Deep Learning Internals

SEMINAR

contents

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

Deep Learning Internals

SEMINAR

course contents

- ✋ Introduction to Deep Learning Day1
- ✋ Deep Neural Network
- ✋ Deep Learning Internals
- ✋ Software Tools Day2
- ✋ Artificial Neural Networks Internals
- ✋ Convolutional Neural Networks Internals
- ✋ Recurrent Neural Network and Sequence Modelling Day3
- ✋ Reinforcement Learning Day4
- ✋ Deep Generative Models
- ✋ Crash Course in GPU Day5
- ✋ Applications
- ✋ Practical Methodology

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

Description:

- You’ve probably heard that Deep Learning is making news across the world as one of the most promising techniques in machine learning, especially for analyzing image data. With every industry dedicating resources to unlock the deep learning potential, to be competitive, you will want to use these models in tasks such as image tagging, object recognition, speech recognition, and text analysis. In this training session you will build deep learning models using neural networks, explore what they are, what they do, and how. To remove the barrier introduced by designing, training, and tuning networks, and to be able to achieve high performance with less labeled data, you will also build deep learning classifiers tailored to your specific task using pre-trained models, which we call deep features. Also, you’ll develop a clear understanding of the motivation for deep learning, and design intelligent systems that learn from complex and/or large-scale datasets.

Key Skills:

- Combine different types of layers and activation functions to obtain better performance
- Describe how these models can be applied in computer vision, text analytics and speech recognition
- Describe how a neural network model is represented and how it encodes non-linear features
- Use pretrained models, such as deep features, for new classification tasks
- You will learn how to Prototype ideas and then productionize
- Explore a dataset of products, reviews and images
- Use Tensorflow and Keras for building detailed CNN based models
- Use Tensorflow and Keras for building detailed RNN based models
- Value of GPU in Deep Learning computation and provisioning rules

Prerequisites:

- This is an advanced level session and it assumes that you have good familiarity with Machine learning.
- Machine Learning Internals
- Working Knowledge of python

Instructional Method:

- This is an instructor led course provides lecture topics and the practical application of 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.

Deep Learning Internals

- ***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***

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

- 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

- ***Software Tools***
  - Tensorflow
    - Installation
    - Sharing Variables

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

        - 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
- ***Artificial Neural Networks Internals***
    - Training a DNN Using Plain TensorFlow
    - Fine-Tuning Neural Network Hyperparameters
    - Training an MLP with TensorFlow’s High-Level API
    - From Biological to Artificial Neurons
    - Deep Feedforward Networks
        - Hidden Units
        - Architecture Design
        - Back-Propagation and Other Differentiation Algorithms
        - Gradient-Based Learning
        - Learning XOR
    - Optimization for Training Deep Models
        - Random or Unsupervised Features
        - Structured Outputs
        - Convolutional Networks
        - Challenges in Neural Network Optimizatio
        - Efficient Convolution Algorithms
        - Variants of the Basic Convolution Function
        - The Convolution Operation
        - Pooling
        - Parameter Initialization Strategies

- Motivation
- How Learning Differs from Pure Optimization
- Basic Algorithms
- Optimization Strategies and Meta-Algorithms
- The Neuroscientific Basis for Convolutional Networks
- Convolution and Pooling as an Infinitely Strong Prior
- Data Types
- Approximate Second-Order Methods
- Algorithms with Adaptive Learning Rates

▪ Regularization for Deep Learning

- Bagging and Other Ensemble Methods
- Dataset Augmentation
- Tangent Distance, Tangent Prop, and Manifold Tangent Classifier
- Parameter Tying and Parameter Sharing
- Semi-Supervised Learning
- Early Stopping
- Sparse Representations
- Regularization and Under-Constrained Problems
- Multi-Task Learning
- Dropout
- Norm Penalties as Constrained Optimization
- Noise Robustness
- Parameter Norm Penalties
- Adversarial Training

▪ ***Convolutional Neural Networks Internals***

- Pooling layer
- Image augmentation
- Convolutional layer
- History of CNNs
- Convolutional layers in Keras
- Code for visualizing an image
- Input layer

- How do computers interpret images?
- Practical example image classification
- Convolutional neural networks
- Dropout

- Attention Mechanism for CNN and Visual Models
  - Types of Attention
  - Glimpse Sensor in code
  - Attention mechanism for image captioning
  - Hard Attention
  - Applying the RAM on a noisy MNIST sample
  - Recurrent models of visual attention
  - Using attention to improve visual models
  - Reasons for sub-optimal performance of visual CNN models
  - Soft Attention
- Build Your First CNN and Performance Optimization
  - Convolution and pooling operations in TensorFlow
  - Convolutional operations
  - Using tanh
  - Convolution operations in TensorFlow
  - Regularization
  - Fully connected layer
  - Weight and bias initialization
  - Pooling, stride, and padding operations
  - CNN architectures and drawbacks of DNNs
  - Applying pooling operations in TensorFlow
  - Using sigmoid
  - Training a CNN
  - Using ReLU
  - Activation functions
  - Building, training, and evaluating our first CNN
    - Creating a CNN model
    - Defining CNN hyperparameters

        - Model evaluation
        - Dataset description
        - Loading the required packages
        - Running the TensorFlow graph to train the CNN model
        - Preparing the TensorFlow graph
        - Loading the training/test images to generate train/test set
        - Constructing the CNN layers
    - Model performance optimization
        - Applying dropout operations with TensorFlow
        - Building the second CNN by putting everything together
        - Appropriate layer placement
        - Which optimizer to use?
        - Creating the CNN model
        - Dataset description and preprocessing
        - Number of neurons per hidden layer
        - Number of hidden layers
        - Batch normalization
        - Memory tuning
        - Training and evaluating the network
        - Advanced regularization and avoiding overfitting
- Popular CNN Model Architectures
    - Architecture insights
    - ResNet architecture
    - AlexNet architecture
    - VGG image classification code example
    - Introduction to ImageNet
    - VGGNet architecture
    - GoogLeNet architecture
    - LeNet
    - Traffic sign classifiers using AlexNet
    - Inception module
- Transfer Learning

- Multi-task learning
- Target dataset is small but different from the original training dataset
- Autoencoders for CNN
- Applications
- Target dataset is large and similar to the original training dataset
- Introducing to autoencoders
- Convolutional autoencoder
- Target dataset is large and different from the original training dataset
- Transfer learning example
- Feature extraction approach
- Target dataset is small and is similar to the original training dataset
- An example of compression

▪ GAN: Generating New Images with CNN

- Feature matching
- GAN code example
- Deep convolutional GAN
- Adding the optimizer
- Training a GAN model
- Semi-supervised learning and GAN
- Pixpix - Image-to-Image translation GAN
- Calculating loss
- Semi-supervised classification using a GAN example
- CycleGAN
- Batch normalization

▪ Object Detection and Instance Segmentation with CNN

- Creating the environment
- Fast R-CNN (fast region-based CNN)
- The differences between object detection and image classification
- Mask R-CNN (Instance segmentation with CNN)

- Cascading classifiers
- Haar Features
- Faster R-CNN (faster region proposal network-based CNN)
- Traditional, nonCNN approaches to object detection
- R-CNN (Regions with CNN features)
- Running the pre-trained model on the COCO dataset
- Why is object detection much more challenging than image classification?
- The Viola-Jones algorithm
- Preparing the COCO dataset folder structure
- Downloading and installing the COCO API and detectron library (OS shell commands)
- Instance segmentation in code
- Haar features, cascading classifiers, and the Viola-Jones algorithm
- Installing Python dependencies (Python environment)

▪ Popular CNN Model Architectures

- Introduction to ImageNet
- VGG image classification code example
- GoogLeNet architecture
- Architecture insights
- Inception module
- AlexNet architecture
- VGGNet architecture
- LeNet
- ResNet architecture
- Traffic sign classifiers using AlexNet

▪ ***Recurrent Neural Network and Sequence Modelling***

▪ Concrete Recurrent Neural Network Architectures

- Simple RNN
- Gated Architectures:LSTM
- Gated Architectures:GRU
- CBOW as an RNN

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

- 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

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

- 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

▪ ***Reinforcement Learning***

- Policy Gradient Methods
- Integrating Learning and Planning
- Model-Free Control
- Exploration and Exploitation
- Markov Decision Processes
- Case Study: RL in Classic Games
- Introduction to Reinforcement Learning
- Planning by Dynamic Programming
- Model-Free Prediction
- Value Function Approximation

▪ ***Deep Generative Models***

- Deep Boltzmann Machines
- Back-Propagation through Random Operations
- Restricted Boltzmann Machines
- Boltzmann Machines for Structured or Sequential Outputs
- Generative Stochastic Networks
- Boltzmann Machines
- Other Boltzmann Machines
- Other Generation Schemes
- Directed Generative Nets
- Boltzmann Machines for Real-Valued Data
- Evaluating Generative Models
- Drawing Samples from Autoencoders
- Deep Belief Networks
- Convolutional Boltzmann Machines

▪ ***Crash Course in GPU***

- Introduction to CUDA and OpenCL
- Fundamentals of GPU Algorithms(Applications of Sort and Scan)
- Dynamic Parallelism
- Optimizing GPU Programs
- The GPU Hardware and Parallel Communication Patterns
- Parallel Computation Patterns
- The GPU programming Model
- Parallel Optimization Patterns
- Fundamentals of GPU Algorithms(Reduce,Scan,Histograms)
- Deep Learning use of GPU

▪ ***Applications***

- Other Applications
- Computer Vision
- Natural Language Processing
- Large Scale Deep Learning
- Speech Recognition

▪ ***Practical Methodology***

- Selecting Hyperparameters
- Debugging Strategies
- Example : Facial Recognition
- Performance Metrics
- Determining Whether to Gather More Data
- Default Baseline Models

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