|  |
| --- |
| **SPA** |
| **Thinking about Data Systems** |
| **Reliable, Scalable and Maintainable Data Applications** |
| **Scaling with the traditional databases** |
| **Big Data Systems** |
| **Desired properties of Big Data Systems** |
| **Data Model for Big Data** |
| **Generalized Big Data System Architecture** |
| **Real time systems** |
| **Difference between Batch processing and Stream Processing** |
| **Difference between real time and streaming systems** |
| **Streaming Data Applications** |
| **Databases and Streams** |
| **Usage patterns of Streaming Data** |
| **Sources of Streaming Data** |
| **Complex Event Processing Systems** |
| **Explore more on the non-functional requirements of Data Intensive Applications** |
| **-** |
| **Generalized Streaming Data Architecture** |
| **Lambda Architecture** |
| **Kappa Architecture** |
| **Streaming Data system Component** |
| **Features of Real time Architecture** |
| **A real-time architecture checklist** |
| **-** |
| **Service Configuration and Coordination Systems** |
| **Maintaining the state** |
| **Apache ZooKeeper** |
| **Data Flow Manager** |
| **Managing distributed data flows with Apache Kafka** |
| **Kafka Fundamentals Overview** |
| **Use-Cases and applications** |
| **Architecture** |
| **Kafka Topics, Producer and Consumer Using CLI** |
| **Programming Kafka** |
| **Simple Kafka Producer** |
| **Simple Kafka Consumer** |
| **Producer, Consumer Configuration** |
| **Producer, Consumer Execution** |
| **Kafka Consumer Groups** |
| **-** |
| **Streaming Data Processor Concepts** |
| **Timing Concepts** |
| **Windowing** |
| **Joins** |
| **✓ questioning-the-lambda-architecture ✓ a-brief-introduction-to-two-data-processing architectures • Explore the Java APIs exposed by the following systems ✓ Apache ZooKeeper ✓ Apache Kafka • Explore the data models of NoSQL data systems ✓ MongoDB ✓ Cassandra Self-study on other frameworks M3: Streaming Data Frameworks** |
| **-** |
| **Key features of Streaming Data Frameworks** |
| **SELF Exploration/Assignment on the following • Apache Flink • Apache Samza • Apache Kafka Streaming • Apache Storm Spark Streaming Guide Flink Docs Samza Docs Kafka Streaming Guide Storm Docs** |
| **Apache Spark Streaming** |
| **Spark Streaming fundamentals** |
| **Motivation** |
| **Difference between Spark Streaming API and Spark API** |
| **Architecture** |
| **Components of Spark Engine** |
| **Spark Application Architecture** |
| **Fault Tolerance** |
| **Comparison with Traditional Streaming Systems** |
| **Spark + Kafka integration** |
| **Structured Streaming** |
| **Developing application in Databricks platform** |
| **Compare the different streaming data platforms and identify the use cases for which they are suitable • Implement the streaming data pipeline using the Kafka Streaming library • Implement a streaming data application with Spark streaming Kafka Streaming Guide Spark Streaming Guide M4: Streaming Analytics** |
|  |
| **Exact Aggregation of Streaming Data** |
| **Registers and Hash Functions** |
| **Study illustrations for Streaming data concepts • Explore algorithms for aggregation of streaming data • Explore more about the streaming data processing algorithms for exact results Class Notes M5: Advanced Streaming Applications** |
| **-** |
| **Necessity of Streaming SQL** |
| **Streaming SQL: Windows** |
| **Streaming SQL: Joins** |
| **Streaming SQL: Patterns** |
| **Streaming SQL for Apache Kafka** |
| **Streaming Analytics with Cloud** |
| **AWS Kinesis** |
| **Data Streams** |
| **Data Firehose** |
| **Data Analytics** |
| **AWS IoT / Streaming Analytics Service** |
| **Channels, Pipelines** |
| **Data stores & data sets** |
| **Streaming ML Frameworks** |
| **Get familiarized with Streaming SQL tools ✓ Kafka Streaming SQL • Build and deploy machine learning models using Spark structured streaming ✓ structured-streaming-ml** |

|  |
| --- |
| **BDS** |
| **Structured Data (Relational Databases), Semi-structured data (Object Stores), and Unstructured Data (File systems) What is Big Data? Characteristics of Big Data. Systems perspective - Processing: In-memory vs. (from) secondary storage vs. (over the) network** |
| **Locality of Reference: Principle, examples Impact of Latency: Algorithms and data structures that leverage locality, data organization on disk for better locality** |
| **Parallel and Distributed Processing: Motivation (Size of data and complexity of processing); Storing data in parallel and distributed systems: Shared Memory vs. Message Passing; Strategies for data access: Partition, Replication, and Messaging.** |
| **Memory Hierarchy in Distributed Systems: In-node vs. over the network latencies, Locality, Communication Cost. Distributed Systems: Motivation (size, scalability, cost-benefit), Client-Server vs. Peer-to-Peer models, Cluster Computing: Components and Architecture** |
| **Big Data Analytics: Requirements, constraints, approaches, and technologies.** |
| **Big Data Systems – Characteristics: Failures; Reliability and Availability; Consistency – Notions of Consistency.** |
| **CAP Theorem and implications for Big data Analytics** |
| **Big Data Lifecycle: Data Acquisition, Data Extraction – Validation and Cleaning, Data Loading, Data Transformation, Data Analysis and Visualization. Case study – Big data application** |
| **Distributed Computing - Design Strategy: Divide-and-conquer for Parallel / Distributed Systems - Basic scenarios and Implications. Programming Patterns: Data-parallel programs and map as a construct; Tree-parallelism, and reduce as a construct; Map-reduce model: Examples (of map, reduce, map-reduce combinations, and Iterative map-reduce)** |
| **Hadoop: Introduction, Architecture, and Map-reduce Programming on Hadoop** |
| **Hadoop: Hadoop Distributed File System (HDFS), Scheduling in Hadoop (using YARN). Example – Hadoop application.** |
| **Hadoop Ecosystem: Databases and Querying (HBase, Pig, and Hive)** |
| **Hadoop Ecosystem: Integration and coordination (Sqoop, Flume, Zookeeper & Oozie)** |
| **NoSQL databases: Introduction, Architecture, Querying, Variants, Case Study.** |
| **Spark: Introduction, Architecture and Features** |
| **Programming on Spark: Resilient Distributed Datasets, Transformation, Spark SQL, Examples** |
| **Machine Learning (on Spark): Regression, Classification, Collaborative Filtering, and Clustering.** |
| **Streaming: Stream Processing – Motivation, Examples, Constraints, and Approaches.** |
| **Streaming on Spark: Architecture of Spark Streaming, Stream Processing Model, Example.** |
| **Cloud Computing: A brief overview: Motivation, Structure and Components; Characteristics and advantages – Elasticity, Dynamic provisioning, Multi-tenancy. Services on the cloud.** |
| **Storage as a Service: Forms of storage on the cloud, Cloud managed NoSQL databases.** |
| **Amazon’s storage services: block storage, file system, and database; EBS, SimpleDB, S3** |
| **Case study – Amazon DynamoDB (Access/Querying model, Database architecture and applications on the cloud).** |

|  |
| --- |
| **Fundamentals of Neural Network:** |
| **Fundamentals of Neural Network: Perceptron, Perceptron learning algorithm, Multilayer Perceptron (MLP), MLP on Boolean, reals and continuous values,** |
| **Fundamentals of Neural Network: MLP as classifiers, MLP as Universal approximators, Issue of Depth and Width,** |
| **Deep Feedforward Neural Network: MLP with hidden Layers, Forward Propagation, Backward Propagation, Training a DNN using Gradient Descent algorithm, Computational Graphs** |
| **Deep Feedforward Neural Network: Activation Functions, Softmax Regression, T1 – Ch4 and Ch3.4** |
| **Optimization algorithms for Deep models: Challenges – Saddle points and plateau, Non-convex optimization intuition, Stochastic Gradient Descent (SGD), Minibatch SGD, Overview of Rprop, Quickprop, Momentum, Nastrov’s Accelarated Momentum, Algorithms with Adaptive Learning Rates, Adagrad, RMSprop, ADAM, T1 – Ch11** |
| **Regularization for Deep models: Model Selection, Underfitting, and Overfitting, L1 and L2 Regularization, Dropout, Challenge - Vanishing and Exploding Gradients, Parameter Initialization, Challenge Covariance Shift, Batch Normalization, T1 – Ch4, 7.5** |
| **Convolutional Neural Network: Basics of Computer Vision and Invariance, Convolutions for Images, Learning a Kernel, Padding and stride, Channels, Pooling, Designing a CNN, T1 – Ch6** |
| **Popular CNN architectures: LeNet, AlexNet, VGG16, Network in Network (NiN), Inception Net, ResNet, DenseNet, Transfer Learning, Applications of CNN, T1 – Ch7** |
| **Sequence Models: Recurrent Neural Networks, Types of Sequences and RNNs, Back-propagation Through Time, Gates and Exploding / Vanishing gradient, T1 – Ch8** |
| **Popular RNN architectures: Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM) Networks, Bidirectional models, Sequence to sequence learning with an RNN encoder and an RNN decoder, T1 – Ch9** |
| **Attention Mechanism: Attention Pooling, Attention Scoring Functions, Multi-Head Attention, T1 – Ch10** |
| **Attention Mechanism: Self-Attention, Positional Encoding, Transformer architecture, Applications of Transformers, T1 – Ch10** |
| **Representation Learning: Review of PCA, Autoencoder, Denoising Autoencoders, Variational Autoencoders, Applications of Autoencoders, T1 – Ch14** |
| **Generative Adversarial Networks: An overview, applications of GAN, T1 – Ch19** |

**NLP**

**1. Natural Language Understanding and Generation • The Study of Language. • Applications of Natural Language Understanding. • Evaluating Language Understanding Systems. • The Different Levels of Language Analysis. • The Organization of Natural Language Understanding Systems.**

**2. N-gram Language Modelling • N-Grams • Generalization and Zeros. • Smoothing • The Web and Stupid Backoff • Evaluating Language Models • Smoothing • The Web and Stupid Backoff**

**3 Neural networks and Neural language Models • Units • The XOR problem • Feed-Forward Neural Networks • Training Neural Nets • Neural Language Models -expand spend more time**

**4. Part-of-Speech Tagging • (Mostly) English Word Classes • The Penn Treebank Part-of-Speech Tag set • Part-of-Speech Tagging • Markov Chains • The Hidden Markov Model • HMM Part-of-Speech Tagging • Part-of-Speech Tagging for Morphological Rich Languages**

**5. Hidden Markov Models and MEMM • The Hidden Markov Model • Likelihood Computation: The Forward Algorithm • Decoding: The Viterbi Algorithm • HMM Training: The Forward-Backward Algorithm • Maximum Entropy Markov Models • Bidirectionality**

**6. Topic Modelling • Mathematical foundations for LDA : Multinomial and Dirichlet distributions • Intuition behind LDA • LDA Generative model • Latent Dirichlet Allocation Algorithm and Implementation • Gibbs Sampling**

**7. Vector semantics and Embedding • Lexical semantics • Vector semantics • Word and Vectors • TFIDF • Word2Vec, Skip gram and CBOW • Glove • Visualizing Embedding’s**

**8. Grammars and Parsing. • Grammars and Sentence Structure. • What Makes a Good Grammar • A Top-Down Parser. • Bottom-Up Chart Parser. • Top-Down Chart Parsing. • Finite State Models and Morphological Processing. • Grammars and Logic Programming.**

**9. Statistical Constituency Parsing • Probabilistic Context-Free Grammars • Probabilistic CKY Parsing of PCFGs • Ways to Learn PCFG Rule Probabilities • Problems with PCFGs • Improving PCFGs by Splitting Non-Terminals • Probabilistic Lexicalized CFGs**

**10. Dependency Parsing • Dependency Relations • Dependency Formalisms • Dependency Treebanks • Transition-Based Dependency Parsing • Graph-Based Dependency Parsing • Dependency parser using neural network**

**11. Encoder-Decoder Models, Attention and Contextual Embeddings • Neural Language Models and Generation • Encoder-Decoder Networks, Attention • Applications of Encoder-Decoder Networks • Self-Attention and Transformer Networks • BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding • Contextual Word Representations: A Contextual Introduction • The Illustrated BERT, ELMo, and co. • XLM**

**12. Word sense disambiguation • Word Senses • Relations between Senses • WordNet: A Database of Lexical Relations • Word Sense Disambiguation • Alternate WSD algorithms and Tasks • Using Thesauruses to Improve Embedding’s • Word Sense Induction**

**13. Semantic web ontology and Knowledge Graph • Introduction to semantic web • Semantic web ontology • Semantic web languages • Ontology Engineering • Ontology Learning • Knowledge graph –construction of graph**

**14. Introduction to NLP Applications • Brief introduction of state of art applications • Text Summarization • Machine Translation**