**COURSE CONTENT**

# DEEP NEURAL NETWORKS LABORATORY

**VII Semester: CSE(AI & ML)**

| Course Code | Category | Hours / Week | | | Credits | Maximum Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | L | T | P | C | CIA | SEE | Total |
| ACAC25 | Core | 0 | 0 | 3 | 1.5 | 30 | 70 | 100 |
| Contact Classes: Nil | Tutorial Classes: Nil | Practical Classes: 36 | | | | Total Classes:36 | | |

**Prerequisite: Python Programming, Foundations of Machine Learning**

## I. COURSE OVERVIEW:

This course will introduce students will get practical implementations of various deep learning models using Python and the PyTorch library. Recommended lectures are: Machine Learning, and basic courses on Linear Algebra, Analysis, Probability & Statistics. While it is not a hard requirement, basic knowledge of Python will be greatly helpful.

## II. COURSE OBJECTIVES:

**The students will try to learn:**

I. The theoretical foundations, algorithms and methodologies of Neural Network.
II. The design of single and multi-layer feed-forward deep networks and tunes various hyper-parameters.
III. Gain the practical knowledge in handling and analyzing real world applications.
IV. The role of neural networks in engineering, artificial intelligence, and cognitive modeling.

## III. COURSE SYLLABUS:

**Week – 1: Build a Machine Learning Model**

Problem Statement: Understand the process of developing a machine learning model with different approaches of training. Learn how machine learning models can be used as a powerful tool used to perform complex problem solving efficiently. Make use of IRIS dataset for classification and Boston Housing dataset for regression.

**Solutions Expected:**

a. Contextualize machine learning.
b. Explore the data and choose the type of algorithm.
c. Prepare and clean the dataset.
d. Split the prepared dataset and perform cross validation.
e. Deploy the model.

**WEEK 2: Build a Multi-Layer Perception**

Problem Statement: Develop a multilayer perceptron neural network and understand how perceptrons are inspired by the human brain and try to simulate its functionality to solve problems. Construct different layers of perceptron and train the model in an iterative manner. Use the regularization of the loss function to prevent overfitting in the model.

**Solutions Expected:**

a. Load the HR dataset and perform data pre-processing like label encoding.
b. Divide the dataset into training and testing to assess the model performance.
c. Build a classification model using various parameters.
d. Make prediction and evaluate the model.

**WEEK 3: Build a feed-forward neural network**

Problem statement: Develop a feed-forward neural network and understand how backpropagation algorithm for artificial neural networks can helps in improving the accuracy of a model and get a good prediction.

**Solutions Expected:**

a. Import the libraries to load IRIS dataset.
b. Prepare the training and testing datasets and initialize weights.
c. Build a back propagation model and create a loop by updating the weights in each iteration.
d. Calculate the accuracy cum loss and plot the mean squared error and accuracy graphs.

**WEEK 4: Activation Functions**

Problem Statement: Without an activation function, a neural network is a linear regression model that doesn't perform complex tasks. It can learn and carry out more complex tasks by performing non-linear transformations. Implement the activation functions such as:

**Solutions Expected:**

a. Implement the code to visualize the binary step activation function.
b. Implement the code to visualize the linear activation function.
c. Implement the code to visualize the sigmoid activation function.
d. Implement the code to visualize the Tanh activation function.
e. Implement the code to visualize the RELU activation function.
f. Implement the code to visualize the Softmax activation function.

**WEEK 5: Unsupervised Training of Neural Networks**

Problem Statement: During my first project, I was working with a bank's marketing division. The director asked me to meet with him to talk about a data science project. I was eager to get started and was hoping to find a problem that would allow me to implement my skills and improve the customer's experience. The meeting started promptly. The director told me that the bank had a lot of data about its customers, but it didn't know how to use it to improve its operations. He wanted to use data science to enhance the company's marketing efforts.

**Solutions Expected:**

a. Consider the case study of unsupervised deep learning on the MNIST dataset.
b. Define the problem and organize a photo gallery.
c. Arrange on the photos based on time, and location.
d. Extract semantic meaning from the images.

**WEEK 6: Auto Encoders**

Problem Statement: Autoencoders are only capable of properly compressing data on the images that they have been trained to use. For instance, if they were trained to use images of cats, then it would not perform well. To improve its performance, the training network will learn how to encode data. This process can help reduce the dimensionality of the data. Develop a neural network using autoencoder, loss function, and a decoding method to replicate the loss with minimum loss.

**Solutions Expected:**

a. Import all the required libraries and begin with simple auto encoder.
b. Implement the code to develop a deep CNN auto encoder.
c. Show how the model performs using denoising auto encoder.

**WEEK 7: Convolutional Neural Networks**

Problem Statement: Demonstrate a simple convolutional neural network (CNN) to classify CIFAR images. Create and train the model to perform the image classification. Make sure that the classes are mutually exclusive and there is no overlap between them**.**

**Solutions Expected:**

a. Download and prepare the CIFAR10 dataset.
b. Perform the dataset verifications and create the convolutional base.
c. Add dense layers, compile, train, and evaluate the model.

**WEEK 8: Fooling Neural Network**

Problem Statement: To train the neural network, it can use weights and biases in the two parameters W and b. For each pixel in an image, the program can link the darkness of the object to the probability that it will represent a particular digit. For each digit, the program shows one image with 28 x 28 pixels. Blue and red pixels represent positive and negative weights, respectively.

**Solutions Expected:**

a. Design and develop a neural network using two important parameters like weights and biases.
b. Implement the code for w has 10 rows of 784 elements each, where w[n] contains the weights for digit n.
c. Reshape the elements into 28 x 28-pixel images each time we display them.

**WEEK 9: Performance Evaluation**

Problem Statement: Performance metrics can be used to evaluate the efficiency of various types of algorithms, such as regression, classification, and ML. We must choose the right ones for our analysis since their impact on the results will be entirely dependent on the chosen metric.

**Solutions Expected:**

a. Select any dataset and develop a CNN model.
b. Perform the image classification for binary datasets.
c. Evaluate the model performance using metrics like confusion matrix and
d. Submit the classification report by including accuracy and loss.

**WEEK 10: Stochastic Encoders and Decoders**

Problem Statement: There are two of the practical applications of auto encoders. One of these is data denoising, while the other is the reduction of dimensionality in data visualization. With the appropriate constraints, autoencoders can learn interesting data projections. Implement the code to demonstrate stochastic encoders and decoders.

**Solutions Expected:**

a. Develop a simple and basic possible autoencoder.
b. Create a separate encoder and decoder model and reconstruct the MNIST digits.
c. Configure the model using binary cross entropy and Adam optimizer.
d. Visualize the reconstructed inputs and encoded representations.

**Week – 11**: **Test Model**

Problem Statement: We do not automatically reboot our understanding of language when we hear a sentence. We rely on our previous knowledge to interpret the words given in an article. One of the most critical characteristics that we have is our memory. One of the first techniques that people might think of when it comes to developing an algorithm is a neural network. Unfortunately, this is not feasible with the traditional NNs. Implement to code to understand how recurrent neural networks contributes in predicting the next possible outcome for a given input.

**Solutions Expected:**

a. Perform the data preparation to fit for the model.
b. Create a RNN model which can take in the input sequence.
c. Train the model and check the accuracy on training data.
d. Check the loss on the validation data and perform the actual training.

**Week – 12:Recognize Text In Images**

Problem Statement: One of the most challenging problems that people might encounter when it comes to developing an unsupervised algorithm is sequence prediction. This is different from other problems such as regression and classification. Instead of relying on the previous knowledge, you need to consider the order of the observations. One of the most important factors that you must consider when it comes to implementing LSTM is the learning of temporal dependence. Implement the code to demonstrate how LSTM can be used to address the problem.

**Solutions Expected:**

a. Understand the memorization task to test the capabilities of LSTM.
b. Test the learned temporal dependency capability of LSTM.
c. Implement some arithmetic tasks to evaluate the interpretation capability of LSTM.

## IV. REFERENCESBOOKS


1. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O.,Graves, A., et al. (2016).Conditional image generation with pixelCNN decoders. In Advances in Neural Information Processing Systems.
2. Wierstra, D., F¨orster, A., Peters, J., and Schmidhuber, J. (2009). Recurrent policy gradients.
3. Sutskever, I., Vinyals, O., and Le, Q. V. “Sequence to sequence learning with neural networks, In Advances in neural information processing systems”, 2014.

## V. WEB REFERENCES


1. http://paulorauber.com/slides/deep_learning_lab.pdf
2. https://www.geeksforgeeks.org/python-keras-keras-utils-to_categorical/
3. https://www.geeksforgeeks.org/deep-convolutional-gan-with-keras/?ref=rp
4. https://tinyurl.com/yk4clsot