

Modern Natural Language Processing with Deep Learning(7-Days)

contents

SEMINAR

- ☞ Description
- ☞ Key Skills
- ☞ Prerequisites
- ☞ Instructional Method
- ☞ course contents

mobile: +91.9880951838
mailto: mohit.riverstone@gmail.com
mailto: mohit@stillwaters.ai
website: www.stillwaters.ai
blog: slowbreathing.github.io

Modern Natural Language Processing with Deep Learning(7-Days)

course contents

SEMINAR

- ☞ Introduction to Deep Learning Day1
- ☞ Deep Neural Network
- ☞ Deep Learning Internals
- ☞ Recurrent Neural Network and Sequence Modelling Day2
- ☞ RNN Internals Day3
- ☞ Gated Units Internals
- ☞ Modern Natural Language Processing(NLP) with Deep Learning-Part-1 Day4
- ☞ Modern Natural Language Processing(NLP) with Deep Learning-Part-2 Day5
- ☞ Modern Natural Language Processing(NLP) with Deep Learning-Part-3
- ☞ Techniques Day6
- ☞ Case Studies
- ☞ Software Tools Day7

mobile:+91.9880951838
mailto:mohit.riverstone@gmail.com
mailto:mohit@stillwaters.ai
website:www.stillwaters.ai
blog:slowbreathing.github.io

Description:

- The main goal of course is to provide a comprehensive details on the recent advances in deep learning applied to NLP. The session presents state of the art of NLP-centric deep learning research, and focuses on the role of deep learning played in major NLP applications including spoken language understanding, dialog systems,lexical analysis, parsing, knowledge graph, machine translation, question answering, sentiment analysis, social computing, and natural language generation (from images). This session is targetted to data scientists, with a technical background in computation, including, post-doctoral researchers, educators, and industrial researchers and anyone interested in getting up to speed with the latest techniques of deep learning associated with NLP. This is an advanced course on natural language processing. Focus will be on few of the most important tactics Neural Machine Translation(NMT), Attention, Bidirectional Encoder Representations from Transformers(BERT) . In modern NLP these are the most important tricks used and this trick underlies most NLP tasks. Various small case studies will be undertaken but the most important being Machine Translation. Unlike other models, These techniques are end-to-end models that can be used for many use cases. These models will be trained differently but models them selves need not be changed. These are really complex models, and getting them to train right or even use right is very difficult. Therefore understanding their inner working is crucial. Refer to the case-study section to see what can be done.

Key Skills:

- Understand Encoder Decoder Architecture
- Understand Neural Machine Translation
- Have an awareness of the hardware issues inherent in implementing scalable neural network models for language data.
- Understand Attention
- Be able to derive and implement optimisation algorithms for these models
- Be able to implement and evaluate common neural network models for language.
- Understand neural implementations of attention mechanisms and sequence embedding models and how these modular components can be combined to build state of the art NLP systems.

Prerequisites:

- Solid knowledge of Linear,Logistic Regression
- Good knowledge of Machine Learning concepts like pipelines, grid search,randomized search, error curves, normalization techniques etc.
- Working Knowledge of Python
- Cursory knowledge of Deep Neural Network

Instructional Method:

- This is an instructor led course provides lecture topics and the practical application of modern NLP with Deep Learning and the underlying technologies. It pictorially presents most concepts and there is a detailed case study that strings together the technologies, patterns and design.

Modern Natural Language Processing with Deep Learning(7-Days)

■ *Introduction to Deep Learning*

- Parameter Hyperspace
- Minimizing Cost Entropy
- Normalized Inputs And Initial Weights
- Measuring Performance
- Transition Into Practical Aspects Of Learning
- Stochastic Gradient Descent
- Training your Logistic Classifier
- Transition: Overfitting -> Dataset Size
- Momentum And Learning Rate Decay
- Supervised Classification
- Solving Problems
- Lather Rinse Repeat
- Optimizing A Logistic Classifier
- Cross Entropy
- What is Deep Learning

■ *Deep Neural Network*

- "2-layer" neural network
- Network Of ReLUs
- Dropout
- Intro to Deep Neural Network
- No Neurons
- Backprop
- Regularization Intro
- Linear Models Are Limited
- The Chain Rule
- Dropout Pt-2
- Regularization

- Training A Deep Learning Network
- ***Deep Learning Internals***
 - How They Work
 - A Simple Predicting Machine
 - Following Signals Through A Neural Network
 - Sometimes One Classifier Is Not Enough
 - Learning Weights From More Than One Node
 - A Three Layer Example with Matrix Multiplication
 - Training A Simple Classifier
 - Backpropagating Errors To More Layers
 - Classifying is Not Very Different from Predicting
 - Making it easy by looking at logic and math
 - Preparing Data
 - Matrix Multiplication is Useful Honest!
 - Neurons, Nature's Computing Machines
 - How Do We Actually Update Weights?
 - Weight Update Worked Example
 - Backpropagating Errors with Matrix Multiplication
 - Backpropagating Errors From More Output Nodes
 - DIY with Python
 - Interactive Python = IPython
 - A Very Gentle Start with Python
 - The MNIST Dataset of Handwritten Numbers
 - Python
 - Neural Network with Python
 - Hand rolled Neural Network
 - Creating New Training Data: Rotations
 - Your Own Handwriting
 - Inside the Mind of a Neural Network
- ***Recurrent Neural Network and Sequence Modelling***
 - Concrete Recurrent Neural Network Architectures

- Simple RNN
- Gated Architectures:LSTM
- Gated Architectures:GRU
- CBOW as an RNN
- Dropout in RNNs
- Gated Architectures:Other Variants
- Recurrent Neural Networks: Modeling Sequences and Stacks
 - Transducer
 - RNN Training
 - RNN Abstraction
 - Common RNN Usage-patterns
 - A Note on Reading the Literature
 - Encoder
 - Multi-layer (stacked) RNNs
 - RNNs for Representing Stacks
 - Acceptor
 - Bidirectional RNNs (biRNN)
- Modeling with Recurrent Networks
 - Acceptors
 - RNN–CNN Document Classification
 - RNNs as Feature Extractors
 - Subject-verb Agreement Grammaticality Detection
 - Arc-factored Dependency Parsing
 - Part-of-speech Tagging
 - Sentiment Classification
- Conditioned Generation
 - Applications
 - Sequence to Sequence Models
 - Syntactic Parsing
 - Morphological Inflection
 - Attention-based Models in NLP

- Computational Complexity
- Conditioned Generation with Attention
- Machine Translation
- Training Generators
- Interpretability
- Other Conditioning Contexts
- Conditioned Generation (Encoder-Decoder)
- Unsupervised Sentence Similarity
- RNN Generators
- **Models for Sequence Analysis**
 - Dissecting a Neural Translation Network
 - Beam Search and Global Normalization
 - Tackling seqseq with Neural N-Grams
 - Implementing a Sentiment Analysis Model
 - Long Short-Term Memory (LSTM) Units
 - Recurrent Neural Networks
 - Implementing a Part-of-Speech Tagger
 - A Case for Stateful Deep Learning Models
 - Solving seqseq Tasks with Recurrent Neural Networks
 - The Challenges with Vanishing Gradients
 - Augmenting Recurrent Networks with Attention
 - Dependency Parsing and SyntaxNet
 - TensorFlow Primitives for RNN Models
 - Analyzing Variable-Length Inputs
- ***RNN Internals***
 - Backward Propagation
 - Unrolling
 - Forward Propagation
 - Matrix and their Shapes
 - GPU Optimization
 - Recurrent Neurons
- ***Gated Units Internals***

- Forward Propagation
 - Significance of Batch and Sequence
 - Recurrence Depth
 - Gates and their significance
 - Feed Forward Depth
 - Introduction
 - Saliency Heatmap
 - Why
 - variants
 - Weight Shapes
 - Bi Directional
 - Backward Propagation
 - Uni Directional
 - Dropouts
- ***Modern Natural Language Processing(NLP) with Deep Learning-Part-1***
 - Pre-trained Word Representations
 - Character-based and Sub-word Representations
 - Sentences, Paragraphs, or Documents
 - Limitations of Distributional Methods
 - Other Algorithms
 - Using Pre-trained Embeddings
 - Choice of Contexts
 - Connecting the Worlds
 - Syntactic Window
 - Unsupervised Pre-training
 - Dealing with Multi-word Units and Word Inflections
 - Random Initialization
 - Word Embedding Algorithms
 - From Neural Language Models to Distributed Representations
 - Multilingual
 - Window Approach

- Distributional Hypothesis and Word Representations
- Supervised Task-specific Pre-training
- Working with Natural Language Data
 - Directly Observable Properties
 - Typology of NLP Classification Problems
 - Distributional Features
 - Ngram Features
 - Features for NLP Problems
 - Features for Textual Data
 - Core Features vs Combination Features
 - Inferred Linguistic Properties
- Language Modeling
 - Using Language Models for Generation
 - Limitations of Traditional Language Models
 - Evaluating Language Models: Perplexity
 - Language Modeling Task
 - Traditional Approaches to Language Modeling
 - Neural Language Models
 - Byproduct: Word Representations
- From Textual Features to Inputs
 - Encoding Categorical Features
 - Odds and Ends
 - Variable Number of Features: Continuous Bag of Words
 - One-hot Encodings
 - Embeddings Vocabulary
 - Example: Part-of-Speech Tagging
 - Feature Combinations
 - Distance and Position Features
 - Padding, Unknown Words, and Word Dropout
 - Dense Encodings (Feature Embeddings)
 - Network's Output
 - Vector Sharing

- Combining Dense Vectors
- Relation Between One-hot and Dense Vectors
- Example: Arc-factored Parsing
- Dimensionality
- Dense Vectors vs One-hot Representations
- Window-based Features
- Using Word Embeddings
 - Odd-one Out
 - Short Document Similarity
 - Word Clustering
 - Retrofitting and Projections
 - Word Analogies
 - Word Similarity
 - Finding Similar Words
 - Similarity to a Group of Words
 - Obtaining Word Vectors
 - Case Study: A Feed-forward Architecture for Sentence Meaning Inference
 - A Textual Similarity Network
 - Practicalities and Pitfalls
 - Natural Language Inference and the SNLI Dataset
- Case Studies of NLP Features
 - Relation Between Words in Context: Arc-Factored Parsing
 - Document Classification: Authorship Attribution
 - Document Classification: Topic Classification
 - Word in Context, Linguistic Features: Preposition Sense Disambiguation
 - Document Classification: Language Identification
 - Word-in-context: Named Entity Recognition
 - Word-in-context: Part of Speech Tagging
- ***Modern Natural Language Processing(NLP) with Deep***

Learning-Part-2

- Deep Learning in Question Answering
 - Deep Learning in Machine Comprehension
 - Deep Learning in Question Answering over Knowledge Base
- Deep Learning in Machine Translation
 - End-to-End Deep Learning for Machine Translation
 - Statistical Machine Translation and Its Challenges
 - Component-Wise Deep Learning for Machine Translation
- Natural Language Understanding:Neural Machine Translation:Attention+(Plus)
 - Pairing Strategies
 - Rare word Translation
 - Input Feeding
 - Performance
 - Alignment Score
 - Global vs Local Weights
 - MultiLingual Model
 - Performance
 - Utilize Monolingual Data
 - Achieving State of the art results
- Natural Language Understanding
 - Introduction
- Deep Learning in Sentiment Analysis
 - Opinion Mining
 - Sentiment-Specific Word Embedding
 - Fine-Grained Sentiment Analysis
 - Document-Level Sentiment Classification
 - Sentence-Level Sentiment Classification
- Natural Language Understanding:Neural Machine Translation:Attention
 - Avoid the curse of length

- Identifying bottlenecks with the vanilla seq-to-seq structure
- Attention
 - Achieving State of the art results
 - Performance
 - BahdanauAttention
 - Soft Alignment
 - Backward Propagation
 - Context Vector
 - Image Text Embedding
 - Alignment
 - Forward Propagation
 - QA
 - LuongAttention
 - Attention Mechanism
 - Image Generation
 - Training
 - Big Wins
- Natural Language Understanding:Neural Machine Translation:GoogleNMT(GNMT)
 - GNMT Decoder
 - GNMT Encoder
 - Achieving State of the art results
- Natural Language Understanding:Neural Machine Translation
 - Backward Propagation
 - Evaluation Metrics
 - Implementation techniques
 - RBMT vs SMT vs NMT
 - Forward Propagation
 - Performance
 - Formulation: Sequence-to-sequence

- Identifying Bottlenecks
- Training
- Big Wins
- Goal: End-to-End
- Encoder-Decoder Architecture
 - LSTM
 - Bi-LSTM
 - GRU
 - CNN
- Decoder
 - Conditional recurrence language model
 - Output
 - Decoder Transition
 - Decoder Strategies
 - Naive Search
 - Beam Search
 - Greedy Search
 - Encoder
 - Context Vector
 - Encoder Transition
 - Inputs
- ***Modern Natural Language Processing(NLP) with Deep Learning-Part-3***
 - BERT (Bidirectional Encoder Representations from Transformers)
 - Take BERT out for a spin
 - Piping out outputs
 - Comparisions with Convnets
 - BERT: From Decoders to Encoders
 - Task specific-Models
 - Two-sentence Tasks

- Masked Language Model
- Transfer Learning to Downstream Tasks
- OpenAI Transformer: Pre-training a Transformer Decoder for Language Modeling
- Piping in Inputs
- Architecture
- BERT for feature extraction
- The Transformer: Going beyond LSTMs
- Transformer
 - Encoding
 - Encoders
 - High Level View
 - Tensors
 - Linear Softmax layer
 - Loss Function
 - Decoder side
 - Decoders
 - The Residuals
- ELMO:Advanced Word Embeddings
 - Training ELMo on corpus
 - Salient features
 - Problem at hand?
 - Loading the ELMo embedding
 - Token Representation:
 - Bidirectional Language Model (biLM)
 - How does it do it? Using Long Short-Term Memory (LSTM)
 - Let's see the architecture:
 - What's already existing?
 - What numbers do they improve on?
 - Deep contextualized word representation
 - Why to look for a new method?
 - Let's dive in crux!

- **Self Attention**
 - Attention is all you need
 - Representing The Order of The Sequence Using Positional Encoding
 - Matrix Calculation of Self-Attention
 - Visual Attention
 - Machine Translation
 - Scaled dot product attention
 - Model Variations
 - Simultaneously Self-Attending to All Mentions for Relation Extraction
 - Applying Attention Throughout the Entire Model
 - Deep Semantic Role Labeling With Self-Attention
 - Detailed Architecture
 - Multi Head Attention
 - Self-Attention in NLP
 - Multi Headed Attention
 - Why Self Attention
- **ULMFIT**
 - Input Dropouts
 - Multi Batch Encoder
 - Weight Dropouts
 - Variable Length BPTT
 - Building Blocks
 - Transfer Learning with ULMFIT
 - Hidden Dropout
 - Encoder Dropouts
 - QRNN
 - AWD_LSTM
- ***Techniques***
 - Multi-LSTM
 - Seq2seq models

- TextCNN
- Text clustering
- BiMulti-LSTM
- Self Attention
- NMT(Neural Machine Translations)
- Seq2Seq with Attention
- TextLSTM
- Word Embeddings
- Encoder-Decoder models
- Neural Network
- BERT(Bidirectional Encoder Representations from Transformers)
- ELMO
- TextRNN
- NMT(Neural Machine Translations) with Attention
- Transformer
- ULMFIT
- Bi-LSTM
- ***Case Studies***
 - Relation extraction
 - Text Summarization
 - Subjectivity analysis
 - Morphological analysis
 - Sentence classification
 - QA(Questions and Answers)
 - Information extraction
 - Chatbot
 - Semantic textual similarity
 - Word sense disambiguation
 - Spelling correction
 - Named entity recognition
 - Entity linking
 - Sarcasm detection

- Grammatical error correction
- Emotion classification
- Document classification
- Sentiment analysis
- Language Modeling
- Topic Modeling
- Slot filling
- Text classification / categorization
- Intent classification
- Paraphrase generation
- Hate speech detection
- ***Software Tools***
 - **Tensorflow**
 - Installation
 - Sharing Variables
 - Creating Your First Graph and Running It in a Session
 - Managing Graphs
 - Visualizing the Graph and Training Curves Using TensorBoard
 - Implementing Gradient Descent
 - Lifecycle of a Node Value
 - Linear Regression with TensorFlow
 - Modularity
 - Saving and Restoring Models
 - Name Scopes
 - Feeding Data to the Training Algorithm
 - **Keras**