GANS FOR CREATIVE APPLICATIONS

TEXT2IMAGE

INTRODUCTION

We aim to learn about Computer Vision and GANs and their applications.

The objective of this project is to create a text2image model that can generate an image from a given text input.

TOPICS COVERED...

WEEK 1

BASIC PYTHON

- Explored libraries such as NumPy, Pandas and Matplotlib

INTRODUCTION TO ML

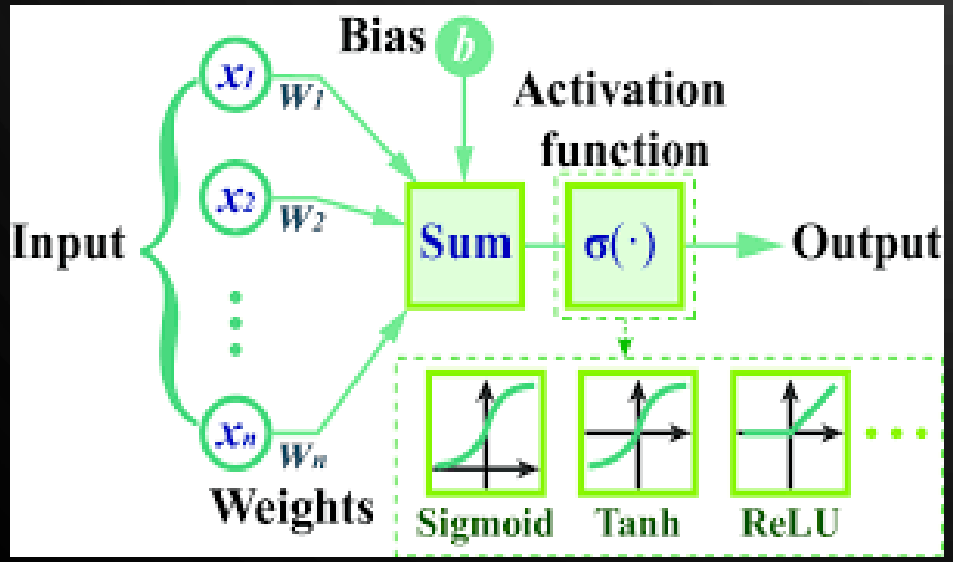
- Explored online resources on Machine Learning and Artificial Intelligence, had interactive discussions on the same.
- We discussed about the difference between AI, ML and DL.
- Machine Learning is an AI application where computers learn from historical data to make future predictions or decisions. It aims to achieve performance on par with or surpassing human capabilities.
- In the first week, we also learnt about perceptron model and forward propagation.

WEEK 1 ASSIGNMENT

The process of computing the output of a neural network when given an input is called Forward Propagation.

Steps:

1. Take Input features
2. Apply Weights and Biases.
3. Pass it through an activation function.
4. Obtain the output predictions.



Main topics Covered in Assignment 1

- NUMPY:
  - Basic logical and algebraic operations on multidimensional arrays.
  - Methods to sort and join multiple arrays .
  - Ways to reshape, slicing and indexing of Arrays
  - Copying Arrays.
- PANDAS:
  - To generate and load Datasets.
  - Function to bin the petal length of IRIS data.

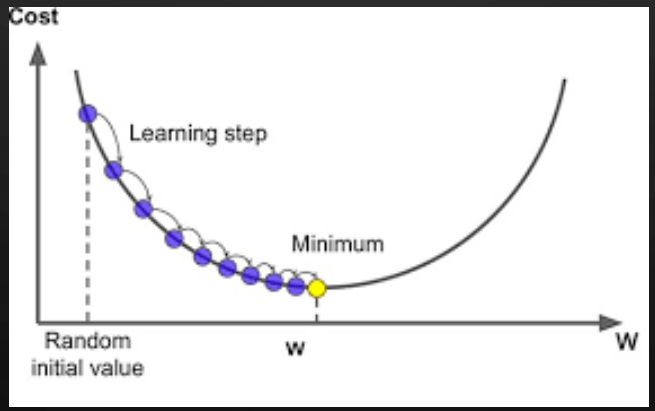
WEEK 2

TENSORFLOW

- Python library specifically designed to help execute ML and AI models.
- Special focus on training and inference of deep neural networks
- Went through online resources and studies examples

GRADIENT DESCENT

- An algorithm that we use to find the minima of our cost function. With each iteration we try to take steps down on our cost function.
- Any Gradient descent algorithm starts with any random point on the cost function, and then according to the slope of the cost function it takes step forward or backward on the x axis. Initially it takes big steps which gradually become smaller as we converge towards the minima this is decided by learning rate.
- Learned about various optimisers like Momentum, SGD, and Adam.



$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

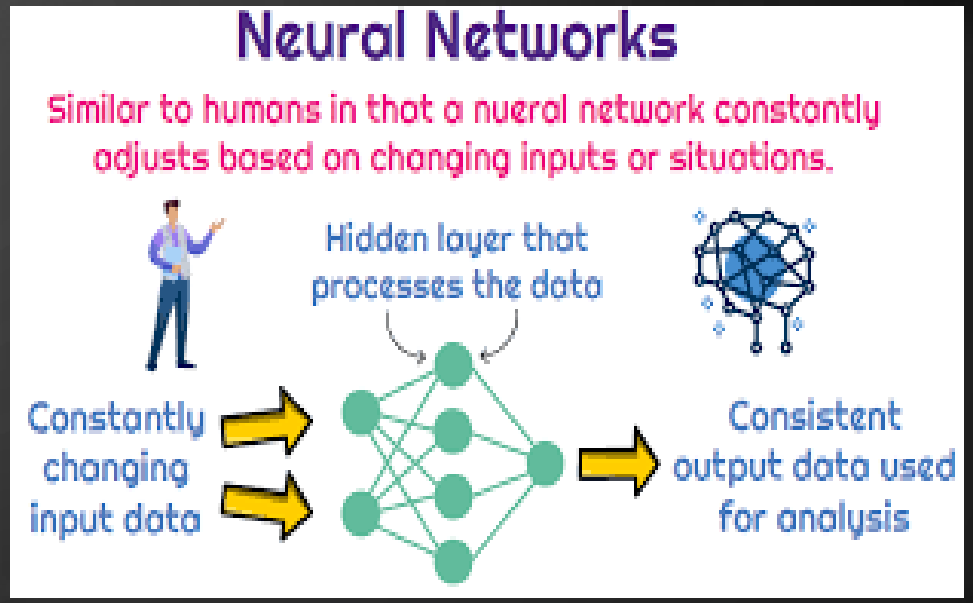
NEURAL NETWORKS

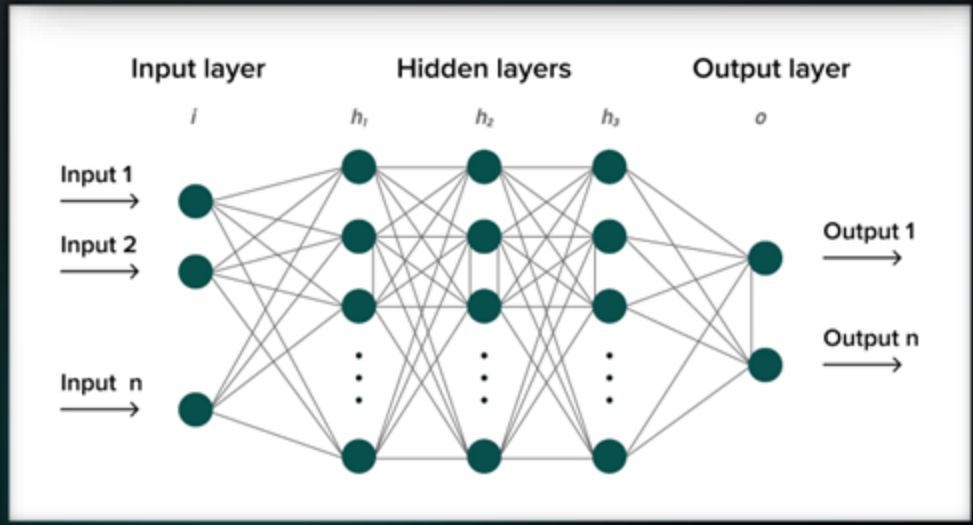
A method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain.

Components of neural network :

1. Input layers
2. Hidden layers
3. Output layers

Neural network training is the process of teaching a neural network to perform a task. Neural networks learn by initially processing several large sets of labeled or unlabeled data. By using these examples, they can then process unknown inputs more accurately.



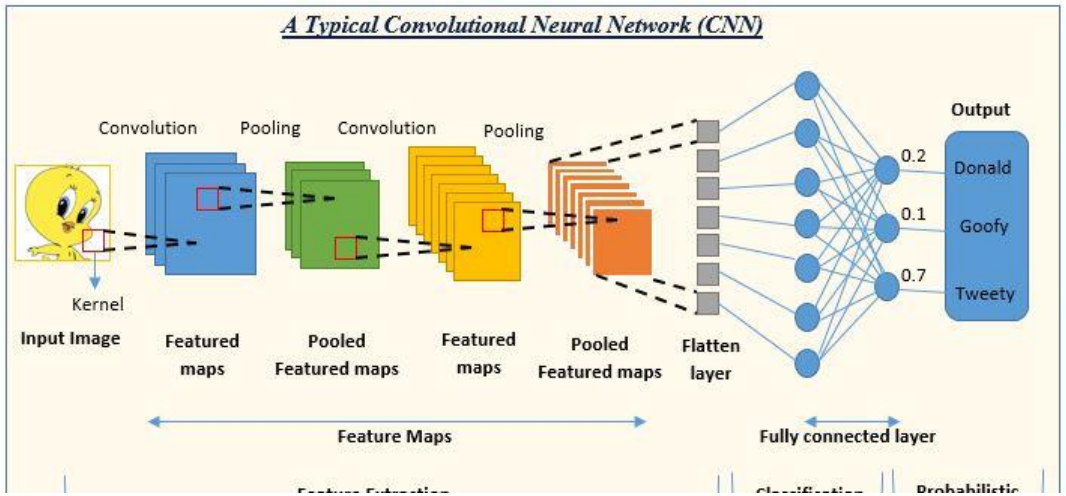


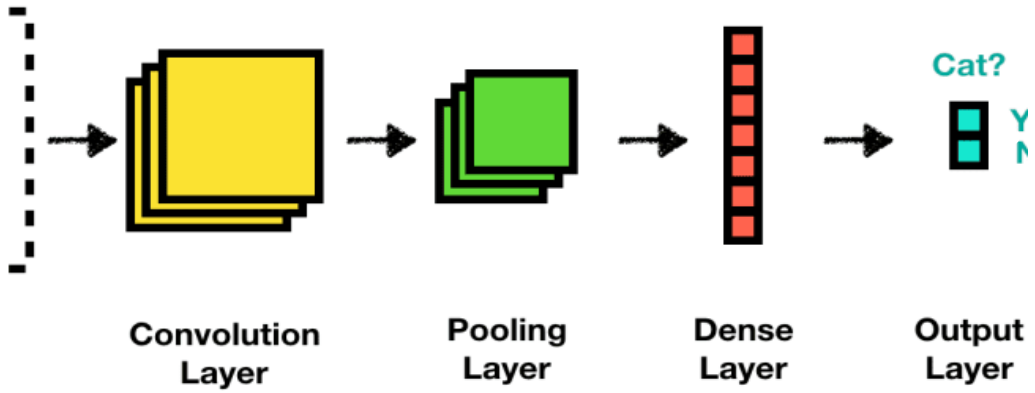
WEEK 2 ASSIGNMENT

Main topics Covered in Assignment 2

- We imported a data set from sklearn_dataset.
- We divided the data set into two parts i.e., training data and test data.
- Built a function named as GDRegressor to perform gradient descent and predict the fitting line.
- Initialised the function by keeping epochs and learning rate, accordingly, fitted the training data in the fit object of function.
- And then we simply calculated r2_score to analyse our model.
- This assignment gave us idea of working of gradient descent. And helped us to learn more about train test splitting of data. In this assignment we used sklearn library (Scikit learn).

WEEK 3





CONVOLUTIONAL NEURAL NETWORKS (CNN)

- A Convolutional Neural Network, also known as CNN or Conv Net, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image.
- A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.
- They are widely used for image classification, object detection, and other computer vision tasks.
- Explored online resources on CNNs.

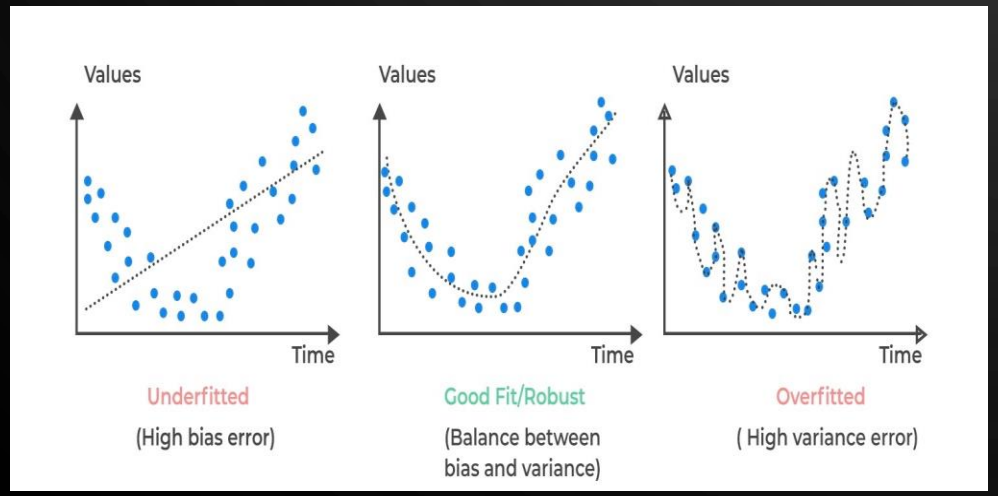
OVERFITTING-UNDERFITTING

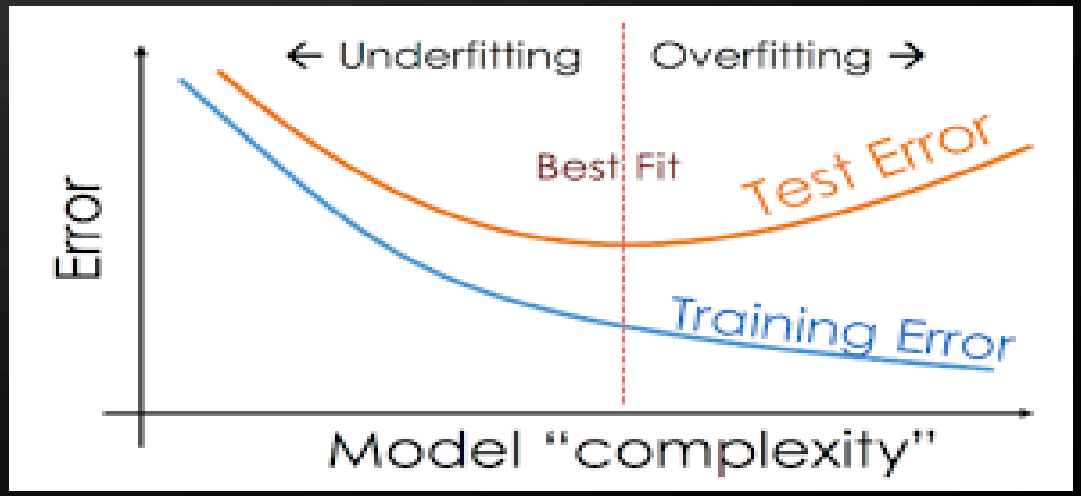
OVERFITTING

- When a model performs very well for training data but has poor performance with test data.
- In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data.
- Caused by large number of layers, nodes in layers and epochs.

UNDERFITTING

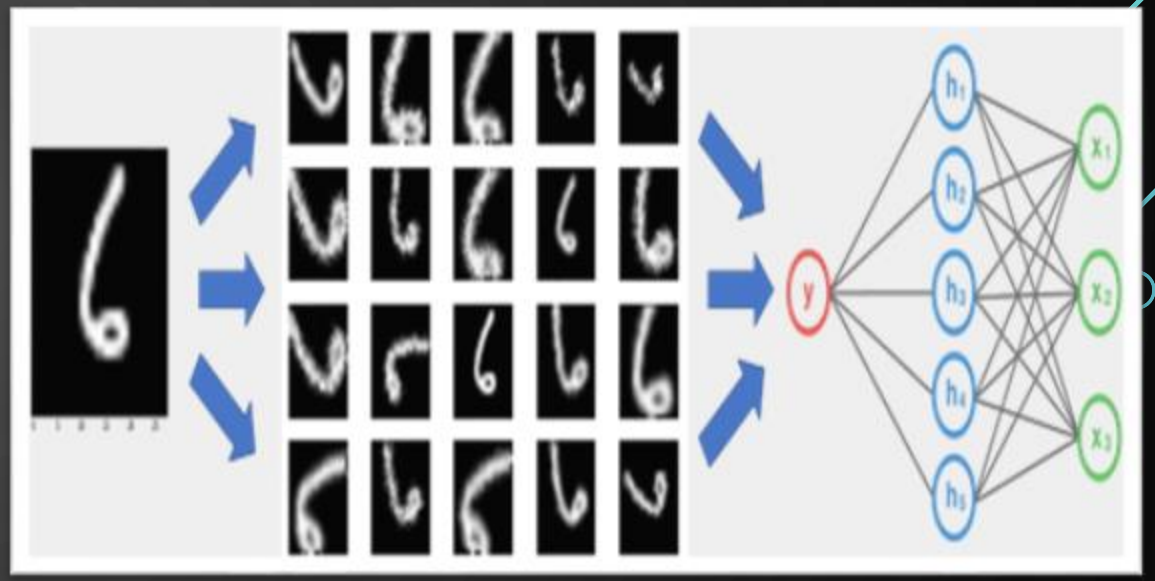
- When a model has not learned the patterns in the training data well and is unable to generalize well on the test set, it is known as underfitting.
- Experimented with different learning rates and layers to try to get best accuracy
- Underfitting is generally caused by the model being too simple, or the dataset not being large enough, or having a large bias.

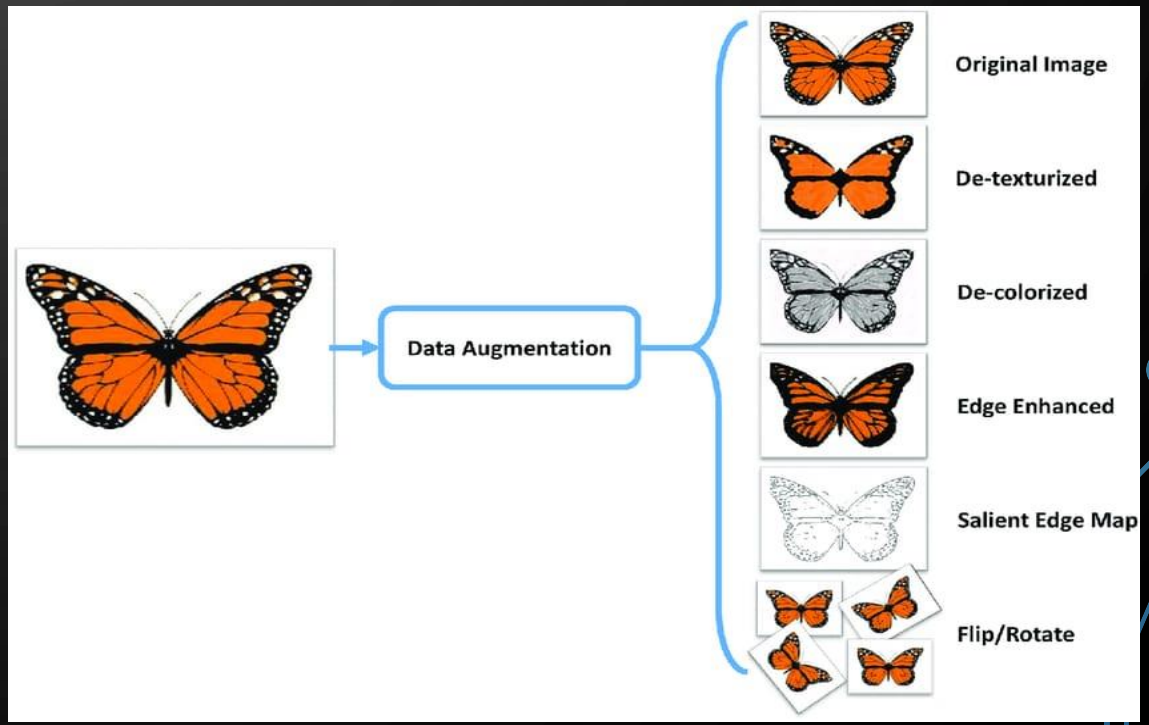




DATA AUGMENTATION

- Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data.
- It reduces the cost of collecting and labelling data.
- Data Augmentation helps in improving model accuracy by adding more training data, preventing data scarcity, increasing the diversity of the dataset, reducing overfitting and underfitting, and helping in generalisation of the model.

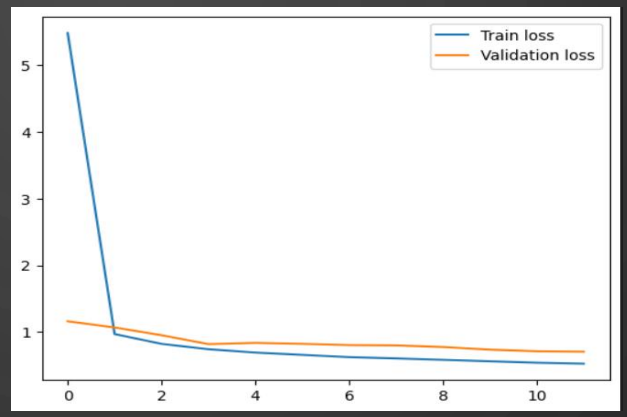


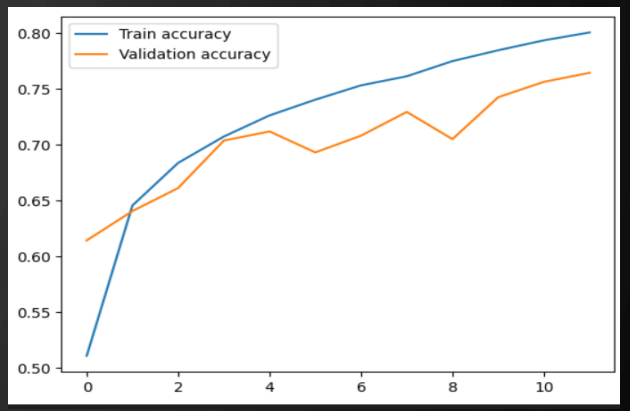


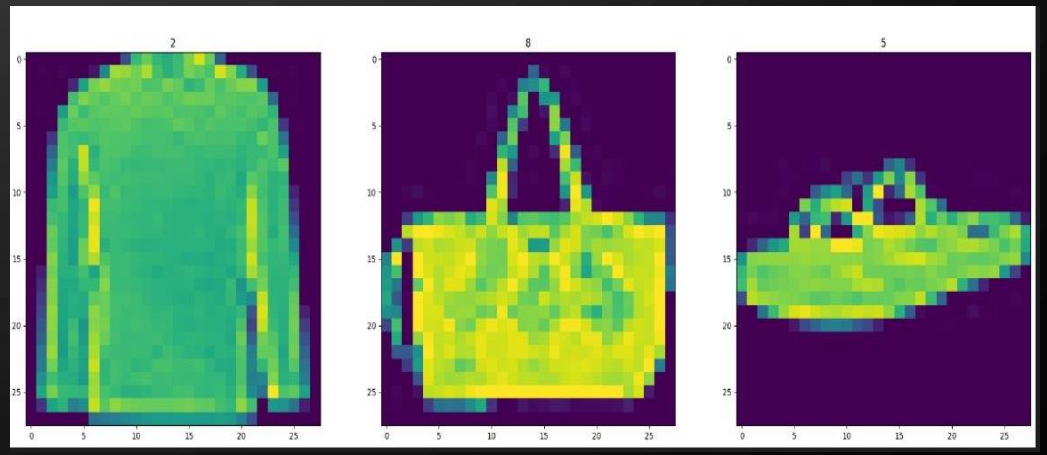
WEEK-3 ASSIGNMENT

Topics that are included in 3rd assignment are mentioned below:-

- Randomly plotting images with label from training dataset.
- Output shape and activation function in different layers of neural network.
- Adjustment of training dataset for different epochs to have minimum loss.
- Plotted graph of training and validation for accuracy and losses.
- Kernal shapes and its multiplication in different layers.





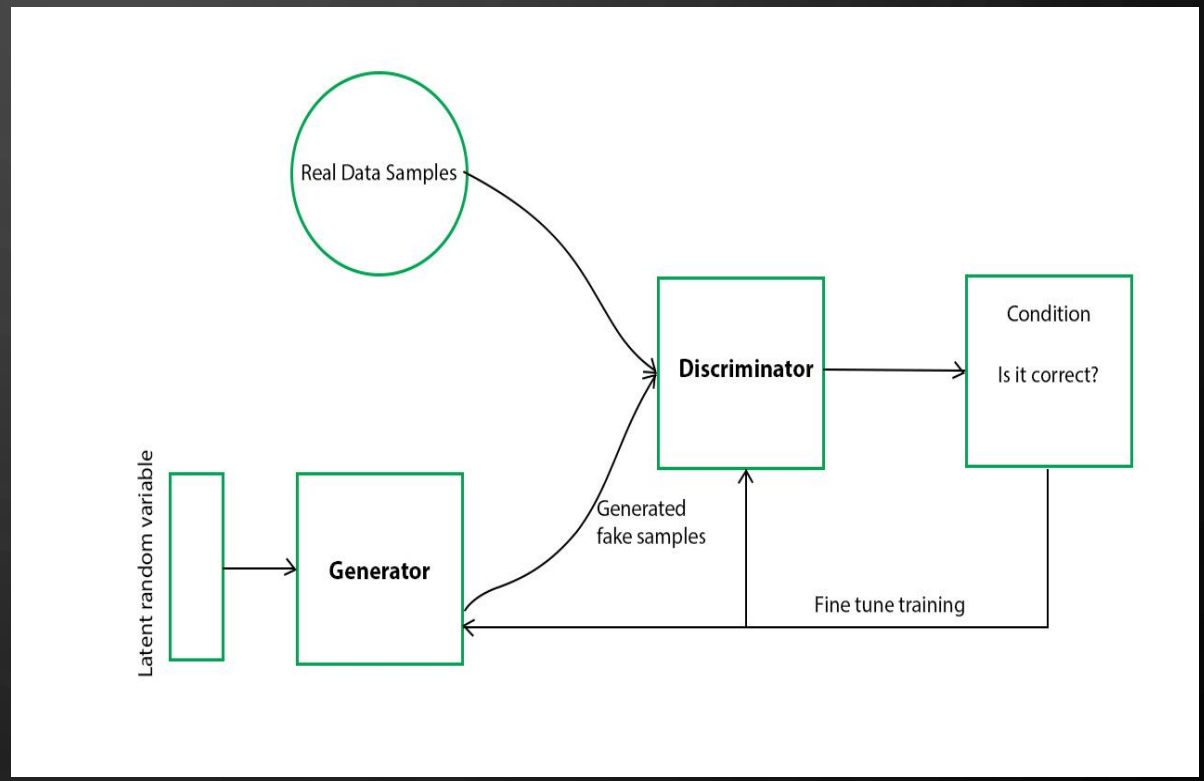


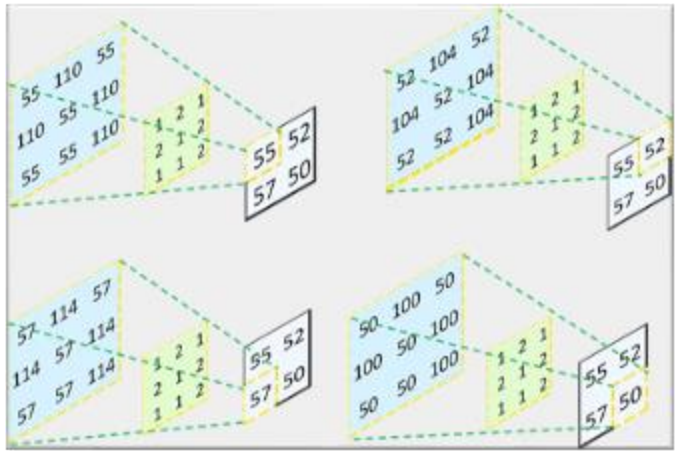
WEEK 4

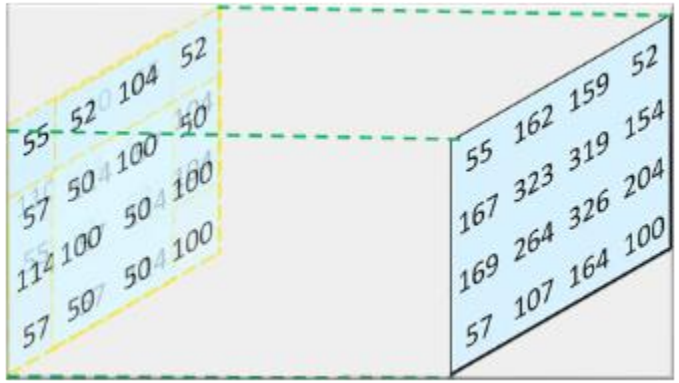
UNDERSTANDING GANS

- Generative Adversarial Networks (GANs) are a type of deep learning model that can generate new data with similar characteristics as the training data.

- The GAN architecture consists of two neural networks: a generator and a discriminator. The generator generates synthetic data while the discriminator evaluates the authenticity of the generated data. The two networks are trained together in a game-theoretic manner, where the generator tries to fool the discriminator and the discriminator tries to correctly identify the synthetic data. This process continues until the generator produces data that is indistinguishable from the real data.







TRANSPOSED CONVOLUTIONS

- Transposed Convolutions are used in GANs to upsample the feature maps produced by the generator. They are also known as deconvolutions or fractionally-strided convolutions. Transposed convolutions can be thought of as the opposite of regular convolutions, and are used to increase the resolution of feature maps.
- Unlike regular (forward) convolutions, which perform a filtering operation to extract features from an input image or volume, transposed convolutions work in the opposite direction. They are used to increase the spatial resolution of feature maps.
- The transposed convolution applies the convolution operation, which involves element-wise multiplication of the transposed kernel with the upsampled input data. The result is summed up to produce a single output value for each location in the output feature map.

WEEK 5

Natural Language Processing

Natural Language Processing is a technique used to deal with human language.

It provides a method for computers to interpret human language as data.

It has many applications such as speech recognition, text to speech named entity recognition and so on

Need For NLP

1. Machine learning models are not capable of processing strings or plain text in their raw form.
2. They require numerical data as inputs to do tasks, such as classification, regression etc.
3. Hence, we can say that to build any model in machine learning or deep learning, the final level data has to be in numerical form because models don't understand text or image data directly as humans do.

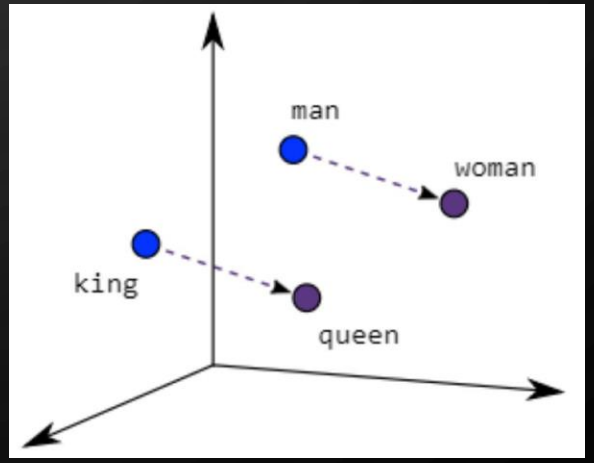
WORD EMBEDDINGS AND VECTORIZATION

Word embeddings
Processing natural language text and extract useful information from the given word, a sentence using machine learning and deep learning techniques requires the string/text needs to be converted into a set of real numbers (a vector).

Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which used to find word predictions, word similarities/semantics.

Word embedding method we used in our assignment- Word2Vec

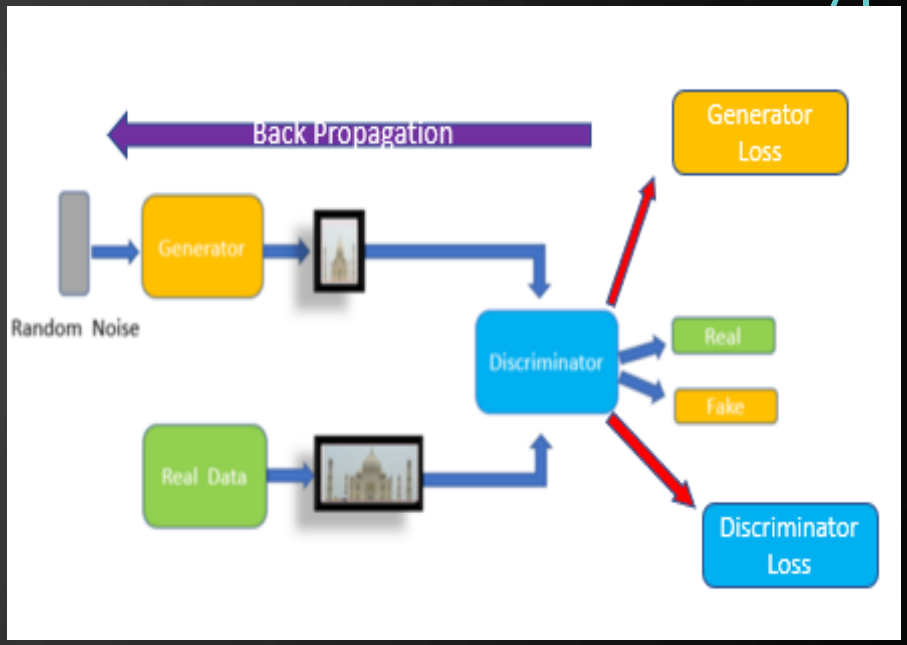
What is Word2Vec ?
It is a word embedding methodology designed to learn continuous word embeddings from large amounts of text data in an unsupervised manner. The model aims to capture the semantic meaning and contextual relationships between words by representing them as dense vectors in a continuous vector space.
As seen in the image below where word embeddings are plotted, similar meaning words are closer in space, indicating their semantic similarity.



WEEK 6

GAN LOSS AND TRAINING

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$



Discriminator loss:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$$

- While training of discriminator, it classifies an image as real or fake.
- It classifies both real and fake image generated.
- For misclassifying an image it penalizes itself by maximizing the function mentioned above.

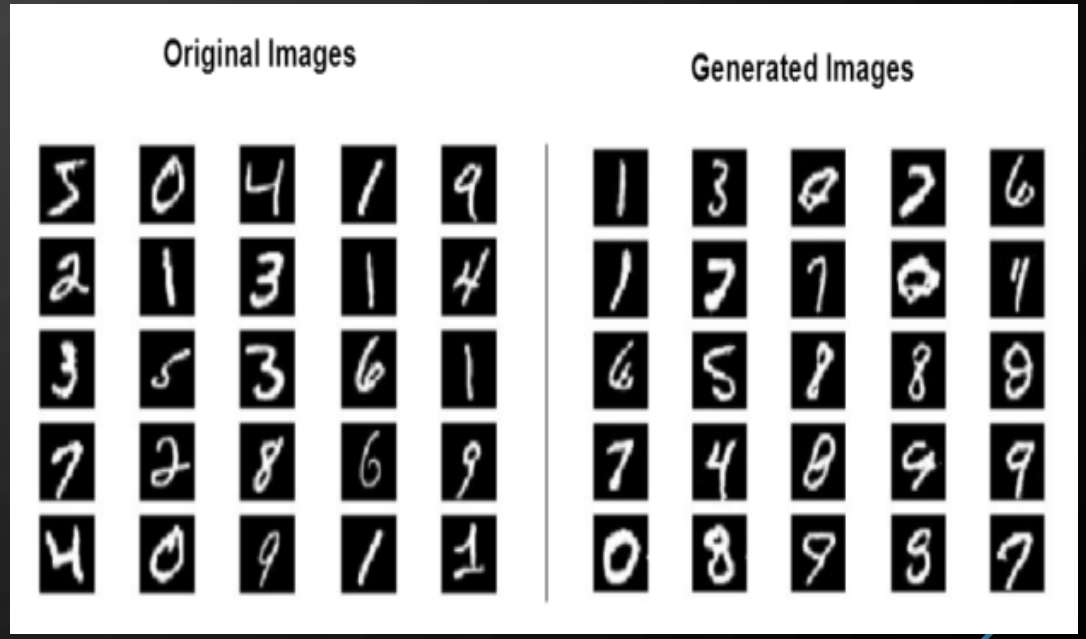
Generator Loss:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

- During generator training, it produces image from random noise.
- Generator tries to fool discriminator, by training model to generate indistinguishable fake image.
- If it fails, it gets penalized by minimizing the above equation.

EXPERIMENTING WITH DCGAN ON HANDWRITTEN DATASET

- DCGAN uses convolutional and convolutional-transpose layers in the generator and discriminator respectively.
- The generator consists of convolutional-transpose layers, batch normalization layers, and ReLU activations.
- The discriminator is made up of strided convolutional layers, batch norm layers and LeakyReLU activations.
- This model has been trained on handwritten dataset. The image on the right shows us the formation of handwritten numbers by the generator after all the epochs.



FINAL WEEK

FINAL ASSIGNMENT

We need to build a GANS model from scratch using the dataset provided -

- Imported the given dataset using numpy.load and resized the images using OpenCV.
- Converted the text in proper format to pass it into the model.
- Build architecture for Generator and Discriminator network for GANS model.
- Created a function for training the GANS model and defined the training loop.
- Generated the images from the trained generator model by providing it with text and analyzed its performance
- Adjusted hyperparameters for better results.