 **Annexure ‘CD – 01’**

**FORMAT FOR COURSE CURRICULUM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **L** | **T** | **P/S** | **SW/FW** | **No. of PSDA** | **TOTAL CREDIT UNITS** |
| 2 | - | 2 | - | - | 3 |

**Course Title: Natural Language Processing with Deep Learning Credit Units: 3**

**Course Level: UG Course Code:**

**Course Objectives:**

* Understand the basic properties of human languages and descriptive and theoretical frameworks for handling these properties;
* Understand the design of tools for basic NLP tasks such as tagging and partial parsing and be able to apply them to text and evaluate their performance;
* Understand and apply various deep learning models to solve natural language processing problems.

**Pre-requisites:**

1. Course of Machine Learning should have been done.

2. Knowledge of Python language is required.

3. Exposure to Linguistics is useful, though not mandatory.

**Course Contents/Syllabus:**

|  |  |
| --- | --- |
|  | **Weightage (%)** |
| **Module I** | **20%** |
| **Descriptors/Topics**  Introduction of Vectors, Word Analogy, Asses Word Vectors using TF-IDF and t-SNE, word vectors from GloVe, word vectors from word2vec, Text Classification with Word Vectors, Basics of Theano and Tensorflow |
| **Module II** | **20%** |
| **Descriptors/Topics**  Introduction of Language Modeling and Neural networks,Bigrams and Language Models, Neural Bigram Model, Neural Network Bigram Model, Improving Efficiency |
| **Module III** | **20%** |
| **Descriptors/Topics**  Introduction of Word Embedding, Continuous Bag of words Model (CBOW), Skip-Gram, Hierarchical Softmax, Negative Sampling, Word2Vec  Introduction to Matrix Factorization, Matrix Factorization Training and Matrix Factorization Model, Regularization for Matrix Factorization, Introduction to Glove: Global Vectors, Unifying Word2Vec and GloVe, Pointwise Mutual Information (PMI) |
| **Module IV** | **20%** |
| Introduction of Parts-of-Speech (POS) Tagging, Neural Network for POS, Parts-of-Speech Tagging Recurrent Neural Network, Parts-of-Speech Tagging Hidden Markov Model , Named Entity Recognition (NER), Named Entity Recognition RNN |
| **Module V** | **20%** |
| **Descriptors/Topics**  Introduction of Recursive Neural Networks, Sentences as Trees, Data Description for RNN, Tree Neural Network (TNN) with Recursion, Trees to Sequences, Recursive Neural Tensor Networks (RNTN) |

**Course Learning Outcomes:**

* Understand the basics properties of human languages, word analogy and text classification.
* Able to implement the different language models and solve problems using neural network techniques
* Understand and implement basic techniques of word embedding.
* Design and Implement different neural network techniques to solve basic NLP tasks such as tagging and partial parsing and compare them to evaluate their performance;
* Able to formulate and apply deep learning techniques to solve NLP problems.
* Identify appropriate deep learning technique given particular requirements imposed by the data to solve the problem.

**Pedagogy for Course Delivery:** The class will be taught using theory, presentations and seminars. In addition to assigning the case studies, the class will be taught audio-visuals aids and problem based methods. Moreover, the course instructor will spend considerable time in understanding the concept of innovation through the eyes of the industry. The instructor will cover the ways to think innovatively liberally using thinking techniques. Delivery of course will be covered using e-content based on 4-quadrant approach.

**List of Professional Skill Development Activities (PSDA):**

**NILL**

**Lab/ Practicals details, if applicable:**

**List of Experiments:** For practical Labs for Deep Learning with NLP, students may use softwares like MATLAB/Octave or Python.

* Perform Classification with word vectors.
* Implement Neural Network Bigram Model.
* Implementation of word2vec using numpy.
* Implementation of word2vec using tensorflow.
* Implement GloVe using numpy gradient descent.
* Implement GloVe using Alternating Least Squares
* Visualizing data with analogies with t-SNE.
* Visualizing country analogies using embedding projectors.
* Implement GloVe using tensorflow gradient descent.
* Perform Pointwise Mutual Information.
* Implement Recursive Neural Tensor Network using tensorflow.

**Assessment/ Examination Scheme:**

|  |  |
| --- | --- |
| **Theory L/T (%)** | **Lab/Practical/Studio (%)** |
| **67%** | **33%** |

**Theory Assessment (L&T):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Continuous Assessment/Internal Assessment 40%** | | | | | **End Term Examination 60%** |
| **Components (Drop down)** | **Attendance** | **Class Test** | **HA** | **Quiz** | **EE** |
| **Weightage (%)** | 5 | 15 | 10 | 10 | 60 |

**Lab/ Practical/ Studio Assessment:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Continuous Assessment/Internal Assessment 40%** | | | | | **End Term Examination 60%** | |
| Components (Drop down | Performance | Lab Record | Viva | Attendance | Practical | viva |
| Weightage (%) | 15 | 10 | 10 | **5** | **30** | **30** |

**Text Reading:**

1. Brownlee, J. (2017). *Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems*. Machine Learning Mastery.
2. Goyal, P., Pandey, S., & Jain, K. (2018). Deep learning for natural language processing. *Deep Learning for Natural Language Processing: Creating Neural Networks with Python [Berkeley, CA]: Apress*, 138-143.

**References:**

1. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book in preparation. (2015).
2. Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends in Machine Learning 2.1 (2009): 1127.
3. Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of Machine Learning Research 12.Aug (2011): 2493-2537.
4. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.