Machine Learning in the context of Big Data

INE410131 - Gerência de Dados para Big Data

Anthony Galtier

Outline

1. Introduction
2. The challenges of Machine Learning with Big Data
3. Manipulations for Big Data
4. Machine Learning Paradigms for Big Data
5. The case of Deep Learning
6. Conclusions

Outline

1. **Introduction**
2. The challenges of Machine Learning with Big Data
3. Manipulations for Big Data
4. Machine Learning Paradigms for Big Data
5. The case of Deep Learning
6. Conclusions

1. The context of Big Data

A broad term

“High volume, high velocity, high variety”

With promises

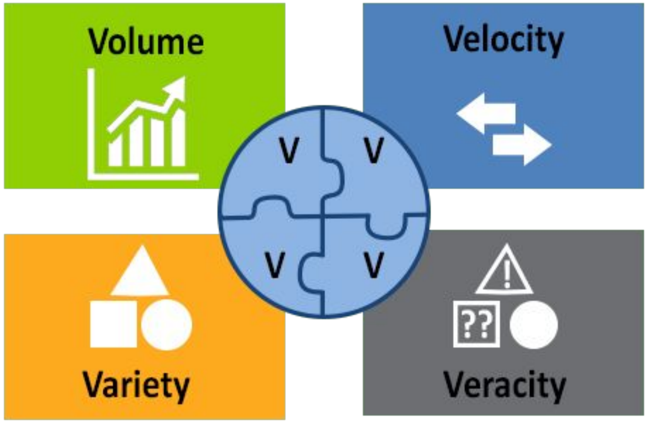
- insight discovery
- improved decision making
- process optimization

...

And challenges

- Storage
- Processing
- Analysis

...



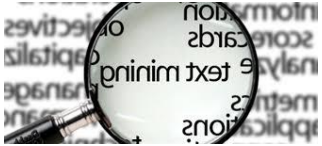
Big Data Vs

1. Big Data Analytics

Data Visualisation

Statistical Analysis

" The ability to extract value from Big Data depends on data analytics"



Text analysis

Machine Learning

Business Intelligence

1. Machine Learning

Automating analytical model building with AI

Algorithms that learn from data, identify patterns, make decisions with minimal human intervention

Supervised learning	Unsupervised learning
Both inputs and outputs are known	Only inputs are known
Finding the best mapping function between input and outputs	Finding the model that will best fit the underlying structure of the data
Regression, Classification	Clustering, Association
Linear regression, KNN, Random Forests, SVM	K-Means, Mean-Shift, DBSCAN, HAC

1. Machine Learning Assumptions

The more data the better the learning? Not necessarily...

Machine Learning: 1950s Internet: late 1980s Big Data: 1990s

=> Machine Learning was not developed in the context of Big Data

Assumptions:

- Data sets fit entirely into memory
- Data are uniformly distributed across all classes
- Statistical properties are similar across a complete dataset

....

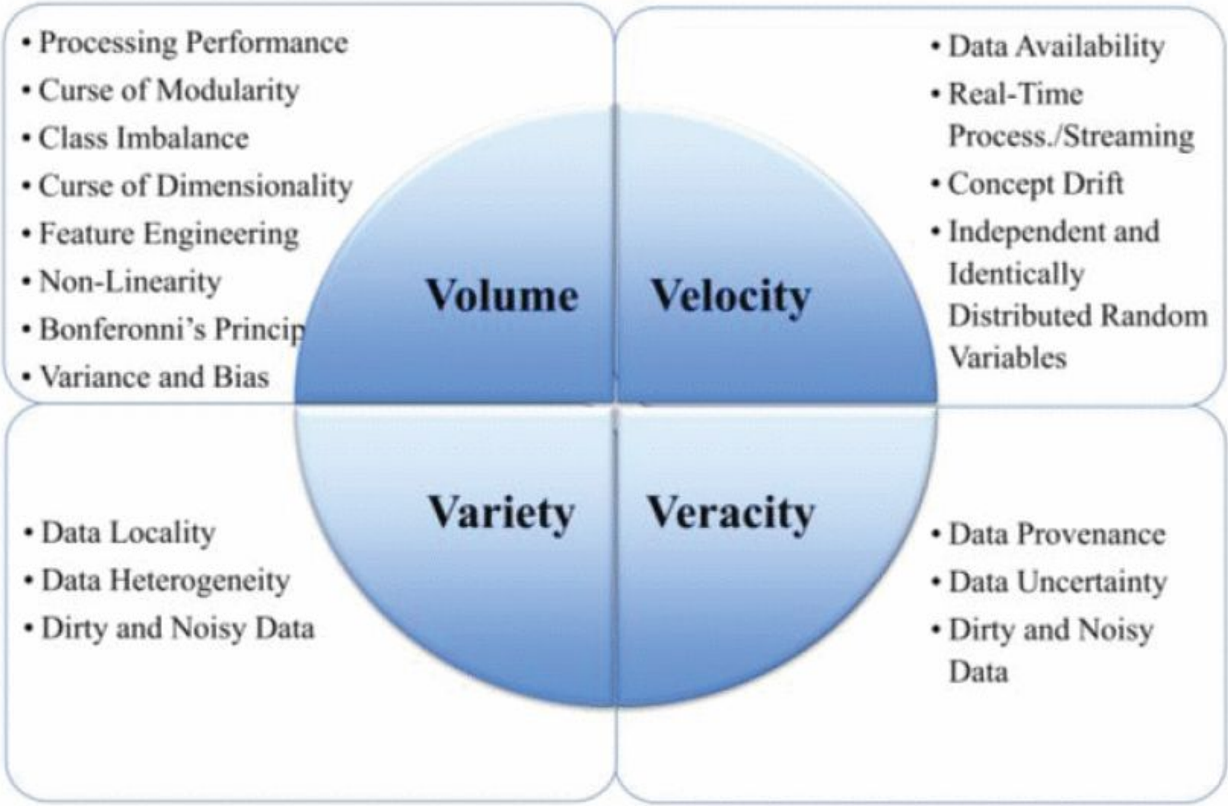
Big Data breaks these assumptions

- New challenges
- New approaches

Outline

1. Introduction
2. **The challenges of Machine Learning with Big Data**
3. Manipulations for Big Data
4. Machine Learning Paradigms for Big Data
5. The case of Deep Learning
6. Conclusions

2. The challenges



2.1. Volume

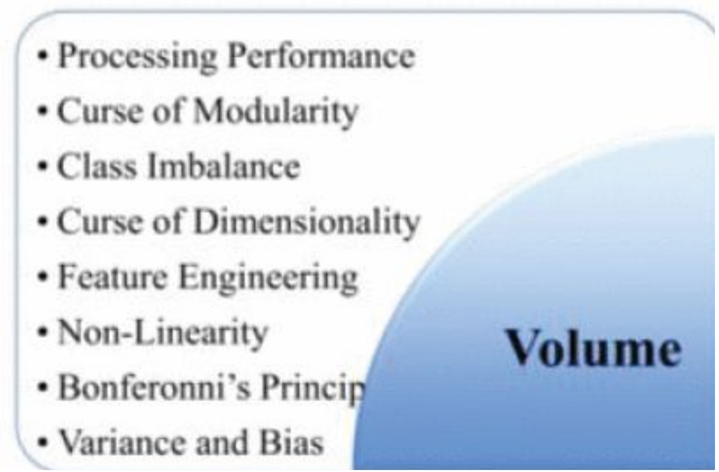
Characteristics

Refers the amount, size, scale of the data

Volume is relative to the type of data:
- Large amount of simple data
- Smaller amount of complex data
- or both

Vertical size vs. Horizontal size

Challenges



2.1.1. Volume: Processing performance

Scale and volume adds computational complexity

ML algorithms generally have high time complexity $\geq O(n^3)$ and space complexity $\geq O(n^2)$

=> Trivial operations can become very costly or infeasible

The performance of ML algorithms becomes increasingly dependent on how the data is stored and moved. Performance increasingly requires:

- Parallelisation
- Partitioning
- Resusing

Which may not always be possible

2.1.2. Volume: The curse of modularity

ML algorithms generally require the following assumption:

"The data being processed fits entirely in memory or in a single file on a disk"

... which is no longer true in the context of Big Data

=> entire families of ML algorithm fail

MapReduce is brought forward as a solution:

efficiently solves this curse for inherently parallel algorithms   K-Mean, Mean-Shift

...but not all ML algorithms are so   Gradient Descent, Expectation Maximisation

2.1.3. Volume: Class Imbalance

The assumption that:

"Data are uniformly distributed across all classes" ...is often broken in the context of Big Data

=> negatively affects many ML algorithms Decision trees, Neural networks, Support Vector Machines

Class imbalance is an active research topic

Japkowicz and Stephen:

Class imbalance problems depend on:

- task complexity
- degree of class imbalance
- size of the training set

2.1.4. Volume: The curse of dimensionality

Refers to the difficulty of working in high dimensional spaces

The Hughes effect

For a fixed-sized training set,

Increasing dimensionality => Decreasing prediction performance

High dimensionality also affects the processing performance of ML algorithms

2.1.5 Volume: Feature Engineering

Originates from the curse of dimensionality

Refers to the process of creating new features to improve the performance of ML algorithms

High-time complexity algorithms: >= $O(n^3)$

A very time-consuming pre-processing step ... and even more so in the context of Big Data

Feature selection

Also becomes a complex task with Big Data

- spurious correlations between features
- Incidental endogeneity

2.1.7. Volume: Variance and Bias

Machine Learning relies on the idea of generalisation, which implies error

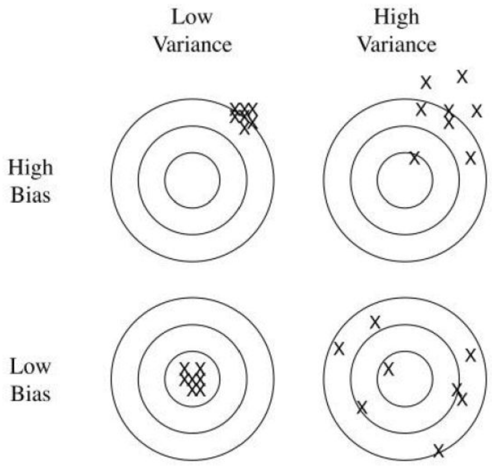
Generalisation error can broken down into two types of error:

Variance and **Bias**

Ideally both types of errors should be minimised.

However, scaling up the volume of data, ML models tend to get too biased on the training data   "Overfitting"

Regularisation techniques for Big Data is still a very open field of research



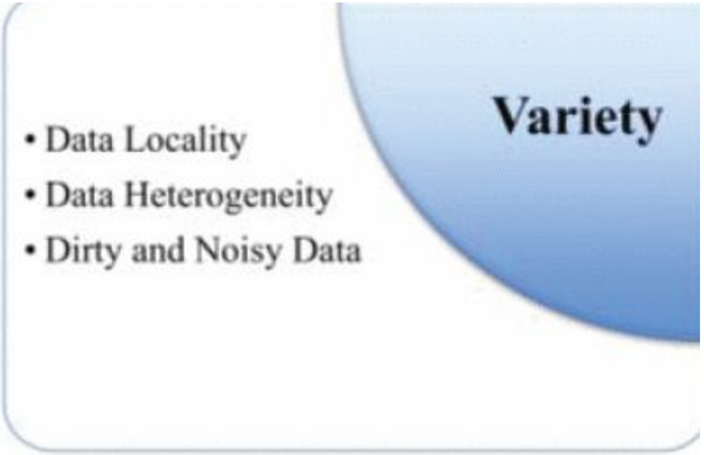
2.2. Variety

Characteristics

Differences in data types, in what the data actually represents and where it comes from

- Structural variety
- Semantic variety
- Variety of sources

Challenges



2.2.1. Variety: Data locality

ML assumes that the entire assume that the entire dataset is found in memory or in a single disk file.

...not the case with Big Data

Data is distributed over many files, in many different physical locations

Bringing data to computation vs. Bringing computation to data

Distributed and parallel computing comes as a solution MapReduce, Hadoop, Spark

...but does not fit all ML models

2.2.2. Variety: Data heterogeneity

Syntactic heterogeneity	Semantic heterogeneity
Refers to the difference in data types, formats, encoding	Refers to the differences in meaning and interpretation
ML algorithms do not recognise theses differences	ML algorithms were not developed to deal with semantically diverse data
=> Pre-processing of the data becomes even more challenging	=> Semantic heterogeneity must be resolved beforehand

2.2.3. Variety: Dirty and noisy data

Data can be characterised according to the following features:

- **Condition:** the readiness of the data for analysis.
- **Location:** where the data physically reside.
- **Population:** the entities and their sets of common attributes

Big data is dirty

- ill-conditioned
- many different locations
- unknown populations

Big data is noisy

- measurement errors
- outliers
- missing values

2.3. Velocity

Characteristics

Data is generated rapidly and often, to realise it's value, needs be analysed just as fast.

Velocity thus both refers to the

- Speed at which data is generated
- Rate at which data must be analysed

Challenges

2.3.1. Velocity: Data availability

ML often assumes that the entire data set be present before learning ...not the case with Big Data

Most models need to be trained again every time new data arrives

In the context of Big Data, new data is constantly generated

=> becomes a very costly and time-consuming operation

ML models must adapt their learning for newly arriving data

<u>Incremental learning</u>

An active research topic

Difficult to adapt ML algorithms

2.3.2. Velocity: Real time processing / Streaming

Data availability challenge + **speed**

Adapting ML to handle
constant streams of data

Performing analyses in
real-time or near-real time

Great business value in real-time processing

fraud detection, trading, surveillance systems

Emergence of streaming systems

...not yet merged with machine learning algorithms

...a very complex task

2.3.3. Velocity: Concept drift

The statistical properties of the target variable, which the model is trying to predict, may change in time.

Example: Energy consumption and demand

Concept drifts can be

- Incremental
- Gradual
- Sudden
- recurring

ML models trained with old data become obsolete

=> Concept drifts need be detected quickly

Concept drift is not a new research topic however, Big Data have increased the frequency of its occurrence

2.3.4. Velocity: i.i.d. Random Variables

Independent and identically distributed (i.i.d.) random variables are a common assumption in ML

- simplifies the underlying methods
- improves convergence

...in reality this is not always true

i.i.d. requires the order of the data to be randomised in the data set

=> can be difficult to achieve with Big Data

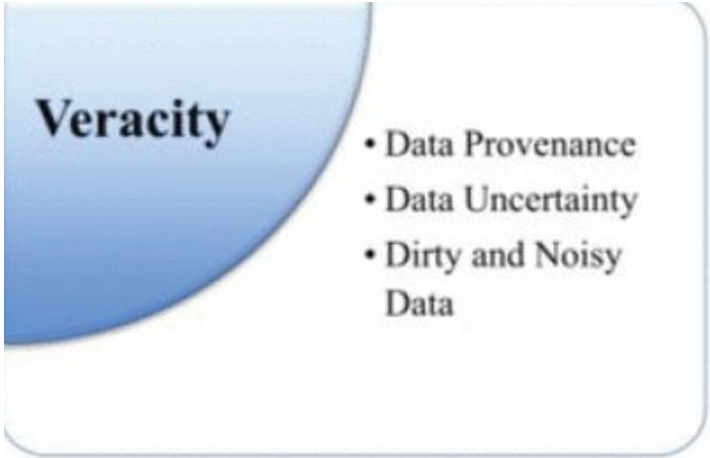
2.4. Veracity

Characteristics

Veracity relates to two aspects:

- Data quality
- Reliability of the data sources

Challenges



2.4.1. Veracity: Data provenance

Refers to the process of tracing and recording the origin and movements of data

Provides a way to establish the veracity of data

Constitutes important contextual information for ML models

=> helps identify the source of processing errors

In the context of Big Data, the size of this metadata can become too large

=> big overhead cost

Solutions such as RAMP exist for certain models ...but not for others

2.4.2. Veracity: Data uncertainty

In the context of Big Data, the means and methods used to collect data introduce uncertainty

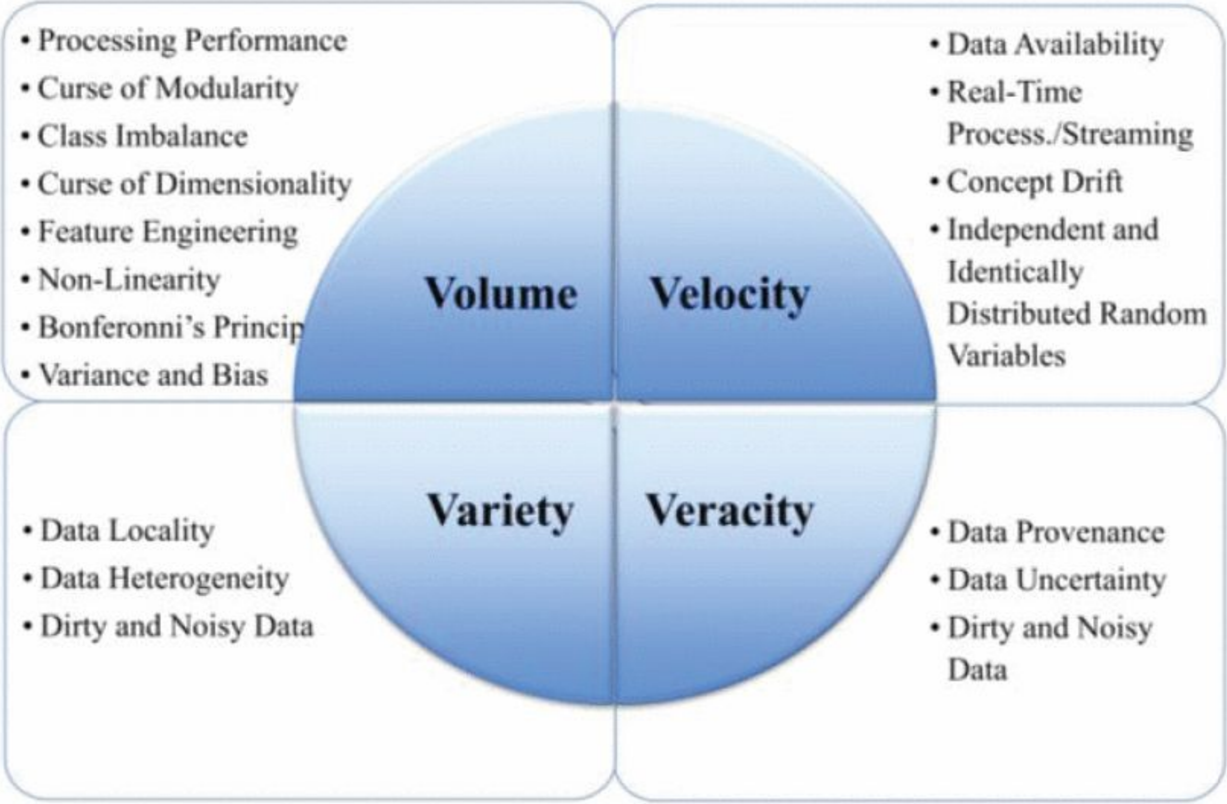
=> impacts the veracity of the dataset

These new forms and means are for example:

- Sentiment data
- Crowdsourced data
- Inherently uncertain data

ML was not designed to handle such imprecise data

2.5. Overview of the challenges



Outline

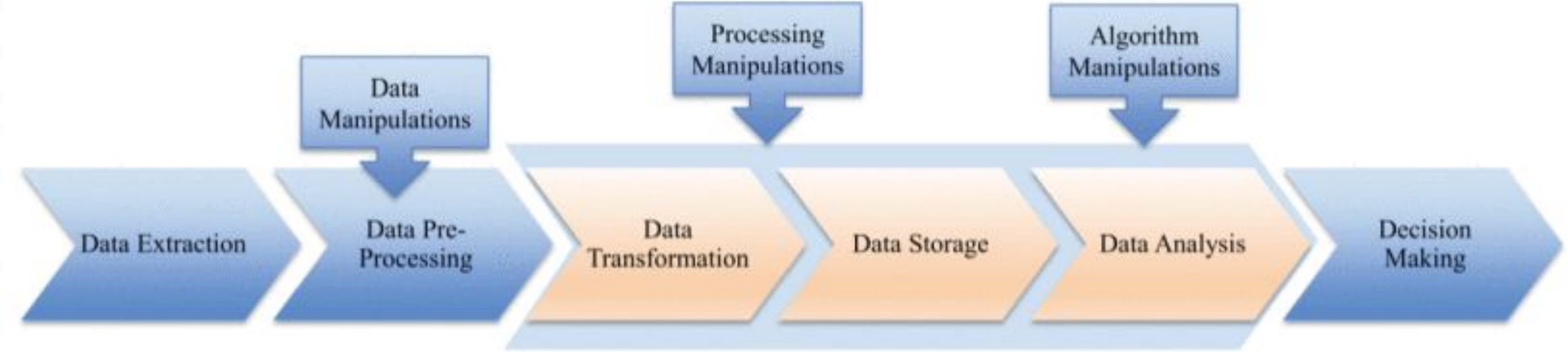
1. Introduction
2. The challenges of Machine Learning with Big Data
3. **Manipulations for Big Data**
4. Machine Learning Paradigms for Big Data
5. The case of Deep Learning
6. Conclusions

3. Manipulations for Big Data

Two approaches:

Developing entirely new algorithms vs **Adapting existing algorithms**

Manipulations to adapt for Big Data:

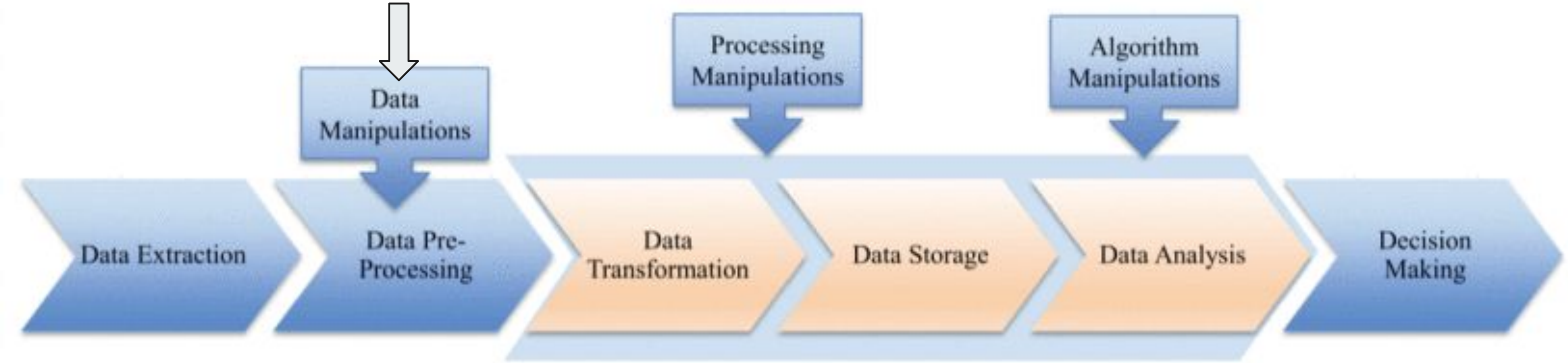


3. Manipulations for Big Data

Two approaches:

Developing entirely new algorithms vs **Adapting existing algorithms**

Manipulations to adapt for Big Data:



3.1. Data manipulations: Dimensionality reduction

Deals with the curse of dimensionality

Mapping high dimensional spaces onto a lower dimensionality one

- Linear mapping techniques
  - PCA
- Non-linear mapping techniques
  - Kernel PCA, Laplacian Eigenmaps, Isomap, LLE

Other techniques,

- Random projections
- Auto-Encoders

Dimensionality reduction improves performance and processing time of ML algorithms

3.2. Data manipulations: Instance selection

Selecting the most representative subset of the data to reduce the “height” of the dataset

Many diverse approaches:

- random selection
- genetic algorithm-based selection
- progressive sampling
- using domain knowledge
- cluster sampling

=>

- Reduces dataset size
- improves processing performance
- eases curse of modularity

However,

How big should the sample be?

What sampling approach to use?

How good will the model be?

Big data challenges remain...

3.3. Data manipulations: Data cleaning

Pre-processing step to remove noise and outliers

Techniques such as,

- smoothing filters
- wavelet transforms

...not new to Big Data

...not suitable for real-time processing of data

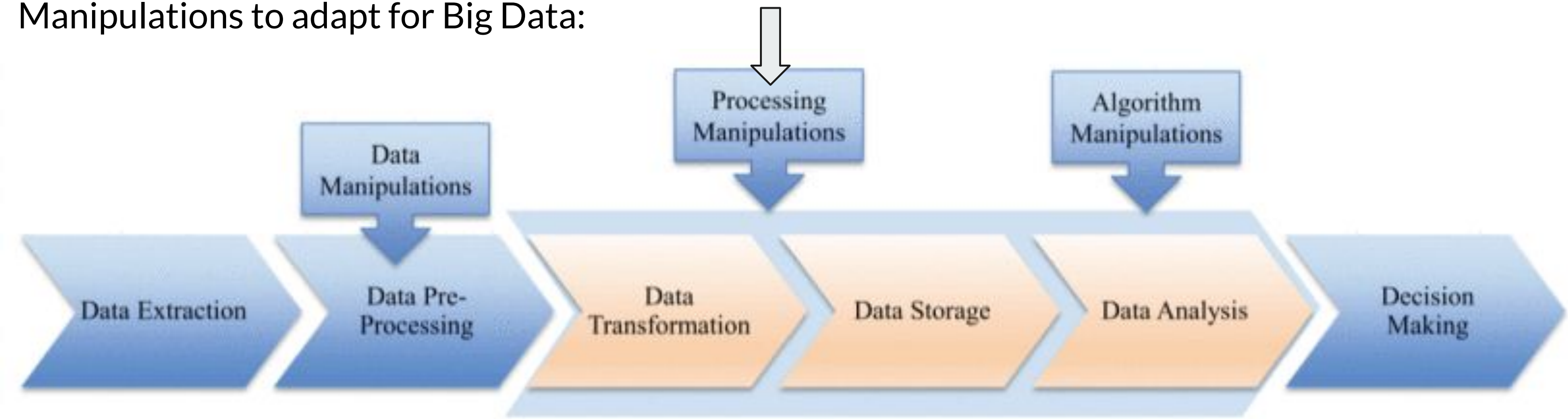
Auto-Encoders also comes as a solution

3. Manipulations for Big Data

Two approaches:

Developing entirely new algorithms vs **Adapting existing algorithms**

Manipulations to adapt for Big Data:



3.4. Processing manipulations: Vertical Scaling

Increasing the capacity of existing hardware or software by adding resources

Vertical scaling ⇔ Scaling up

- Multi-core CPUs
- Supercomputers
- GPUs
- FPGAs

...usually discarded in the context of Big Data

However, can be useful for ML

GPUs can be suitable for parallelizable ML algorithms

FPGAs are very performant for scanning large amounts of network data

3.5. Processing manipulations: Horizontal Scaling

Batch-oriented systems

Processing large amounts of data at once

More concerned with throughput rather than latency

Batch-oriented systems are based on Google's MapReduce paradigm — Hadoop, NIMBLE

Extensions developed to deal with iterative algorithms — Haloop, Twister

Batch-oriented systems effectively tackle:

- The curse of modularity
- Data locality issues

only partially tackles the curse of dimensionality

=> graph based solutions

3.5. Processing manipulations: Horizontal Scaling

Stream-oriented systems

Operate on one element or small data set in real-time or near real-time

Operations performed are less complex

Graph based topology

Micro-batches

Streaming systems mitigate processing time but are only suitable for very simple ML algorithms

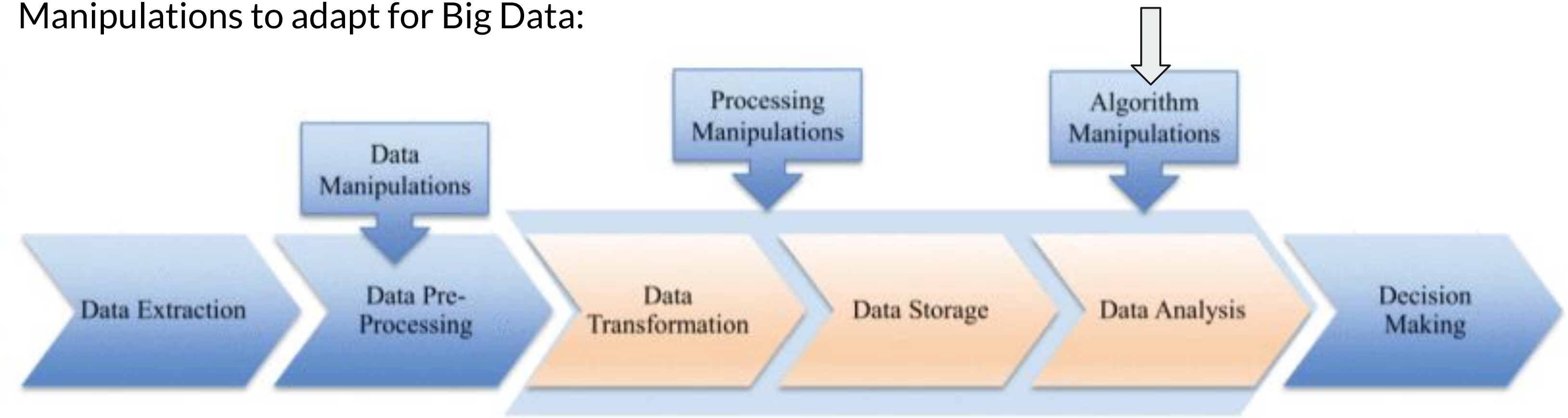
...still a lot of open research

3. Manipulations for Big Data

Two approaches:

Developing entirely new algorithms vs **Adapting existing algorithms**

Manipulations to adapt for Big Data:



3.6. Algorithm manipulations

Algorithm modifications

Modifying algorithms to improve their performance

- **Pegasos** optimised SVM algorithm for large-scale text processing
- **Regularization paths** optimises linear models for large and sparse datasets

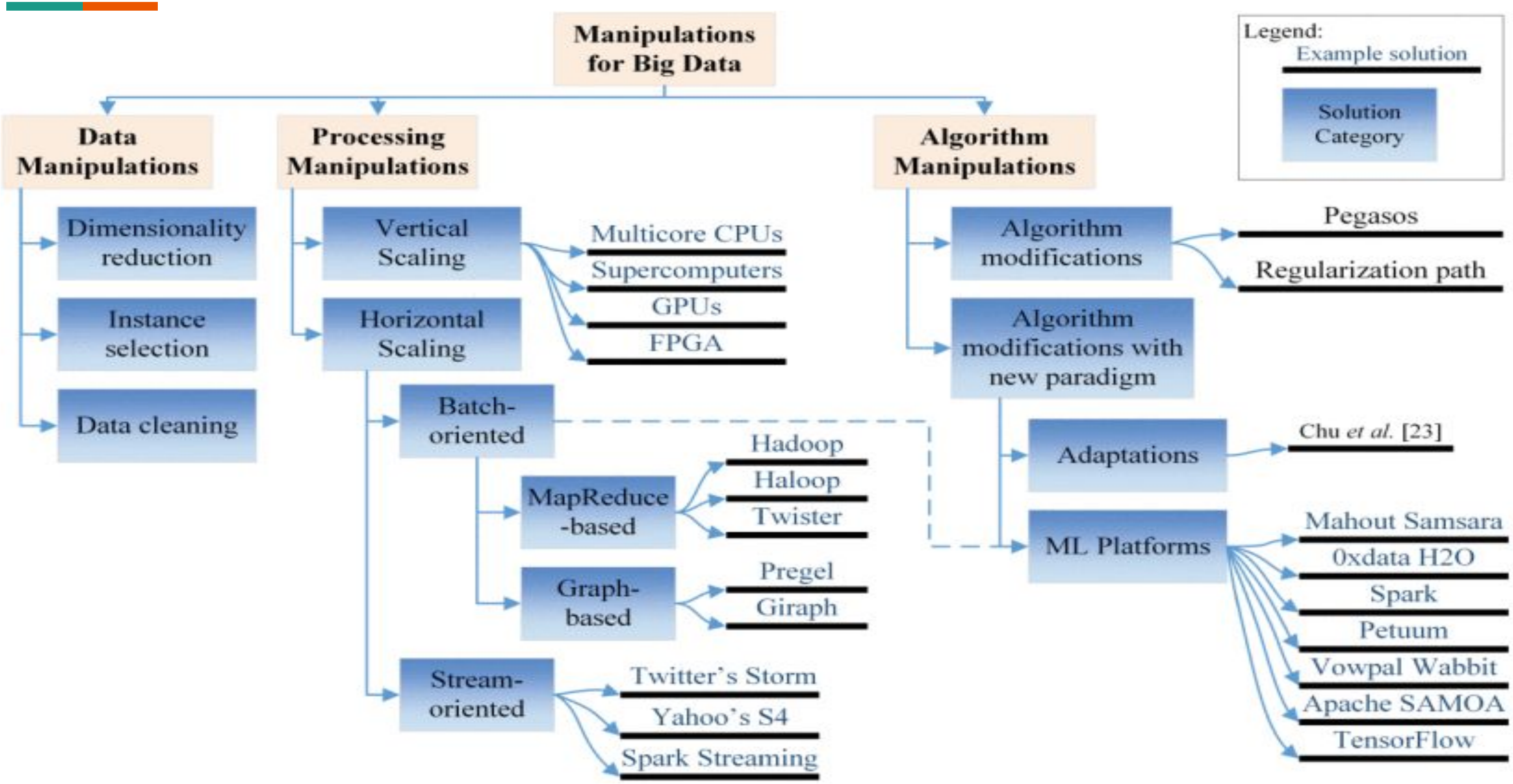
Algorithm modifications with new paradigms

Parallelise algorithms to use MapReduce Naive Bayes, GDA, K-Means, NN, SVM

ML platforms combine algorithm adaption with new computing paradigms

mahout Spark MLlib H$_2$O.ai Apache SAMOA Scalable Advanced Massive Online Analysis TensorFlow

3.7. Overview of manipulations for Big Data



Outline

1. Introduction
2. The challenges of Machine Learning with Big Data
3. Manipulations for Big Data
4. **Machine Learning Paradigms for Big Data**
5. The case of Deep Learning
6. Conclusions

4. Machine Learning paradigms for Big Data

1. Online Learning
2. Local Learning
3. Transfer Learning
4. Lifelong Learning
5. Ensemble Learning

4.1. Online learning

An alternative to batch learning

Uses data streams for learning

“Learn as you go”

=> useful when it is computationally infeasible to train over the entire dataset

Pros

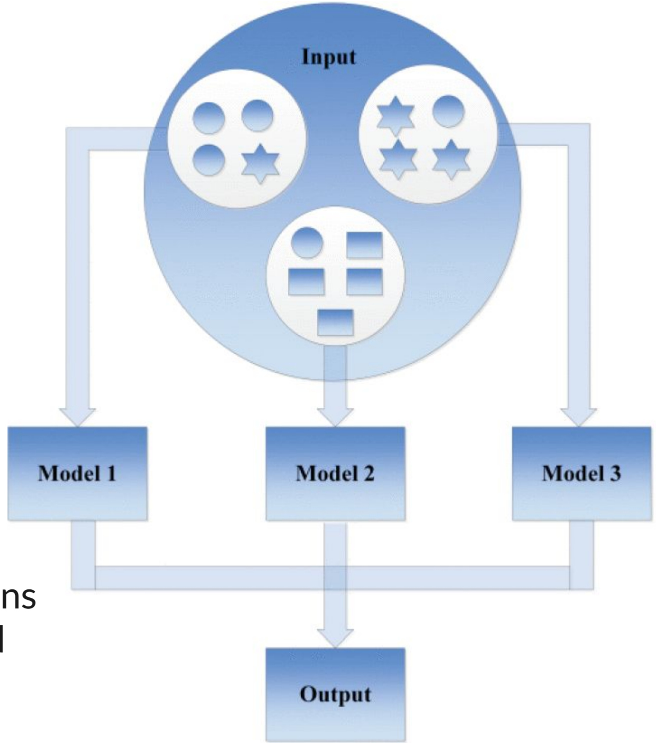
- Enables the processing of large volumes
- Facilitates real-time processing
- Remedies the curse of modularity
- Able to learn from non-i.i.d. data

Cons

- Curse of dimensionality remains
- Feature engineering is difficult
- Variety issues are unresolved

4.2. Local learning

1. Separate the input space into clusters
2. Build a separate model for each cluster
   => reduces overall cost and complexity



Pros

Alleviates:

- Curse of modularity
- Cass imbalance
- Variance and bias
- Data locality

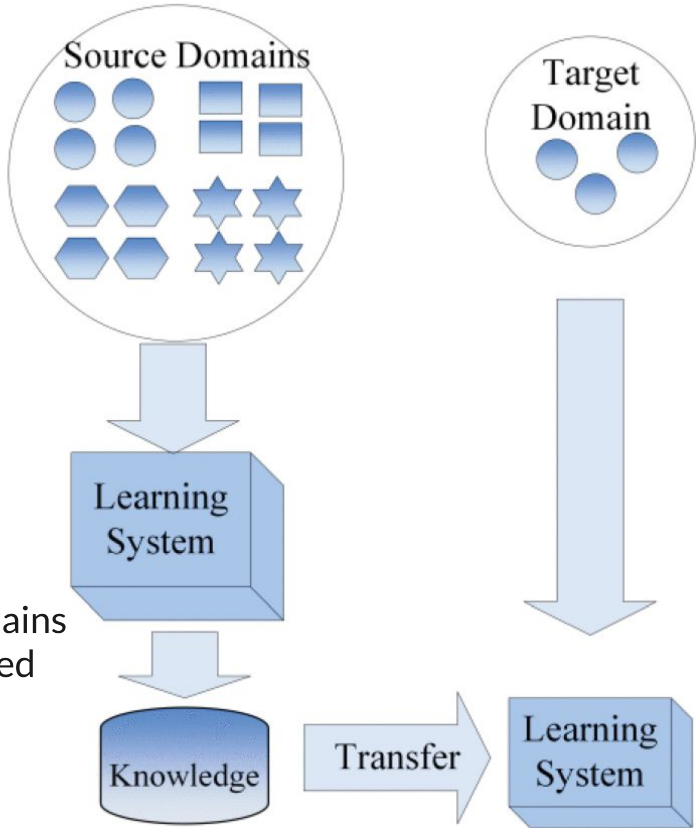
Cons

- Curse of dimensionality remains
- Velocity issues are unresolved

4.3. Transfer learning

Seeks to improve learning on a target domain by training the model with datasets from other domains

The training set domain is not necessarily the same as the test set domain



<u>Pros</u>

Alleviates:

- Curse of modularity
- Data heterogeneity
- Dirty and noisy data

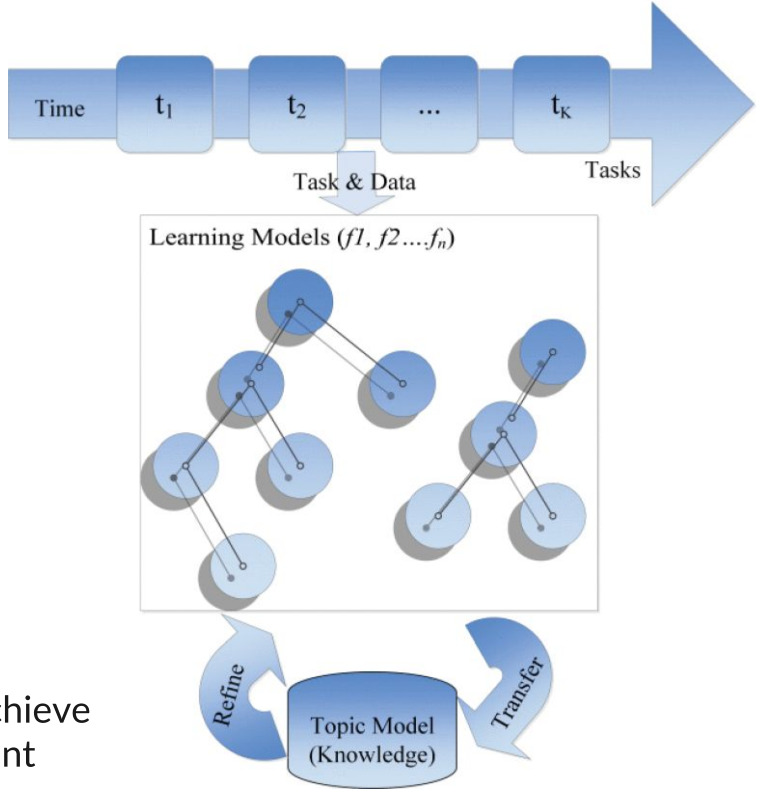
<u>Cons</u>

- Curse of dimensionality remains
- Velocity issues are unresolved

4.4. Lifelong learning

Mimics human learning
Relies on a “Knowledge Model” collecting and combining learning outputs from various learning models

Related online learning
=> continuous form of learning
Related to transfer learning
=>includes a multitude of domains



Pros

- Real-time processing
- Processing time
- Data availability
- Concept drift
- Class imbalance
- Data variety

Cons

A very difficult vision to achieve
=> Still in early development

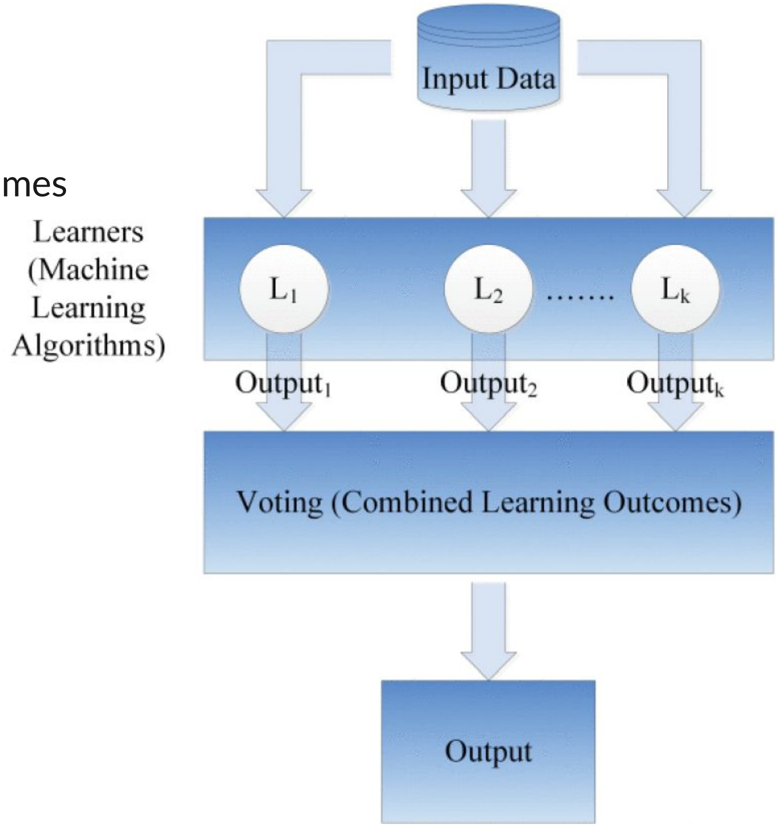
4.5. Ensemble learning

Combines multiple learners to obtain better learning outcomes

Overall output is determined by a voting system
=> improves overall accuracy

The data set can be split to train the different learners
=>can deal with voluminous datasets

Useful to identify the best performing ML algorithms



<u>Pros</u>

- Better accuracy
- Deals better with concept drift
- Curse of modularity

<u>Cons</u>

- Variety issues
- Velocity issues

Machine Learning as a service

Beyond this manipulations and new paradigms there exists proprietary services attached to large scale cloud services to perform machine learning:

Outline

1. Introduction
2. The challenges of Machine Learning with Big Data
3. Manipulations for Big Data
4. Machine Learning Paradigms for Big Data
5. **The case of Deep Learning**
6. Conclusions

5.1. Deep learning

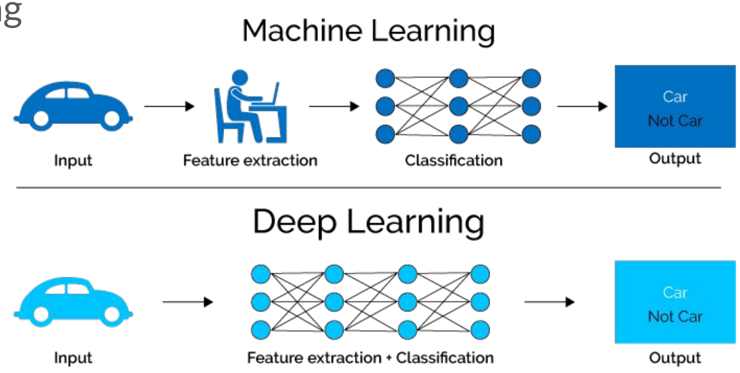
One of the most currently remarkable machine learning techniques
    image analysis   speech recognition   text understanding

Vaguely inspired by biological nervous systems

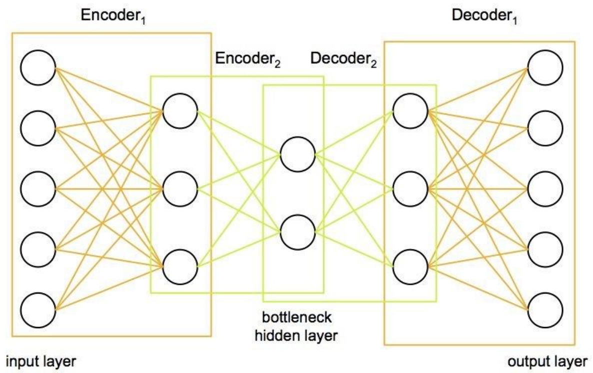
Automatically learns data representations and features
    classification   pattern recognition

Relies on a hierarchical architecture “layers”
    ->progressive abstraction of the data

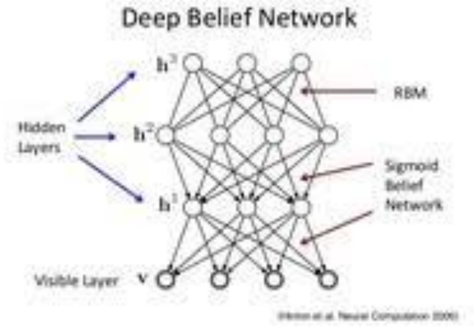
Suitable for supervised, unsupervised or semi-supervised problems

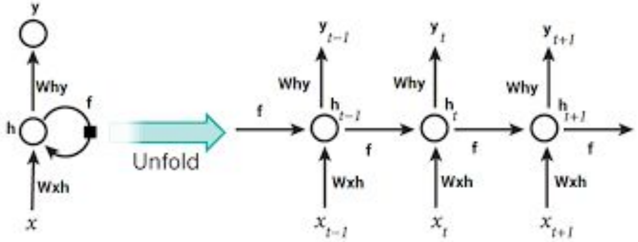


5.2. Typical Deep learning models

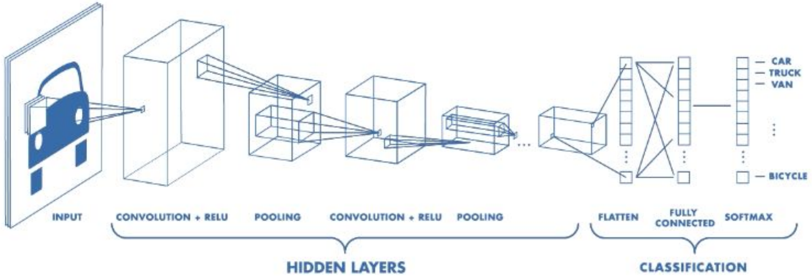


Stacked autoencoders





Recurrent Neural Networks

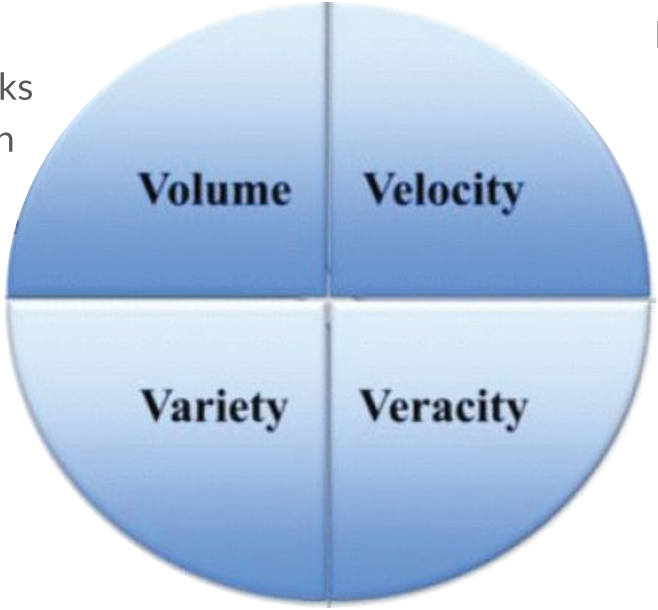


Convolutional Neural Networks

5.3. Deep learning models for Big Data feature learning

Deep stacking networks
Large-scale deep learning frameworks
Model compression and optimisation

Incremental deep learning models



Multi-model deep learning
Multi-source deep learning

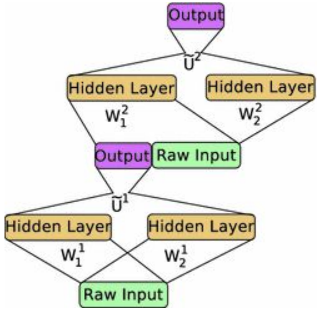
Denoising auto-encoder model
Non-local auto-encoder model

5.4. Volume: Deep learning with huge amounts of data

Large scale deep learning models
few hidden layers + large number of neurons -> millions of parameters
Three approaches:

<u>Parallel deep learning models</u>

Deep stacking networks



Software frameworks:
Distbelief

<u>GPU-based implementations</u>

Great computing power
Big memory bandwidth
=> suitable for parallel computing

Custom high performance computers

FPGA approaches

<u>Optimised deep learning models</u>

Model compressions

Low rank factorisation

Hash Trick compression

5.5. Variety: Deep learning with heterogeneous data

Objects in Big Data sets are often multi model multimedia clips webpages

Multi-model deep learning models have been proposed for specific tasks

- Audio-video object feature learning uses separate RBMs for audio and video
- Text-image recognition two deep BMs learn features from text and image respectively
- Human pose estimation multi-source deep learning model
- Chinese dialogue recognition etc...

=> Always follows the same model: **learn specific features** and then **combine**

5.5. Velocity: Deep learning with real-time data

Deep learning models have a huge amount of parameters

=> training is a very long task

How to adapt deep learning to incremental learning methods?

- Incremental back propagation
- Online deep learning
- Structure based incremental autoencoders

5.6. Veracity: Deep learning with low-quality data

Most deep learning models are designed for high quality data

Some models have recently been proposed:

Denoising autoencoder	capable of learning features from imprecise data
Imputation autoencoder	capable of learning features from incomplete data
Deep imputation network	stacking imputation autoencoders
Non-local autoencoder	only learns reliable features

...remains an open field of research

Outline

1. Introduction
2. The challenges of Machine Learning with Big Data
3. Manipulations for Big Data
4. Machine Learning Paradigms for Big Data
5. The case of Deep Learning
6. **Conclusions**

Conclusions

✓ High degree of resolution

* Partial resolution

APPROACHES			CHALLENGES																	
			VOLUME								VARIETY			VELOCITY				VERACITY		
			Processing Performance	Curse of Modularity	Class Imbalance	Curse of Dimensionality	Feature Engineering	Non-linearity	Bonferonni''s Principle	Variance and Bias	Data locality	Data Heterogeneity	Dirty and noisy Data	Data availability	Real-time Processing/Streaming	Concept drift	I.i.d	Data Provevance	Data Uncertainty	Dirty and Noisy Data
MANIPULATIONS	Data Manipulations	Dimensionality Reduction	✓			✓														
MANIPULATIONS	Data Manipulations	Instance Selection	✓	✓																
MANIPULATIONS	Data Manipulations	Data Cleaning											✓							✓
MANIPULATIONS	Processing Manipulations	Vertical Scaling	✓															*		
MANIPULATIONS	Processing Manipulations	Hori-zontal Scaling: Batch-oriented	✓	✓		*					✓							*		
MANIPULATIONS	Processing Manipulations	Hori-zontal Scaling: Stream-oriented	✓	✓										✓	✓			*		
MANIPULATIONS	Algorithm Manipulations	Algorithm Modifications	✓	*		*					✓				✓					
MANIPULATIONS	Algorithm Manipulations	Algorithm Mod. with new Paradigm	✓	*		*					✓				✓					
LEARNING PARADIGMS		Deep Learning					✓	✓				✓	*						*	*
LEARNING PARADIGMS		Online Learning	✓	✓	*						✓		*	✓	✓	*	✓			*
LEARNING PARADIGMS		Local Learning	✓	✓	✓					✓	✓									
LEARNING PARADIGMS		Transfer Learning			✓							✓	*						*	*
LEARNING PARADIGMS		Lifelong Learning	✓		✓							✓	*	✓	✓	*			*	*
LEARNING PARADIGMS		Ensemble Learning	✓	✓												✓				

References

Alexandra L'heureux, Katarina Grolinger, Hany F. Elyamany and Miriam A. M. Capretz. "Machine Learning With Big Data: Challenges and Approaches", IEEE ACCESS, Vol. 5, June 7, (2017).

Qingchen Zhang, Laurence T .Yangab, Zhikui Chen, Peng Li, "A survey on deep learning for big data", ELSEVIER, Vol, 42, July 2018

O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, K. Taha, "Efficient machine learning for big data: A review", Big Data Res., vol. 2, no. 3, pp. 87-93, Sep. 2015.

J. Dean, S. Ghemawat, "MapReduce: Simplified data processing on large clusters", Proc. 6th Symp. Oper. Syst. Design Implement., pp. 137-149, 2004.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal, "Optimization Methods for Large-Scale Machine" Learning, *SIAM Rev.*, 60(2), 223–311

X. Chen, X. Lin, Big data deep learning: challenges and perspectives, IEEE Access 2 (2014) 514–525.

V. Sze, Y. Chen, T. Yang, J. Emer, Efficient processing of deep neural networks: a tutorial and survey, 2017, arXiv:1703.09039.

Questions?