-based methods such as [Support Vector Machines](https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/) (SVM) and [Random Forests](https://www.analyticsvidhya.com/blog/2020/12/lets-open-the-black-box-of-random-forests/).

The main motive of the extractive method is to maintain the original meaning of the text. Also, this method works well when the input text/content is already in a well-structured manner, both physically and logically, just like the content in newspapers.

**7. Explain Abstractive summarization ?**

**Ans:** Abstractive Summarization is a task in Natural Language Processing (NLP) that aims to generate a concise summary of a source text. Unlike extractive summarization, abstractive summarization does not simply copy important phrases from the source text but also potentially come up with new phrases that are relevant, which can be seen as paraphrasing. Abstractive summarization yields a number of applications in different domains, from books and literature, to science and R&D, to financial research and legal documents analysis.

To date, the most recent and effective approach toward abstractive summarization is using transformer models fine-tuned specifically on a summarization dataset.

**8. Explain Beam search ?**

**Ans:** Beam search is a heuristic search algorithm that explores a graph by expanding the most optimistic node in a limited set. Beam search is an optimization of **best-first search** that reduces its memory requirements.

Best-first search is a graph search that orders all partial solutions according to some heuristic. But in beam search, only a predetermined number of best partial solutions are kept as candidates. Therefore, it is a **greedy** algorithm.

Beam search uses **breadth-first search** to build its search tree. At each level of the tree, it generates all successors of the states at the current level, sorting them in increasing order of heuristic cost. However, it only stores a predetermined number (β), of best states at each level called the **beamwidth**. Only those states are expanded next.

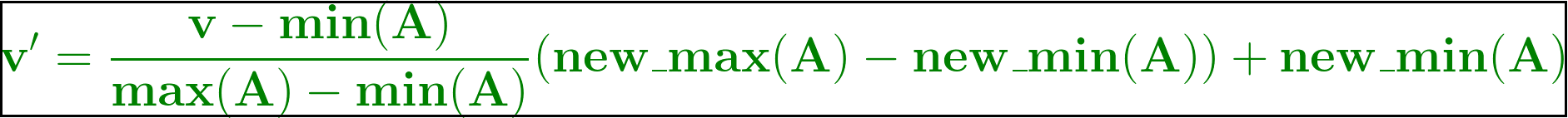
The greater the beam width, the fewer states are pruned. No states are pruned with infinite beam width, and beam search is identical to breadth-first search. The beamwidth bounds the memory required to perform the search. Since a goal state could potentially be pruned, beam search sacrifices completeness (the guarantee that an algorithm will terminate with a solution if one exists). Beam search is not optimal, which means there is no guarantee that it will find the best solution. In general, beam search returns the first solution found. Once reaching the configured maximum search depth (i.e., translation length), the algorithm will evaluate the solutions found during a search at various depths and return the best one that has the highest probability.

The beam width can either be **fixed** or **variable.** One approach that uses a variable beam width starts with the width at a minimum. If no solution is found, the beam is widened, and the procedure is repeated.

**9. Explain Length normalization ?**

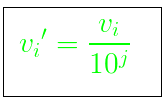
**Ans:** Document length normalization adjusts the term frequency or the relevance score in order to normalize the effect of document length on the document ranking.

One simple length normalization formula is to divide the number of occurrences by the length of the document.

[**Min-Max normalization**](https://www.geeksforgeeks.org/data-normalization-in-data-mining/)**:**In this technique of data normalization, a linear transformation is performed on the original data. The minimum and maximum value from data are fetched and each value is replaced according to formula.   


Where A is the attribute data,  
Min(A), Max(A) are the minimum and maximum absolute values of A respectively.  
v’ is the new value of each entry in data.  
v is the old value of each entry in data.  
new\_max(A), new\_min(A) is the max and min values of the range(i.e boundary value of range required) respectively.

[**Normalization by decimal scaling**](https://www.geeksforgeeks.org/data-normalization-in-data-mining/)**:**It normalizes by moving the decimal point of values of the data. To normalize the data by this technique, we divide each value of the data by the maximum absolute value of the data  The data value, *vi*, of data, is normalized to *vi‘* by using the formula below :



[**Z-score normalization or Zero mean normalization**](https://www.geeksforgeeks.org/data-normalization-in-data-mining/)**:**In this technique, values are normalized based on mean and standard deviation of the data A. The formula used is:

A picture containing diagram, line, screenshot, font

Description automatically generated

v’, v is new and old of each entry in data respectively. σA, A is the standard deviation and mean of A respectively.

**10. Explain ROUGE metric evaluation ?**

**Ans:** ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations.

It works by comparing an **automatically produced summary** or **translation** against a set of **reference summaries** (typically human-produced).

It refers to how much of the **reference summary**the **system summary** is recovering or capturing. If we are just considering the individual words, it can be computed as:

0*yhcByAEK3i1SVRhu