HPC and Data Analytics

Riccardo Zanella - r.zanella@cineca.it

SuperComputing Applications and Innovation Department



Table of Contents

Cineca TRAINING High Performance Computing 2017

Machine Learning Background

Efficient machine learning with Python?

Intel Data Analytics Acceleration Library (DAAL)

NVIDIA-based solutions



Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.



Cineca TRAINING High Performance Computing 2017

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

So we need to identify:

the class of tasks T

Cineca TRAINING High Performance Computing 2017

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

So we need to identify:

- the class of tasks T
- the measure of performance P



Cineca TRAINING High Performance Computing 2017

Definition of Learning Algorithm [Mitchell 1997]¹

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

So we need to identify:

- the class of tasks T
- the measure of performance P
- the source of experience E







Cineca TRAINING High Performance Computing 2017



task class T: playing checkers



Cineca TRAINING High Performance Computing 2017



144

- task class T: playing checkers
- performance measure P: fraction of games won against opponents



- task class T: playing checkers
- performance measure P: fraction of games won against opponents
- training experience E: playing practice games against itself

Example: handwritten characters recognition

82944649709275159133 2359176282507497832 1/836105100112730465 26472899307102835465

ienere Anna Arge. Tota Anna Arbert		
		Now press neque de part al 3 apresance
	Short Kenah Kulu di Pajuk	was over d'adat d'adim lines l'ode
Physical addresses approach the scale		1256 78 90 Jam Broke K SO
Capa, Seaan, Soning any Arang Ari A Mahdalay (1971) (In a sing panelik Ari)), Sea Ana Anisa Ari Anisa Anisa Anisa Arian Anisa Anisa Anisa Anisa Anisa Anisa Anisa Anisa Anisa Anisa	gegegen Geogram (ed.) 164 - San Brender, Strand 164 - San Brender, San Brender, 165 - San Brender, San Brender, 1	Event denne for som for site over yn hendere formy van providen for enerthe formy van providen for hendere for som for som terminet
		Bien Juriserend



Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017



99111

 task class T: recognizing and classifying handwritten characters within images

Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017

1183610	9709295159183 2822507497832 9300112730465 9307102035465
secon dana Ina Ange Una Ange ang Angeran dang Angeran ang Angeran Ange	Van 1993 Participa partica (S. Province Man Low Kadat Katu Vice Colo 123157999 (Jon Marchine K S.
 pp and the strength Theory, Theorem 2014 Theory and the strength of the probability of the first strength of the probability of the strength of the strenghold of the strength of the strenghold of the strength of the	Eland Hanna (2) Long (10) (2) only you have a second for the seco

99111

- task class T: recognizing and classifying handwritten characters within images
- performance measure P: fraction of characters correctly classified

5/32

Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017

1359176282	9295159123 2507497832 00112730465 07102035465
Mar Sangar Den Bangar Den Bangar Mar Bangar	S President faith of the formation of the second se

99111

- task class T: recognizing and classifying handwritten characters within images
- performance measure P: fraction of characters correctly classified
- **training experience E:** a database of handwritten characters with given classifications

▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3...\}$ each example is a (input,target) pair: $Z_i = (X_i, Y_i)$





Cineca

High Performance

- ► **training experience E:** a number of training examples $E = \{z_1, z_2, z_3...\}$ each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ task class T: a decision function f able to predict unknown Y from known X



Cineca

High Performance

- ▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3...\}$ each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ task class T: a decision function f able to predict unknown Y from known X
- **performance measure P:** a loss function *L* to measure the (non-symmetric) distance $L(f, Z_i) = d(f(X_i), Y_i)$

Cineca

High Performance

- ▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3...\}$ each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ task class T: a decision function f able to predict unknown Y from known X
- **performance measure P:** a loss function *L* to measure the (non-symmetric) distance $L(f, Z_i) = d(f(X_i), Y_i)$

Examples:

- regression
 - X is a real-valued scalar or vector
 - ► *Y* is a scalar real value
 - f is able to predict Y_i value from X_i
 - L is usually the euclidean norm

26171

Cineca

High Performance

- ▶ **training experience E:** a number of training examples $E = \{z_1, z_2, z_3...\}$ each example is a (input,target) pair: $Z_i = (X_i, Y_i)$
- ▶ task class T: a decision function f able to predict unknown Y from known X
- **performance measure P:** a loss function *L* to measure the (non-symmetric) distance $L(f, Z_i) = d(f(X_i), Y_i)$

Examples:

- regression
 - X is a real-valued scalar or vector
 - ► *Y* is a scalar real value
 - f is able to predict Y_i value from X_i
 - L is usually the euclidean norm
- classification
 - X is a real-valued scalar or vector (features)
 - Y is an integer (label) corresponding to a class index
 - f is able to provide the probability of X_i being in class Y_i
 - L is usually the negative log-likelihood

Cineca

High Performance

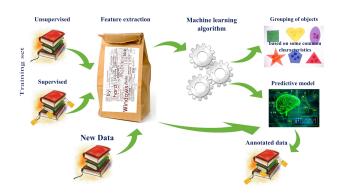


Cineca TRAINING High Performance Computing 2017



• Application of computer-enabled algorithm to a data set to find a pattern





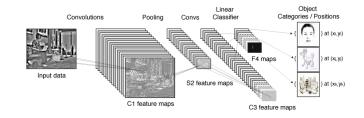
- Application of computer-enabled algorithm to a data set to find a pattern
- Wide range of tasks: segmentation. classification, clustering, supervised/unsupervised learning



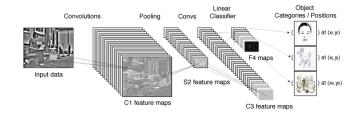


- Application of computer-enabled algorithm to a data set to find a pattern
- Wide range of tasks: segmentation. classification, clustering, supervised/unsupervised learning
- Various algorithms: association rules, decision trees, SVM



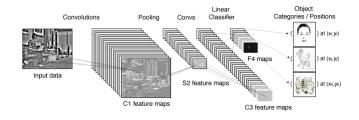


Cineca TRAINING High Performance Computing 2017

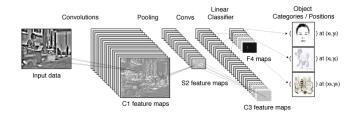


Application of an Artificial Neural Network to a data set to find a pattern

99111



- Application of an Artificial Neural Network to a data set to find a pattern
- Multiple hidden layers (to mimic human brain processes associated to vision/hearing)



- Application of an Artificial Neural Network to a data set to find a pattern
- Multiple hidden layers (to mimic human brain processes associated to vision/hearing)
- Big data sets and relevant number of variables

Framework desired features

We are interested in:

- classical machine learning algorithms
- deep learning approach (especially convolutional neural network)



Cineca

High Performance

Framework desired features

We are interested in:

- classical machine learning algorithms
- deep learning approach (especially convolutional neural network)
- high level language (Python)
- little/no programming effort

Cineca

High Performance

Framework desired features

We are interested in:

- classical machine learning algorithms
- deep learning approach (especially convolutional neural network)
- high level language (Python)
- little/no programming effort
- integration with existing pipelines
- multi-core CPU and/or many-core GPU support



Cineca

High Performance

Intel Data Analytics Acceleration Library (DAAL)

Cineca TRAINING

High Performance Computing 2017





Intel Data Analytics Acceleration Library (DAAL)



Functions for machine learning, deep learning, data analytics





- Functions for machine learning, deep learning, data analytics
- Optimized for Intel architecture devices (processors, coprocessors, and compatibles)





- Functions for machine learning, deep learning, data analytics
- Optimized for Intel architecture devices (processors, coprocessors, and compatibles)
- C++, Java and Python APIs





- Functions for machine learning, deep learning, data analytics
- Optimized for Intel architecture devices (processors, coprocessors, and compatibles)
- C++, Java and Python APIs
- Connectors to popular data sources including Spark and Hadoop



- Functions for machine learning, deep learning, data analytics
- Optimized for Intel architecture devices (processors, coprocessors, and compatibles)
- C++, Java and Python APIs
- Connectors to popular data sources including Spark and Hadoop
- Open source version under Apache 2.0 license

Cineca TRAINING High Performance Computing 2017

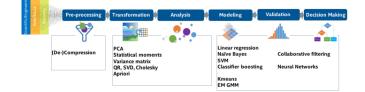
Intel Data Analytics Acceleration Library (DAAL)



- Functions for machine learning, deep learning, data analytics
- Optimized for Intel architecture devices (processors, coprocessors, and compatibles)
- C++, Java and Python APIs
- Connectors to popular data sources including Spark and Hadoop
- Open source version under Apache 2.0 license
- Paid versions include premium support.



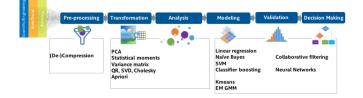
Cineca TRAINING High Performance Computing 2017





11/32

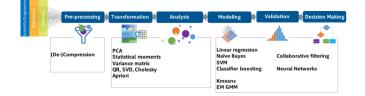
Cineca TRAINING High Performance Computing 2017



Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix



Cineca TRAINING High Performance Computing 2017



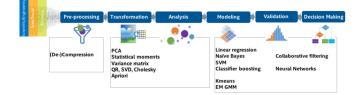
Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD



99111

Cineca TRAINING High Performance Computing 2017

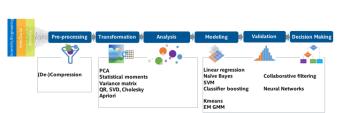


Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

99111



Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

Classification: Naive Bayes, K-Nearest Neighbors, SVM, multiclass classification

Cineca

High Performance

Computing 2017

Pre-processing Transformation Modeling Validation Decision Making Analysis PCA Linear regression (De-)Compression Naïve Bayes **Collaborative filtering** Statistical moments SVM Variance matrix **Classifier boosting** Neural Networks QR, SVD, Cholesky Apriori Kmeans EM GMM

Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

Classification: Naive Bayes, K-Nearest Neighbors, SVM, multiclass classification

Neural Networks: layers of type: fully-connected, activation, convolutional, normalization, concat, split, softmax, loss function



Cineca

High Performance

Computing 2017

Pre-processing Transformation Validation Decision Making Analysis Modeling PCA Linear regression (De-)Compression Naïve Bayes **Collaborative filtering** Statistical moments SVM Variance matrix **Classifier boosting** Neural Networks QR, SVD, Cholesky Apriori Kmeans EM GMM

Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

Classification: Naive Bayes, K-Nearest Neighbors, SVM, multiclass classification

Neural Networks: layers of type: fully-connected, activation, convolutional, normalization, concat, split, softmax, loss function

Clustering: K-Means, EM for GMM



Cineca

High Performance

Computing 2017

Cineca TRAINING High Performance Computing 2017





R = F(D.....D.)

Batch processing

All data is stored in the memory of a single node. An Intel DAAL function is called to process the data all at once.



Cineca TRAINING High Performance Computing 2017



R = F(D.....D.)



 $S_{i+1} = T(S_i, D_i)$ $R_{i+1} = F(S_{i+1})$

Batch processing

All data is stored in the memory of a single node. An Intel DAAL function is called to process the data all at once.

Streaming processing

All data does not fit in memory, or when data is arriving piece by piece. Intel DAAL can process data chunks individually and combine all partial results at the finalizing stage.

Cineca TRAINING High Performance Computing 2017



R = F(D,.....D.)



 $\begin{array}{l} S_{i+1} = T(S_i,D_i) \\ R_{\cdots} = F(S_{\cdots}) \end{array}$



Batch processing

All data is stored in the memory of a single node. An Intel DAAL function is called to process the data all at once.

Streaming processing

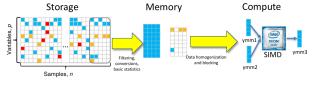
All data does not fit in memory, or when data is arriving piece by piece. Intel DAAL can process data chunks individually and combine all partial results at the finalizing stage.

Distributed processing

Intel DAAL supports a model similar to MapReduce. Slaves in a cluster process local data (map stage), and then the master process collects and combines partial results from slaves (reduce stage).

DAAL data flow

Cineca TRAINING High Performance Computing 2017



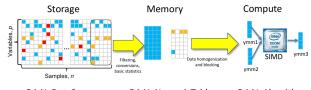
DAAL DataSource

DAAL NumericTable

DAAL Algorithm

DAAL data flow

Cineca TRAINING High Performance Computing 2017



DAAL DataSource

DAAL NumericTable

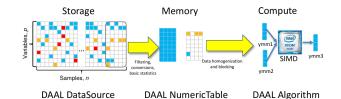
DAAL Algorithm

Data sources:

- ► file based (CSV, binary)
- database query (ODBC, SQL)
- Python: numpy array interoperability

DAAL data flow

Cineca TRAINING High Performance Computing 2017



Data sources:

- file based (CSV, binary)
- database query (ODBC, SQL)
- Python: numpy array interoperability

Data structures:

- numeric tables
 - homogeneous data: dense, sparse, packed, triangular matrix, symmetric matrix
 - heterogeneous data: SOA vs AOS
- tensors (n-dimensional matrix)



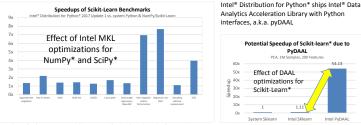
Official Intel benchmark results (I)

Cineca TRAINING High Performance

Computing 2017

(intel) SC16

Skt-Learn* Optimizations With Intel® MKL... And Intel® DAAL



System info: 32x intel[®] Xeon[®] CPU ES-2008 v3 @ 2.30GHz, disabled HT, 64GB RAM; Intel[®] Distribution for Python[®] 2017 Gold; Intel[®] MKL 2017.0.0; Ubuntu 14.04.4175; Numpy 1.11.1; scikit-learn 0.17.1

Software and workdadu used is performance texts may have been optimized for performance only on 1441 incorporations. Any Change to any of those factors may cause the results to vary. Tou should consult other informance texts to assist you in fully exakating your contemplated parchases, including the performance of that pocket, when combined with other products. - Any Annanes text the powers of their expective ones. Benchmark Socret Information and performance texts to assist you in fully exakating your contemplated parchases, including the performance of that pocket, when combined with other products. - Other branch and manuse use the powers of their expective ones. Benchmark Socret Information

Optimization Notice: Intel Complem may or may not optimize to the same degree for non-intel microprocesson for optimizations that are not unique to intel microprocesson. These optimizations include \$322, \$323, and \$3243, and \$3243,

[Supercomputing 2016 (SC16), November 13-18, 2016, Salt Lake City]



Official Intel benchmark results (II)

Cineca TRAINING High Performan

(intel) SC16

High Performance Computing 2017

Distributed Parallelism

- Intel[®] MPI* accelerates Intel[®] Distribution for Python (mpi4py*, ipyparallel*)
- Intel Distribution for Python also supports
 - PySpark* Python* interfaces for Spark*, an engine for large-scale data processing
 - Dask* flexible parallel computing library for numerical computing

PyDAAL Implicit ALS with Mpi4Py* 6x 5x Scales with MPI. Spark, Dask and other distributed computing 4x engines 2x 1.7x 2.2x 3.0x 5.3x 2 nodes 4 nodes 8 nodes 16 nodes

Configuration Info: Hardware (each node): Intel(R) Xeon(R) CPU E5-2697 v4 (b 2.30GHz, 2x18 cores, HT is ON, RAM 128GB; Versions: Oracle Linux Server 6.6, Intel® DAAL 2017 Gold, Intel® MPI 5.1.3; Interconnect: 1 GB Ethernet.

Software and workloads used in performance tesh may have been optimised for performance only on bitel micropromons. Performance tesh, such as 35mek and MobileNak, are measured using specific computer systems, components, unhaver, operations and functions have dwargs and of these factors may cause the multist vary. To as board consult of the informations and performance tesh to solit you in USy evaluation your contempliated purchases, includes the performance of that product when combined with other products. The branch and some are the property of the impeditor waves. These provides of the information and the product systems from the source interformation and performance tesh.

Optimization Notice: Itel® complem may or agents to the same degree for non-Itel® interprocession for optimizations that are not unique to Itel interprocession. These optimizations include SE2, SEE3, and SSEE1 instruction sets and other optimizations include for a same degree for non-Itel® interprocession for optimizations that are not unique to Itel interprocession. These optimizations include SE2, SEE3, and SSEE1 instruction sets and other optimizations include sets and other optimizations include sets and other optimizations in the product sets and sets are interded for use with held interprocession. These optimizations in the product sets and other optimizations in tell interprocession. These optimizations in the product sets and sets a

[Supercomputing 2016 (SC16), November 13-18, 2016, Salt Lake City]

