

Domain specific libraries for Deep Learning

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

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.

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



Cineca TRAINING High Performance Computing 2017



task class T: playing checkers



Cineca TRAINING High Performance Computing 2017



444

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

Cineca TRAINING High Performance Computing 2017



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

Cineca TRAINING High Performance

Computing 2017

Example: handwritten characters recognition

82944649709275159133 2359176282507497832 1/836105100112730465 26472899307102835465

interference in a policitati Interference in Magain Tables Auflideren anbeitet	
	Very prog negade part at 3 apresent
	the order d'adat d'adain l'here d'ada
elje tene adaren. Aj nare deravel	256.56 7890 Jon Small fremantik & 50
Taken, Trener, Second as a discussion of generative frequencies of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the discussion of the di	Electric de construction de construction de la construction de la construction de la construction de la construction de constr
in the system of . New property Reads,	touldement de Vous en merris
	Bien purspierent



Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017



99111

task class T: recognizing and classifying handwritten characters within images

5/32

Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017

1359176	19709295159183 52822507497832 03100112730465 19307102035465
word Anna Internation Maps State Annalase strengt Annalase Strengt Annalase Strengt Annalase Strengt Annalase Strengt Annalase Strengt	Van 1995 Personal personal (* 1999) 1999 Marine Kasar Kasar (* 1997) 1995 Table (* 1997)
Tables Treases Trease and a property of property from a solitor for any end of property of the soliton of the soliton from the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton of the soliton	Example Research (R) Source (R) Source (R) production (R) Source (R) Source (R) (marticle) (R) Source (R) Source (R) Research (R) None (R) Source (R) Research (R) None (R) Source (R) (R) Source (R) Source

99111

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

Example: handwritten characters recognition

Cineca TRAINING High Performance Computing 2017

1183610	9709295159103 2822507497832 93100112730465 9307102035465
and the second second	
The American States of Sta	New press Prequented part at the profine
Santi Sanah Kala Kingan Dina Sangan	was brief & adat & adain Visit Gole
trips transactioners, appears, terminal	125656 7890 Jon Sund frankle & So
$\label{eq:constraints} \begin{split} & Tables, (Protect,) \\ & for any exponent energy of a particle of the state of th$	Eland Erena de la como fine al colo par enclara lonno in provinti de provente de 53 forma des como
Son progenite Gaude,	toublement to your an mound

444

- 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

26776

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



Cineca TRAINING High Performance

Computing 2017



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



Cineca TRAINING High Performance



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



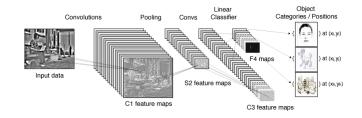
Cineca TRAINING High Performance



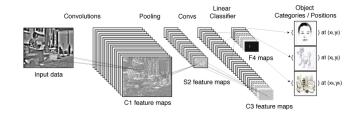
- Application of computer-enabled algorithm to a data set to find a pattern
- Wide range of tasks: segmentation. classification, clustering, supervised/unsupervised learning
- Various algorithms: association rules, decision trees, SVM



Cineca TRAINING High Performance Computing 2017



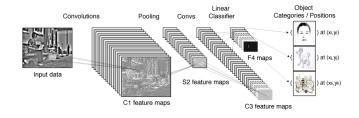
Cineca TRAINING High Performance Computing 2017



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

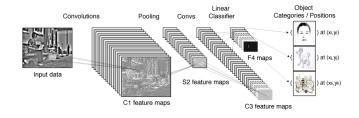
99111

Cineca TRAINING High Performance Computing 2017



- 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)

Cineca TRAINING High Performance Computing 2017



- 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

Cineca TRAINING High Performance

Computing 2017

We are interested in:

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

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)

D ...

Cineca TRAINING

High Performance Computing 2017

UATA SOURCES	Pre-processing	Transformation	Analysis	Modeling	Validation	Decision Making
Business Scientific Engineering Web/Social	 Decompression Filtering Normalization 	 Aggregation Dimension Reduction 	 Summary Statistics Clustering. 	 Machine Learning Parameter Estimation Simulation 	 Hypothesis testing Model errors 	Forecasting Decision Trees Etc.



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

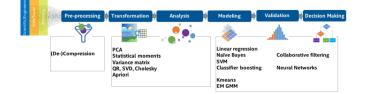
Intel Data Analytics Acceleration Library (DAAL)



- 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.

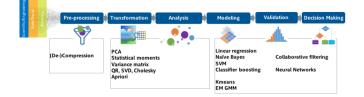


Cineca TRAINING High Performance Computing 2017





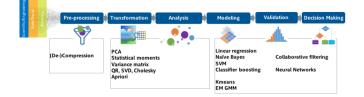
Cineca TRAINING High Performance Computing 2017



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



Cineca TRAINING High Performance Computing 2017



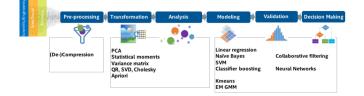
Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD



99111

Cineca TRAINING High Performance Computing 2017



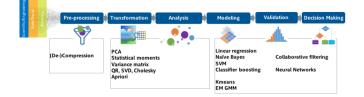
Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

99111

Cineca TRAINING High Performance Computing 2017



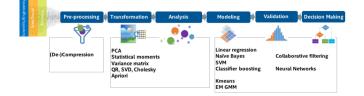
Statistics: min, max, mean, standard deviation, correlation, covariance matrix, correlation distance matrix, cosine distance matrix

Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

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

Cineca TRAINING High Performance Computing 2017



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

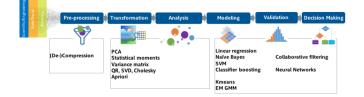
Factorizations: Cholesky, QR, SVD

Dimensionality Reduction: PCA

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 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

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 TRAINING 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.

99111

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

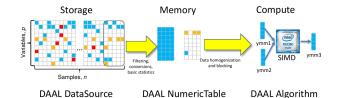
All data does not fit in memory, or when data is arriving piece by piece. Intel DAAL can process data chunks individually and combine all partial results at the finalizing stage.

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

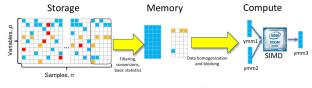




13/32

DAAL data flow

Cineca TRAINING High Performance Computing 2017



DAAL DataSource

DAAL NumericTable

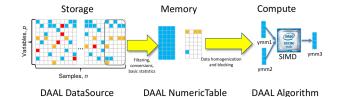
DAAL Algorithm

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)



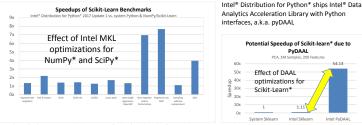
Official Intel benchmark results (I)

Cineca TRAINING High Performance

Computing 2017

(intel) SC16

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



System info: 32x intel[®] Xeon[®] CPU E5-2098 v3 @ 2.30GHz, disabled HT, 64GB RAM; Intel[®] Distribution for Python[®] 2017 Gold; Intel[®] MKL 2017.0.0; Uburtu 14.04.4 LTS; Numpy 1.11.1; scikit-learn 0.17.3

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 Performance Computing 2017

(intel) SC16

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 (p 2.30GHz, 2x18 cores, HT is DN, 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 tests may have been optimised for performance only on hitel micropromoson. Performance tests, such as Stanuk and MobileNak, are measured using specific computer systems, components, unbrave, operations and functions. Nay charge tax ory of those factors may have been optimised for performance only on hitel micropromoson. Performance tests, such as Stanuk and MobileNak, are measured using specific computer systems, components, unbrave, operations and functions. Nay charge tax ory of those factors may charge tax or the standard system component of the performance of that product when combined with other products. The branch and shares are the property of the impeditor uses. Standard system component on the performance of the performance of the product when combined with other products.

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 and sets are interded for use with held interprocession. These optimizations in the product sets and sets are interded for use with held interprocession. These optimizations in the product sets are interded for use with held interprocession. These optimizations in the product sets are interded for use with held interprocession. These optimizations in the product sets are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations in the product sets are interded for use with held interprocession. These optimizations in the product sets are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are interded for use with held interprocession. These optimizations are

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



pyDAAL installation

Cineca TRAINING High Performance Computing 2017

Requirements:

- Intel Math Kernel Library (MKL): for BLAS and LAPACK
- Integrated Performance Primitives (IPP) for data compression/decompression
- Threading Building Blocks (TBB) for multicore and many-core parallelism

pyDAAL installation

Cineca TRAINING High Performance Computing 2017

Requirements:

- Intel Math Kernel Library (MKL): for BLAS and LAPACK
- Integrated Performance Primitives (IPP) for data compression/decompression
- Threading Building Blocks (TBB) for multicore and many-core parallelism

Installation methods:

- 1. anaconda Intel channel (Linux)
- 2. Intel distribution (Windows, Linux, OS X)
- 3. build from source