UDC 004.8

A REVIEW OF THE USAGE OF MACHINE LEARNING IN REAL-TIME SYSTEMS

N.H. Abroyan, R.G. Hakobyan

National Polytechnic University of Armenia

In this work, we supply a general overview over the usage of machine learning techniques in real-time systems. At present, there is a tendency of a full or partial replacement of a human's intellectual work by computer programs in every sphere and, for that, there is a need to imitate a human brain i.e. create something like artificial intelligence. On the one hand, machine learning has been quite popular and successfully used in various spheres in recent years. Moreover, the discovery and usage of deep neural networks has immensely increased the efficiency of machine learning usage. On the other hand, as the amount of data greatly increases and changes in quality over time, the usage of real-time systems becomes more and more widespread. So it is quite effective and convenient to use machine learning in real time systems for elaborating a huge amount of newly generated data. Although nowadays there are several machine learning algorithms for classification, regression, clustering etc, their traditional usage as supervised or unsupervised machine learning approach in real-time systems will not be efficient enough because of some nuances that we are going to talk about in this work.

Keywords: machine learning, classification, regression, real-time system, supervised learning, unsupervised learning, semi-supervised learning.

Introduction. Interest in machine learning has grown exponentially over the past two decades, mostly due to a couple underlying factors. First, the expansion of computers, the internet, and the information economy have generated increasing volumes and varieties of data, many of which are unstructured (i.e. they cannot be processed by computers without first requiring human effort to structure them into machine-readable form). At the same time, computational processing has become cheaper and more powerful, enabling to carry out faster and more complex mathematical calculations and increasingly affordable data storage. Machine learning is a subfield of computer science that evolved from the study of pattern recognition and the computational learning theory in artificial intelligence. In 1959 Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed". Machine learning algorithms iteratively learn from data by generalizing their experience into models. These models allow computers to find insights that might be difficult or impossible for humans to find. They learn from previous computations to produce reliable decisions and results.

A system is said to be real-time if the total correctness of an operation depends not only upon its logical correctness, but also upon the time in which it is performed [1]. So

real-time systems have some extra special properties which need some sort of different attitude and usage of machine learning algorithms, and we are going to study it.

Supervised or unsupervised learning. Traditionally, there have been two fundamentally different types of tasks in machine learning [2].

The first task is supervised learning. Let $X = (x_1, \dots, x_n)$ be a set of n examples (or points), where $x_i \in X$ for all $i \in [n] := \{1, \dots, n\}$. Our goal is to learn a mapping from x to y, given a training set made of pairs (x_i, y_i). Here, $y_i \in Y$ are called the labels or targets of the examples x_i. A standard requirement is that the pairs (x_i, y_i) are sampled independently and identically distributed from some distribution, ranging over $X \times Y$. The task is well defined, since a mapping can be evaluated through its predictive performance on the test examples. When the labels are continuous, the task is called regression. When y takes values in a finite set (discrete labels), the task is called classification [2]. A graphical example of classification is presented in Fig. 1. In this figure training examples are introduced in the form of circles and triangles, which means that in the given training set, all examples are already differentiated.

The second is unsupervised learning. Here also, typically, it is assumed that the points are drawn independently and identically distributed from a common distribution on X. It is often convenient to define the $(n \times d)$ matrix $X = (x_i^T)_{i \in [n]}^T$ that contains the data points as its rows. The goal of unsupervised learning is to find an interesting structure in the data X. It has been argued that the problem of unsupervised learning is fundamental in terms of estimating a density, which is likely to have generated X. However, there are also weaker forms of unsupervised learning, such as quantile estimation, clustering, outlier detection, and dimensionality reduction [2].

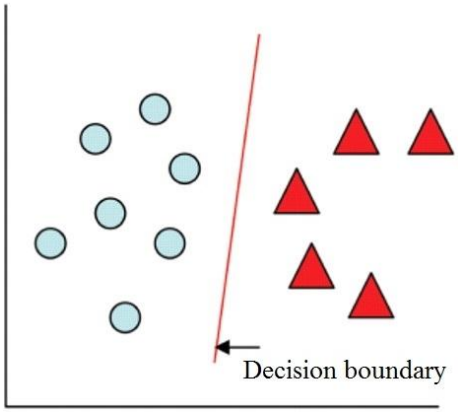

Fig. 1. Graphical presentation of an example of supervised learning

A graphical example of clustering is presented in Fig. 2. In this figure, all training examples are presented in the exact form of circles, which means that initially there is no any difference among them. Here, our task is to determine some regularity and classify them into groups. Thus for instance, the density of circles (i.e. training examples) could serve as a classification regularity in area or space.

For real-time systems, before choosing one of those tasks (supervised or unsupervised), there are two main facts that should be considered.

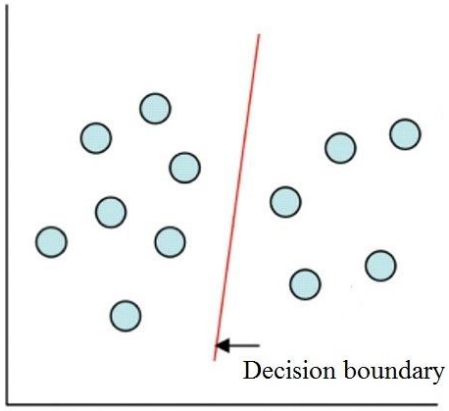

Fig. 2. Graphical presentation of an example of unsupervised learning

- The training model should take into account the recent history when it makes its predictions. A good example is the weather; if it has been sunny and 25 degrees the last two days, it is unlikely that it will be -5 and snow the next day.
- The training model should be updatable. That is, our model should "evolve" based on the real-time data that we receive. A good example might be a retail sales model that remains accurate as the business gets larger.

These two phenomena sound like the same thing, but they are potentially very different. The central question is whether the underlying source generated data is changing. In case of the weather for the previous few days (also considering the historical climate statistics), you can usually predict the weather with a high accuracy for the next day, and your prediction, given the recent history, will be nearly the same from year to year. Here, the characteristics of climate does not change or changes insignificantly (climate also tends to change during many years and a reason for that could be, for instance, the global warming). Eventually, the same model for the last year will work for this year. In the case of the business, the underlying source is changing; the business is growing, and our prediction of the sales, given the previous few days of sales, is probably going to be different from that of the last year. Therefore, the last year's data, when the business was small, is not completely relevant to this year's data, when the business is larger. We need to update the model (or scrap it completely and retrain) to get something that works. The first case, where the prediction is conditioned by history, has not special specific properties to review, supervised learning will work finely. In the second case, where there is a need to update the model or retrain completely, we deal with non-stationary data, and here the approaches are as follows:

- Using the incremental method. These are machine learning algorithms that learn incrementally over the data. That is, the model is updated each time it sees a new

training instance. There are incremental versions of Support Vector Machines and Neural networks. Bayesian Networks can be made to learn incrementally.

- Using periodic re-training with a batch algorithm. Perhaps this is a more straightforward solution. Here, we simply buffer the relevant data and retrain our model by some period.

In case of using supervised learning, if our data is changing in quality over time and we want our predictions to remain accurate, there is a need of doing predictions manually and appending them to our training set. This approach is not an effective one, because over time, there is a need of human intervention. On the other hand, using unsupervised learning is not always acceptable. For instance, in financial data, there should be at least some labeled data in order to do sensible prediction. For that reason, we can choose something between supervised and unsupervised learning, which is called semi-supervised learning.

Semi-supervised learning is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information – but not necessarily for all examples. Often, this information will be the targets associated with some of the examples. In this case, the data set $X=(x_i)_{i\in[n]}$ can be divided into two parts: the points $X_l:=(x_1, ... ,x_l)$, for which labels $Y_l:= (y_1, ... , y_l)$ are provided, and the points $X_u:= (x_{l+1}, ... ,x_{l+u})$, the labels of which are not known [2]. Using semi-supervised learning, there can be a way to predict based on the initial labeled data and always renew the real-time unlabeled data. A graphical example of semi-supervised learning is shown in Fig. 3. In this figure, we have presented both labeled (circles and triangles) and unlabeled (dots) data. This means that some of our training examples are already labeled, but the others are not labeled and there is a need to find regularities among them and classify them into groups.

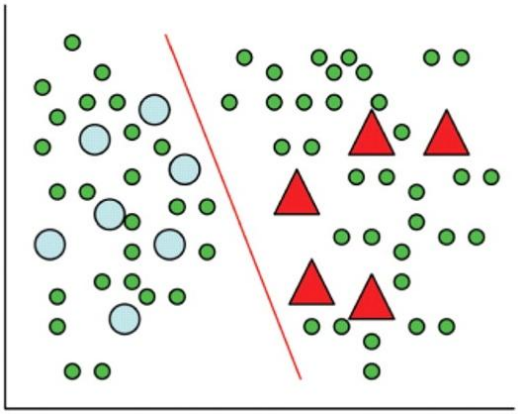

Fig. 3. Graphical presentation of an example of semi-supervised learning

The first usage of semi-supervised learning is known as self-learning. That was the earliest idea about using unlabeled data in classification of self-learning, which is also

known as self-training, self-labeling or decision-directed learning. This is a wrapper-algorithm that repeatedly uses a supervised learning method. It starts by training on the labeled data only. In each step a part of the unlabeled points is labeled according to the current decision function; then the supervised method is retrained, using its own predictions as additional labeled points. This idea can be found in literature (e.g., Scudder (1965); Fralick (1967); Agrawala (1970)) [2].

Thus, one should not be too surprised that for semi-supervised learning to work, certain assumptions will have to be held. One of such most popular assumptions can be formulated as follows. If two points x_1, x_2 are close, so should be the corresponding outputs y_1, y_2. Clearly, without such assumptions, it would never be possible to generalize from a finite training set to a set of possibly infinitely many unseen test cases [2].

Considering this, there is a need of generalization of the smoothness assumption that is useful for semi-supervised learning, which is called "semi-supervised smoothness assumption". While in the supervised case, according to our prior beliefs, the output varies smoothly with the distance, we now also take into account the density of the inputs. The assumption is that the label function is smoother for semi-supervised learning in high-density smoothness assumption regions than in low-density regions. If two points x_1, x_2 in a high-density region are close, so should be the corresponding outputs y_1, y_2. Note that by transitivity, this assumption implies that if two points are linked by a path of high density (e.g., if they belong to the same cluster), their outputs are likely to be close. If, on the other hand, they are separated by a low-density region, their outputs need not be close. Note that the semi-supervised smoothness assumption applies to both regression and classification [2].

In case of using machine learning in real-time systems too, there are two main things to consider:

- Data Horizon: How quickly do we need the most recent datapoint to become part of our model? Does the next point need to modify the model immediately?
- Data Obsolescence: How long does it take the data to become irrelevant to the model? Good examples come from economics; generally, newer data instances are more relevant. However, in some cases data from the same quarter from the previous year are more relevant than the previous quarter of the current year.

Keeping performance high. To improve the prediction accuracy, there is a need for having many mutual exclusive training features. The increase of the features' number leads to a decrease in the machine learning algorithm performance. On the other hand, it is obvious that in real-time, the systems' performance is very important. So there is a need of wisely choosing the features. We need a minimum number of features, which can ensure a high rate of accuracy. To achieve that, we can use deep learning.

Deep learning (more correctly deep machine learning) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures, or otherwise composed of multiple non-linear transformations. During the past several years, the

techniques developed from deep learning research have already been impacting a wide range of areas and aspects of machine learning and artificial intelligence [3]. There are several definitions of deep learning and one of them is that deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction [4].

During the last several years, many universities' and information technology companies' researchers have demonstrated the empirical success of deep learning in different applications of computer vision, phonetic recognition, voice search, conversational speech recognition, speech and image feature coding, semantic utterance classification, natural language understanding, hand-writing recognition, audio processing, information retrieval, robotics etc. [3].

Deep learning algorithms are based on distributed representations. The underlying assumption behind distributed representations is that the observed data are generated by the interactions of many different factors at different levels. Deep learning adds the assumption that these factors are organized into multiple levels, corresponding to different levels of abstraction or composition. The varying numbers of layers and layer sizes can be used to provide different amounts of abstraction [5]. Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones. These architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out the features useful for learning [5]. Many deep learning algorithms are applied on unlabeled data (which is usually more abundant than labeled data), making this an important benefit of these algorithms. The deep belief network is an example of a deep structure that can be trained in an unsupervised manner [5]. One of the most popular algorithms of deep learning is deep neural networks. An example of deep neural network is presented in Fig. 4. Another method of keeping performance high is parallelization. For increasing a program's performance, there is a need to parallelize that program, especially the algorithms that are used in that program [6]. So, high performance can also be achieved by parallelizing the known machine learning algorithms or evaluating new ones by using parallelization methods. There could be different approaches to parallelizing of machine learning algorithms. Both SIMD (single instruction – multiple data) and MIMD (multiple instruction – multiple data) parallelization types may work here. One of the MIMD parallelization ways can be the modification of the algorithm in the way of those operations in loops, and frequently executed instructions satisfy Bernstein's conditions. Every iteration can be performed independently from the previous one. In this case parallelization can be done through a pipeline mechanism [6]. Another way of MIMD parallelization is parallelization through multithreading. Threading provides a mechanism for programmers to divide their programs into more or less independent tasks with the property that when one thread is blocked another

thread can be run [6].

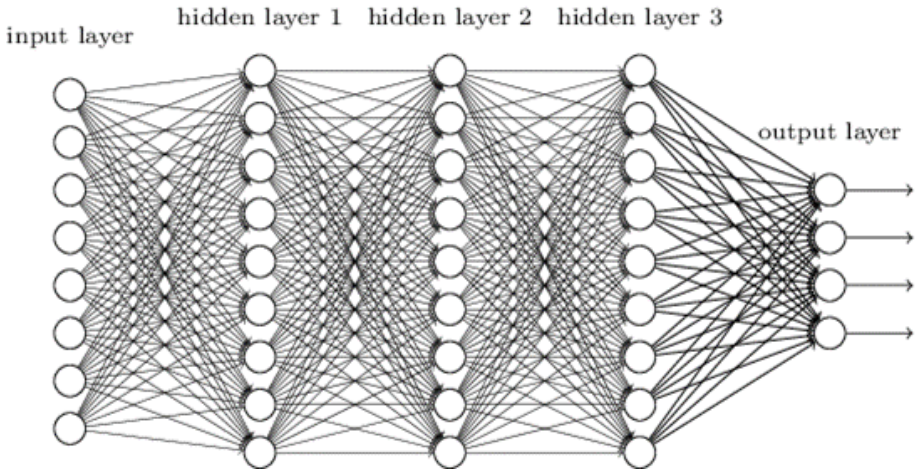

Fig. 4. An example of deep neural network

SIMD parallelization also would be effective for machine learning algorithms since many algorithms involve operations with matrixes. Many operations with matrixes can be parallelized quite well and in many cases it is done through GPU (graphical processing unit). Today, parallel GPUs have begun making computational inroads against the CPU (central processing unit), and a subfield of research, dubbed GPU Computing or GPGPU for General Purpose Computing on GPU, has found its way into fields like machine learning, oil exploration, image processing, linear algebra, statistics, 3D reconstruction, stock options pricing determination etc. Their highly parallel structure makes them more effective than general-purpose CPUs for SIMD parallelizations.

Conclusion. In this work, we introduced a general overview on the usage of machine learning in real-time systems. We showed that for better result it would be more effective to use the semi-supervised learning method. For real-time systems there is a need to take into account such factors as model adaptiveness, data change over time, data obsolescence, etc. For ensuring high performance in real-time systems, there is a need for choosing features by wisely using deep learning. Also parallelization of machine learning algorithms is also an acceptable way of keeping performance high.

References

1. **Shin K.G., Ramanathan P.** Real-time computing: a new discipline of computer science and engineering // Proceedings of the IEEE.- Jan. 1994.- 82 (1).- P. 6–24.

2. **Olivier Chapelle, Bernhard Scholkopf, Alexander Zien.** Semi-Supervised Learning.- Cambridge, Massachusetts: The MIT Press, 2006.- 528 p.
3. **Deng L., Yu D.** Deep Learning: Methods and Applications // Foundations and Trends in Signal Processing.- 2014.- 7.- P. 3–4.
4. **Song H.A., Lee S.Y.** Hierarchical Representation Using NMF // Neural Information Processing. Lectures Notes in Computer Sciences 8226.- Springer, Berlin Heidelberg, 2013.- P. 466–473.
5. **Bengio, Y., Courville A., Vincent P.** Representation Learning: A Review and New Perspectives // IEEE Transactions on Pattern Analysis and Machine Intelligence.- 2013.- 35 (8).- P. 1798–1828.
6. **Abroyan N.H., Hakobyan R.G.** Parallelization of Sorting Algorithms // Computer Science and Information Technologies.- 2015.- P. 201-205.

Received on 07.04.2016.
Accepted for publication on 20.05.2016.

ԻՐԱԿԱՆ ԺԱՄԱՆԱԿԱՅԻՆ ՀԱՄԱԿԱՐԳԵՐՈՒՄ ՄԵՔԵՆԱՅԱԿԱՆ ՈՒՍՈՒՑՄԱՆ ՕԳՏԱԳՈՐԾՄԱՆ ԸՆԴՀԱՆՈՒՐ ԱԿՆԱՐԿ

Ն.Հ. Աբրոյան, Ռ.Գ. Հակոբյան

Ներկայացվում է իրական ժամանակային համակարգերում մեքենայական ուսուցման օգտագործման ընդհանուր բնութագիրը: Այսօր գրեթե բոլոր ոլորտներում նկատվում է մարդու ինտելեկտուալ աշխատանքը համակարգչային ծրագրային միջոցներով լրիվ կամ մասնակիորեն փոխարինելու միտում: Դրա համար անհրաժեշտ է նմանակել մարդու ուղեղի աշխատանքը, այսինքն՝ ստեղծել արհեստական բանականությանը նման բան: Մի կողմից՝ վերջին մի քանի տարիների ընթացքում մեքենայական ուսուցումը շատ տարածված է եղել և հաջողությամբ օգտագործվել այդ նպատակով: Ավելին, նեյրոնային խոր ցանցերի հայտնաբերումից և կիրառումից հետո մեքենայական ուսուցման արդյունավետությունը մեծապես աճել է: Մյուս կողմից, քանի որ տվյալների քանակը գնալով աճում է, և դրանք կրում են որակական փոփոխություններ, ուստի իրական ժամանակային համակարգերի օգտագործումն ավելի լայն տարածում է գտնում: Այսպիսով, նոր ստացվող տվյալների մշակման համար մեքենայական ուսուցման օգտագործումն իրական ժամանակային համակարգերում լինում է բավականին արդյունավետ և նպատակահարմար: Չնայած նրան, որ ներկայումս կան մեքենայական ուսուցման մի քանի ալգորիթմներ՝ նախատեսված դասակարգման, նվազարկման, կլաստերացման համար, սակայն դրանց ավանդական օգտագործումը, որպես վերահսկվող կամ չվերահսկվող մեքենայական ուսուցման մոտեցում, իրական ժամանակային համակարգերում չի կարող լինել բավարար արդյունավետ՝ որոշակի առանձնահատկությունների պատճառով, որոնք դիտարկվում են այս աշխատանքում:

Առանցքային բառեր. մեքենայական ուսուցում, իրական ժամանակային համակարգ, վերահսկվող ուսուցում, չվերահսկվող ուսուցում, կիսավերահսկվող ուսուցում:

ОБЩИЙ ОБЗОР ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ

Н.О. Аброян, Р.Г. Акопян

Рассматривается общая характеристика использования машинного обучения в системах реального времени. Люди стараются посредством компьютерных программ полностью или частично имитировать работу человеческого мозга – создать искусственный интеллект. С одной стороны, в последние годы машинное обучение получило широкое распространение и удачно применялось для этой цели. Более того, использование глубоких нейронных сетей привело к повышению производительности машинного обучения. С другой стороны, так как количество данных со временем увеличивается и они претерпевают качественные изменения, применение систем реального времени становится актуальным и распространенным. Таким образом, применение машинного обучения в системах реального времени для обработки получаемых данных становится достаточно продуктивным и удобным. Несмотря на то, что в настоящее время существует ряд алгоритмов машинного обучения для классификации, регрессии, кластеризации и т.п., их классическое использование в качестве контролируемого или неконтролируемого машинного обучения неэффективно в системах реального времени ввиду некоторых особенностей, которые приведены в этой работе.

Ключевые слова: машинное обучение, система реального времени, контролируемое обучение, неконтролируемое обучение, полуконтролируемое обучение**.**