Deep Learning applications and challenges in Big Data

Pouya Kousha
Fall 2018 – CSE 5194
Prof. DK Panda
Agenda of Presentation

- Observe novel applicability of DL techniques in Big Data Analytics.
- Applications of DL techniques for common Big Data Analytics problems.
  - Semantic indexing
  - Discriminative tasks
  - Semantic tagging
- Overview of DL challenges in Big Data Analytics
  - Streaming data
  - High dimensional data
  - Large scale models
  - Domain adaptation
Overview: Big Data Challenges

- Variety
- Volume
- Velocity
- Veracity
- Data quality and validation
- Feature engineering
- …
Applications of DL in Big Data Analytics

- Primary focus of DL is to extract meaningful features (high-level representation) in data.
  - Labeled data
  - Unlabeled data
- For a Big Data application, it’s more useful to extract representations and patterns from unsupervised data.
- Classically, Big data uses shallow training architectures which are able to capture local relationships.

However

- DL algorithms are performing better at extracting non-local and global relationships and pattern in the data.
- Output of DL algorithms can be used to develop more discriminative and complex models on Big Data.
Benefits of using learnt representations of DL

1. Relatively simple linear models can work effectively with the knowledge obtained from the more complex and more abstract data representations.

2. Increased automation of data representation extraction from unsupervised data enables its broad application to different data types.
   - Image
   - Text
   - video

3. Relational and semantic knowledge can be obtained at the higher levels of abstraction and representation of the raw data.
   - Makes it more global features

4. DL algorithms and architecture are more suitable for issues related to Volume and Variety of Big Data Analytics.
Goal: Using semantics to label data for using in information retrieval and storing.

Gained importance since large amount of data (text, images, video, and etc.) are being generated across various domains.

Previous solutions are facing two challenges
- Massive volume of data (volume)
- Different data representations (variety)
Using DL for semantic indexing

- Instead of using raw data for data indexing, DL can be used to generate high-level abstract data representation.
  - DL reveals complex association and factors for unsupervised data.

- High-level abstract data representation should
  - Be meaningful
  - Demonstrate rational and semantic association
  - Doesn’t depend on data

- Data representation plays an important role in the indexing of the data.
  - Let the similar representation to be stored closer to each other in memory.

- BUT, how to represent it?
Goal of document representation is to create a representation that condenses specific and unique aspects of the document, e.g. document topic.

Historic models are mostly based on word count and occurrence of words:
- Consider individual words to be dimensions, with different dimensions being independent.
- As example TF-IDF and BM25

Often observed that the occurrence of words are highly correlated.

Using Deep Learning techniques to extract meaningful data representations
- Leads to the reduction of the dimensions of the document data representations.
Using vector representation

- One option is “Vector representation” enables faster search and information retrieval.
- Each point is presented by a vector representation in a vector space.
- The data instances that have similar vector representations are likely to have similar semantic meaning.
- Allowing for a vector-based comparison which is more efficient than comparing instances based directly on raw data.
words that are similar end up clustering nearby each other.

https://www.tensorflow.org/tutorials/representation/word2vec
Word2vec

- A example of vector representation.
- Given a dictionary (corpus) as input with billions of words, generate a vector representation of words where
  - Words with similar semantic meanings are close neighbors to each other
  - Semantic indexing helps predict a target word for a given context

Examples:
- Paris belongs to France and Berlin belongs to Germany.
- Adam was studying in the library.

- Word vectors with such semantic relationships could be used to improve many existing Natural Language Processing (NLP) applications.
Binary Codes for document


- A DL model where
  - Lowest layer represents the word-count vector of a document
  - the top layer represents a learned binary code for that document.

- By using short binary codes as addresses, we can perform retrieval on very large document sets in a time that is independent of the size of the document set using only one word of memory to describe each document.

- Slow training time but fast inference time

Wrap up: semantic indexing

- Deep Learning algorithms make it possible to learn complex nonlinear representations between word occurrences.
- With DL, one can leverage unsupervised data to have access to a much larger amount of input.
- By using DL, we don’t have to worry about generating huge labeled data to capture complex representations.
- To accelerate training and improve accuracy, one can use a mixture of labeled data and unlabeled data as input for deep learning.
- Remember, similar to textual data, DL can be used for other kinds of data.
Discriminative tasks and semantic tagging

- Proposed method:
  1. Use DL algorithms to extract complicated non-linear features from the raw data
  2. Then use simpler linear models to perform discriminative tasks with output of step 1

- Advantages:
  - Extracting features with Deep Learning adds non-linearity to the data analysis, associating the discriminative tasks closely to Artificial Intelligence.
  - Applying relatively simple linear analytical models on the extracted features is more computationally efficient, which is important for Big Data Analytics.

- Massive amount of data develop some non-linear features learnt through DL, for further study data analysts can apply it on simpler linear model for further analysis.
Discriminative analysis can be the primary purpose of data analysis or can be used for tagging on the data for the purpose of searching.

- As example, searching of audio/video files by speech.
- Google’s image search

To deal with searching through large scale image data, one approach to consider is to automate the process of tagging images and extracting semantic information from the images.

- DL helps in discriminative task of semantic tagging for data.
Semantic indexing vs semantic tagging

- In **semantic indexing**, the focus is on using the Deep Learning abstract representations directly for data indexing purposes.

- In **semantic tagging**, the abstract data representations are considered as features for performing the discriminative task of data tagging.

- This tagging on data can also be used for data indexing as well, but the primary idea here is that Deep Learning makes it possible to tag massive amounts of data by applying simple linear modeling methods on complicated features that were extracted by Deep Learning algorithms.
Examples of semantic tagging

- ImageNet was an example of using DL for object detection in images. (already presented)

- DL algorithms can be trained without using labelled data.
  - Google and Stanford formulated a very large deep neural network that was able to learn very high-level features, such as face detection or cat detection from scratch (without any priors) by just using unlabeled data
  - Their work was a large scale investigation on the feasibility of building high-level features with Deep Learning using only unlabeled (unsupervised) data, and clearly demonstrated the benefits of using Deep Learning with unsupervised data.
  - based on these features their approach also outperformed the state-of-the-art and recognized 22,000 object categories from the ImageNet dataset.

- Conclusion: using features extracted from a given dataset to successfully perform a discriminative task on another dataset is possible in DL.
Example: action scene recognition

- Detecting the activity or scene in the given context.
- Deep Learning can be used for action scene recognition as well as video data tagging.
DL has shown remarkable results in extracting useful features (representation) for performing discriminative tasks on image/video data. These features (representations) are useful for data tagging and can be used in search engines. High-level complex data representations are useful for the application of simpler linear models for Big Data analytics. Future work should determine appropriate objectives in learning good representations.
So far, we focused on emphasizing the applicability and benefits of Deep Learning algorithms for Big Data Analytics.

However, there are some challenges posed by characteristics of Big Data to DL:
- learning with streaming data
- dealing with high-dimensional data
- scalability of models
- distributed computing
Incremental learning for non-stationary data

- One of the challenges in Big Data is to handling stream data and fast-moving input data.
- DL needs to handle these challenges as well.
- Some example solutions use:
  - Incremental feature learning
  - Denoising autoencoders
Denoising autoencoders are a variant of autoencoders which extract features from corrupted input, where the extracted features are robust to noisy data and good for classification purposes.

In a denoising autoencoder, there is one hidden layer which extracts features
- Initial Number of nodes = number of extracted features.

Incrementally, the samples that don’t follow objective function are reused for adding new nodes to the hidden layer based on those samples

Incoming new data samples are used to jointly retrain all the features.
**Incremental feature learning (cond.)**

- This approach is useful in applications where the distribution of data changes with respect to time in massive online data streams.
- Similar features are merged to produce a more compact set of features.
- Incremental feature learning and extraction can be generalized for other Deep Learning algorithms, such as RBM.
- It’s possible to handle new incoming stream of an online large scale data.
- It avoids expensive cross-validation analysis in selecting the number of features in large-scale datasets.
- Monotonically adding features can lead to having a lot of redundant features and overfitting of data.
High dimensional data

- Some DL algorithms can become very computationally expensive when dealing with high dimensional data like image.
- The cause of slow down is the long training time through hierarchy of DL algorithm.
- High dimensional data will make it more complicated to train DL model.
- In case of large volume of data in Big Data, these DL algorithms can be bottlenecks.

- An example of existing solution:
  - Convolutional neural networks
Convolutional neural networks

- Used in ImageNet to achieve better results.
- The neurons in the hidden layers units do not need to be fully connected to previous layer.
- Instead they can be connected to the neurons that are in the same spatial area.
- The resolution of the image data is also reduced when moving toward higher layers in the network.
- Scales up efficiently on high dimensional data.
Main question: how to scale up the achievements to a larger model and larger dataset?

- Support for model parallelism
  - Within a node (multi-threading)
  - Across machines (MPI)

- Support for data parallelism

- Using high speed communication infrastructure

- Handle communication, synchronization, and details of parallelism within framework.
  - Increases productivity as well as performance!

- Example of such work
  - DistBelief
  - COTS HPC

https://xiandong79.github.io/Intro-Distributed-Deep-Learning
Challenges in large-scale models

- Determining the optimal number of model parameters in such large-scale models and improving their computational practicality pose challenges in DL for Big Data Analytics.
- Handling massive volumes of data
- Large-scale DL models for Big Data Analytics have to face with other Big Data problems
  - Domain adaptation
  - Streaming data
- Above challenges motivates the need for better large-scale DL algorithms and architecture.
Domain adaptation

- Problem: the distribution of the training data (from which the representations are learnt) is different from the distribution of the test data (on which the learnt representations are deployed).
- A known problem in Big data due to variation of input data types and domains.
  - Volume + Variety
- What representations should be shared among different domains?
- Poses a challenge in utilization of the entire Big Data input dataset vs portion of it.
  - What volume of data in enough for training to have ample representations?
Future works should focus on addressing one or more of these problems often seen in Big Data, thus contributing to the Deep Learning and Big Data Analytics research corpus.

We need to adapt DL for challenges associated with Big Data:
- high dimensionality
- streaming data analysis
- scalability of Deep Learning models
- improved formulation of data abstractions
- distributed computing
- criteria for extracting good data representations
- domain adaptation
- semantic indexing
Each of these areas generate new challenges and propose solutions for each other.