PRIN17 - Multicriteria Data Structures and Algorithms: from compressed to learned indexes, and beyond
Meeting PRIN - 09.02.22

UNIMI, Dipartimento Informatica

<table>
<thead>
<tr>
<th>Componenti Unità</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marco Frasca</td>
</tr>
<tr>
<td>Giorgio Valentini</td>
</tr>
<tr>
<td>Dario Malchiodi</td>
</tr>
<tr>
<td>Marco Mesiti</td>
</tr>
<tr>
<td>Paolo Perlasca</td>
</tr>
<tr>
<td><strong>Giosuè Marinò</strong></td>
</tr>
<tr>
<td><strong>Alessandro Petrini</strong></td>
</tr>
<tr>
<td><strong>Jessica Gliozzo</strong></td>
</tr>
</tbody>
</table>

RTD B (coordinator)
PO
PA
PA
RU
Collaboratore
Assegnista
Dottoranda
Past Attempt. (UNIPI) Rank integer by simulating the PGM-index through a (Deep) Neural Network (NN) to evaluate potential extensions
Past Attempt. (UNIPI) Rank integer by simulating the PGM-index through a (Deep) Neural Network (NN) to evaluate potential extensions

- PROBLEM INPUT: sorted sequence of integers $X$ [extendable to other type of elements]
Past Attempt. (UNIPI) Rank integer by simulating the PGM-index through a (Deep) Neural Network (NN) to evaluate potential extensions

- PROBLEM INPUT: sorted sequence of integers $X$ [extendable to other type of elements]
- PROBLEM OUTPUT: a learned index to perform the $\text{rank}[x] := |y \in X | y \leq x|$, $x \in X$
Past Attempt. (UNIPI) Rank integer by simulating the PGM-index through a (Deep) Neural Network (NN) to evaluate potential extensions

- PROBLEM INPUT: sorted sequence of integers $X$ [extendable to other type of elements]
- PROBLEM OUTPUT: a learned index to perform the $\text{rank}[x] := |\{y \in X| y \leq x\}, x \in X$

The network can refine the weights when initialized with PGM slopes and intercepts at the lowest level, but cannot learn it from the beginning
Past Attempt. (UNIPI) Rank integer by simulating the PGM-index through a (Deep) Neural Network (NN) to evaluate potential extensions

- PROBLEM INPUT: sorted sequence of integers $X$ [extendable to other type of elements]
- PROBLEM OUTPUT: a learned index to perform the $\text{rank}[x] := |y \in X | y \leq x|, x \in X$

The network can refine the weights when initialized with PGM slopes and intercepts at the lowest level, but cannot learn it from the beginning

- No short-mid term extensions are planned for this problem
Same problem. (UNIPI) Rank strings through DNN
Same problem. *(UNIPI)* Rank *strings* through DNN

- PROBLEM INPUT: sorted sequence of strings $X$
Same problem. (UNIPI) Rank strings through DNN

- PROBLEM INPUT: sorted sequence of strings $X$
- PROBLEM OUTPUT: a learned index to perform the $\text{rank}[x] := |\{y \in X| y \leq x\}|$, $x \in X$
Same problem. (UNIPI) Rank strings through DNN

- PROBLEM INPUT: sorted sequence of strings $X$
- PROBLEM OUTPUT: a learned index to perform the $\text{rank}[x] := |\{y \in X \mid y \leq x\}|$, $x \in X$

As baseline we consider Fast Succinct Trie (FST) [CACM 21]
Learning issues

- We cannot solve the problem in an exact way with DNN
Learning issues

- We cannot solve the problem in an exact way with DNN
- Usually we minimize the average error, no possibility to control the maximum error
Learning issues

- We cannot solve the problem in an exact way with DNN
- Usually we minimize the average error, no possibility to control the maximum error
- Training can be slow (we need to overfit)
Learning issues

- We cannot solve the problem in an exact way with DNN
- Usually we minimize the average error, no possibility to control the maximum error
- Training can be slow (we need to overfit)

- We compare with FST having the same average error

Short-term studies: characterize the cases where NN do better than FST
Mid-term studies: investigate the usage of multiple NNs
Learning issues

- We cannot solve the problem in an exact way with DNN
- Usually we minimize the average error, no possibility to control the maximum error
- Training can be slow (we need to overfit)

- We compare with FST having the same average error
- In some cases NNs do better
- Further experiments are needed

Short-term studies: characterize the cases where NN do better than FST
Learning issues

- We cannot solve the problem in an exact way with DNN
- Usually we minimize the average error, no possibility to control the maximum error
- Training can be slow (we need to overfit)

- We compare with FST having the same average error
- In some cases NNs do better
- Further experiments are needed

Short-term studies: characterize the cases where NN do better than FST

Mid-term studies: investigate the usage of multiple NNs
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing reasearch
- Preliminary results (ne parlerà Dario)
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
- Preliminary results (ne parlerà Dario)

Bloom Filter

- Space efficient probabilistic data structure for representing a set $X \subset S$
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
- Preliminary results (ne parlerà Dario)

**Bloom Filter**

- Space efficient probabilistic data structure for representing a set $X \subset S$
- Support membership queries: given $y \in S$, $y$ belongs to $X$?
  - False negatives: **NO**
  - False positives: **YES**
**Another research activity in T1** (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
- Preliminary results (ne parlerà Dario)

**Bloom Filter**

- Space efficient probabilistic data structure for representing a set $X \subset S$
- Support membership queries: given $y \in S$, $y$ belongs to $X$?
  - False negatives: **NO**
  - False positives: **YES**
- It has been shown that embedding a ML oracle into the filter can help reducing its space/false positive rate (LBF)
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
- Preliminary results (ne parlerà Dario)

**Bloom Filter**

- Space efficient probabilistic data structure for representing a set \( X \subset S \)
- Support membership queries: given \( y \in S \), \( y \) belongs to \( X \)?
  - False negatives: **NO**
  - False positives: **YES**
- It has been shown that embedding a ML oracle into the filter can help reducing its space/false positive rate (LBF)
- We are studying the role of classifiers in LBF
Another research activity in T1 (UNIPA): study the impact of ML models in Learned Bloom Filters (LBFs)

- Recent works have shown that ‘classical’ BF might be improved in terms of space occupancy by using ML models
- Ongoing research
- Preliminary results (ne parlerà Dario)

**Bloom Filter**

- Space efficient probabilistic data structure for representing a set $X \subset S$
- Support membership queries: given $y \in S$, $y$ belongs to $X$?
  - False negatives: **NO**
  - False positives: **YES**

- It has been shown that embedding a ML oracle into the filter can help reducing its space/false positive rate (LBF)
- We are studying the role of classifiers in LBF
- ... and their dependence on the query distribution
Short-term research objectives on LBF (UNIPA):

- Try answer the question "Which classifier?" and "Complex is (always) better?"
Short-term research objectives on LBF (UNIPA):

- Try answer the question "Which classifier?" and "Complex is (always) better?"

- ... and "To which extent LBF are sensitive to the query distribution"?
Short-term research objectives on LBF (UNIPA):

- Try answer the question "Which classifier?" and "Complex is (always) better?"

- ... and "To which extent LBF are sensitive to the query distribution"?

- When test and query distribution are drawn from the same distribution, it is proven that the FPR on query set is expected to be very close to the test one
Short-term research objectives on LBF (UNIPA):

- Try answer the question "Which classifier?" and "Complex is (always) better?"

- ... and "To which extent LBF are sensitive to the query distribution"?

- When test and query distribution are drawn from the same distribution, it is proven that the FPR on query set is expected to be very close to the test one

- And in the other cases?
NN compression

The aim of our unit in this task is to study techniques to reduce the model space requirements while possibly preserving (or even improving) its accuracy.
The aim of our unit in this task is to study techniques to reduce the model space requirements while possibly preserving (or even improving) its accuracy.

Understand the capability of different compression techniques and (possibly) characterize when it is convenient to use them.

Connection Pruning, Weight quantization, Sparsified NN training, Knowledge distillation, Filter/channel removal, optimization frameworks...

Combined approaches

Which method to use?
Which layer type to compress?
Compressed ML models

NN compression

• The aim of our unit in this task is to study techniques to reduce the model space requirements while possibly preserving (or even improving) its accuracy.

• Understand the capability of different compression techniques and (possibly) characterize when it is convenient to use them.
  
  • Connection Pruning, Weight quantization, Sparsified NN training, Knowledge distillation, Filter/channel removal, optimization frameworks ...
  
  • Combined approaches
**Compressed ML models**

**NN compression**

- The aim of our unit in this task is to study techniques to reduce the model space requirements while possibly preserving (or even improving) its accuracy.

- Understand the capability of different compression techniques and (possibly) characterize when it is convenient to use them:
  - Connection Pruning, Weight quantization, Sparsified NN training, Knowledge distillation, Filter/channel removal, optimization frameworks...
  - Combined approaches

- Which method to use?
**Compressed ML models**

**NN compression**

- The aim of our unit in this task is to study techniques to reduce the model space requirements while possibly preserving (or even improving) its accuracy.

- Understand the capability of different compression techniques and (possibly) characterize when it is convenient to use them.
  - Connection Pruning, Weight quantization, Sparsified NN training, Knowledge distillation, Filter/channel removal, optimization frameworks ...
  - Combined approaches

- Which method to use?

- Which layer type to compress?
NN compression: old results.

- Applied pruning and different techniques of weight quantization and evaluated their performance.
Compress ML models

**NN compression: old results.**

- Applied pruning and different techniques of weight quantization and evaluated their performance

- Introduced a compact format (HAM, sHAM) for compressed fully connected layers, able to take advantage from both sparse and repeated weights
NN compression: old results.

- Applied pruning and different techniques of weight quantization and evaluated their performance
- Introduced a compact format (HAM, sHAM) for compressed fully connected layers, able to take advantage from both sparse and repeated weights
- Validated our approach on two publicly available CNN for multi-class classification and for regression, on 4 different datasets

Preliminary results published in [1] and [2]


**NN compression: old results.**

- Applied pruning and different techniques of weight quantization and evaluated their performance

- Introduced a compact format (HAM, sHAM) for compressed fully connected layers, able to take advantage from both sparse and repeated weights

- Validated our approach on two publicly available CNN for multi-class classification and for regression, on 4 different datasets

- Compression ratio up to 165 while preserving accuracy


NN compression: extensions and short-term objectives

- Extending the set of NN compression techniques (done)
Compressed ML models

NN compression: extensions and short-term objectives

- Extending the set of NN compression techniques (done)
- Considering also the compression of convolutional layers (done)
NN compression: extensions and short-term objectives

- Extending the set of NN compression techniques (done)
- Considering also the compression of convolutional layers (done)
- Compare sHAM with the state of the art methods for sparse and quantized matrix compression (partially done)
**NN compression: extensions and short-term objectives**

- Extending the set of NN compression techniques (done)
- Considering also the compression of convolutional layers (done)
- Compare sHAM with the state of the art methods for sparse and quantized matrix compression (partially done)
- Build-up an automatic optimizer across the range of possible compression techniques so as to respect a given constraint of the maximum space occupied by the compressed model (depends on the completion of previous points)
NN compression: extensions and short-term objectives

- Extending the set of NN compression techniques (done)
- Considering also the compression of convolutional layers (done)
- Compare sHAM with the state of the art methods for sparse and quantized matrix compression (partially done)
- Build-up an automatic optimizer across the range of possible compression techniques so as to respect a given constraint of the maximum space occupied by the compressed model (depends on the completion of previous points)
- Improve sHAM in the coding phase
Application of NN compression in Medicine: Breast cancer diagnosis

- Based on the count of specific cells types from breast tissue images
Compressed ML models

Application of NN compression in Medicine: Breast cancer diagnosis

- Based on the count of specific cells types from breast tissue images
- PathoNet [SciRep21] is a CNN model proposed for the automatic detection and count of pathogenic cells
Application of NN compression in Medicine: Breast cancer diagnosis

- Based on the count of specific cells types from breast tissue images
- PathoNet [SciRep21] is a CNN model proposed for the automatic detection and count of pathogenic cells
- We preserve its accuracy by reducing up to 4× its memory footprint, up to 6× its disk requirement, with a negligible execution time increase
Application of NN compression in Medicine: Breast cancer diagnosis

- Based on the count of specific cell types from breast tissue images
- PathoNet [SciRep21] is a CNN model proposed for the automatic detection and count of pathogenic cells
- We preserve its accuracy by reducing up to $4 \times$ its memory footprint, up to $6 \times$ its disk requirement, with a negligible execution time increase
- Allows the deployment and usage on edge/low resource devices
Compressed ML models

Application of NN compression in Medicine: Breast cancer diagnosis

- Based on the count of specific cells types from breast tissue images
- PathoNet [SciRep21] is a CNN model proposed for the automatic detection and count of pathogenic cells
- We preserve its accuracy by reducing up to $4 \times$ its memory footprint, up to $6 \times$ its disk requirement, with a negligible execution time increase
- Allows the deployment and usage on edge/low resource devices
- Potentially extends its application to poor geographic areas
- Manuscript to be completed