DPU

Practical No: 1

Title: Study of Deep learning Packages: Tensorflow, Keras, Theano and PyTorch Document the distinct features and functionality of the packages.

Aim: Study and installation of following Deep learning packages.

i) Tensor Flow
ii) Keras
iii) Theno
iv) PyTorch

Theory:

It is comman to hear the terms "deep learning", "machine learning and "artifical Intelligence" used interchangeably that leads to potential confusion. Deep learning and Machine learning are the part of artificial intelligence family, though deep learning is also subset of machine.

Deep learning imitates the human brain's neural pathway in processing data, using it for decision-making, detecting objects recognizing speech and translating languages It learns without human supervision or

or intervention pulling from unstructured and unlabeled data.

Keras

Keras is an effective high-level neural network Application Programming Interface API written in Python.

This open source neural network library is designed to provide fast experimentation with deep neural networks and it can run on top of CNTK, TensorFlow and Theano

Keras focuses on being modular, user friendly and extensible

It doesn't handle low-level computation instead it hands them off to another library called backend.

PyTorch

PyTorch is relatively new deep learning framework based on Torch

PyTorch has reputation for simplicity ease of use flexibility, efficient, memory usage and dynamic computational graphs.

It also feel native, making coding more manageable and increasing processing spe

TensorFlow

TensorFlow is a symbolic math library used for neural networks and is best suited for dataflow programming across a range of tasks.
It offers multiple abstraction levels for building and training models
A promising and fast growing entry in the world of deep learning
TensorFlow offers a flexible comprehensive ecosystem of community resources and libraries and tools that facilitate building and deploying machine learning apps.

Theano

Theano used to be one of the more popular deep learning libraries, an open source project that lets programmer define, evaluate and optimize mathematical expression including multi-dimensional arrays and matrix-valued expression.

	Keras	PyTorch	TensorFlow
API Level	High	Low	High & low
Architecture	simple, concise readable	Complex, less readable	Not easy to use.

DPU

(2)

Datasets	Smaller datasets	Larger datasets	Larger datasets
Debugging	Simple network so, debugging is not often needed	Good debugging capabilities	Difficult to conduct debugging
Popularity	Most popular	Third most popular	Second most popular
written In	Python	Lua	C++ CUDA Python

Conclusion

Hence we have studied TensorFlow, PyTorch keras Theano all packages for Deep learning.

Practical No 2

Title: Implementing Feed Forward Neural Network.

Aim: Implementing Feed Forward neural networks with Keras and TensorFlow

a). Import the neccessary packages
b). Load the traning and testing data (MNIST / CIFAR10)
c). Define the network architecture using Keras.
d). Train the model using SGD.
e). Evaluate the network
f) Plot the traning loss and accuracy

Theory

Feed Forward Neural Network allows signals to travel one approach only from the input to output. There is no feedback (loops) such as the output of some layer does not influence that same layer

Feed Forward networks tends to be simple networks that associates inputs with outputs It can be used in pattern recognition This type of organization is represented as bottom-up or top-down.

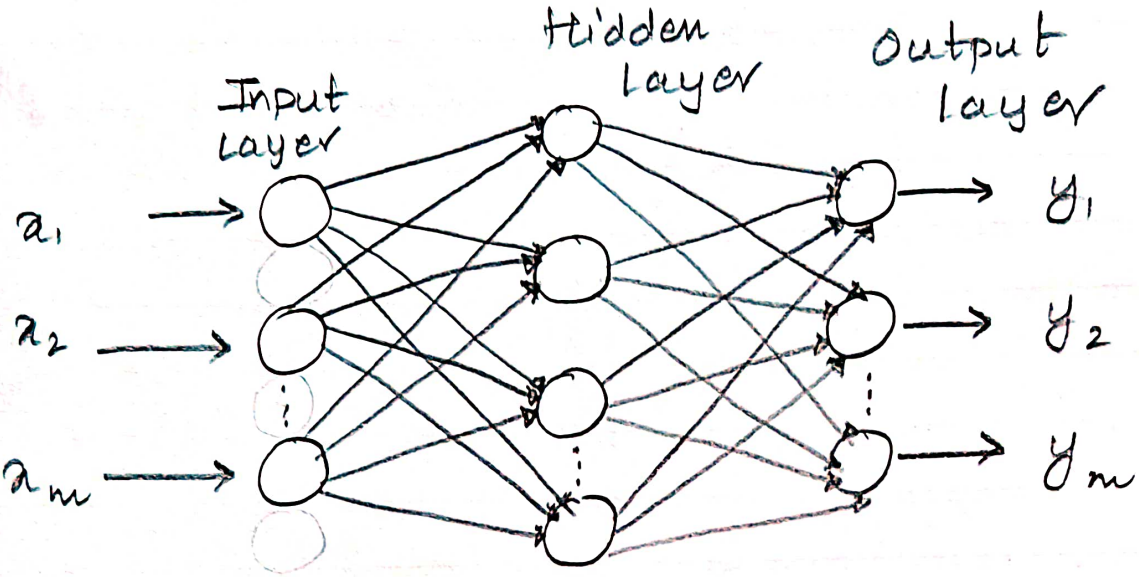


Fig 1 Feed Forward Neural Network.

* Cross-Entropy Loss

Cross entropy loss refers to contrast between two random variable; It measures them in order to extract the difference in the information they contain showcasing the results

We use this type of loss function to calculate how accurate our deep learning model is by defining the difference between the estimated probability with our desired outcome.

$$\text{Cross-Entropy} = L(y,t) = -\sum_i t_i \ln y_i$$

* Difference between sigmoid and softmax

Softmax	Sigmoid
Used for multi-classification in logistic regression model	Used for binary classification in logistic regression model
The probabilities sum will be 1	The probabilities sum need not be 1
Used in the difference layers of neural	Used as activation function while

network	building neural networks
The high value will have the higher probability than other values	The high value will be have the higher probability but not the higher probability.

* optimizer in Deep Learning

Optimizer are algorithms or methods used to minimize an error function (loss function) or to maximize the efficiency of production.

optimizer are mathematical functions which are dependent on model's learnable parameter i.e weights and Biases

Optimizers help to know how to change weights and learning rate of neural network to reduce the losses.

Conclusion

Hence we have implemented Feed Forw Neural Network using MNIST dataset usin Keras.

Practical No 3

Title: Build the Image classification model.

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

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

Theory:

Image classification is where a computer can analyse an image and identify the 'class' the image falls under. A class is essentially a label, for instance, 'car', 'animal', 'building' and so on
Simply put, image classification is where machine can look at an image and assign a label to it. It is a key part of computer vision, allowing computer to see the world as we do

Image classification has few uses - and vast potential as it grows in reliability.

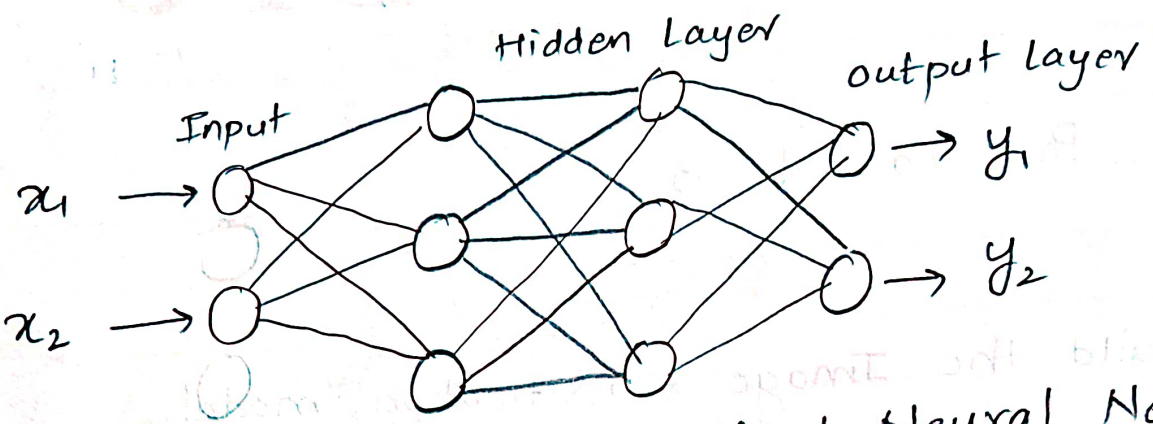


Fig 1 Artifical Neural Network

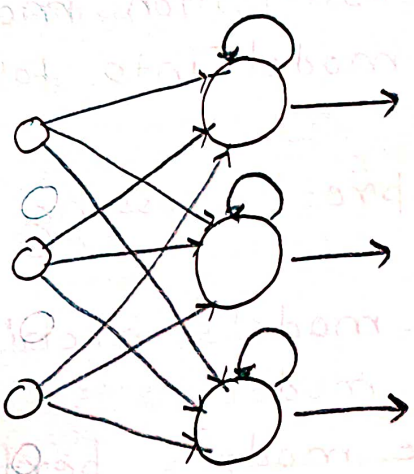


Fig 2 Recurrent Neural Network.

Fully connected

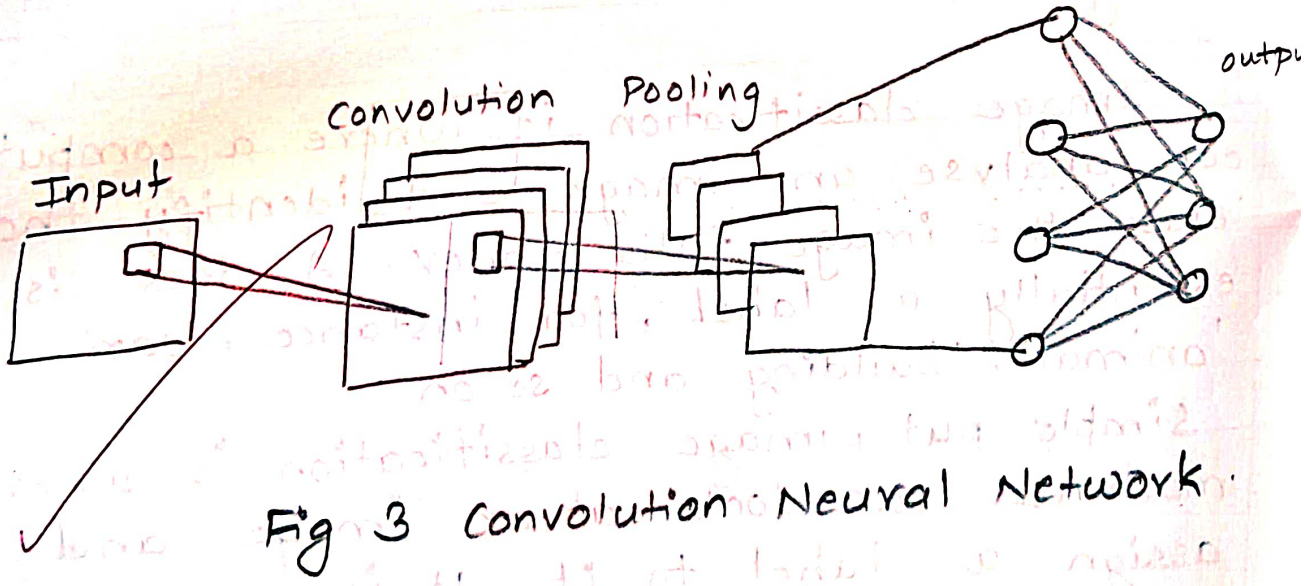


Fig 3 Convolution Neural Network.

Here are just a few examples of what makes it ~~beautiful~~ useful.
self-driving cars use image classification to identify what's around them i.e trees, people, traffic lights and so on

Types of Neural Networks

1). Artifical Neural Network.
ANN is also known as FeedForward Neural network because inputs are processed only in the forward direction
ANN consist 3 layer - Input, Hidden & output
The input layer accepts the inputs the hidden layer processes the inputs and the output layer produces the result.

2). Recurrent Neural Network
RNN has a recurrent connection on a hidden state.
This looping constrait ensures that sequential information is captured in the input data.

3). convolution Neural Network.
CNN are all rage in the deep learning community right now.
These CNN models are being used across different application and domains, and

and they're especially prevalent in i
and video processing projects.

* Architecture of CNN
A convolution neural network, ha
the following layers that are useful
for various deep learning algorithms.
Let us see the working of these laye
taking an example of the image
having dimension 12 x 12 x 4.

Input Layer
This layer will accept the imag
of width 12 height 12 and depth 4

Convolution layer
It computes the volume of image
by getting the dot product between t
image filters possible and the image
patch

Pool Layer
This function mainly reduces the volu
of the intermediate outputs

Conclusion
Hence we have implemented Image
classification model

Practical No 4.

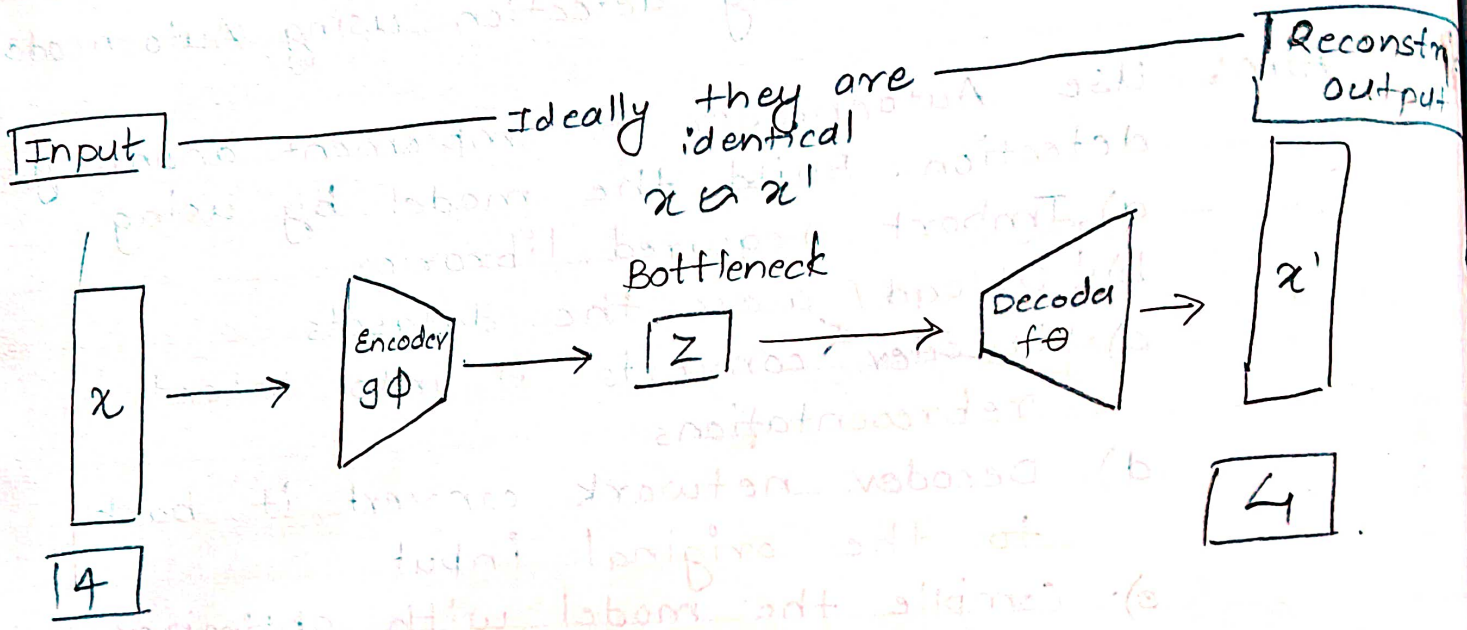
Title: ECG Anomaly detection using Autoencoder

Aim: Use Autoencoder to implement anomaly detection. Build the model by using
a) Import required libraries
b) Upload / access the datasets
c) Encoder converts it into latent representations
d) Decoder network convert it back to the original input
e) Compile the model with optimizer Loss and Evaluation Metrics.

Theory.

Anomaly detection a.k.a outlier detection has been an active research area for several years, due to its broad application in a large number of key domains such as risk management, compliance, security, financial surveillance health and medical risk, and AI safety

Autoencoder encodes the input values x using function f. It then decodes the encoded values f(x) using a function g



Anomoly Detection using Autoencoder.

to create output values identical to the input values.
Autoencoder's objective is to minimizes reconstruction error between the input and output.
when a representation allows a good reconstruction of its input, then it has retained much of the information present in the input.

Types of Anomaly Detection Algorithm.

1). K-Nearest Neighbor
K-NN is one of the simplest supervised learning algorithm and methods in machine learning. It stores all the available examples and then classifies the new ones based on similarities in distance metrics.

2). Local outlier Factors (LOF).
The LOF is a key anomaly detection algorithm based on a concept of a local density. It uses the distance between the k nearest neighbors to estimate the density LOF compares the local density of an item to local densities of its neighbors.

3). K-means
K means is a very popular clustering algorithm

in the data mining area. It creates k group from a set of items so that elements of a group are more similar

4). Support Vector Machine

A SVM is also one of the most effective anomaly detection algorithms SVM is a supervised machine learning technique mostly used in classification problems.

5). Neural Network Based Anomaly Detection

Neural Networks are quite popular algorithms initially designed to mimic biological neurons.

Conclusion

Hence we have implemented Anomaly detection using Autoencoders

Practical No 5

Title: Implement the continuous Bag of Words (CBOW) model

Aim: Implement the CBOW Model. Stages can be
a). Data preparation
b). Generate training model data
c). Train model
d). output

Theory

Natural Language Processing (NLP)

Natural Language Processing refers to Aritifical Intelligence method of communication with an intelligent system using Natural language such as English
Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instruction, when you want to hear decision from dialogue based clinical expert system etc

Word Embedding in NLP is a technique where individual words are represented as

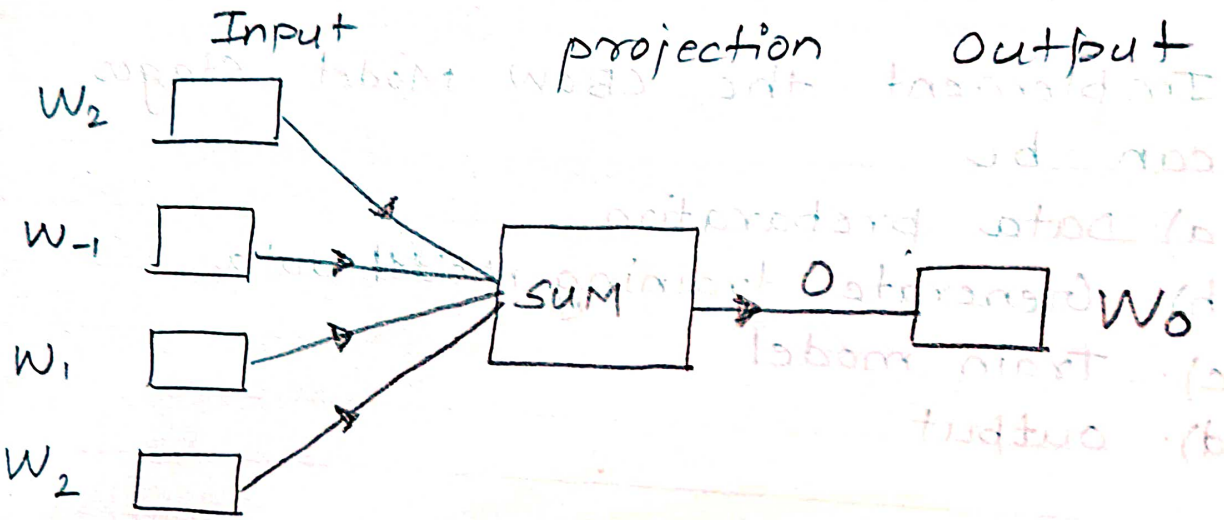


Fig 1 CBOW architecture

real-valued vectors in a lower dimensional space and captures inter-word semantics.

Word2Vec

Word2Vec is a method to construct an word embedding
It can be done CBOW and skip gram
The effectiveness of word2Vec comes from its ability to group together vectors of similar words
Given a large dataset word2Vec can make strong estimates about a word's meaning based on their occurrence in the text
These estimates yield word associations with other words in the corpus.

Continuous Bag of Words

The CBOW model tries to understand the context of the words and takes this ~~this~~ as input. It tries to predict words that even are contextually accurate

The CBOW model architecture tries to predict the target word by trying to understand the context of surrounding words.
Eg: Consider a sentence 'It is a pleasant day'. The model converts this sentence into pairs in the form (context word,

target word)
The user will have to set the window size. If the window for the context word is 2 then the word pairs would look like this : ([it a], is), ([is, pleasant], a), ([a, day], pleasant). With these word pairs, the model tries to predi the target word considered the context wor

If we have 4 context words used for predicting one target word the input layer will be in the form of four 1xw input vectors.
These input vectors will be passed to the hidden layer where it is multiplied by WxN matrix
Finally the 1xN output from the hidden layer enters the sum layer where an element-wise summation is performed on the vectors before a final activation is performed and the outputs is obtained.

Conclusion

Hence we have implemented CBOW model.

Practical No 6

Title: Object detection using Transfer learning of CNN architecture

Aim: Object detection using Transfer learning of CNN architecture
a). Load in a pre-trained CNN model trained on a large dataset
b). Freeze parameter
c). Add custom classifier
d). Fine-tune hyperparameter.

Theory

Transfer Learning

In transfer learning, the knowledge of an already trained machine learning model is applied to a different but related problem. For example if you trained a simple classifier to predict whether an image contains a backpack you could use the knowledge that the model gained during its training to recognize other objects like sunglasses
With transfer learning we basically try to exploit what has been learned in one task to improve generalization in

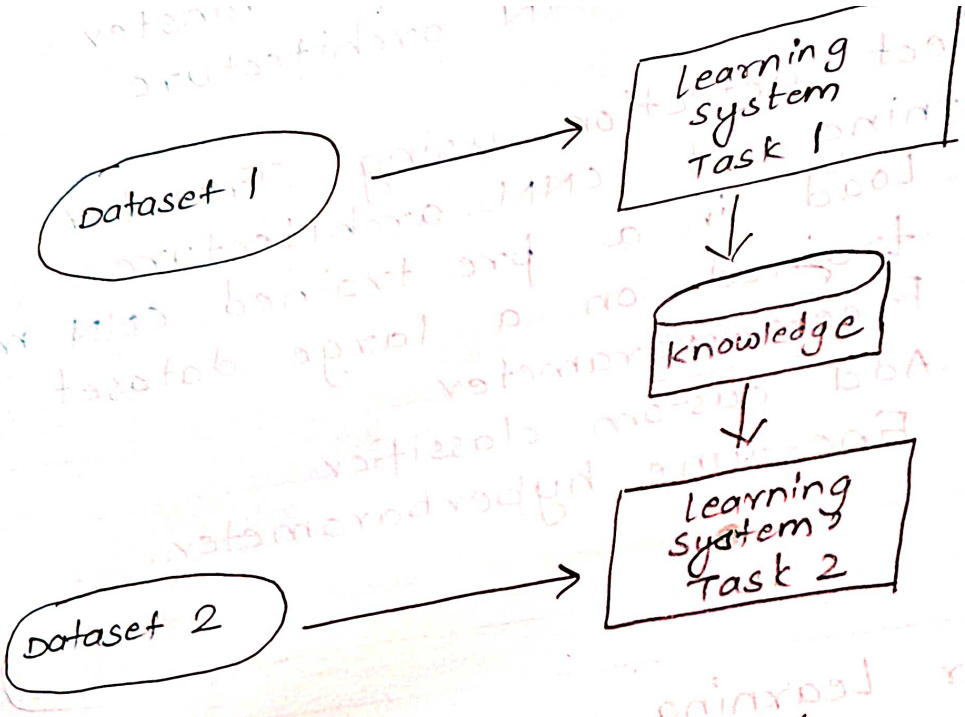


Fig 1 Transfer learning

another. We transfer the weights that ~~has~~ a network has learned at "task A" to a new "task B"
The general idea is to use the knowledge a model has learned from the task with a lot of available labeled training data in a new task that doesn't have much data.
Instead of starting the learning process from scratch, we start with pattern learned from solving a related task.
Transfer learning is mostly used in a computer vision and natural language processing tasks like sentinment anblysis due to huge amount of computational power required

Approaches to Transfer learning

1). Training a model to reuse it
2). Use a Pre-trained model
3). Feature extraction
4). Popular Pre-trained model

Types of Transfer learning

1). Domain Adaptation
2). Domain Confusion
3). Multitasking learning
4) One shot learning

5). Zero-shot learning

Steps for transfer learning
1). obtain pre-trained model
2). Create a base model
3). Freeze layers
4). Train Add new trainable layers
5). Train the new layers
6). Fine-tune your model

Application of Transfer learning
1). Transfer learning for NLP
2). Transfer learning for Audio/Speech
3). Transfer learning for Computer vision

Transfer learning Advantages
1). Helps solve complex real-world problems
2). Tackle problems like having little or almost no labeled data availability
3). Ease transfering knowledge from one model to another based on domains and tasks
4). Provides a path towards achieving advancement in NLP

Conclusion
Hence we have implemented object detection using Transfer Learning of CNN architecture