Activities report from August 2012 to April 2013

Telmo Amaral

26th April 2013

Project “Reusable Deep Neural Networks: Applications to Biomedical Data”  
(PDTC/EIA-EIA/119004/2010)

Instituto de Engenharia Biomédica (INEB)  
Rua Dr. Roberto Frias, 4200-465, Porto, Portugal
Contents

1 Introduction 3

2 Introduction to deep learning 3

3 Visit to LISA lab 3
   3.1 Input from Prof. Bengio .......................... 3
   3.2 Activities and notes .............................. 5

4 Notes on existing software 5

5 Introduction to convolutional neural networks 7

6 Notes on UCI biomedical datasets 7

7 Notes on relevant conferences 8

A Slides: Introduction to deep learning 10

B Slides: Introduction to convolutional neural networks 14

C Spreadsheet: Notes on UCI biomedical data sets 17

References 19
1 Introduction

This report describes several activities I have undertaken since the beginning of my contract in August 2012 until the end of the project’s first year, at the start of May, 2013. Each of the following sections addresses an individual activity. Activities are ordered more or less chronologically.

Besides the tasks described here, work related with the training of restricted Boltzmann machines (RBMs) and stacked auto-encoders is described in detail in two separate technical reports [1, 2].

2 Introduction to deep learning

My work in the project started in August 2012, with a review of introductory materials about deep learning, which included the first three sections of the 2009 article/book “Learning deep architectures for AI” by Bengio [4], as well as the 2010 review paper “Deep machine learning – a new frontier in artificial intelligence research” by Arel et al. [3].

Those readings allowed me to become acquainted with the concepts of shallow versus deep architectures, the motivations for the use of deep architectures (such as their inspiration from nature and their theoretical advantages in terms of efficiency), and the limitations associated with deep architectures until the breakthrough led by Hinton et al. [8] in 2006.

I prepared an overview of these topics and presented it in the project meeting of October 12, 2012. This presentation included a first list of some publicly available software implementations of deep learning models. It is included in Appendix A of this report.

3 Visit to LISA lab

In January 2013, for a period of four weeks, I visited the LISA machine learning laboratory1 at the University of Montreal, headed by Prof. Yoshua Bengio. LISA is one of the most important research groups worldwide in the area of deep learning, routinely producing state-of-the-art research.

3.1 Input from Prof. Bengio

Prof. Bengio offered some comments of particular relevance to our project, as well as some feedback on the work carried out to date within the project, regarding the training of RBMs, which I have demonstrated to him. His comments are summarised in the following paragraphs.

Machine learning models in general need large amounts of training data. Often, there is not a lot of data available in biomedical applications, especially

---

data annotated by experts. In order to train deep learning models aimed at biomedical applications, it would be important to have at least a large amount of non-labelled data, for unsupervised pre-training.

Computer vision is where deep learning excels the most. A drop in object classification error rate from about 30% to less than 16% has been recently achieved using deep convolutional nets, as reported by Krizhevsky et al. [9].

The unsupervised training of auto-encoders could offer an opportunity for the use of alternative cost functions, given that auto-encoders are trained to reconstruct their own inputs. Moreover, auto-encoders are much simpler to train than RBMs. These remarks by Prof. Bengio prompted our own work with auto-encoders, started by Luis Alexandre and continued by myself and Chetak Kandaswamy, as described in a separate technical report [2].

Regarding the unsupervised training of RBMs, division of the training data into mini-batches is important not only to avoid operations with very large matrices, but also in terms of efficiency. Using all the available training data at each training epoch (iteration) would be excessive, because we don’t need to update the model’s parameters precisely in the right direction at each epoch; a smaller amount of data is sufficient to update the parameters in the right general direction, and much less time-consuming. An analogy can be drawn with a person going from a point A to a point B in a number of steps: the person doesn’t need to worry about moving precisely in the direction of point B at each step taken; a number of roughly precise steps will lead to B just the same.

A clarification regarding the sampling mechanism used during the training of RBMs: what we’re using is so-called block Gibbs sampling, as opposed to Gibbs sampling “one bit at a time”. Block Gibbs sampling allows to more efficiently explore the modes of the multi-dimensional data distribution, since all bits are samples in parallel. This is possible in restricted Boltzmann machines, but not in unrestricted ones.

On the subject of sampling from a trained RBM, in theory, when starting from a random visible vector, it is necessary to perform a large number of Gibbs sampling steps to obtain a good sample, like we did in our implementation. However, once this “burn in” is achieved, each new step, if done without restarting from a random vector, is enough to yield a good sample. In practice, if we start not from a random vector but from a training vector, we can assume that the “burn in” is already done and start drawing samples right away. These techniques can greatly speed up the generation of samples.

The fact that the type of RBM we’ve implemented is meant for use with binary data doesn’t preclude its use with real-valued data scaled between 0 and 1. Our discouraging results with real-valued data, in particular with the Iris data set, could be due to the choice of learning rate. In some cases it is important to adjust the learning rate throughout the training. (Later, in our work with auto-encoders, we have implemented adaptive learning rates.)

In case we pursue experiments with RBMs, we should now focus on using more efficient code, so that we may use data sets with large numbers of features and examples.
3.2 Activities and notes

During my visit, I did a more in-depth reading of some parts of Prof. Bengio’s 2009 book [4], in particular the sections on neural networks for deep architectures (Section 4) and on energy-based models and Boltzmann machines (Section 5). Prof. Bengio’s lecture notes “Introduction to Gradient-Based Learning” proved to be very useful as an introduction to gradient descent learning methods. The material I read on convolutional neural networks later helped me to prepare the presentation described in Section 5.

I spent some time also with the previously mentioned paper by Krizhevsky et al. [9], which reports recent developments on the use of convolutional neural networks, as well as with the pre-print of a new review on deep learning by Bengio et al. [5], which is now publicly available. While not precluding the reading of the 2009 book, this article contains a lot of information on recent developments and research directions.

In order to demonstrate my Matlab implementation of RBM training algorithms to Prof. Bengio, I spent some time tidying up its experimental scripts and collecting results into a report. This ultimately became the technical report on experiments with RBMs being submitted separately [1].

I read some introductory materials on the Python programming language, the NumPy numerical library, as well as Theano, a Python library that facilitates the development of deep learning models, while giving the option of performing their training on one or more graphical processing units (GPUs). The LISA lab was the ideal place to become acquainted with these programming technologies, as Theano has been developed by researchers at LISA and is extensively used in their work.

Throughout the visit, I collected some notes on various software implementations of deep learning models that are publicly available. Those notes are presented in Section 4.

I had the chance to attend a seminar presented by a leading researcher of the Canadian company D-Wave Systems Inc., on the use of quantum computing for training deep learning models. This company specialises in the development and commercialisation of quantum computers, an emerging field with potential deep learning applications. I attended also a brief “tea talk” about a paper co-authored by Prof. Pedro Domingos, a Portuguese researcher based in the U.S. who works with deep architectures [6].

4 Notes on existing software

The following is a list of some software implementations of deep learning models that are publicly available, collected during my introductory readings and also during my visit the LISA lab. In each case, the group or researcher responsible for the software is given (with a link to the relevant web page), as well as the

---

programming language and relevant libraries used, and the models that are implemented.

1. LISA lab
   Deep Learning Tutorials
   http://deeplearning.net/tutorial/
   Python / Numpy / Theano
   • logistic regression
   • multilayer perceptrons
   • deep convolutional networks
   • autoencoders, denoising autoencoders
   • stacked denoising autoencoders
   • restricted boltzmann machines
   • deep belief networks

2. LISA lab
   Wiki, “fundamental research projects”
   http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/
   C++ / PLearn
   • deep belief networks
   • stacked autoassociators

3. Rasmusberg Palm
   DeepLearnToolbox
   https://github.com/rasmusbergpalm/DeepLearnToolbox
   Matlab
   • deep belief networks
   • stacked autoencoders
   • convolutional neural networks
   • convolutional autoencoders
   • vanilla neural networks

4. Ruslan Salakhutdinov, Geoff Hinton
   Training a deep autoencoder or a classifier on MNIST digits [7]
   Matlab
   • restricted boltzmann machines
     - binary hidden and binary visible
     - Gaussian hidden and binary visible
   • deep autoencoders
   • deep belief networks ?
5. Ruslan Salakhutdinov
Learning Deep Boltzmann Machines
http://www.utstat.toronto.edu/~rsalakhu/DBM.html
Matlab
  • deep boltzmann machines

6. Hugo Larochelle
Efficient Learning of Deep Boltzmann Machines [10]
an enhancement of 5
http://www.dmi.usherbrooke.ca/~larocheh/code/dbm_recnet.tar.gz
Matlab
  • deep boltzmann machines

7. Andrej Karpathy
matrbm
a simplified version of 4
http://code.google.com/p/matrbm/
Matlab
  • restricted boltzmann machines
  • deep belief networks (of stacked RBMs)

5 Introduction to convolutional neural networks

Following up on my visit to the LISA lab, I prepared an introduction to convolutional neural networks, focusing on the motivation for their use, the main differences from traditional networks, the concepts of shared weights and multiple feature maps, down-sampling layers, and a description of the LeNet-5 model. This introduction was presented in the project meeting of February 22, 2013 and is included in Appendix B of this report.

6 Notes on UCI biomedical datasets

In order to form an idea of the type of biomedical data that we could use in our experiments while relying only on CPU power (as opposed to using GPUs), I searched the Machine Learning Repository maintained by the University of California - Irvine (UCI) 4 for data sets that fulfilled a number of characteristics, namely: being life sciences related; being frequently and recently used; being appropriate for classification learning; and being neither too small (less than 1000 examples) nor too big (say below 10000 examples).

A spreadsheet gathering the collected information on 62 available life sciences data sets is included in Appendix C. The data sets are ordered by decreasing number of papers citing them. This number of citations, as well as the range of

dates of those citations, provides a measure of how popular and recently used each data set is.

Taking into account the characteristics we desired, the five data sets highlighted in yellow seemed to be the most interesting ones, namely: Mushroom data; Splice-junction gene sequence data; Abalone data; Thyroid disease data; and Yeast data. These findings were discussed in the project meeting of March 15, 2013. We have subsequently used two of these data sets in our experiments with auto-encoders.

7 Notes on relevant conferences

I gathered in the table shown below information on conferences that are potentially interesting for publication of our work. Conferences are categorised into machine learning (ML), computer vision (CV), and biomedical engineering (BME). This list was discussed in the project meeting of April 19, 2013, but it should be kept up to date.
<table>
<thead>
<tr>
<th>Deadline</th>
<th>Conference</th>
<th>Year</th>
<th>ML</th>
<th>CV</th>
<th>BME</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-11-10</td>
<td>ISBI</td>
<td>2013</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2012-11-15</td>
<td>AISTATS</td>
<td>2013</td>
<td>x</td>
<td></td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>2012-11-15</td>
<td>CVPR</td>
<td>2013</td>
<td>x</td>
<td>x</td>
<td></td>
<td>IEEE conference.</td>
</tr>
<tr>
<td>2012-12-07</td>
<td>ESANN</td>
<td>2013</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-01-15</td>
<td>ICLR</td>
<td>2013</td>
<td>x</td>
<td></td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>2013-02-04</td>
<td>EMBC</td>
<td>2013</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2012-10-01</td>
<td>ICML</td>
<td>2013</td>
<td>23</td>
<td>x</td>
<td></td>
<td>(2)</td>
</tr>
<tr>
<td>2012-12-15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-03-01</td>
<td>ICANN</td>
<td>2013</td>
<td>1</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-04-07</td>
<td>ICANN</td>
<td>2013</td>
<td>6</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-04-12</td>
<td>ICCV</td>
<td>2013</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-04-19</td>
<td>ICIP</td>
<td>2013</td>
<td>Germany</td>
<td>8</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-04-22</td>
<td>ECML/PKDD</td>
<td>2013</td>
<td>Czech Republic</td>
<td>7</td>
<td>x</td>
<td>(3)</td>
</tr>
<tr>
<td>2013-04-24</td>
<td>BMVC</td>
<td>2013</td>
<td>UK</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-04-30</td>
<td>ProMI</td>
<td>2013</td>
<td>India</td>
<td>2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-05-11</td>
<td>ALT</td>
<td>2013</td>
<td>Singapore</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-05-13</td>
<td>UKCI</td>
<td>2013</td>
<td>UK</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-05-24</td>
<td>IDEAL</td>
<td>2013</td>
<td>China</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-05-31</td>
<td>APCA</td>
<td>2013</td>
<td>Italy</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-05-31</td>
<td>NIPS</td>
<td>2013</td>
<td>USA</td>
<td>17</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-06-01</td>
<td>ICTET</td>
<td>2013</td>
<td>India</td>
<td>7</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-06-15</td>
<td>CIARP</td>
<td>2013</td>
<td>Cuba</td>
<td>4</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-06-15</td>
<td>ICONIP</td>
<td>2013</td>
<td>South Korea</td>
<td>5</td>
<td>x</td>
<td>Bengio as plenary speaker.</td>
</tr>
<tr>
<td>2013-06-15</td>
<td>PRM</td>
<td>2013</td>
<td>Russia</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2013-08-25</td>
<td>ICPRAM</td>
<td>2014</td>
<td>France</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-12-20</td>
<td>ICPR</td>
<td>2014</td>
<td>Sweden</td>
<td>11</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

Numbers next to x's are Field Ratings from Microsoft Academic Search

(1) AISTATS may not be very relevant, because its former Learning Workshop now became a separate conference, ICLR.

(2) ICML 2013 had three submission / reviewing cycles. Next year there may be even more. The idea seems to be to make the conference work also a journal.

(3) ECML/PKDD 2013 accepted submissions also for a new "journal track": papers could be submitted to one of two journals and, if accepted, would receive a presentation slot at the conference.
A  Slides: Introduction to deep learning
Shallow vs. deep architectures

Motivation for deep architectures
- Efficiency
- Inspiration from nature

Limitations and recent breakthrough

Work directions

Outline

1. Shallow vs. deep architectures
2. Motivation for deep architectures
   - Efficiency
   - Inspiration from nature
3. Limitations and recent breakthrough
4. Work directions

Introduction to deep architectures and planning of work directions

Telmo Amaral
12th October 2012

Depth of a flow graph

- Learning architectures can be represented as flow graphs.
- Depth: length of longest path from an input to an output.
- E.g., multi-layer perceptrons (MLPs) have depth 2.

Another shallow example

- Support vector machines (SVMs) have depth 2 as well.

Two deeper architectures

- A convolutional neural network (CNN):
- A deep belief network (DBN):

Local generalisation

- Principle exploited by majority of learning algorithms.
- Locally capture the variations: if two inputs are close, the corresponding outputs should also be close.
- But there should be at least as many training examples as ups and downs in the target function.

Non-local generalisation – distributed representations

- Trivial example: number 9 represented as 10010010 10010010 vs. 1 0 0 0 1.
- Potential to capture exponentially more variations for the same number of free parameters – better generalisation.
Deep architectures and work directions

**Non-local generalisation – distributed representations**

- Some families of functions with \( n \) inputs can be compactly represented with \( O(n) \) nodes where depth is \( d \), but require \( O(2^n) \) nodes where depth is \( d-1 \).
- The existence of a compact, deep representation suggests an underlying structure in the function: a shallow, structured representation won’t be able to generalise well.

**The brain has a deep architecture**

- E.g., the visual cortex signals flow through a hierarchy of areas.
- Each area represents information at a higher level of abstraction.
- Representations are sparse, but still exponentially efficient: \% of simultaneously active neurons is a very large number, and inputs are represented by the activation of features that are not mutually exclusive.

**Cognitive processes seem deep**

- Cognitive processes involve the hierarchical organisation of ideas.
- Humans first learn simpler concepts, then compose them into more abstract ones.
- Engineers typically solve complex problems by breaking them up into simpler ones.

**Problems before 2006**

- Deep NNs tended to yield worse training and test errors that shallow NNs (up to 2 orders less).
- Exceptions: CNNs, though tailored for very specific applications.

**Key principles that constituted a breakthrough**

- Pre-training of layers through unsupervised learning: representation learned at each layer can be used as input to the next layer, allowing the extraction of gradually higher abstractions.
- Fine-tuning of all layers through supervised learning (including additional top layers that produce predictions).

**Seminal papers**

- Pre-training achieved by treating layers as restricted Boltzmann machines (RBMs).

**Performance of deep NNs**

- Better than shallow NNs in:
  - vision tasks;
  - natural language processing (NLP) tasks.
- Better than SVMs in vision tasks.
- Allow dataset sizes untreatable by SVMs in NLP tasks.
Some available software
- Rasmus B. Palm’s DeepLearningToolbox, a Matlab deep learning toolbox, including DBNs; stacked and convolutional autoencoders; CNNs and vanilla NNs.
- Hugo Larochelle’s ML Python, a Python machine learning library, which can be used for deep learning research, featuring RBMs and autoencoders.
- Ruslan Salakhutdinov’s and Geoff Hinton’s Matlab code for training a deep autoencoder made of stacked RBMs. Andrej Karpathy’s Matlab code for the same purpose.
- Ruslan Salakhutdinov’s Matlab code for training Deep Boltzmann Machines (DBMs, an alternative to DBNs). Hugo Larochelle’s Matlab code for efficient learning of DBMs (apparently a continuation of the same work).

Main directions of work
- Use existing toolboxes such as ML Python and DeepLearningToolbox to test DNNs with different types of building blocks (e.g., RBMs and auto-associators) and the more traditional risk functionals (e.g., MSE and cross-entropy).
- Develop our own implementation of DNNs, in order to experiment with risk functionals based on more sophisticated principles (e.g., error density, MEE, Z-EDM, and EXP).
B Slides: Introduction to convolutional neural networks
Introduction to convolutional neural networks (CNNs)

Telmo Amaral

22nd February, 2013

Motivation

- Many neurally inspired models emulate the behavior of the visual system, such as:

- CNNs were the only exception to the difficulty in training deep neural networks before the era of unsupervised pre-training.
- CNNs are variants of MLPs inspired from biology.
- Cells within the visual cortex are sensitive to small sub-regions of the visual field, called receptive fields.
- Receptive fields are tiled to cover the whole visual field.

Main differences from traditional MLPs

- Local connectivity
- Shared weights
- Multiple feature maps
- Down-sampling layers

Local / sparse connectivity pattern

- Each hidden unit is connected to a local subset of units on the layer below.
- So, spatially local filters are learnt.
- Stacking layers leads to (non-linear) filters that are increasingly global in relation to the bottom layer.

Skewed weights

- Certain weights are constrained to be identical: the same filter is replicated across each layer of weights.
- A layer of replicated filters determines the values of a layer of hidden units, to form a feature map.

Why interesting?

- Invariance to translation: filter replication allows features to be detected regardless of their position in the visual field.
- Weight sharing greatly reduces the number of free parameters to learn.
- Gradient descent can still be used to learn shared parameters, with a small adaptation:
  - compute partial derivatives of loss function with respect to each connection, as in a conventional MLP with no sharing;
  - add partial derivatives of cost function with respect to weight.

Skewed weights

A feature map $h$ is obtained by convolving filter $W$ with image $x$ on the layer below $(x$ can itself be a feature map), then adding a scalar bias $b$, and applying a non-linearity such as $tanh$:

$$h = tanh(W * x + b)$$

Recalling 1D or 2D convolution:

$$o[m, n] = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} f(u, v) g[u - m, v - n]$$

$$o[m, n] = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} f(u, v) g[u - m, v - n]$$
For a richer representation, each hidden layer is composed of a set of feature maps, \( \{ h^k \}, k = 0..K \).

- \( W^{kl}_{ij} \) is the weight connecting any pixel on feature map \( k \) of a given layer to pixel \( ij \) (in filter coordinates) on feature map \( l \) of the layer below.

- If a given layer has \( K \) feature maps, the layer below has \( L \) feature maps, and all filters cover \( I \times J \) pixels, then there are \( K \times L \times I \times J \) different weights between the two layers.

- Lower layers alternate between convolution and down-sampling.
- Upper layers are fully connected and equivalent to a traditional MLP.
- (With other types of network it could be difficult to learn a reduced representation starting from 10k or 20k pixels, but with the combination of convolution down-sampling that type of learning becomes easier.)

(Image sources)
- http://deeplearning.net/tutorial/lenet.html
C Spreadsheet: Notes on UCI biomedical data sets
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Types</th>
<th>Default Task</th>
<th>Attribute Types</th>
<th># Instances</th>
<th># Attributes</th>
<th># citations between and reported classification accuracies</th>
</tr>
</thead>
<tbody>
<tr>
<td>M. Tuberculosis Genes</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>150</td>
<td>54</td>
<td>1988, 1989, 2005</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>266</td>
<td>9</td>
<td>1988, 2005</td>
</tr>
<tr>
<td>Heart Disease</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>303</td>
<td>73</td>
<td>1988, 2004</td>
</tr>
<tr>
<td>Skin Cancer</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>307</td>
<td>35</td>
<td>1988, 2004</td>
</tr>
<tr>
<td>Soybean (Small)</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>47</td>
<td>35</td>
<td>1987, 2004</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin (Diagnostic)</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>369</td>
<td>31</td>
<td>1993, 2004</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin (Prognostic)</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Regression</td>
<td>699</td>
<td>10</td>
<td>1993, 2004</td>
</tr>
<tr>
<td>Relational</td>
<td></td>
<td></td>
<td></td>
<td>298</td>
<td>34</td>
<td>1993, 2004</td>
</tr>
<tr>
<td>Diabetes</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>4177</td>
<td>8</td>
<td>1995, 2004</td>
</tr>
<tr>
<td>Liver Disorders</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>345</td>
<td>7</td>
<td>1990, 2004</td>
</tr>
<tr>
<td>Horse Colic</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>768</td>
<td>8</td>
<td>1990, 2004</td>
</tr>
<tr>
<td>Parkinsons Telemonitoring</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>1200</td>
<td>22</td>
<td>1997, 2004</td>
</tr>
<tr>
<td>Primary Tumor</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>1404</td>
<td>8</td>
<td>1997, 2004</td>
</tr>
<tr>
<td>Arrhythmia</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>452</td>
<td>271</td>
<td>1998, 2005</td>
</tr>
<tr>
<td>SPET Heart</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>267</td>
<td>2</td>
<td>2001, 2002</td>
</tr>
<tr>
<td>Dermatology</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>366</td>
<td>33</td>
<td>1988, 2004</td>
</tr>
<tr>
<td>Echocardiogram</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>132</td>
<td>12</td>
<td>1989, 2002</td>
</tr>
<tr>
<td>Haberman's Survival</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>306</td>
<td>0</td>
<td>1999, 2003</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>32</td>
<td>56</td>
<td>1992, 2003</td>
</tr>
<tr>
<td>Post-Operative Patient</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>350</td>
<td>8</td>
<td>1993, 2000</td>
</tr>
<tr>
<td>Survival (Series)</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Real</td>
<td>210</td>
<td>5</td>
<td>1997, 2004</td>
</tr>
<tr>
<td>Contraceptive Method Choice</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Real</td>
<td>1473</td>
<td>9</td>
<td>1997, 2004</td>
</tr>
<tr>
<td>SPECT Heart</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>247</td>
<td>1</td>
<td>2001, 2004</td>
</tr>
<tr>
<td>Acute Inflammations</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer</td>
<td>120</td>
<td>6</td>
<td>2009, 2000</td>
</tr>
<tr>
<td>Aridione</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Real</td>
<td>900</td>
<td>1000</td>
<td>2008, 2009</td>
</tr>
<tr>
<td>Breast Tissue</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Real</td>
<td>306</td>
<td>10</td>
<td>2010, 2009</td>
</tr>
<tr>
<td>Cardiacnography</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Real</td>
<td>2126</td>
<td>23</td>
<td>2010, 2000</td>
</tr>
<tr>
<td>Dermoscopy</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>503</td>
<td>10</td>
<td>2010, 2000</td>
</tr>
<tr>
<td>Dorothea</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>1960</td>
<td>10000</td>
<td>2008, 2009</td>
</tr>
<tr>
<td>EPO (Anemia Liver Patient Dataset)</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>583</td>
<td>10</td>
<td>2012, 2008</td>
</tr>
<tr>
<td>KEGG Metabolic Relation Network (Undirected)</td>
<td>Multivariate, Univariate, Text</td>
<td>Classification, Regression, Clustering</td>
<td>Integer, Real</td>
<td>65954</td>
<td>29</td>
<td>2001, 2009</td>
</tr>
<tr>
<td>KEGG Metabolic Relation Network (Directed)</td>
<td>Multivariate, Univariate, Text</td>
<td>Classification, Regression, Clustering</td>
<td>Integer, Real</td>
<td>5144</td>
<td>24</td>
<td>2001, 2009</td>
</tr>
<tr>
<td>Mammographic Mass</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>961</td>
<td>6</td>
<td>2007, 2009</td>
</tr>
<tr>
<td>Shundhurd plant species leaves data</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>1600</td>
<td>64</td>
<td>2009, 2009</td>
</tr>
<tr>
<td>p53 Mutants</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>16772</td>
<td>5409</td>
<td>2010, 2009</td>
</tr>
<tr>
<td>Parkinsons</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>287</td>
<td>23</td>
<td>2008, 2008</td>
</tr>
<tr>
<td>Parkinsons Telemonitoring</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>5875</td>
<td>26</td>
<td>2009, 2009</td>
</tr>
<tr>
<td>Plants</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>21262</td>
<td>70</td>
<td>2008, 2009</td>
</tr>
<tr>
<td>PullChem Bioassay Data</td>
<td>Multivariate, Data-Generator</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>72</td>
<td>1992, 2000</td>
<td></td>
</tr>
<tr>
<td>Quadruped Mammals</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>210</td>
<td>7</td>
<td>2013, 2009</td>
</tr>
<tr>
<td>Sponges</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>76</td>
<td>45</td>
<td>2009, 2009</td>
</tr>
<tr>
<td>Mosquito Acid Signaling Network</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Integer, Real</td>
<td>300</td>
<td>43</td>
<td>2008, 2009</td>
</tr>
<tr>
<td>Coveytype</td>
<td>Multivariate</td>
<td>Classification</td>
<td>Categorical, Integer</td>
<td>581012</td>
<td>54</td>
<td>1998, 2009</td>
</tr>
<tr>
<td>Diabetes</td>
<td>Multivariate, Time-Series</td>
<td>Classification</td>
<td>Categorical, Integer</td>
<td>20</td>
<td>2001, 2009</td>
<td></td>
</tr>
<tr>
<td>E. Coli Genes</td>
<td>Relational</td>
<td></td>
<td></td>
<td>122</td>
<td>4</td>
<td>1999, 2000</td>
</tr>
<tr>
<td>IEG Database</td>
<td>Multivariate, Time-Series</td>
<td>Classification</td>
<td>Categorical, Integer, Real</td>
<td>164800</td>
<td>8</td>
<td>2010, 2009</td>
</tr>
<tr>
<td>Localization Data for person activity</td>
<td>Multivariate, Sequential, Time-Series</td>
<td>Classification</td>
<td>Real</td>
<td>164800</td>
<td>8</td>
<td>2010, 2009</td>
</tr>
</tbody>
</table>
References


