



Title :- Study of deep learning packages: Tensorflow, Keras, Theano & PyTorch Document the distinct features & functionality of the packages

Aim: study & installation of following Deep learning Packages

(i) Tensorflow

(ii) Keras

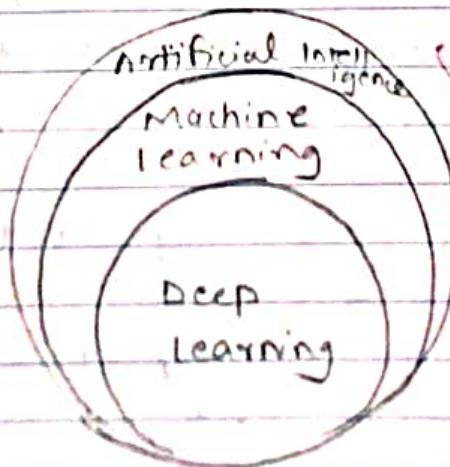
(iii) Theano

(iv) PyTorch

Theory :-

Q.1 What is Deep Learning ?

→ Deep learning is a subset of machine learning (ML), where artificial neural networks - algorithms modeled to work like the human brain - learn from large amounts of data.



Q.3 What are various packages in python for supporting Machine Learning libraries & which are mainly used for Deep learning?

- Python libraries that are used in Machine learning are
- (i) Numpy - Python library for large multi-dimensional array & matrix processing, with the help of large collection of high-level mathematical functions.
 - (ii) Scipy - It contains different modules for optimization, linear algebra, integration & statistics. Scipy is also very useful for image manipulation.
 - (iii) Scikit Learn - It is built on top of two basic python libraries, viz Numpy & Scipy. It supports most of the supervised & unsupervised learning algorithms.
 - (iv) Theano - Theano is a popular python library that is used to define, evaluate & optimize mathematical expression involving multi-dimensional arrays in an efficient manner.
 - (v) Tensorflow - It is very popular open-source library for high performance numerical computational developed by the Google Brain team in Google. Tensorflow is widely used in the field of deep learning research & application.
 - (vi) Keras - It provides many inbuilt methods for grouping, combining & filtering data. It is high-level neural networks API capable of running on top of Tensorflow, CNTK or Theano.
 - (vii) PyTorch - It has an extensive choice of tools & libraries that support computer vision, natural language processing (NLP) & many more ML programs.



PICT, PUNE

(Viii) Pandas:- It is a popular library for data analysis. It provides high-level data structure & wide variety of tools for data analysis.

(ix) Matplotlib - It is very popular python library for data visualization. It is 2D plotting library used for creating 2D graphs & plots.

Q.3 Compare: Tensorflow / keras / Theano & PyTorch on following points:

- (1) Available functionality (2) GUI status (3) Versions
- (4) Features (5) Compatibility with other environments.
- (6) Specific application domains.

Parameters	Tensorflow	Keras	Theano	PyTorch
(1) Available functionality	It is symbolic math library used for neural network library that lets networks & is designed to best suited for provide fast dataflow programming across a range of tasks deep neural network & it including can run on	It is open-source neural network library that lets programmers based on Torch, define, evaluate & optimize models across a range of tasks deep neural network & it including	It is open-source project deep learning framework that lets programmers based on Torch, define, evaluate & optimize models across a range of tasks deep neural network & it including	It is new framework designed to programmers based on Torch, define, evaluate & optimize models across a range of tasks deep neural network & it including



Parameters	Tensorflow	Keras	Theano	PyTorch
2. GUI status	Tensorflow GUI is tool available for developing through an training, testing interface & visualization studio radio deep learning models with Tensorflow.	Keras API very simplifies complex Keras workflows.	It allows you to define optimise & evaluate model	It seeks to make it easier to build & com- pare models in PyTorch which is an IDE for PyTorch
3. Versions.	Tensorflow 2.8.0 & Tensorflow 2.8.0 is the latest version	Keras release 1.0.5 is the current API versions.	Theano - PyMC 1.12.1 is the latest stable release	
4. Features.	(1) models can developed easily (2) complex numeric computation can be done (3) consist of ML API's (4) easy deployment & computation using. CPU & GPU models (5) pre-trained models.	(1) large data set (2) output of Numpy computation can be layer (3) consist of ML Numpy data (4) evaluation	(1) tight integration with Numpy (2) transparent use of GPU (3) train from scratch with TorchScript (4) speed & stability op. (5) pre-trained dynamic code generation	(1) TorchScript graph computation (2) dynamic differentiation (3) automatic differentiation (4) Python support (5) easy to scale timization on cloud platforms (4) dynamic code generation (5) extensive unit testing & self-verification



PICT, PUNE

Parameters	Tensorflow	Keras	Theano	PyTorch
(5) compatibility with other environments	follows semantic versioning 2.0 for its public API. Each release has the MINOR PATCH.	Needs the Python & tensorflow pre-installed for Keras	Needs conda installation & created with no CUDA drivers	IF conda-env is created with no Python packages pre-installed, compatibility of the Python-PyTorch theano python installed correctly library with V 3.8.10 & higher-
(6) specific Application domains.	Image classification object-detection pose estimation speech gesture recognition	Premade architecture with pre-trained weights.	Python library that allows to evaluate mathematical operations.	Deep learning used CNN to develop image classification, object detection & generative application

Q.4 Enlist the models datasets & pre-trained models, libraries & extensions, tools related to Tensorflow also discuss any two case studies like (PayPal, Intel, etc). related to Tensorflow

- -Tensorflow datasets. - All datasets are exposed as if data-
- Pre-trained models-

- (1) Image classification models
- (2) Object detection models
- (3) Text classification models
- (4) Text-embedding models
- (5) Audio speech synthesis models
- (6) Audio embedding models.

- libraries & extensions.

- (1) Decision forests
- (2) Tensorflow Hub
- (3) Model optimization
- (4) Tensorflow Recommenders.
- (5) Lattice
- (6) Tensorflow Graphics
- (7) Tensorflow Federated
- (8) Probability
- (9) Tensor2Tensor
- (10) TensorFlow Privacy
- (11) Dopamine
- (12) TRFL
- (13) Mesh Tensorflow.
- (14) Ragged Tensors
- (15) Unicode Ops
- (16) Magenta
- (17) Nucleus
- (18) Sonnet
- (19) Model card toolkit
- (20) Cloud.
- (21) Tensorflow Text

- case studies.

(1) Intel optimizes Tensorflow Inference performance on the Xeon Scalable processor.

- Intel's partnership with Google has resulted in up to 2.8x inference performance improvement across different models to benefit a wide range of customers running Tensorflow on Intel platforms.



(2) Paypal uses Tensorflow to stay at the cutting edge of fraud detection.

- Using Tensorflow, deep learning & generative modeling, Paypal has been able to recognize complex temporally varying fraud patterns to increase fraud decline accuracy while improving experience of users through increased precision in identification.

Q.5 Explain the Keras ecosystem (Kerastuner, kerasNLP, kerasCV, AutoKeras & model optimization). Also explain following concepts related to Keras.

1. Developing sequential model.
2. Training & validation using the inbuilt functions
3. Parameters Optimization.

Keras ecosystem -

(1) Keras Tuner -

- Kerastuner is an easy-to-use, scalable hyperparameter optimization framework that solves the pain points of hyperparameter search. Easily configure your search space with define-by-run syntax, then leverage one of the available search algorithms to find best hyperparameters values for your models.

(2) KerasNLP -

- KerasNLP is a simple & powerful API for building NLP models. KerasNLP provides modular building blocks following standard Keras interfaces that allows you to quickly & flexibly iterate on your task. KerasNLP is maintained by

the Keras team directly.

(3) Keras CV

- Keras CV is a repository of modular building blocks that applied computer vision engineers can leverage to quickly assemble production-grade, state-of-the-art training & inference pipelines for common use cases such as image classification, object detection, image segmentation, image data augmentation etc.

(4) Auto Keras -

- Auto Keras is an AutoML system based on Keras. It is developed by PADA lab at Texas A&M University. The goal of autoKeras is to make machine learning accessible for everyone.

(5) Model optimization toolkit

- The tensorflow model optimization toolkit is a set of utilities to make your inference model faster, more memory efficient & more power-efficient, by performing post-training weight quantization & pruning - dware training

Developing sequential Model

- A sequential model is appropriate for a plain stack of layers where each layers has exactly one input tensor & one output tensor
- You can create sequential model by passing a list of layers to the sequential constructor. Its layers are accessible via the `layers` attributes.



PICT, PUNE

- Training & validation using the inbuilt functions -
 - The typical end-to-end workflow looks like, consisting
 - Training
 - Validation on a holdout set generated from the original training data,
 - Evaluation on test data.

To train a model with `fit()`, you need to specify a loss function an optimizer & optionally some metrics to monitor. At the end of each epoch, the model will iterate over the validation dataset & compute the validation loss & validation metrics.

• Parameter optimization

- `Keras Tuner` is an easy-to-use, scalable hyperparameter optimization framework that solves the pain points of hyperparameter search. Easily configure your search space with a define-by-run syntax, then leverage one of the available search algorithm to find the best hyperparameter values for your models.

Q.6 Explain simple Theano program

→ Theano is a Python library that allows you to define, optimize & evaluate mathematical expressions involving multi-dimensional array efficiently.

for e.g. :- Adding two matrices -

import numpy

import theano.tensor as T
from theano import function

$M = T.dmatrix('x')$

$y = T \cdot \text{dmatix}('y')$

$z = x + y$

$F = \text{function } ([x, y], z)$

$F([30, 50], [2, 3]), [[60, 70], [3, 4]])$

Output -

array $([[90, 120], [5, 7]])$

Q.3 Explain PyTorch Tensor & also explain Uber's Pyro, Tealab Autopilot

• PyTorch Tensors -

- Tensors are the central data abstraction in PyTorch. This will be created from torch.Tensor class.

• Creating Tensors -

The simplest way to create tensor is with torch.empty() call:

import torch

import math

$x = \text{torch.empty}(3, 4)$

print(type(x))

O/P \rightarrow <class 'Tensor'>

- If you want to initialize your tensor with some value then common cases are all zeros, all ones, or random values & the torch module provides factory methods for all these cases

• Uber's Pyro



- Researchers from Uber, Northeastern & Stanford came together to form an active probabilistic programming community around their packages Pyro & ProbTorch
 - Pyro is a universal probabilistic programming language (PPL) written in python & supported by PyTorch on the backend.
 - Pyro enables flexible & expressive deep probabilistic modeling, unifying the best of modern deep learning & Bayesian modeling.
-
- Tesla autopilot
 - Autopilot advanced safety & convenience features are designed to assist you with the most burdensome part of driving.
 - Autopilot introduces new features & improves existing functionality to make your Tesla safer & more.
 - Autopilot enables your car to steer, accelerate & brake automatically within its lane.

Conclusion:-

Tensorflow, PyTorch, Keras & Theano all these packages are installed & ready for deep learning applications. As per application domain & dataset we can choose the appropriate package & build required type of Neural Network.

Y81
978



PICT, PUNE

Title - Implementing feedforward neural networks

Aim - Implementing feedforward neural networks with Keras & TensorFlow.

- a. Import the necessary packages
- b. Load the training & testing data (MNIST / CIFAR10)
- c. Define the network architecture using keras
- d. Train the model using SGD
- e. Evaluate the network
- f. Plot the training loss & accuracy.

Theory

Q1 What is feedforward Neural Network?

- - A feedforward neural network is artificial neural network in which the connections between nodes class does not form a cycle.
- The feedforward model is the simplest form of neural network as information is only processed in one direction.

Q.2 How the feedforward Neural Network works?

- A feedforward neural network is commonly seen in its simplest form as a single layer perceptron. In this model a series of input enter the layer & are multiplied by the weights.
- Each value is then added together to get a sum of the weighted input values.
- If the sum value is above specific threshold, the output value produced is often +, whereas if the sum falls below the threshold, the output value is -.

Q.3 Endist atleast three real time scenarios where feed forward neural network is used.

- (1) Pattern Recognition (2) speech Recognition (3) Handwritten character recognition (4) Data compression

Q.4 Explain the components of feedforward neural network.

- - Feedforward neural network has following components
 - Input layer
 - This layer comprises neurons that receive the input & transfer them to the different layers in the network.
 - Output layer
 - This layer is the forecasted feature that depends on the type of model being built
 - Hidden layer - The hidden layer positioned between the input & output layer. The number of hidden layers depends on the type of the model.



- Neuron weights - The strength or the magnitude of connection between two neurons is called weight. The values of the weights is usually small & fits within range of 0 to 1.

- Neurons.

The feedforward network has artificial neurons, which are an adaption of biological neurons.

- Activation function - This is decision making center at the neuron output. The neurons finalize linear or non-linear decision based on the activation function.

Q5 What is cost function in feedforward neural network.

→ The cost function is an important factor of a feedforward neural network. Generally, major adjustment to weights and biases have little effect on the categorized data points.

$$C(w, b) = \frac{1}{m} \sum_{i=1}^m \|y_i - o_i\|^2$$

where,

w = weights collected in network

b = biases

n = no. of training inputs

o = output vectors

x = input

$\|v\|$: usual length of vector v

Q.6 What is loss function in feedforward neural network.

→ A neural network's loss function is used to identify if the learning process needs to be adjusted. As many neurons as there are classes in the output layer. To show the difference between the predicted & actual distributions of probabilities.

Q.7 What is cross entropy loss.

→ The cross-entropy loss for binary classification.

$$L(\theta) = \begin{cases} -\log(\hat{y}) & \text{if } y=1 \\ -\log(1-\hat{y}) & \text{if } y=0 \end{cases}$$

- The cross-entropy loss with multi-class categorization.

$$L(\theta) = - \sum_{i=1}^K y_i \log(\hat{y}_i)$$

Q.8 What is kernel concept related to feedforward Neural Network.

→ Kernel i.e. weights used to scale input & hidden node values; each connection typically holds a different weight.

Q.9 Describe MNIST & CIFAR-10 dataset.

→ MNIST dataset - The MNIST dataset of handwritten digits has a training set of 70,000 example & each row of the matrix corresponds to a 28×28 image.



- CIFAR 10 dataset

- CIFAR 10 is an established computer vision dataset used for object recognition.
- Furthermore, the data were converted from RGB to gray, normalized & rounded to 2 decimal places.

Q.10 Explain use & parameter setting related to feed forward network implementation for following libraries sklearn:

- (1) LabelBinarizer (sklearn preprocessing)
- (2) classification_report (sklearn-metrics) & tensorflow, keras : models, layers, optimizers, datasets, backend and set to respective values.

(1) Sklearn : LabelBinarizer -

- The LabelBinarizer will be used to one-hot encode our integer labels as vector labels.

(2) Sklearn : classification report

The classification report function will give us a nicely formatted report displaying the total accuracy of our model, along with a breakdown on the classification accuracy for each digit

(3) Tensorflow keras model

Model groups layers into an object with training & inference features.

keras.model C

* args, *kwargs

)

where,

args → inputs, output, name of the model

(4) Tensorflow . keras . model optimizers

- Base class for optimizers

```
tf.keras.optimizers.Optimizer()
```

```
name, gradient_aggregator = None, gradient_transforms  
= None * * kwargs)
```

(5) Tensorflow . keras . layers

- This is class from which all layers inherit.

```
tf.keras.layers.Layer()
```

```
trainable = True, name = None, dtype = None,  
dynamic = False * * kwargs.  
)
```

- Users will just instantiate layer & then treat it as a callable

(6) Tensorflow . keras . dataset -

- It is small numpy dataset for debugging / testing

- modules →

(1) boston-housing

(2) cifar10

(3) cifar100

(4) fashion-mnist

(5) imdb

(6) mnist

(7) reuters



IIT, PUNE

(7) TensorFlow · keras · backend,

- keras backend API

• Modules - experimental

• Functions

(1) clear_sessions(...)

(2) epsilon(...)

(3) floatx(...)

(4) get_uid(...)

(5) image_data_format(...)

(6) is_keras_tensor(...)

(7) rnn(...)

Q.11 What is mean by flattening the dataset and why it is

needed. related to standard neural network implementation

→ - flattening the dataset means we are going to flatten our pooled feature map into a column like

- The reason to do this is that we're going to need to insert this data into an artificial neural network

Q.12 What is significance of optimizer in training model.

→ - optimizers are algorithms or methods used to change the attribute of your neural network such as weight & learning rate in order to reduce the losses.
optimization algorithms or strategies are responsible for reducing the losses & to provide the most accurate results possible.



IIT, PUNE

Q.13 What is Epoch in fit command in training.

- - An epoch is a single iteration through the training data.
- Each and every sample from your training dataset will be used once per epoch, whether it is for training or validation.

Conclusion:

A feed forward neural network implemented using mnist dataset & model. Label scenarios & SGD & accuracy at 80.22% was obtained.

✓✓✓

✓✓✓



Assignment No. 3

Title: Build the Image classification model

Aim: Build the Image classification model by dividing the model into following 4 stages:

- (a) Loading & pre-processing the image data
- (b) Defining the model's architecture
- (c) Training the model
- (d) Estimating the model's performance

Theory:

Q.1 What is Image classification problem?

- - Image classification problem, which is the task of assigning an input image one label from a fixed set of categories
- This is one of the core problems in computer vision, that despite its simplicity, has a large variety of practical applications

Q.2 Why to use deep learning for Image classification?

state 2 compare Type of Neural networks used for the Image classification

→ Deep learning excels in recognizing objects in image as it's implemented using 3 or more layers of artificial neural networks where each layer is responsible for extracting one or more features of the image

• Convolution Neural Networks (CNN):

- It is a subtype of neural networks that is mainly used for the applications in image & speech recognition.

- Its built-in convolution layer reduces the high dimension



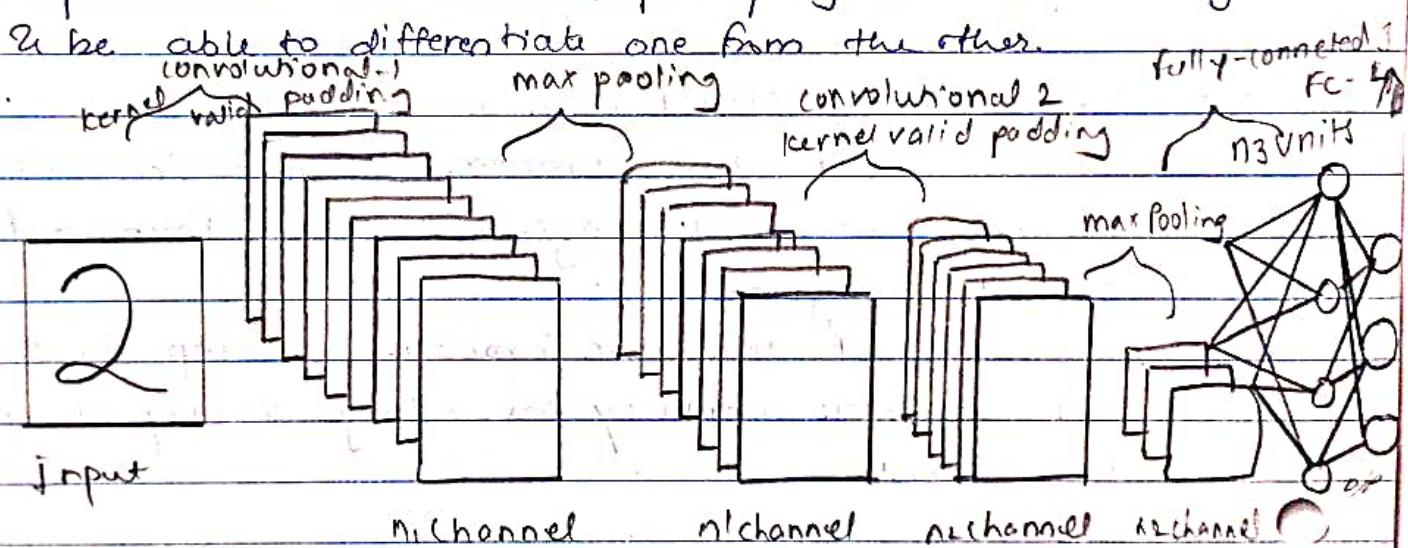
lity of images without losing its information.

- The CNN receives an input feature-map : a 3D matrix where the size of the first 2 dimensions corresponds to the length & width of the images in pixels.

Q.3 What is CNN?

- Convolution Neural Network (CNN) is a deep learning algorithm which can take in an input image/assign importance to various aspects/objects in the image

to be able to differentiate one from the other.



Q.4 Explain convolution operation & convolution kernel related to deep learning

→ convolutional operation

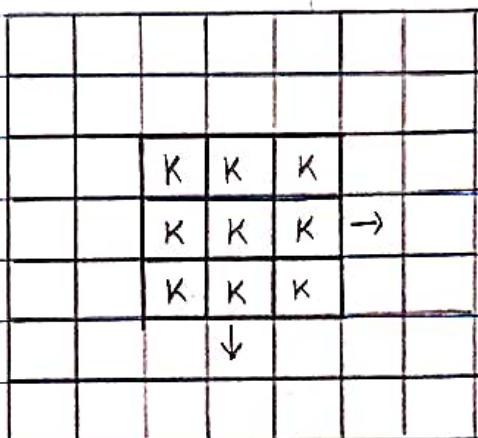
- The name "convolutional Neural Network" indicates that the network employs a mathematical operations called convolution.

- Convolution is a specialised kind of linear operation.



- Convolution Kernel

- A kernel is small 2D matrix whose contents are based upon the operations to be performed.
- The movement of kernel is always from left to right & top to bottom.
- A kernel maps on the input image by simple matrix multiplication & addition, the output obtained is of lower dimensions & therefore easier to work with.



Kernel movement

Q5 Explain how kernel operates on the Input image by taking sample matrix.

- - stride defines by what step does the kernel move, for example stride of 1 makes kernel slide by one row / column of a time & stride of 2 moves kernel by 2 rows/ columns.
- For input image with 3 or more channels such as RGB a filter is applied.



Input matrix

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

Kernel

0	-1	0
-1	5	-1
0	-1	0

Result

-45	12
22	10

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

0	-1	0
-1	5	-1
0	-1	0

-45	103
22	10

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

0	-1	0
-1	5	-1
0	-1	0

-45	103
22	10

45	12	5	17
22	10	35	6
88	26	51	19
9	77	42	3

0	-1	0
-1	5	-1
0	-1	0

-45	103
-176	133

- First entry in the convoluted result is calculated as:

$$45 \times 0 + 12 \times (-1) + 5 \times 0 + 22 \times (-1) + 10 \times 5 + 35 \times (-1) + \\ 88 \times 0 + 26 \times (-1) + 51 \times 0 = -45$$

Q.6 Explain the types of convolution and convolution layers related to CNN.

→ Three main types of convolution layers related to CNN:

(1) Convolutional layers:

- The convolutional layer is the core building block of CNN, and it is where the majority of computation occurs.



- It requires few components which are input data, a filter & feature map.
- After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing non-linearity to the model.

(2) Pooling layer -

- Pooling layer, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input.
- Similar to the convolution layer, the pooling operation sweeps a filter across the entire input but the difference is that this filter does not have any weights.

- There are two main types of pooling:

- (i) Max Pooling - As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- (ii) Average Pooling - As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

(3) fully connected layer -

- The pixel values of the input image are not directly connected to the output layer in partially connected layers.
- While convolution & pooling layers tend to use ReLU



functions, fc layers usually leverage a softmax activation function to classify inputs approximately producing a probability from 0 to 1

Q.7 Explain how feature extraction is done with convolution layers?

- features are extracted by taking the convolution of portion of data sample under consideration.
- The amount of data portion that the filter traverses each time is proportional to the stride length & padding value.
- The convolution output is passed through an activation unit called ReLU.
- The output of ReLU is then passed through pooling layer
- Thus, this layer reduces the size of data sample

Conclusion:

CNN model was built & implemented cifar10 dataset was used & accuracy of 0.689 with respect to accuracy of 55% of object

✓
13/9

Assignment No. 4

Title :- ECG - Anomaly detection using Autoencoders.

Aim:- Use Autoencoders to implement anomaly detection. Build the model using

- (a) Import required libraries
- (b) Upload / access the dataset
- (c) Encoder convert it into latent representation
- (d) Decoder networks convert it back into original input
- (e) Compile the model, with optimizer, loss & evaluation metrics.

Theory:

Q1 What is anomaly detection?

- - Technique that uses AI to identify abnormal behaviour as compared to an established pattern
- Autoencoders are the type of neural networks that can be used to learn hidden encodings in the input data, which can be used for anomaly detection.

Q2 What are Autoencoders?

- - Autoencoders are the type of ANN used to learn data encodings in an unsupervised manner.
- The aim of an encoder architecture is to learn lower dimensional representation for high dimensional data.
- Typically dimensionality reduction by training the network to capture the most important parts of input images.



Q.3 Application of autoencoders in deep learning.

- (1) Dimensionality Reduction
- (2) Feature extraction
- (3) Image denoising
- (4) Image compression
- (5) Image Search
- (6) Anomaly detection
- (7) Missing value imputation

Q.4 Different type of anomaly detection algorithm in deep learning.

- (1) Isolation forest (2) local factor (3) Robust covariance
- (4) one-class SVM (5) One-class SVM (using score)

Q.5 What is the difference between novelty detection & anomaly / outlier detection?

- - You have dataset that contains only good data & you are trying to check if any new observations that are similar to data.
- The goal is to check if new observations are outliers
- Outliers is unsupervised anomaly detection algorithm
- Novelty is semi-supervised anomaly detection algorithm.

Q.6 Architecture of autoencoders :-

- There are mainly three components of autoencoders.
- (1) Encoder - An encoder is a feedforward fully connected neural network that compresses the input into



latent space representation & encodes the input image as a compressed representation in a reduced dimensions. The compressed image is a distorted version of the original image.

(2) code: The part of the neural network that contains reduced representation of the input that is feed into the decoder.

(3) decoder: Decoder is also a feedforward network like an encoder and has a similar structure to encoder. The network is responsible for reconstructing the input back to the original dimensions of the code.

Q7 What is reconstruction error?

- The difference between input representation & output representation is called reconstruction error
- one of the predominant parameters used in unsupervised anomaly detection methods

Q8 What is MinMaxScaler in sklearn?

→ MinMaxScaler scales all the data features in range [0,1] or in the range [-1,1] if dataset contains negative values.

Syntax:-

scalar = preprocessing.MinMaxScaler();

minmax = scalar.fit_transform(X);



Q.9 What is anomaly score?

→ The likelihood that a point is an outlier is measured using some statistical score is called anomaly score.

Q.10 Describe ECG dataset.

→ The dataset is composed of two collection of heartbeat signals derived from two famous heartbeat classification.

(1) The MIT-BIH Arrhythmia Dataset

(2) The PTB Diagnostic ECG Dataset

Q.11 Keras optimizers:

→ During training of Neural Network, one of the main optimization problem is to optimizer loss function.
- Optimized loss function will arrive at a ideal weights.
The methods which are used for optimization are called optimizers.

Different optimizers available in keras.

(1) SGD (2) RMSprop (3) Adam (4) Adadelta (5) Adagrad
(6) Adamax (7) Nadam (8) Ftrl

Q.12 Keras Dense & Dropout!

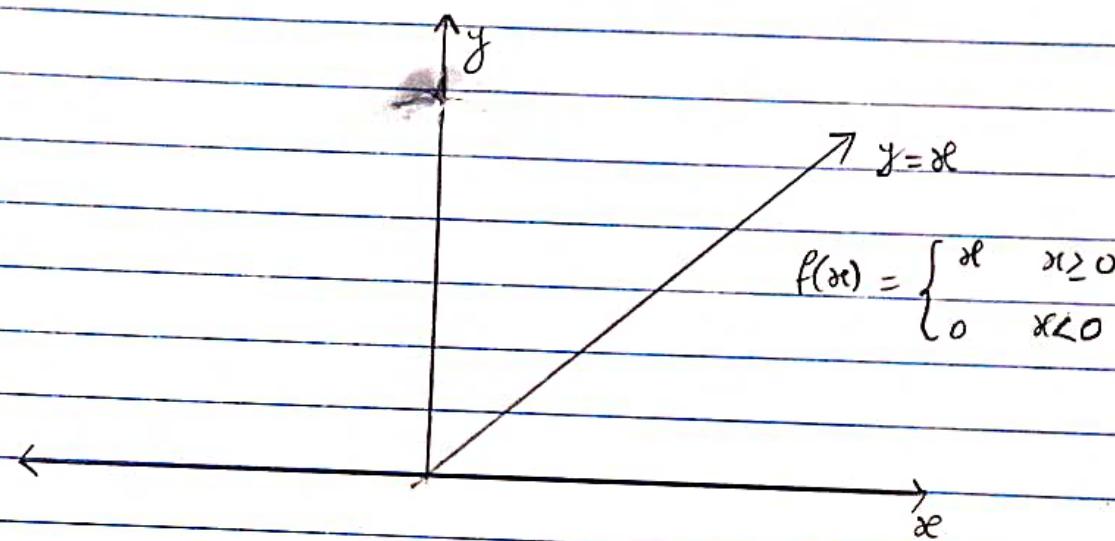
→ Dropout layer ignores a set of neurons randomly
- This is normally done to avoid the overfitting
- The dense layer is a normal fully connected layer in a neural network.

Q.13

What is ReLU Activation Function?

→ ReLU is a non linear activation function that is used in multi-layer neural network or deep learning neural network

$$f(n) = \max(0, n)$$



Conclusion:-

CNN model was build & implemented ~~Cifar10 dataset~~
was used & accuracy of ~~44.9~~ 0.689 with
respect to ~~accuracy of 55% of object~~

Use of autoencoder to implement anomaly detection
was carried out with the model of ECg dataset
using minmax scalar the accuracy ~~94.2%~~
 0.941 was obtained.

~~94.2%~~
0.941



IIT, DUNE

Assignment No. 5

Title: Implement continuous bag of words Model

Aim: Implement the continuous bag of word model stages can be

- (a) Data preparation
- (b) Generate Training data.
- (c) Train model
- (d) Output.

Theory :

Q.1 What is NLP ?

→ - The branch of AI concerned with giving computers the ability to understand text & spoken words in much the same way human beings can

Q.2 What is Word embedding related to NLP?

→ - Word embedding in NLP is a technique where words are represented as real valued vectors in lower-dimensional space & captures inner semantics.
- Each word is represented by real valued vectors with tens or hundreds of dimensions.

Ex:- (1) TFIDF (Term frequency inverse document frequency)
(2) Bag of words.
(3) word2vec.

Q3 Explain word2vec techniques.

→ - Developed by Google in 2013 to train word embedding and is based on the distributional hypothesis.



- It uses cosine similarity metric to measure semantic similarity.

- there are two variations of word2vec

1. continuous bags of word.

2. skip gram

Q.4 Application of word Embeddings.

→ - Analyzing survey responses.

- music / videos recommendation system

- Analyzing verbatim comments.

Q.5 Explain CBOW Architecture?

→ - CBOW model tries to understand the context of the words and tries to predict the words which are contextually accurate.

- Let's say we have q content word used for predicting one target word - the input layer will be in the form of four ($1 \times w$) input vectors.

Q.6 What is Tokenizer?

→ Tokenizer is used in lexical analysis, to break the raw text into small chunks.

Q.7 Lambda layer in keras:

→ - lambda layers in keras is used to wrap the arbitrary expression

→ Used while designing sequential as well as functions

API

- lambda layers are best suited for simple operation
or quick experimentation

Conclusion

The continuous bag of ' was built & implemented.

✓
N/A

11/10

Assignment NO 6

Title: Object detection using transfer learning of CNN Architecture

Aim: Object detection using transfer learning of CNN Architecture.

- (a) Load into a pretrained model trained on a large dataset -
- (b) Freeze parameters in a models lower conventional layers.
- (c) Add custom classification layers with several layers of trainable parameters to model.
- (d) Train classifier layers on training data available for tasks.

Theory:

Q.1 What is Transfer Learning?

- - Method used in machine learning in which pretrained model is used on the new problem.
- In transfer learning, machine exploits the knowledge gained from previous task to improve generalization about another

Q.2 What are pretrained Neural Network Models?

- - Pretrained Neural Network are the neural network models which are trained on a one task and then used in a different task
- Pretrained model used in Computer Vision
 - (1) VGG-16 (2) ResNet-50 (3) InceptionV3 (4) EfficientNet
- Pre-trained model used in NLP:-

- (1) GPT - 3 (2) BERT (3) ELMo (4) XLMNet (5) ALBERT

Q.3 Introduction to PyTorch:-

- - PyTorch is an open source machine learning library for python. It is used for application such as natural language processing.
- It is initially developed by Facebook AI research group.
- Uber's pyro software for probabilistic programming is built on it.

Features:

- (1) Easy interface (2) Python usage (3) Computational graphing

Pytorch is known for having three levels of abstraction as given below:

- (1) Tensor: Imperative n-dimensional array which runs on GPU
- (2) Variable: Node in computational graph.
- (3) Module: Neural new layer which will store state & learnable weights

Q.4 Advantages of transfer learning?

- Better Initial model:- In any other type of learning you need to build the model without any knowledge. Transfer learning offers the better starting point & can perform task at some level of without even training.

- High learning rate : Transfer learning offers higher learning rate during training.
- High accuracy after training
- Faster training : The learning can achieve the desired performance faster than traditional learning methods since it leverages a pretrained model

Q.5 Application of transfer learning

- (1) Name entity recognition (2) Intent classification
- (3) sentiment analysis (4) cross lingual learning
- (5) sequence labeling (6) image recognition
- (7) Model Training in autonomous driving industry
- (8) spam filtering

Q.6 Explain CalTech-UCI image dataset

- CalTech-UCI dataset consists of the pictures of the object belonging to 10 classes. Each image is labelled with the single object
- Each class contains roughly 40 to 800 images totally around 9K images.

Q.7 Explain Imagenet dataset

- ImageNet is a large visual dataset designed for use in visual object recognition software research
- The dataset contains more than 14 million images
- The dataset contains more than 1 million hand-annotated pictures
- The dataset contains more than 20000 categories

Q.8

List down basic steps of transfer learning

- 1. Obtain the pre-trained model
- 2. Create base model
- 3. Add freeze layers from pre-trained model
- 4. Add new trainable layers
- 5. Fine Tune your model

Q.9

What is data augmentation?

- - Data augmentation is a process of artificially increasing the amount of data by generating new datapoints from existing data

Q.10

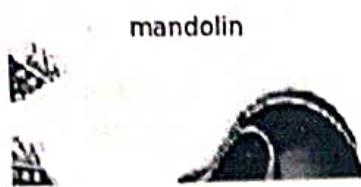
Why preprocessing is needed on input data in transfer learning

- - The idea of transfer learning is to use pretrained model on target data
- To get the better performance of pretrained model
Raw data is preprocessed & normalized before providing as an input to the freeze layers in pretrained model

Q.11

Explain VGG16 model in PyTorch

- - VGG16 is a convolutional neural network that is 16 layer deep.
- VGG16 pretrained model has 138 million parameters.
- VGG16 pretrained version is a neural network which is trained on 14 million images from Imagenet dataset.



Conclusion: Object detection using transfer learning implemented. The dataset used was Caltech which contains images of 100 categories. The pretrained model used for transfer learning VGG. The dataset was split into train validation & test data & accuracy turned out to be 100%.

