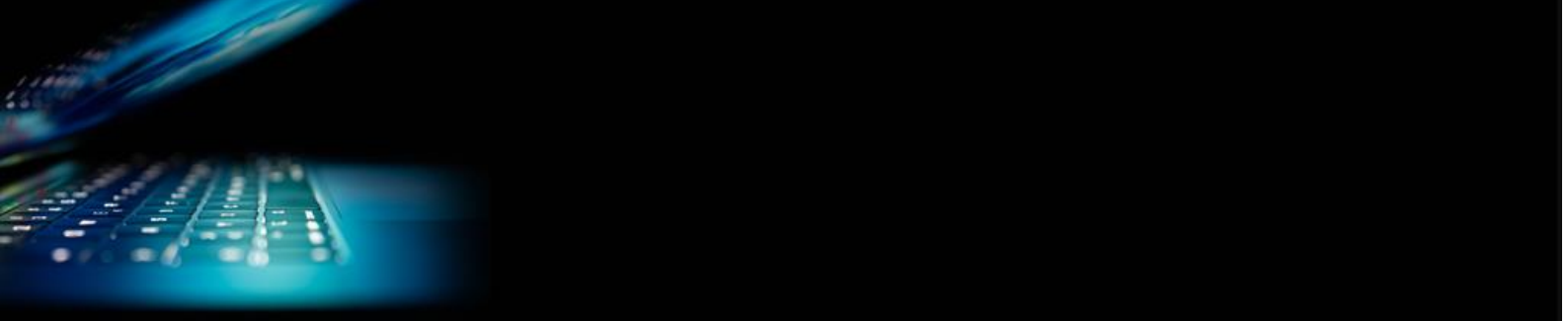


Deep Learning and NLP





Natural Language Processing with Deep Learning in Python

WHAT YOU WILL LEARN

- Understand and implement word2vec
- Understand the CBOW method in word2vec
- Understand the skip-gram method in word2vec
- Understand the negative sampling optimization in word2vec
- Understand and implement GloVe using gradient descent and alternating least squares
- Use recurrent neural networks for parts-of-speech tagging
- Use recurrent neural networks for named entity recognition
- Understand and implement recursive neural networks for sentiment analysis
- Understand and implement recursive neural tensor networks for sentiment analysis

Natural Language Processing in TensorFlow

WHAT YOU WILL LEARN

- Build natural language processing systems using TensorFlow
- Process text, including tokenization and representing sentences as vectors
- Apply RNNs, GRUs, and LSTMs in TensorFlow
- Train LSTMs on existing text to create original poetry and more

Modules

- Module 1 : Introduction to Natural Language Processing
 - Getting Started
 - Knowing each other
 - Welcome to the Course
 - About the Course
 - Introduction to Natural Language Processing
 - Exercise : Introduction to Natural Language Processing
 - Podcast with NLP Researcher Sebastian Ruder
- Module 2 : A Refresher to Python
 - Installation steps for Linux
 - Installation steps for Mac
 - Installation steps for Windows
 - Packages Installation
 - Introduction to Python
 - Variables and Operators
 - Exercise : Variables and Operators
 - Python Lists
 - Exercise : Python Lists

- Dictionaries
 - Exercise : Dictionaries
 - Conditional Statements
 - Exercise : Conditional Statements
 - Loops
 - Exercise : Loops
 - Functions
 - Python Functions Practice
 - Exercise : Functions
 - Packages
 - Exercise : Packages
 - Files
 - Exercise : Files
- Module 3 : Learn to use Regular Expressions
 - Welcome to Module
 - Understanding Regular Expression
 - Implementing Regular Expression in Python
 - Exercise : Implementing Regular Expression in Python
 - Regular Expressions in Action
- Module 4 : First Step of NLP - Text Processing
 - Welcome to Module
 - Tokenization and Text Normalization
 - Exercise : Tokenization and Text Normalization
 - Exploring Text Data
 - Part of Speech Tagging and Grammar Parsing
 - Exercise : Part of Speech Tagging and Grammar Parsing
 - Implementing Text Pre-processing Using NLTK
 - Exercise : Implementing Text Pre-processing Using NLTK
 - Natural Language Processing Techniques using spaCy
- Module 5 : Extracting Named Entities from Text
 - Welcome to Module
 - Understanding Named Entity Recognition
 - Exercise : Understanding Named Entity Recognition
 - Implementing Named Entity Recognition
 - Exercise : Implementing Named Entity Recognition
 - Named Entity Recognition and POS tagging using spaCy
 - POS and NER in Action : Text Data Augmentation
 - Assignment: Share your learning and build your profile
- Module 6 : Feature Engineering for Text
 - Introduction to Text Feature Engineering
 - Count Vector, TFIDF Representations of Text
 - Exercise : Introduction to Text Feature Engineering
 - Understanding Vector Representation of Text
 - Exercise : Understanding Vector Representation of Text
 - Understanding Word Embeddings
 - Word Embeddings in Action - Word2Vec
-

- Word Embeddings in Action - GloVe
- Module 7 : Mastering the Art of Text Cleaning
 - Introduction to Text Cleaning Techniques Part 1
 - Exercise : Introduction to Text Cleaning Techniques Part 1
 - Introduction to Text Cleaning Techniques Part 2
 - Exercise : Introduction to Text Cleaning Techniques Part 2
 - Text Cleaning Implementation
 - Exercise : Text Cleaning Implementation
 - NLP Techniques using spaCy
- Module 8 : Project I - Social Media Information Extraction
 - Project I - Social Media Information Extraction
- Module 9 : Interpreting Patterns from Text - Topic Modelling
 - Introduction to Topic Modelling
 - Exercise : Introduction to Topic Modelling
 - Understanding LDA
 - Exercise : Understanding LDA
 - Implementation of Topic Modelling
 - Exercise : Implementation of Topic Modelling
 - LSA for Topic Modelling
- Module 10: Project II - Categorization of Sports Articles
 - Understanding the Problem Statement
 - Importing Dataset
 - Text Cleaning and Pre-processing
 - Categorizing Articles using Topic Modelling
- Module 11.1 : Machine Learning Algorithms
 - Types of Machine Learning Algorithms
 - Logistic Regression
 - Decision Tree
 - Naive Bayes
 - SVM (Support Vector Machine)
 - Random Forest
- Module 11.2 : Understanding Text Classification
 - Overview of Text Classification
 - Exercise : Overview of Text Classification
 - Assignment: Share your learning and build your profile
- Module 12.1 : Introduction to Deep Learning (Optional)
 - Getting started with Neural Network
 - Exercise : Getting started with Neural Network
 - Understanding Forward Propagation
 - Exercise : Forward Propagation
 - Math Behind Forward Propagation
 - Exercise : Math Behind Forward Propagation
 - Error and Reason for Error

- Exercise : Error and Reason for Error
 - Gradient Descent Intuition
 - Understanding Math Behind Gradient Descent
 - Exercise : Gradient Descent
 - Optimizer
 - Exercise : Optimizer
 - Back Propagation
 - Exercise : Back Propagation
 - Why Keras?
 - Exercise : Why Keras?
 - Building a Neural Network for Text Classification
 - Why CNN?
 - Exercise : Why CNN?
 - Understanding the working of CNN Filters
 - Exercise : Understanding the working of CNN Filters
 - Introduction to Padding
 - Exercise : Introduction to Padding
 - Padding Strategies
 - Exercise : Padding Strategies
 - Padding Strategies in Keras
 - Exercise : Padding Strategies in Keras
 - Introduction to Pooling
 - Exercise : Introduction to Pooling
 - CNN architecture and its working
 - Exercise : CNN architecture and its working
- Module 12.2 : Deep Learning for NLP
 - Deep Learning for NLP Part 1
 - Exercise : Deep Learning for NLP Part 1
 - Deep Learning for NLP Part 2
 - Exercise : Deep Learning for NLP Part 2
 - Text Generation Using LSTM
 - Exercise : Text Generation Using LSTM
- Module 13 : Project III – SMS Spam Classification
 - Dataset download
 - Text Cleaning
 - Feature Engineering
 - Advanced Feature Engineering
 - Combining Features
 - ML Classifier
 - Spam Classification using Deep Learning
- Module 14 : Project IV – Hate Speech Classification
 - Project III
- Module 15 : Project V – Building Auto-Tagging System
 - Overview of Auto-Tagging System
 - Introduction to Dataset and Performance Metrics
-

- Auto-Tagging Implementation Using Machine Learning Part-1
 - Auto-Tagging Implementation Using Machine Learning Part-2
 - Auto-Tagging Implementation Using Deep Learning

 - Module 16 : Recurrent Neural Networks
 - Why RNN
 - Introduction to RNN: Shortcomings of an MLP
 - Introduction to RNN: RNN Architecture
 - Training an RNN: Forward propagation
 - Training an RNN: Backpropagation through time
 - Need for LSTM/GRU
 - Long Short Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
 - Project: Categorisation of websites using LSTM and GRU I
 - Dataset and Notebook
 - Project: Categorisation of websites using LSTM and GRU II

 - Module 17 : Introduction to Language Modelling in NLP
 - Overview : Language Modelling
 - What is a Language Model in NLP?
 - N-gram Language Model
 - Implementing an N-gram Language Model - I
 - Implementing an N-gram Language Model - II
 - Neural Language Model
 - Implementing a Neural Language Model

 - Module 18 : Sequence-to-Sequence Modelling
 - Intuition Behind Sequence-to-Sequence Modelling
 - Need for Sequence-to-Sequence Modelling
 - Understanding the Architecture of Sequence-to-Sequence
 - Understanding Functioning of Encoder and Decoder
 - Case Study: Building an Spanish to English Machine Translation Model
 - Pre-processing of Text Data
 - Converting Text to Integer Sequences
 - Model Building and Inference

 - Module 19 : Project VI - Summarization of Customer Reviews
 - Introduction
 - Pre-processing and Feature Creation
 - Model Building and Summary Generation

 - Module 20 : Project VII - Build your first Chatbot
 - Introduction
 - About this module
 - Overview of Conversational Agents
 - Project - Foodbot
 - Overview of Rasa Framework
 - System Setup
 - Rasa NLU: Understanding user intent from a message
-

- Rasa NLU: Extracting intents from a user's message
 - Rasa Core: Making your chatbot conversational
 - Working with Zomato API
 - Create a Workspace in Slack
 - Deploying to Slack
 - Assignment: Share your learning and build your profile
- Module 21 : Bonus Section (Advance NLP tools)
 - Getting started with Bonus Section
 - Text Classification & Word Representations using FastText (An NLP library by Facebook)
 - Introduction to Flair for NLP: A Simple yet Powerful State-of-the-Art NLP Library
 - Introduction to StanfordNLP: An Incredible State-of-the-Art NLP Library for 53 Languages (with Python code)
 - A Step-by-Step NLP Guide to Learn ELMo for Extracting Features from Text
 - Tutorial on Text Classification (NLP) using ULMFiT and fastai Library in Python
 - 8 Excellent Pretrained Models to get you Started with Natural Language Processing (NLP)
 - Geo-coding using NLP by Shantanu Bhattacharyya and Farhat Habib
 - Demystifying the What, the Why and How of Chatbot by Sonny Laskar
 - Sentiment Analysis using NLP and Deep Learning by Jeeban Swain
 - Identifying Location using Clustering and Language Model - By Divya Choudhary
 - Building Intelligent Chatbots from Scratch

Live Projects -

1) Stock Prices Predictor

This is another interesting machine learning project idea for data scientists/machine learning engineers working or planning to work with finance domain. Stock prices predictor is a system that learns about the performance of a company and predicts future stock prices. The challenges associated with working with stock prices data is that it is very granular, and moreover there are different types of data like volatility indices, prices, global macroeconomic indicators, fundamental indicators, and more. One good thing about working with stock market data is that the financial markets have shorter feedback cycles making it easier for data experts to validate their predictions on new data. To begin working with stock market data, you can pick up a simple machine learning problem like predicting 6 month price movements based on fundamental indicators from an organizations' quarterly report. You can download Stock Market datasets from [Quandl.com](https://www.quandl.com) or [Quantopian.com](https://www.quantopian.com).

2) Human Activity Recognition using Smartphone Dataset

The smartphone dataset consists of fitness activity recordings of 30 people captured through smartphone enabled with inertial sensors. The goal of this machine learning project is to build a classification model that can precisely identify human fitness activities. Working on this machine learning project will help you understand how to solve multi-classification problems.

One can become a master of machine learning only with lots of practice and experimentation. Having theoretical surely helps but it's the application that helps progress the most. No amount of theoretical knowledge can replace hands-on practice. There are many other machine learning projects for beginners like the ones mentioned above that you can work with. However, it will help if you familiarize yourself with the above listed projects first.

If you are a beginner and new to machine learning then working on machine learning projects designed by industry experts at DeZyre will make some of the best investments of your time. These machine learning projects have been designed for beginners to help them enhance their applied machine learning skills quickly whilst giving them a chance to explore interesting business use cases across various domains – Retail, Finance, Insurance, Manufacturing, and more. So, if you want to enjoy learning machine learning, stay motivated, and make quick progress then DeZyre's machine learning interesting projects are for you. Plus, add these machine learning projects to your portfolio and land a top gig with a higher salary and rewarding perks.

3) Learn to build Recommender Systems with Movielens Dataset

From Netflix to Hulu, the need to build an efficient movie recommender system has gained importance over time with increasing demand from modern consumers for customized content. One of the most popular datasets available on the web for beginners to learn how to build recommender systems is the Movielens Dataset which contains approximately 1,000,209 movie ratings of 3,900 movies made by 6,040 Movielens users. You can get started working with this dataset by building a world-cloud visualization of movie titles to build a movie recommender system.

4) Social Media Sentiment Analysis using Twitter Dataset

Social media platforms like Twitter, Facebook, YouTube, Reddit generate huge amounts of big data that can be mined in various ways to understand trends, public sentiments and opinions. Social media data today has become relevant for branding, marketing, and business as a whole. A sentiment analyser learns about various sentiments behind a “content piece” (could be IM, email, tweet or any other social media post) through machine learning and predicts the same using AI. Twitter data is considered as a definitive entry point for beginners to practice sentiment analysis machine learning problems. Using Twitter dataset, one can get captivating blend of tweet contents and other related metadata such as hashtags, retweets, location, users and more which pave way for insightful analysis. Twitter dataset consists of 31,962 tweets and is 3MB in size. Using Twitter data you can find out what the world is saying about a topic whether it is movies, sentiments about US elections or any other trending topic like predicting who would win the FIFA world cup 2018. Working with the twitter dataset will help you understand the challenges associated with social media data mining and also learn about classifiers in depth. The foremost problem that you can start working on as a beginner is to build a model to classify tweets as positive or negative

5) AutoBot

Build a system that can have a conversation with you. The user types messages, and your system replies based on the user's text. Many approaches here ... you could use a large twitter corpus and do language similarity

