Lab 1 : Study of Deep learning Packages: Tensorflow, Keras, Theano and PyTorch. Document the distinct features and functionality of the packages.

1. **What is Deep Learning? How is it different from Machine Learning?**

- **Answer**: Deep learning is a subset of machine learning that uses many layers of “neurons” to learn patterns in data on its own, especially for complex tasks like image recognition. Regular machine learning often needs us to choose features manually.

2. **What are neural networks, and how do they work?**

- **Answer**: Neural networks are models inspired by the human brain. They have layers of connected nodes that process data. As data moves through these layers, the network adjusts and learns to make better predictions.

3. **Explain the difference between a Perceptron and a Multilayer Perceptron.**

- **Answer**: A perceptron has only one layer and can handle simple tasks. A multilayer perceptron (MLP) has extra layers, allowing it to solve more difficult problems.

4. **What are activation functions? Why are they used?**

- **Answer**: Activation functions add flexibility to neural networks, helping them learn complex patterns. Common ones are ReLU (used to set negative values to zero) and Sigmoid (which makes output between 0 and 1).

5. **What is backpropagation? Why is it important in neural networks?**

- **Answer**: Backpropagation is how the network learns from mistakes. It works by adjusting weights in the network, helping the model improve with each round of training.

6. **What are some common deep learning frameworks, and what makes them different?**

- **Answer**: Main frameworks include:
  - **TensorFlow**: Great for large projects and works on many devices.
  - **Keras**: Simple and quick, built on TensorFlow.
  - **Theano**: Fast with math but older.
  - **PyTorch**: Flexible and popular for research.

7. **What is TensorFlow, and what are its main advantages?**

- **Answer**: TensorFlow is a Google-built deep learning tool. It's powerful, works on large data.

8. **What is Keras, and how does it relate to TensorFlow?**

- **Answer**: Keras is a tool that runs on top of TensorFlow, making it easier to build models without much coding.

9. **What is meant by overfitting, and how can it be avoided in deep learning?**

- **Answer**: Overfitting is when a model learns too many details from training data and struggles with new data. To avoid it, we use methods like "dropout" (turning off some neurons randomly) or use more data.

10. **What is the purpose of using optimizers in deep learning?**

- **Answer**: Optimizers help the model find the best values for weights, making the model's predictions more accurate. Examples include SGD and Adam.

Lab 2: Implementing Feedforward neural networks with Keras and TensorFlow

a. Import the necessary packages

b. Load the training 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 training loss and accuracy

1. **What is a Feedforward Neural Network?**

- **Answer**: A Feedforward Neural Network is the simplest type of neural network. Data flows in one direction, from input to output, without looping back. It's often used for tasks like image and text classification.

2. **What is the difference between training and testing data?**

- **Answer**: Training data is used to help the model learn patterns. Testing data is separate and used to check if the model can make correct predictions on new, unseen data.

3. **Why do we split data into training and testing sets?**

- **Answer**: Splitting helps us see if the model can generalize well to new data, avoiding overfitting (learning too much from training data alone).

4. **What is SGD (Stochastic Gradient Descent)?**

- **Answer**: SGD is a way to train models by adjusting weights to minimize error. It updates weights using small batches of data, which makes learning faster and uses less memory.

5. **How do we define the architecture of a neural network in Keras?**

- **Answer**: In Keras, we set up layers of the network, like the input, hidden, and output layers, choosing things like the number of neurons and activation functions.

6. **What is accuracy, and why do we use it?**

- **Answer**: Accuracy is the percentage of correct predictions out of all predictions made. It’s used to measure how well the model performs on a task.

7. **Why do we plot training loss and accuracy?**

- **Answer**: Plotting helps us see if the model is learning well. A dropping loss shows the model is learning, and rising accuracy shows it's getting better at making correct predictions.

8. **What does it mean to evaluate a model?**

- **Answer**: Evaluating a model means checking its performance on new data to see how well it generalizes. This helps us understand if the model will perform well in real situations.

9. **What is a hidden layer, and why is it important?**

- **Answer**: Hidden layers are layers between input and output that help the model learn complex patterns. More hidden layers allow the network to solve harder problems.

10. **What is a loss function, and why is it used in training?**

- **Answer**: A loss function measures how far off the model’s predictions are from the actual results. During training, the model tries to minimize this loss to improve its accuracy.

Lab 3: Build the Image classification model by dividing the model into following 4 stages:

a. Loading and preprocessing the image data

b. Defining the model's architecture

c. Training the model

d. Estimating the model's performance

1. **What is Image Classification?**

- **Answer**: Image classification is the task of identifying objects or features in an image, assigning it to a category. For example, a model might classify an image as a "cat" or "dog."

2. **Why do we need to preprocess image data?**

- **Answer**: Preprocessing prepares images so the model can understand them better. This often includes resizing, normalizing colors, and converting them to a format the model can read.

3. **What does it mean to define a model's architecture?**

- **Answer**: Defining the architecture means deciding on the network's structure, like how many layers it has, the type of layers (like convolutional layers for images), and activation functions.

4. **Why do we divide the model-building process into stages?**

- **Answer**: Dividing into stages—like loading data, defining the model, training, and testing—makes it easier to design, debug, and improve each step separately.

5. **What is model training in image classification?**

- **Answer**: Model training is when the model learns patterns from labeled images (images with correct answers) to make predictions on new, unseen images.

6. **Why do we check a model's performance on new data?**

- **Answer**: Checking performance on new data shows if the model can generalize well, meaning it performs accurately even on images it hasn't seen before.

7. **What are convolutional layers, and why are they important in image classification?**

- **Answer**: Convolutional layers are special layers that help a model focus on small details like edges and textures in images. They're essential for handling image data effectively.

8. **What is accuracy in the context of image classification?**

- **Answer**: Accuracy is the measure of how many images the model classified correctly out of all tested images. Higher accuracy means the model is performing well.

9. **What is a validation set, and why do we use it?**

- **Answer**: A validation set is a part of the data used during training to check the model's performance. It helps us adjust settings (like layer numbers or learning rate) without overfitting.

10. **Why is it useful to plot accuracy and loss over time?**

- **Answer**: Plotting accuracy and loss shows if the model is learning correctly. If accuracy improves and loss decreases, it indicates that the model is learning well.

Lab 4: Use Autoencoder to implement anomaly detection. Build the model by using:

a. Import required libraries

b. Upload / access the dataset

c. Encoder converts it into latent representation

d. Decoder networks convert it back to the original input

e. Compile the models with Optimizer, Loss, and Evaluation Metrics

1. **What is Anomaly Detection?**

- **Answer**: Anomaly detection is identifying unusual data points that don't fit with the rest of the data. It's often used to find errors or detect fraud.

2. **What is an Autoencoder?**

- **Answer**: An autoencoder is a type of neural network that learns to compress data into a smaller form and then reconstruct it. It's useful for tasks like anomaly detection.

3. **How does an Autoencoder work?**

- **Answer**: An autoencoder has two main parts:

  - **Encoder**: Compresses the input data.

  - **Decoder**: Rebuilds the data from the compressed form. Differences between the original and rebuilt data can help detect anomalies.

4. **Why are Autoencoders used for Anomaly Detection?**

- **Answer**: Autoencoders learn typical patterns in data. When they see unusual patterns (anomalies), they can't reconstruct them well, helping to identify unusual or faulty data.

5. **What is Latent Representation in an Autoencoder?**

- **Answer**: Latent representation is the compressed form of data created by the encoder. It captures important patterns in a smaller size, useful for efficient data storage and anomaly detection.

6. **What are Loss Functions, and why are they used in Autoencoders?**

- **Answer**: Loss functions measure how different the reconstructed data is from the original. The autoencoder adjusts to reduce this difference, helping it learn better.

7. **What is the purpose of the Optimizer in an Autoencoder?**

- **Answer**: The optimizer adjusts the autoencoder's settings (weights) to minimize the loss, helping it improve its reconstructions over time.

8. **What is Unsupervised Learning, and how does it relate to Autoencoders?**

- **Answer**: Unsupervised learning finds patterns in data without labels. Autoencoders are a type of unsupervised learning, learning patterns without needing labeled data.

9. **How does anomaly detection with autoencoders differ from regular classification?**

- **Answer**: Anomaly detection identifies unusual patterns, while classification assigns data to a specific category. Autoencoders help by identifying unusual data instead of labeling it.

10. **What does it mean to evaluate an Autoencoder model?**

- **Answer**: Evaluating an autoencoder means checking how well it reconstructs data. For anomaly detection, we test if it can correctly identify normal data and anomalies.

Lab 5: Implement the Continuous Bag of Words (CBOW) Model. Stages can be:

a. Data preparation

b. Generate training data

c. Train model

d. Output

1. **What is Natural Language Processing (NLP)?**

- **Answer**: NLP is a field in computer science focused on making computers understand and work with human language, like translating text or detecting sentiment.

2. **What is the Continuous Bag of Words (CBOW) model?**

- **Answer**: The CBOW model is a way to predict a word based on its surrounding words. It's used to learn word relationships, which helps in tasks like translating languages or finding similar words.

3. **How does CBOW differ from Skip-gram?**

- **Answer**: In CBOW, the model predicts a target word from surrounding context words. In Skip-gram, it does the opposite: predicting surrounding words from a single target word.

4. **What are word embeddings?**

- **Answer**: Word embeddings are ways to represent words as numbers so that computers can understand relationships between them. Words with similar meanings have closer embeddings.

5. **Why are word embeddings useful in NLP?**

- **Answer**: Word embeddings capture relationships between words (like “king” and “queen”), allowing models to understand context and similarities, which improves language-based tasks.

6. **How does the CBOW model learn word relationships?**

- **Answer**: CBOW takes surrounding words as input, learns patterns, and predicts the target word, adjusting itself until it makes accurate predictions. This process helps it understand word context.

7. **What is data preparation in CBOW?**

- **Answer**: Data preparation involves cleaning text (like removing punctuation) and converting words to numbers so that the model can learn from the data.

8. **Why do we need training data in CBOW?**

- **Answer**: Training data provides examples of words in sentences, helping the CBOW model learn typical patterns and relationships between words.

9. **What is the purpose of generating training data pairs in CBOW?**

- **Answer**: Generating pairs (target word and context words) helps the model learn by giving it clear examples of which words go together.

10. **What is the output of a CBOW model?**

- **Answer**: The output of a CBOW model is a prediction of the target word based on context words, and the trained word embeddings that can be used in other NLP tasks.

Lab 6: Object detection using Transfer Learning of CNN architectures

a. Load in a pre-trained CNN model trained on a large dataset

b. Freeze parameters (weights) in model's lower convolutional layers

c. Add custom classifier with several layers of trainable parameters to model

d. Train classifier layers on training data available for task

e. Fine-tune hyper parameters and unfreeze more layers as needed

1. **What is Transfer Learning?**

- **Answer**: Transfer learning is using a pre-trained model (one that's already been trained on a large dataset) for a new but similar task. It helps save time and improves performance, especially on smaller datasets.

2. **Why is Transfer Learning useful in deep learning?**

- **Answer**: Transfer learning allows us to use knowledge from a model trained on one task to perform another task, reducing the amount of data and training time needed.

3. **What is a Convolutional Neural Network (CNN)?**

- **Answer**: CNNs are a type of neural network especially good for image data. They use filters to detect features like edges and textures, which helps recognize objects in images.

4. **How does Transfer Learning work with CNNs?**

- **Answer**: In transfer learning with CNNs, we use a model trained on a large dataset like ImageNet, freeze its early layers, and add new layers for our specific task, training only those layers.

5. **What does it mean to "freeze" layers in a CNN?**

- **Answer**: Freezing layers means locking the weights of those layers so they don't change during training. This is often done with the early layers in transfer learning to keep basic features the model already learned.

6. **Why do we add custom layers in Transfer Learning?**

- **Answer**: Adding custom layers lets us tailor the model to our specific task, like detecting a particular object. These layers learn specific patterns relevant to the new data.

7. **What is object detection?**

- **Answer**: Object detection is identifying and locating objects in an image, such as finding cars in a street photo. It’s used in applications like self-driving cars and facial recognition.

8. **How does object detection differ from image classification?**

- **Answer**: Image classification assigns a single label to an image (e.g., "cat" or "dog"). Object detection, however, finds multiple objects within an image and their locations.

9. **Why is fine-tuning important in Transfer Learning?**

- **Answer**: Fine-tuning means slightly adjusting the pre-trained model for better performance on new data. It helps the model adapt to specifics of the new task.

10. **What is hyperparameter tuning, and why is it done in object detection?**

- **Answer**: Hyperparameter tuning involves adjusting settings like learning rate and batch size to improve model performance. It ensures the model learns efficiently for the specific task.