Received July 1, 2015, accepted July 17, 2015, date of publication August 21, 2015, date of current version September 9, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2471178

Cognition-Based Networks: A New Perspective on Network Optimization Using Learning and Distributed Intelligence

MICHELE ZORZI[1], (Fellow, IEEE), ANDREA ZANELLA[1], (Senior Member, IEEE), ALBERTO TESTOLIN[2], MICHELE DE FILIPPO DE GRAZIA[2], AND MARCO ZORZI[2,3]

[1]Department of Information Engineering, University of Padua, Padua 35131, Italy
[2]Department of General Psychology, University of Padua, Padua 35131, Italy
[3]IRCCS San Camillo Foundation, Venice-Lido 30126, Italy

Corresponding author: Michele Zorzi (zorzi@dei.unipd.it)

This work was supported by Fondazione CaRiPaRo through the Project A Novel Approach to Wireless Networking based on Cognitive Science and Distributed Intelligence within the framework "Progetti di Eccellenza 2012."

ABSTRACT In response to the new challenges in the design and operation of communication networks, and taking inspiration from how living beings deal with complexity and scalability, in this paper we introduce an innovative system concept called COgnition-BAsed NETworkS (COBANETS). The proposed approach develops around the systematic application of advanced machine learning techniques and, in particular, unsupervised deep learning and probabilistic generative models for system-wide learning, modeling, optimization, and data representation. Moreover, in COBANETS, we propose to combine this learning architecture with the emerging network virtualization paradigms, which make it possible to actuate automatic optimization and reconfiguration strategies at the system level, thus fully unleashing the potential of the learning approach. Compared with the past and current research efforts in this area, the technical approach outlined in this paper is deeply interdisciplinary and more comprehensive, calling for the synergic combination of expertise of computer scientists, communications and networking engineers, and cognitive scientists, with the ultimate aim of breaking new ground through a profound rethinking of how the modern understanding of cognition can be used in the management and optimization of telecommunication networks.

INDEX TERMS Cognitive networks, deep learning, hierarchical generative models, optimization.

I. INTRODUCTION

Traditionally, the ISO/OSI system architecture has been the cornerstone of network design, due to its modularity that enables the optimization of individual sets of functionalities and guarantees scalability. While such an ordered and simple structure has successfully served the needs of the Internet users up to now, the always increasing number and variety of services deployed over the network, and the effort of the Internet service providers to continuously improve the quality of the services offered to their customers, are challenging the current network architecture, which suffers from ossification in the underlying infrastructure and does not appear capable of scaling up with the growing complexity of the upcoming communication scenarios.

This trend is indeed expected to accelerate in future fifth-generation (5G) mobile systems that, though not yet fully specified, will certainly pose extreme challenges in terms of *heterogeneity* of both device capabilities and traffic; *scalability* in terms of number of functions and parameters within a single node, and of number of nodes in the system; *efficient use of the resources*, such as bandwidth and energy; and effective management of *Quality of Experience* (QoE) [1]–[3].

A few examples that illustrate well the difficulties that need to be solved by future network technologies are the following.

a) Massive Access: Current cellular networks have been designed to serve a small number of users per cell at high bit rate, particularly in the downlink. This paradigm is being challenged by massive access scenarios, characterized by a very large number of devices in each cell, and prevalent uplink traffic. An example is the machine-to-machine scenario, which requires the base stations to guarantee access to a very large number of machine-type devices that sporadically transmit very short packets [4], [5]. Another example is the simultaneous upload of pictures or videos taken by people attending a public event and wishing to share the moment with others, using social media or cloud services. With current technologies and protocols, massive access will generate overwhelming signaling overhead and cause service outages.

b) *Mobility in HetNets:* The demand for high-speed connectivity is also expected to dramatically increase in the near future. A promising approach to tackle this problem is to place pico and/or femto base stations within a macro cell, a paradigm known as Heterogeneous Networks (HetNets). While HetNets can provide better coverage and higher connection speed to users, the higher density of base stations with different coverage areas and backhaul connectivity raises new technical challenges for both the mobile users and the network operators, which include the design of efficient handover policies, resource allocation/reservation schemes, service migration strategies, and so on [6], [7].

c) *Concurrent Multimedia Flows:* The popularity of smartphones and tablets has generated a growing demand for mobile multimedia content. A wireless access point, hence, may be required to serve simultaneously multiple video connections with dynamic rates via scalable coding and different quality requirements, data transactions of various kinds, several file transfers and possibly some voice connections and audio streaming. To provide the maximum QoE to the end user while using resources efficiently, the new generations of communication systems have to differentiate the services not only by class of application, but even per flow within each class, thus providing *content-based service optimization* [8], [9].

d) *Self-Organizing Networks:* The ever increasing topological and functional complexity of networks, in particular cellular networks, calls for automated mechanisms for handling some fundamental networking operations, in particular the *Self-Configuration* of system parameters when network elements are added or removed from the system (e.g., IP addresses, neighbours list, frequency allocation, propagation channel model, antenna tilt); the *Self-Optimization* of the network performance when the system is operational (coverage, capacity, energy consumption); and the *Self-Healing* of the system, i.e., the recovery of the network functionalities and services after faults or failures of some components [10]–[12]. These capabilities, collectively referred to as *Self-X*, are fundamental in future wireless networks to ensure high quality of the mobile services while reducing capital expenditures (CAPEX) and operational expenditures (OPEX) of the radio access network. However, the realization of Self-X mechanisms that are scalable and fully autonomic is still an open research challenge.

There is hence a need to manage more efficiently the available resources, taking into account the vast variety of traffic features and of their performance requirements, as well as the extreme heterogeneity of device capabilities and of communications technologies.

Recent trends in networking have shown that crossing the boundaries of the layering architecture can lead to much higher efficiency than respecting the orthodox layered model, and cross-layer approaches have been proposed and shown to provide very good results, especially in resource-challenged environments [13], [14], although one should always be careful when and how to use cross-layer techniques [15]. Furthermore, newly emerged networking paradigms and capabilities, including Software Defined Networking (SDN) and Network Function Virtualization (NFV) [16], open up unprecedented opportunities towards new systems and applications.

However, greater flexibility in network management implies more degrees of freedom in the setting of the network parameters and, consequently, a much bigger optimization space, which will call for more advanced (and complex) optimization strategies. If in addition we try to use the abundance of sensory data already present (or easily obtainable) in networks and devices, the dimensionality of the problem quickly becomes very large, making traditional approaches insufficient and calling for disruptive paradigms.

As a response to these challenges, and inspired by how the nervous system of living beings deals with complexity and scalability, we introduce the new concept of COgnition-BAsed NETworkS (COBANETS), intelligent communication systems which are much more than just a collection of smart or cognitive nodes, and instead include a network-wide cognitive infrastructure for learning, modeling and optimization, and data representation. Advanced machine learning techniques, in particular unsupervised deep learning and probabilistic generative models (suitable for scenarios with massive unlabeled data, such as those previously described), along with network optimization at all layers of the protocol stack and corresponding reconfiguration through SDN tools, are the key building blocks of our approach, which significantly departs from state-of-the-art solutions in cognitive networking.

The conceptual design and practical implementation of cognition-based networks have been elusive for years. In this paper, we claim that this vision is now at hand, because of the following key enabling factors:

i) The recent advances in machine learning and cognitive science, with the development of deep unsupervised learning networks which have been successfully applied to solve extremely difficult classification problems.

ii) The impressive performance improvements of processing units, with the commercial diffusion of parallel computing architectures that are particularly suitable for running very-large-scale deep learning models, such as those based on powerful Graphics Processing Units (GPUs).

iii) The rapidly growing popularity of new networking paradigms, such as SDN and NFV, that have the potential to overcome the ossification in the underlying infrastructure of the Internet and to enable a more dynamic and flexible management of the network, thus making it possible to actuate network-wide optimization strategies.

Despite the conjunction of these favorable factors, building the grand vision of a learning network, able to adapt to changing conditions and to serve multiple communication services, still remains a great challenge, which requires pushing the research significantly beyond the current state-of-the-art. The path however is littered with a number of roadblocks, among which we identify the following as the most critical:

i) No attempt has been made so far to design and implement a *practical* network-wide learning framework for network optimization.
ii) Little is known about what can be fundamentally achieved using such a learning approach for network optimization.
iii) Although the excellent performance of deep networks in many challenging machine learning tasks provides strong evidence on their suitability for learning and adaptation, their application to network optimization has not yet been widely explored.
iv) A large-scale testbed that incorporates deep learning concepts and uses them for network optimization through SDN tools does not yet exist.

We remark that the availability of large datasets of experimental data, and the possibility of testing the proposed algorithms in real scenarios, are essential elements for the proper design of a cognition-based network. Indeed, the proposed approach gravitates around the possibility of turning the complexity of the system into an advantage, rather than an obstacle, by exploiting the inner capabilities of deep learning architectures to capture and discriminate hidden features of the complex multidimensional signals that are observed in real scenarios, whose richness cannot be fully replicated by any mathematical or even simulation model.

In the rest of this position paper, we address these roadblocks and describe our vision on how to move forward towards the practical realization of the COBANETS concept. The paper is organized as follows. In Sec. II we will quickly survey the recent history of cognitive networking and of machine learning methods applied to network optimization. Furthermore, the section offers a brief introduction of unsupervised learning techniques, which are at the core of the COBANETS framework. The reasons behind this choice are discussed in Sec. III, which describes in more details the COBANETS concept, and the specific properties of Generative Deep Neural Networks that make them particularly attractive as the enabling elements of a cognitive architecture. In Sec. IV, as a concrete example, we illustrate the potential of the proposed concept by describing some preliminary studies where we applied deep learning neural networks to gain richer context information on video flows, thus making it possible to design resource management algorithms capable of maximizing the connection performance while guaranteeing high QoE to the final users. Sec. V discusses the most relevant research challenges opened by the proposed approach, and finally Sec. VI concludes the paper with a short summary of the study and some final considerations.

II. STATE OF THE ART

In order to set the stage for the description of the proposed approach, we first provide a brief overview of the recent history of cognitive networking and machine learning approaches, with particular focus on deep learning and generative models which are the basic building blocks of our approach.

A. COGNITIVE RADIOS AND NETWORKS

Cognition as a way to deal with the challenges of future networks has been suggested several times in the past. The pioneering work in [17] and [18] proposed to apply cognition to special communications devices, called cognitive radios, able to learn and adapt to the environment, with the goal of providing reliable communication and efficient utilization of the radio spectrum. This concept of adaptability at the physical layer was later extended to a paradigm called cognitive radio network [19], where the spectrum owned by the so-called primary users (i.e., the legitimate users of a licensed band) is also exploited by secondary cognitive radios to communicate while coexisting with the primary users.

A popular application of this concept is dynamic spectrum access, motivated by the observation that licensed bands are heavily underutilized and there is room for their opportunistic usage [20]. Several other works in this area have dealt more recently with networking aspects (such as medium access and routing protocol design) based on this primary/secondary paradigm, for example in the context of Cognitive Radio Ad Hoc Networks [21], where a population of secondary users in ad hoc mode opportunistically coexist with primary users in a cellular environment. In all cases, primary users are assumed to use legacy technology and to be unaware of the cognitive operation, whereas secondary users are equipped with a frequency-agile transceiver and have the intelligence necessary to perform spectrum sensing, dynamic spectrum access, as well as possibly other advanced functionalities.

Even though in most of the existing papers on cognitive radio and cognitive radio networks the ''cognitive'' aspects are focused on sensing, channel selection, and adaptive communications, both Mitola [17] and Haykin [18] actually gave a broader definition of cognitive radio, which includes aspects that relate to the true essence of cognition, such as intelligent observation, learning, and decision-making. From this viewpoint, the existing studies that have addressed these cognition issues in networks, though certainly interesting and valuable in their own right, have only scratched the surface of what promises to be a rich research area with high potential for innovation [22]. In this direction, cognitive networks for wireless systems [23]–[26] and the Knowledge Plane (KP) for the Internet [27] have been proposed as new paradigms in which the concepts of cognition, learning and adaptability are applied in an end-to-end fashion to the whole protocol stack.

The vision of the KP given in [27] is quite abstract, and more conceptual than practical, but contributes to better shape the general idea of cognitive network by defining the functionalities and properties that are expected from such a system. The KP is seen as a separate construct, parallel to the existing data and management planes, that shall become the depository of a distributed and system-wide intelligence, which is here intended as the capability to "abstract and isolate high level goals from low level actions, to integrate and act on imperfect and conflicting information, and to learn from past actions to improve future performance" [27, p. 3]. In [27], the authors also attempt a speculative characterization of the KP architecture in functional terms, claiming that the KP structure shall be: *distributed*, in order to somehow reflect and respect the structure and dynamics of the current Internet, which is the result of the combination of a multitude of networks of the most disparate sizes, interconnected in a loosely hierarchical topology; and *compositional*, i.e., composed by modules of different sizes that can combine into more complex modules or decompose themselves in simpler entities, as appropriate.

In order to be able to achieve its global monitoring and control objective, the KP architecture shall make it possible to access *sensors* distributed across the system, which shall produce observations of the current status of the system that are connected with relevant performance indices, and to operate on *actuators*, that implement control actions according to certain policies. This concept is taken up in [24]–[26], where the authors identify the *Software Adaptable Network* (SAN) as a fundamental building block of a cognitive network. Practically, the concept of SAN is that of a network able to modify one or several layers of the protocol stack in its nodes according to the adaptation strategies decided by a cognition entity, with the aim of achieving end-to-end objectives. The interaction with the SAN is supposed to be provided through an Application Programming Interface (API), which is used by the cognitive engine to both act on the modifiable elements of the network and collect reports on the network status, which are then fed to the learning process, closing the cognition cycle.

A somewhat similar architecture is proposed in [28] and [29], where the authors define a framework for a *cognitive resource manager* (CRM) that aims at enabling autonomic optimization of the whole communication stack of a wireless system by interacting with and dynamically adapting the layered protocol stack. The CRM is described as a sort of "micro kernel" that can be enhanced with additional software modules implementing different types of algorithms for data management, representation, and learning, such as neural networks, Bayesian reasoning, genetic algorithms, and so on. The interaction of the CRM with the different and heterogeneous link layer technologies is realized by means of the Unified Link Layer API (ULLA), first proposed in the European project GOLLUM.[1] The CRM framework hence shares the general structure and purpose of the Knowledge Plane, but with the extra requirement of real-time functioning, which makes its actual implementation much more complicated.

B. THE ROLE OF MACHINE LEARNING

Recent research that makes explicit use of Machine Learning (ML) tools in a networking context includes [30] that proposes a framework for cognitive inference in the presence of partial and noisy observations, and [31] that introduces the concept of "docitive networks," where nodes effectively teach other nodes for improved performance, whereas a recent survey on learning techniques for cognitive radio networks is provided in [32]. A generic architecture model is formalized in [33], which focuses on the learning engine of cognitive networks as a way to improve capacity maximization and dynamic spectrum access.

An example of application to data routing and clustering in sensor and ad hoc networks can be found in [34], where ML approaches are surveyed and compared. In [35] the authors study the optimization of routing and scheduling through reinforcement learning, showing that good performance can only be achieved by using a joint approach that simultaneously acts on the route selection and the scheduling policy applied by the nodes along the path.

Other researchers have used supervised learning methods to address various classification tasks. Decision trees have been used for system failure diagnosis [36], and for file type classification [37]. In [38] the authors survey a number of learning strategies able to classify IP packets and identify the application that generates network traffic, using supervised, unsupervised and hybrid approaches. In [39], the authors show how Bayesian techniques can greatly improve the accuracy when analyzing Internet traffic based on packet header-derived discriminators, whereas [40], [41] evaluate several different algorithms working on empirical data, in terms of accuracy and complexity.

Reference [42] proposes a computer program able to automatically design end-to-end congestion-control algorithms based on human-supplied specifications, such as throughput maximization and/or delay minimization, and to outperform the best-known human-designed techniques in many different scenarios. This type of approach shows that a unified framework endowed with enough intelligence has the potential to do better than customized solutions, and could be complemented with the deep learning approach we propose. Another approach to improve TCP performance based on both prior file transfer history and measurements of simple path properties is presented in [43].

Similarly, in [44] the authors propose a predictive model of Internet video QoE based on a data-driven approach and on the analysis of metric interdependencies and complex relationships by means of decision trees.

[1]http://www.ist-gollum.org

C. SELF-ORGANIZED NETWORKS

In the last years, growing attention has been devoted to the development of machine learning mechanisms to provide *Self-X* capabilities in future wireless networks [45]. The general idea underlying the Self-Organization concept is to replicate in man-made systems the capabilities of some biological systems (e.g., flocks of cranes, schools of fish, swarms of insects) to autonomously adapt to the dynamics of the surrounding environment, in order to achieve a desired objective. Although a formal and widely accepted definition of Self-Organization is still lacking, in [46] the authors have attempted a reasoned synthesis of the different interpretations of this concept that have been proposed in the literature, defining self-organization in a system as an adaptive and autonomous functionality that is ''scalable, stable and agile enough to maintain its desired objective(s) in the face of all potential dynamics in its operating environment'' [46, p. 339]. Therefore, self-organized networks shall not just be able to *autonomously adapt* to changing conditions, but also to *learn* based on experience. Machine learning mechanisms naturally take a primary role in this context for what concerns the learning aspect, but can also play a significant part in network optimization, in particular when combined with reinforcement learning [47].

As an example, reinforcement learning, in combination with fuzzy logic, is proposed in [48] and [49] to optimize the downtilt of the antennas of a Long-Term Evolution (LTE) base station, in order to achieve the self-configuration, self-optimization, and self-healing functionalities. The scheme is proved to outperform other heuristic algorithms and to be robust to environmental noise. In [50], the authors present different learning strategies to maximize the coverage and optimize the capacity of a wireless cellular network, using a Fuzzy Q-Learning based solution. They observed that the stability of the proposed solution (which, according to [46], is a fundamental requirement for a truly self-organizing system) depends upon the number of agents (base stations in this case) that simultaneously try to adjust their antennas downtilt: when a single agent at a time can adjust its parameters, the stability and performance of the whole system are eventually maximized, but the convergence time can become very long; conversely, when all agents can take actions simultaneously, the convergence process is sped up, but at the cost of larger oscillations and reduced performance gains. Hence, a clustering mechanism that combines the benefits of both strategies is proposed to find a satisfactory tradeoff between the two approaches.

Despite the appreciable body of literature on Self-X mechanisms, practical solutions that exhibit the scalability, stability and agility requirements for proper self-organization of upcoming wireless systems are still missing.

D. UNSUPERVISED LEARNING

The above applications of ML to networking problems are meant to solve specific issues, and make use of either supervised learning (in which correct inputs/outputs are explicitly presented and/or suboptimal actions are explicitly corrected) or some form of reinforcement learning (where an agent receives a reward based on the action it chooses, trying to find a balance between exploration and exploitation) [51]. These learning approaches are typically effective in the presence of a well-defined goal (as in the above examples) and of a tight action-reward feedback loop through which the agent is informed about the goodness of a certain action, which is thereby learned for future use.

However, there is a broad range of situations where learning is fully unsupervised, and its only objective is that of *building rich internal representations of the sensory world*. For instance, although humans often learn through supervision (e.g., by teacher instruction) or reinforcement (e.g., by understanding the effects of actions), we also continuously perform unsupervised learning, in which we receive a variety of stimuli from the environment and we gradually develop a worldview that constitutes the background upon which we build our cognitive activities [52], [53].

Importantly, once the system has developed expressive internal representations of the data, supervised tasks can be more easily carried out by introducing additional modules, which directly operate on such high-level representations and can yield very high classification performance [54], [55]. This framework is often associated with the notion of representation learning [56], which is in turn applied to implement effective transfer learning algorithms [57], where abstract knowledge extracted from one domain is readily re-used to solve many different supervised tasks. Conversely, knowledge obtained through supervised learning is necessarily goal-specific and does not easily transfer to novel tasks. Finally, when applied to real-world problems, unsupervised learning can exploit the huge amount of data that comes without any label to build rich internal representations.

It is worth noting that a pure supervised learning approach applied to a sufficiently complex problem may even fail to achieve adequate performance, because it might not be able to build an appropriate representation of the data distribution that can be effectively used to support discriminative tasks (see [58] for an example of this behavior in the context of an image classification problem consisting in estimating the numerosity of a visual set).

E. GENERATIVE MODELS AND DEEP LEARNING ARCHITECTURES

In the context of unsupervised learning, a generative model is a probabilistic model of how the underlying physical properties of the world cause sensory data, and is built by extracting a useful set of features from the input space that allows to accurately reconstruct the input information and to support similarity judgments among different (and possibly novel) patterns [59]. In other words, the learner estimates a model representing the probability distribution of the data and uses this knowledge to test hypotheses about how to best interpret the (possibly noisy and unreliable) information coming from the external environment. The learning goal is therefore to discover the latent structure of the data distribution, wich can be accomplished without any external supervision or reward since the learning signal is provided by the discrepancy between the input pattern and the corresponding reconstruction performed by the model.

Notably, generative models can be efficiently implemented in stochastic recurrent neural networks, such as the Boltzmann Machine [60]. Boltzmann Machines have been recently formalized within the framework of probabilistic graphical models [61], thereby allowing the exploitation of powerful analytical and computational tools to improve learning algorithms [62], [63]. One version known as Restricted Boltzmann Machine (RBM), which is schematically represented in Fig. 1, is particularly attractive because training can be very fast using the contrastive divergence learning algorithm [62]. Learning involves the iteration between a positive and a negative phase. During the positive phase (*inference*), visible units are clamped to the values of the data observed in the training set. The network then propagates activations to hidden units, according to the weights of the connections. The entire vector of hidden unit activations constitutes an *internal representation* of the pattern observed in the visible units. During the negative phase, instead, hidden units are fixed and activations are propagated backward to the visible units in a similar fashion, in order to accurately *reconstruct* the original input vector. The objective of the (unsupervised) learning process is to find a good set of weights to obtain accurate reconstructions of the input patterns.

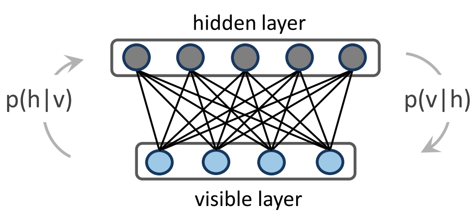


FIGURE 1. The structure of a Restricted Boltzmann Machine.

These advances have made it possible for the very first time to effectively stack together several basic modules, like the RBM, in order to learn multi-layer architectures [64], which paved the way for the introduction of a variety of so-called *deep learning* models [65], [66] that are now the focus of many academic and industrial research groups.

Deep learning architectures make it possible to efficiently encode complex probability distributions using multiple levels of representation [67], where basic features extracted at lower levels are successively combined to form more complex, high-level features, thereby providing a practical way to build *hierarchical generative models* from the training data. Therefore, unlike ''shallow'' architectures, deep learning systems exploit feature reuse in order to process information through multiple stages of transformation and representation, which seems to be a strategy adopted also by the primate cerebral cortex [68]. Note that this approach is currently the state-of-the-art in cognitive science modeling [58], [67].

In addition, the introduction of extremely powerful parallel computing architectures, such as the CUDA framework [69], now makes it possible to efficiently build very-large-scale deep learning models containing millions of connection weights [70], [71], which can be trained in an unsupervised way using the huge number of patterns contained in modern digital datasets [72].

By exploiting these advances, deep learning algorithms led to impressive performance gains in many difficult ML tasks, such as object recognition [73], speech processing [74], natural language modeling [75], predicting the activity of potential drug molecules [76] and studying the effects of mutations in DNA [77], just to name a few.

Although most recent research on deep learning has focused on the use of supervised techniques (see [78] for a review), unsupervised (i.e., generative) deep learning remains the only choice when data cannot be easily labeled (as in our scenarios of interest), and represents a key research frontier for the future. Indeed, although supervised algorithms such as error back-propagation represent a powerful way to train multi-layer architectures, it is evident that ''we discover the structure of the world by observing it, not by being told the name of every object'' [78, p. 442]. Moreover, generative neural networks have also been recently extended to model more complex sequential data, where the temporal dimension can be efficiently processed by exploiting distributed representations of the contextual information [79]–[81].

Finally, although the COBANET architecture described in this paper is mostly concerned with the problem of building useful representations of the networking environment, its overall performance is also heavily dependent on the set of control policies that are used to govern the system. Although there exist several ''engineered'' solutions to define which actions should be performed at each timestep in order to optimize the behavior of the system, human-programmed approaches become infeasible in many complex scenarios, and better performance can be obtained by exploiting learning-based methods. Learning optimal control strategies is usually achieved by means of trial-and-error procedures, which allow to learn how specific actions affect the system behavior based on feedback received from the environment. Interestingly, some recent research has shown how deep learning approaches might be effectively combined with such reinforcement learning strategies to build control systems approaching human-level performance [82]. In particular, the rich and abstract representations created by a deep network were given as input to a control module implementing a variant of the Q-learning algorithm, thereby allowing the agent to learn a meaningful mapping between observation and actions. Such tight synergy between ''perceptual'' and ''control'' modules could greatly improve the overall behavior of a cognitive system, for example providing a principled way to shape the internal representation of the deep network to give prominence to the environmental features that are of most value for the control task.

III. THE COBANETS CONCEPT

From the above review of the related literature, we learn that, although the powerful paradigm of bringing cognitive processes into networks has been suggested in various forms in the past fifteen years or so, the idea has not yet found its way into a comprehensive and practical design, and even less so to a large-scale application in real systems. We believe the main reasons for this are to be found in the lack of a sufficiently general tool to implement intelligence in a scalable way, and the lack of actionable schemes able to effectively implement decisions in complex systems, possibly combined with the lack of a broader view of the cognition-based system beyond the ad hoc application of specific ML techniques to a limited set of functionalities.

New paradigms that have emerged only very recently in the areas of machine learning and cognitive science (deep networks and generative models, i.e., the intelligence) and networking (software defined networks, i.e., the actionable schemes) make this the right time for a disruptive change of paradigm and for realizing the ambition to bring cognitive networking techniques to the next level, by moving from the limited scope of a set of specific applications towards the development of a comprehensive framework in which large-scale unsupervised learning is the stepping stone and a key enabler of a wide range of optimization techniques for the whole network, as well as for its individual components.

According to these premises, the COBANETS concept focuses on generative models, and in particular Generative Deep Neural Networks (GDDNs), and network virtualization paradigms as key enabling factors for the development of a groundbreaking novel approach to network optimization.

In the remainder of this section, we describe in more details the characteristics that make GDNN extremely appealing in this context and, then, we give a broad description of the system architecture we envision. In Sec. IV, we will substantiate our arguments by describing a possible implementation of a subset of the COBANETS principles in a practical scenario.

A. GDNN: GENERATIVE DEEP NEURAL NETWORKS

At an intuitive level, the unsupervised training of a deep neural network builds an inner model of unlabeled input signals that is independent of any specific concept defined by the user. Unlike in typical supervised learning tasks (i.e., classification or regression), the system is not forced to learn an appropriate output response for a given input pattern. Instead, here the learning objective is to extract a useful set of features from the input space, which allow to accurately represent and reconstruct the input information and to support similarity judgments among different (and possibly novel) patterns. Any input signal applied to the visible layer of a deep neural network is indeed mapped to a certain configuration of the neurons in the deeper layer (features), whose joint statistical distribution can capture the highly non-linear and complex interactions between the observed examples. The values taken by these higher-level hidden units provide a more abstract representation of the input data, according to the model learned by the network [53]. The generative model obtained from unsupervised training of a deep neural network (hereafter called Generative Deep Neural Network – GDNN) supports the comprehensive learning framework represented in Fig. 2, and is characterized by the following specific properties.

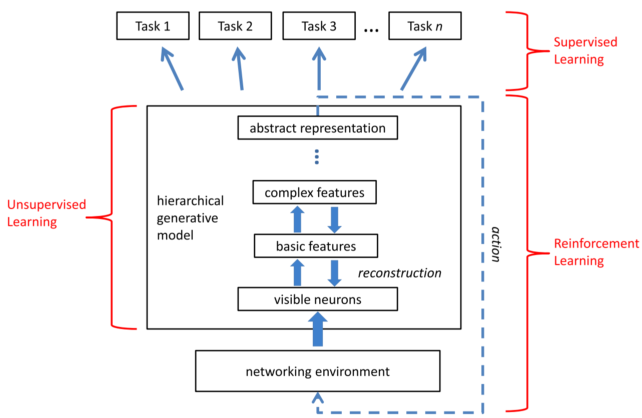


FIGURE 2. Schematic representation of a learning framework based on a Generative Deep Neural Network (GDNN).

1) GENERATIVE PROPERTY

GDNNs are trained to minimize the error between the observed data and its estimate obtained from the internal model (i.e., top-down reconstruction of the data). If the generative model includes the temporal dimension (for example, by learning data patterns presented according to a sequential structure [81]) it can be readily used to make predictions about the upcoming input information based on the recent history of the system, thereby exhibiting a proactive and anticipatory behavior that can be exploited to further improve information encoding [83]. Similarly, it is possible to estimate missing or noisy input terms, and to detect anomalies or unexpected patterns in the input signal.

2) HIERARCHICAL FEATURE EXTRACTION

The internal representations extracted by unsupervised learning are not tied to a specific discriminative task and turn out to be generally more informative than those obtained with supervised training. Moreover, hierarchical architectures capture the underlying factors of variability present in the data distribution by exploiting multiple levels of representation, which allow to extract increasingly more complex and abstract features at the deepest layers of the network [53], [67], [72]. This property is particularly useful because it makes it possible to train a single GDNN and use its top-level, abstract internal representations in place of the original input to train multiple supervised networks for specific tasks, as shown in Fig. 2. Moreover, the performance of such networks is typically better than that obtained by operating directly with the original sensory patterns [53], [56], [58].

3) COMPACT DATA REPRESENTATION

Deep unsupervised learning can also be interpreted as a particular type of efficient coding strategy, where the redundancies present in the input space are compactly described by using a reduced set of latent factors [64]. Within this perspective, deep networks represent a promising framework to study data compression and dimensionality reduction and to achieve scalability.

4) SYNERGY WITH REINFORCEMENT LEARNING

The generative approach also offers new insights about the possible role of actions performed by an agent on the environment. For instance, specific actions might allow to actively search for information that can disambiguate competing hypotheses and improve the internal model of the external world [84]. Moreover, the abstract representations obtained through deep learning appear to be a much better guide for reinforcement mechanisms than the raw data [82].

These properties of GDNNs can be exploited to develop a system architecture capable of efficiently dealing with the scalability, management, and multipurpose optimization challenges offered by the next generations of communication systems and services.

For example, in a HetNet scenario we may train a GDNN to learn the model of the wireless channel from the samples of the signal power received from the base stations, and then exploit the generative property of the model to predict the evolution of the wireless channel and proactively adapt all protocol layers accordingly. Also, endowing GDNNs with reinforcement learning features, we can develop generative models that link actions (e.g., settings of system parameters) and effects (e.g., the corresponding performance), thus making it possible to automatically find optimization actions tailored to the specific operational scenario, according to the Self-Configuration and Self-Optimization paradigms.

A similar approach can be applied to the massive access scenario, where a GDNN may be used to learn the number of active nodes and their traffic patterns by just observing the overall channel activity. Furthermore, the GDNN may be used to infer the nature of the traffic generated by the devices, thus making it possible to discriminate not only between different classes of traffic sources (e.g., alarms versus periodic metering data), but even between different streams of the same class (e.g., between sensory data with higher or lower priority depending, for instance, on their temporal trends). By considering the traffic generated by multiple sources, the GDNN may also reveal the presence of inter-flow correlations due, for instance, to wide-range environmental or social phenomena that impact the readings of multiple sensors and meters deployed in the same geographical area. For example, a GDNN that is fed with the readings of all the sensors and power meters of a building may capture correlations that depend on the specific type of building (e.g., office or residential). Such a richer context information can then be exploited to maximize the QoE offered to the final users, or to optimize the usage of the transmission resources by applying differential encoding between the actual value read by the sensors and the estimate obtained by the GDNN from the context.

The capability of GDNNs to build rich context models can also be exploited to detect different types of anomalies that can occur in (wireless) networks as a consequence of malfunctioning, faults, malicious attacks, or natural disasters. For instance, a base station can feed a GDNN with uplink and downlink traffic parameters (number of connected customers, data packet size, packet generation rate, queue length, packet loss rate, and so on) in order to build a model of its ''normal'' behavior. A classifier can then be trained to recognize any anomaly, such as traffic surges or fallings, sudden variations of the number of customers or access requests, and so on. The data that has generated the anomaly detection can then be passed to expert systems for diagnosing the nature of the problem, so that proper reactions can be promptly and automatically undertaken, according to the Self-Healing paradigm.

Yet another example of the type of optimization that the COBANETS vision can enable is given in Sec. IV, where we describe some recent results concerning the management of video streaming flows in communication networks.

B. GENERAL ARCHITECTURE

The COBANETS concept is based on the idea that GDNNs can be employed to cope with the growing complexity of modern communication systems and services, and to enable optimizations at different levels, from local operations performed by end devices (such as modulation, channel access, buffer management, etc), up to system-wide policies that affect multiple data flows (resource reservation, routing, content caching management, security, and so on).

To fully express the potential of GDNNs, we envision an architecture that enables network-wide observation and sensing at multiple levels, including quantities such as protocol parameters and state variables, traffic conditions, channel statistics, transmission and error events, interference, and so on. In short, the architecture shall possess the capability of sensing its own state and to share this information among the different cognitive entities, in order to achieve the system-level self-awareness that Mitola claims to be a fundamental requirement of any cognitive system [17]. From this perspective, the COBANETS architecture can be embedded within the general framework of the Knowledge Plane described in [27], of which it shares the distributed and compositional characteristics.

The architecture shall also provide the flexibility required for the practical implementation of the proposed approach. In other words, the COBANETS architecture shall embed the SAN concept presented in [24]–[26] that, however, can now be rooted into the emerging SDN and NFV paradigms, which will be the basic building blocks of COBANETS. An advanced SDN controller, indeed, may be able to collect the inter-device data generated by the cognitive nodes in the system, and make decisions for system-wide optimization. SDN protocols, such as OpenFlow [85], can be used to implement the optimization actions on the different nodes, thus realizing the functionalities of the API defined in [24], or the ULLA described in [28], whereas the modifiability of the SAN elements finds its correspondent in the virtualization of the network elements as per the NFV concept. In addition, the controller may instruct the nodes to instantiate generative deep learning modules for the optimization of local functionalities (e.g., PHY and MAC), using inter-device data only, thus fulfilling the distributed and collaborative characteristics of the cognitive resource manager described in [28] and [29].

The Cognitive Engine, receiving input signals from a number of peripheral sensorial units (cognitive devices), can progressively build a representation of the surrounding world and learn the effect of different actions (e.g., settings of protocol parameters, enforcement of system policies, management of network resources, and so on) that will be actuated by means of the SDN and NFV protocols.

According to this vision, the COBANETS architecture loosely fits within the previous cognitive networking architectures, but brings in some disruptive elements of innovation when identifying GDNNs as the basic learning structure to provide context awareness, information representation, and inference capabilities to the system, and proposing the SDN and NFV paradigms as ways to practically implement the proposed architecture in future networks.

Clearly, there is still a long way to go to turn this broad and very general vision into a practical and well defined system architecture. In the remainder of this section, we describe some of the main components that we believe may help reach this ultimate objective. These components, which shall all be part of the final COBANETS architecture, are here presented in order of increasing generality and scope, thus reflecting the natural path we envision for the development of the COBANETS concept.

1) FUNCTIONAL ABSTRACTION AND OPTIMIZATION

Important components of COBANETS will be generative models that provide an informative representation of fundamental elements and functionalities of a communication network, including traffic sources, radio channel, MAC protocols, and so on. These representations can be used to train different types of classifiers that will provide detailed context information, thus making it possible to develop context-aware optimization strategies, with an approach similar to [86] and [87] where the authors applied this method to classify different video flows according to their (estimated) rate-distortion characteristics, by only analyzing the size of the video frames sent over the network. We hence envision generative models that can learn the traffic patterns of different sources, thus providing alternative and more expressive representations of the data source. In parallel, we may have GDNNs that can capture the interdependencies among the parameters within a certain protocol layer (e.g., at the physical layer the transmission parameters of a mobile node and the interference from adjacent cells, or at the MAC layer the packet inter-arrival times and the number of retransmissions). These generative models can be used to predict the offered traffic in the near future and/or to train classifiers to get more detailed context information, for instance the type of application(s) generating the data flows, the operational scenarios (indoor, urban, vehicular, rural), or the congestion level of a certain connection. This context information, in turn, may be used to optimize some network functionalities (e.g., handover, content caching, transmit rate, and so on).

It is worth remarking that the concept of *optimization* in COBANETS shall be intended in a network-wide perspective. The optimization goal, indeed, will generally entail multiple objectives simultaneously, possibly including end-user QoE, service provider revenues, network elements utilization, and so on. Furthermore, the optimization objectives shall be subject to feasibility constraints. Referring to [88], we can hence define the optimization as the problem of finding a vector of decision variables (actions) which satisfies a given set of constraints and makes all the objective functions take values in an acceptable performance region, as defined by the system designer. Such a system-wide multi-objective optimization goal can sometimes be broken down to local optimization functions, which can be addressed independently by local entities (e.g., maximization of the physical layer throughput). In general, however, local optimizations shall be intended as functional to the achievement of system-wide goals. For instance, if the system-wide objective of a wireless system is to provide a minimum guaranteed quality level to mobile video customers, the target quality level of the customers with better link quality may actually be lower than what would be potentially achievable, in order to leave more resources to the other customers in the system.

2) INTEGRATION OF DIFFERENT GENERATIVE MODELS

In analogy with the sensory segregation and integration observed in the brain, the specialized modules operating in different domains, as described above, should be combined in a learning architecture capable of building more abstract representations of the world. Implementing this strategy is very challenging, because it requires to carefully engineer the scope of each sub-module and to integrate the internal representations created by different models without disrupting domain-specific knowledge. A possible solution can be to concatenate the representation of different models and to jointly train an additional generative model with the task of reconstructing this composite input, thereby learning useful correlations among the abstract representations provided by different sensory domains [89]. Another approach consists in mapping the abstract representations learned by the sub-models into layer-specific performance indices and then training a combined generative model using such indices. These two approaches, as well as others, need to be deeply studied and compared in terms of complexity and efficiency, with the goal of identifying the best solution.

3) INTER FLOW OPTIMIZATION

As an intermediate step to system-wide optimization, we believe that COBANETS shall make it possible to jointly optimize multiple functionalities with local scope (e.g., within a single node). Current networks employ adaptive solutions at various layers of the protocol stack, such as adaptive modulation, beamforming and MIMO at the PHY layer, contention window adaptation at the MAC layer, congestion window control at the transport layer, or video coding scalability at the application layer. A proper model-learning algorithm, however, can identify the cross-relationships between the protocols and allow for their joint optimization in a cross-layer fashion. Unlike in traditional cross-layer optimization, where prior knowledge of some explicit interdependencies among protocols is assumed, the approach based on generative model learning has the potential to discover and exploit hidden relations among the different parameters, which can be specific for a certain application scenario or user profile and, hence, are not replicable in other contexts. For example, daily habits of users (e.g., watching movie trailers on the smartphone while commuting by train) may be reflected in specific inter-relations among the type of traffic generated by the device, the interference produced by other devices, the radio channel characteristics, and the geographical location. Therefore, COBANETS shall entail generative models capable of capturing these multifold correlations, thus supporting the design of optimization strategies that are adapted to a specific device and scenario.

4) SYSTEM LEVEL OPTIMIZATION

As mentioned, the final objective of COBANETS shall have a global scope, and shall refer to the whole system. A possible global optimization may include the routing strategy to be applied to the different flows crossing the network, according to the nature of the data (machine-type data, video streaming, web browsing), the characteristics of the user (static, mobile), the congestion on the links, and so on. Scheduling policies in the switches, resource allocation at the base stations, and transport protocol parameters can also be jointly optimized for the specific context. Pursuing such a multidimensional optimization of the whole protocol stack is a formidable task, which is likely impossible to solve directly in real time using traditional approaches. However, we believe that GDNNs can indeed enable the development of an innovative scalable approach to the above problem, by taking advantage of the data provided by the single agents of the system and collected by the cognition-based architecture designed, thus making it possible for a centralized network controller to autonomously derive strategies for the maximization of multi-objective functions, and to actuate such strategies in the network elements by means, e.g., of SDN. Therefore, the last components of the COBANETS architecture are represented by GDNNs with global scope, which will be likely fed by and incorporate GDNNs with local scope, though the actual structure of such components is still to be defined and is in fact a very interesting research challenge.

IV. A PRACTICAL EXAMPLE: COGNITIVE VIDEO TRAFFIC CONTROLLER

To exemplify the potential of the cognition-based approach described in this paper, in this section we report some results we obtained in a preliminary study that addressed the optimization of the QoE perceived by the customers of a video streaming service in the presence of network congestion. The scenario discussed below provides an illustrative case to exemplify the main ideas underlying the proposed framework, even though not all of the COBANETS features described above are explicitly included. In particular, we only used a single-layer RBM to build abstract representations of the training patterns, as the use of hierarchical generative models was found not to provide any gain for the limited dimensionality of these inputs. In addition, in this simple case we did not explore the possibility to actively learn optimal action schemes from environmental feedback (i.e., reinforcement learning). Nevertheless, this example provides evidence about the soundness of the COBANETS technical approach, in which a representation built via unsupervised generative learning can be successfully exploited towards a specific task via supervised learning.

The problem of providing quality-fair delivery of multimedia contents to mobile users has been addressed in many other works in the literature (see [90]–[92]). Many of these works assume that the rate-distortion curve of each video is known in advance. Such a characterization, however, is typically computationally expensive and generally impractical to be performed in real-time, e.g., by means of deep packet inspection techniques. Instead, machine learning approaches can be of use in this respect.

The problem of automatic video processing is closely related to that of image recognition, with the additional complexity given by the temporal dimension of the data. In the so-called "content-based" video retrieval [93], for instance, a range of different techniques can be applied depending on the task of interest, e.g., video indexing, scene recognition and/or classification, object tracking, and motion detection. In recent years, deep learning has been successfully applied in several machine vision tasks, achieving state-of-the-art performance [64], [73]. Although the main application of these systems has been primarily focused on still frames, there have also been successful extensions to the temporal domain [94].

All the above-mentioned machine learning methods, however, are usually applied at the pixel level, or to some higher-level representations obtained after additional pre-processing of the raw images. Nevertheless, for the task of classifying different videos depending on the dynamics of their content, we assume that the relevant information is still preserved after the video has been encoded to be sent on a transmission channel.

In the remainder of this section we elaborate on this intuition and show how raw data, readily available at the network

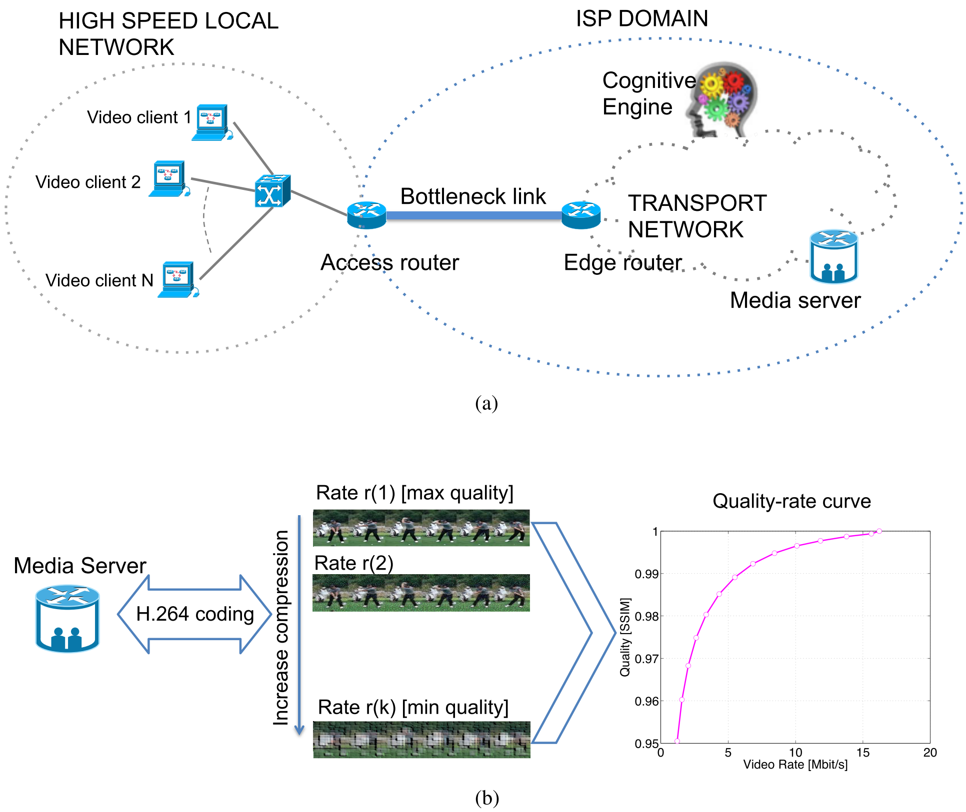


FIGURE 3. Reference scenario of the cognitive video traffic controller. (a) Reference network topology: video clients require video streaming services to a remote video server, through a shared bottleneck link. A cognitive engine, coupled with all network elements and with a SDN/NFV controller, manages the functionalities of all network elements. (b) Graphical representation of a video server storing multiple versions of each video clip with different compression levels, in H.264 format. The right-hand side graph shows the quality-rate curve of a sample video, where the quality is expressed in terms of SSIM.

level, can be used to train, in an unsupervised manner, a generative model that captures the main features of the input data (and, in turn, of the original video sources), thus offering a richer representation of the data source. Then, we show how such a representation can be used, in a supervised framework, to successfully classify the different frame sequences, according to their inner features. Finally, coherently with the COBANETS vision, we exploit such a better context knowledge to optimize the QoE offered to the final user, by tuning the amount of resources allocated to the different users according to their actual needs, as estimated by our cognitive framework.

In the following, we briefly summarize the approach and the results we presented in [86], [87], [95], [96], and refer the reader to the original papers for additional details and an in-depth discussion.

The reference scenario considered in the study is sketched in Fig. 3. The upper part of the figure shows N users that require video streaming services from a remote media server. To exemplify a critical network condition, we assume that all video flows go through a common link with limited capacity, which is hence the bottleneck for the video streaming service. The *Cognitive Engine* represents the entity in charge of managing the network of the Internet Service Provider (ISP) in an ‘‘intelligent’’ manner. To this end, we assume that the Cognitive Engine can communicate with and configure all the network elements (e.g., by means of an OpenFlow–like protocol [85]).

As shown in the lower part of Fig. 3, we assume that the video server stores multiple copies of each video, encoded at different quality levels. In our study, we consider the popular H.264-AVC video encoding standard [97], though the approach can be extended to any other type of video encoder. The different source rates correspond to different perceived quality levels, here measured in terms of the average Structural SIMilarity (SSIM) index, which is a full reference metric that measures the image degradation in terms of perceived structural information change, thus leveraging the tight inter-dependence between spatially close pixels which contain the information about the objects in the visual scene [98]. The range of the SSIM index goes from 0 to 1, and fairly good video qualities are associated with values greater than 0.95 [99]. The right-most plot in Fig. 3(b) shows an example of the quality-rate characteristic of one of the sample videos we considered in our study.

When the bottleneck link is saturated, additional video requests cannot be accepted unless the rates of active videos are decreased to leave resources for the new flows. Therefore, we assume that the Cognitive Engine intercepts all requests for new video streaming sessions generated by the end users and implements *Video Admission Control* (VAC) and *Resource Management* (RM) algorithms to manage the different flows.

Scaling down the rate of a video stream, however, decreases the video quality perceived by the end user, according to a quality-rate relation that is specific for each single video. In [96], we showed that, after a suitable rescaling and normalization of the source rate, the SSIM-to-bitrate curve of a video can generally be well approximated by a polynomial function. Knowing such a polynomial would make it possible to dynamically choose the quality level that best fits the connection conditions. Unfortunately, calculating the SSIM for all possible encoding rates of each video is computationally prohibitive in realistic scenarios.

A possible way to overcome this obstacle is to resort to generative models for learning expressive representations of video clips from some easily accessible features. Following this idea, in [87] we proposed an approach, based on RBMs (see Fig. 1), to reliably estimate the coefficients of the polynomial that approximates the SSIM-to-bitrate curve of a video *without* processing the actual content of the video frames, but only considering the *size of the encoded video frames*. The rationale is that the SSIM-to-bitrate function of a video is closely related to the dynamics of its content, and this information is reflected in the structure of the corresponding sequence of frame sizes after encoding. Indeed, the content of a video influences the structure of its compressed version. For example, highly dynamic videos, with complex spatial and temporal structure, will likely result in larger frame sizes, with lower temporal correlation, while more static video sequences will likely be encoded in frames with smaller size, with a more regular pattern. The GDNN can then extract these informative features from the sequence of the frame sizes of encoded videos, and such features can successively be used to infer the relevant characteristics of the video sequences.

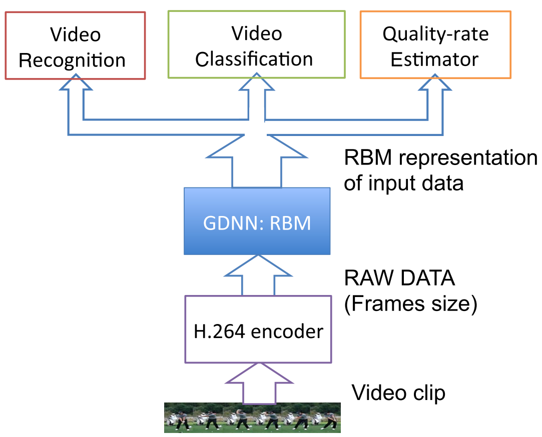


FIGURE 4. Graphical representation of video source modeling using a Restricted Boltzmann Machine (RBM), and of three specific tasks that can be realized on top of it.

To test the effectiveness of this approach, we collected a training dataset containing the frame sizes of the different Groups of Pictures (GOPs) of the test videos. These "raw" data have then been used to train an RBM in an unsupervised fashion. The RBM captures the latent features in the input data and provides a high-level representation of such data. This richer representation can then be exploited by different supervised learning algorithms for various purposes, as graphically exemplified in Fig. 4 where we show three possible tasks that can be performed on top of the RBM, namely *Video Recognition*, *Video Classification*, and *Quality-rate Estimation*.

The Video Classification task aims at assessing whether the internal representations learned by the RBM make it possible to assign each GOP to the video sequence it belongs to. To succeed in this task, the RBM has to extract descriptive features that make it possible to discriminate the content of the videos starting from the information provided by the frame sizes, which is a challenging task. The Video Classification task, instead, aims at clustering the input GOPs into a small number of classes, each containing videos with similar quality-rate characteristics. The Quality-rate Estimation task, finally, aims at the estimate of the coefficients of the polynomial curve that approximates the actual SSIM-to-bitrate characteristic of each single video.

To test the approach, we considered several video clips with CIF and HD resolutions, taken from a reference video database.[2] Each video has been encoded with the Joint Scalable Video Model (JSVM) reference software into H.264-AVC format at $C = 18$ increasing compression levels, which correspond to as many quality levels. Note that there are no scene transitions within each video sequence. The SSIM of a frame encoded at compression level c is obtained by comparing the decoded frame with the full quality version of the same frame. For practical reasons, we take the average values of the SSIM index for each video.

We denote by $r_v(c)$ the transmit rate of video $v \in \{1, \ldots, V\}$ encoded at rate $c \in \{1, \ldots, C\}$, with $r_v(1)$ being the maximum (i.e., full quality) rate. To ease the comparison between different video clips, we found that it is convenient to rescale the source rate as follows

$$\rho = \log(r_v(c)/r_v(1)). \tag{1}$$

The resulting index is called *Rate Scaling Factor* (RSF).

Fig. 5 reports the classification accuracies computed on the training set and on the separate test set for both Video Recognition and Video Classification tasks. The difficulty of the first task (video recognition) is confirmed by the poor performance of the classifier that operates directly on the raw data patterns, i.e., on the vectors containing the frame sizes of the different GOPs. However, the internal representations learned by the RBM model capture some critical features of the data, thereby providing a significant improvement of the classification accuracy. The second task is apparently easier, but the performance obtained by using the internal representations of the RBM is still better than that achieved with raw data.

[2]http://media.xiph.org/video/derf/; ftp://132.163.67.115/MM/cif

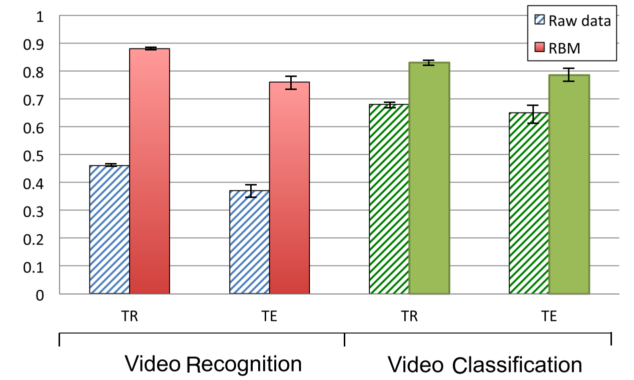


FIGURE 5. **Classification accuracy for the Video Recognition and Video Classification tasks, computed on the training (TR) and test (TE) datasets.**

The third task, i.e., the estimation of the SSIM-to-bitrate coefficients, is the most interesting for the purpose of video service optimization. To evaluate the quality of the estimation, we compute the Root Mean Square Error (RMSE) between the exact SSIM curve and the curve generated using the coefficients estimated by the classifier.

Fig. 6 shows the mean estimation accuracy obtained for some HD videos in the test set. We can appreciate how the RBM model is indeed capable of capturing critical features of the data, thereby providing an increased estimation accuracy for almost all test videos.

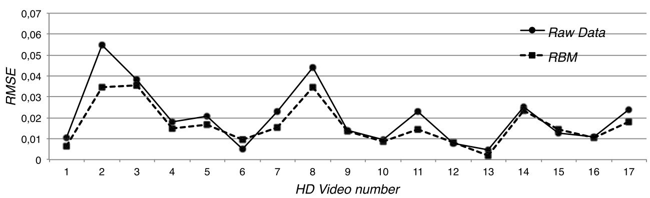
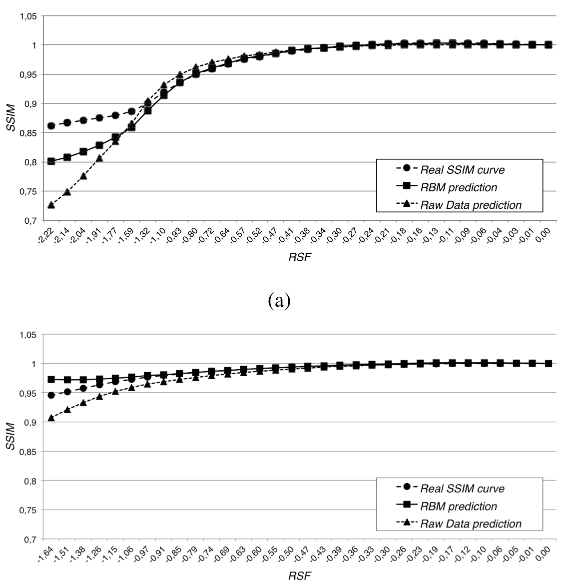


FIGURE 6. **Coefficients prediction error in terms of mean RMSE between the actual and predicted quality-rate curve for different HD videos.**



(a)

(b)

FIGURE 7. **Accuracy of SSIM-to-RSF estimates, for two sample videos. (a) Video number 7. (b) Video number 10.**

Fig. 7 compares the actual SSIM-to-RSF curve (●) with the estimates obtained by applying the linear classifier directly on the raw data patterns (▲) and on the internal representation learned by the RBM (■). By visual inspection, we can see that the estimate provided by the RBM is closer to the actual curve, thus resulting in better performance of QoE-aware VAC and RM algorithms.

We remark that these results are obtained without resorting to computationally expensive procedures, such as deep packet inspection or image processing, but rather exploiting the information that is embedded in easily accessible parameters, such as the packet size. It is hence conceivable that the video server, or the Cognitive Engine, employs such an RBM to enrich each video clip with additional information such as, for instance, the inner representation of the GOP sizes given by the RBM, or the quality-rate class the GOP belongs to, or even the estimate of the coefficients of the SSIM-to-RSF polynomial curve. This information can then be used by the Cognitive Engine to allocate the network resources to the different video clients, according to the characteristics of each single video. For instance, the Cognitive Engine can instruct the edge router to differentiate the share of the bottleneck link assigned to each user in order to provide equal QoE to all active users, while minimizing the call blocking probability (see [87]). Another possibility is that the Cognitive Engine recognizes the presence of a bottleneck link and reconfigures the network to differentiate the path of the different video flows using alternative (e.g., wireless) access networks to reach the final users, or that it generates other instances of the video servers in different points of the network, to better distribute the load.

These examples show that the potential of generative models is greatly expanded by the possibility of managing the entire system, or even part of it, in a flexible and dynamic manner, as per the SDN and NFV paradigms. How to fully unleash this disruptive potential, however, is still an open research question.

V. RESEARCH CHALLENGES

Despite some initial studies, which provided encouraging results, the application of the generative deep learning principles to network optimization is still in its infancy. The design of an effective GDNN-based framework for network optimization poses a number of new scientific and technical challenges that need to be systematically addressed, in order to reach the expected gains.

In the following we discuss what are, in our opinion, some of the most appealing challenges raised by this exciting scenario.

A. DATA COLLECTION AND SHARING

A key enabler for the optimization approach proposed in this paper is the ability to collect data from various layers of the protocol stack, the environment, and even the final user, and to share these data at the network level. In general, we envision several types of data that can be collected, namely:

i) intra-device data (collected within each single device, e.g., protocol parameters or location), to be used in local optimizations (e.g., energy efficiency of a node);
ii) inter-device data (exchanged between devices, e.g., traffic patterns or queue lengths at routers), to be used for optimization on a wider scale (e.g., maximization of the number of flows the system can serve);
iii) user-profile data (which represent the user's preferences) to define the Quality of Experience objective function to be used in the optimization.

Finding which data is most useful in the GDNN and for network optimization, studying the granularity and frequency at which these data need to be collected, and defining practical methods for representing, storing and retrieving such data at both the device and the system level are all open research problems.

B. DATA REPRESENTATION AND SYNCHRONIZATION

Another open issue of the proposed optimization framework is the choice of the format of the data patterns that should be given as input to the GDNNs. In the context of network optimization, indeed, the sensory data might come in many different formats, which should nevertheless be encoded as activation values in the input layer of the network. This implies the need to carefully design a variety of encoding modules that should be used to transform the collected data into a unified representation, which should preserve as much as possible the inherent structure present in the data. This problem becomes even more challenging when considering data coming from heterogeneous devices and/or abstraction layers and from different time scales, or collected with different sampling frequencies or even asynchronously. Therefore, some effort needs to be devoted to the identification of a solution for the data representation problem, which will also allow to better understand which are the most critical dimensions of the data domain (i.e., the most informative input signals) that should be given to the learning system.

C. EXPLOITING LONG-TERM SPATIO-TEMPORAL RELATIONSHIPS

An important component of cognition is the ability to adapt based on behaviors that have been observed and learned in the past and are likely to be encountered again. How to include knowledge of the long-term spatio-temporal behavior of the network parameters (such as congestion or channel characteristics) into the optimization framework is an open research issue. Different strategies to include the time dimension into the generative models can be considered. A possible way is to build input vectors that collect the system parameters sampled at different time scales, in order to provide a representative example of the time evolution of the system. Another promising possibility is to use more complex generative models that are inherently sequential, such as the Recurrent Temporal Restricted Boltzmann Machine [79] or similar models that can be even combined into hierarchical architectures [80]. Further research is needed to gain a deeper understanding of these and other approaches and to find the best solution for the different optimization goals.

D. MULTI-OBJECTIVE OPTIMIZATION STRATEGIES

The final objective of COBANETS shall be the automatic management of complex systems, in which individual agents may have both selfish objectives and common social goals to pursue (the latter possibly encouraged by game-theoretic or trust and reputation-based incentives). This problem may be approached using multi-objective optimization techniques, or by properly defining utility functions that jointly account for multiple objectives, appropriately weighed, or through a hierarchical organization of the goals. The specific properties of the generative models shall likely be combined with reinforcement learning mechanisms to automatically learn the best strategies in such a complex scenario.

E. IDENTIFICATION OF DOMAIN-SPECIFIC DEEP ARCHITECTURES

A crucial aspect to improve the performance and the scalability of many learning systems is to identify a useful set of constraints that can facilitate learning, for instance by reducing the complexity of the model or by improving convergence. For example, the most successful deep architectures for visual object recognition have been designed to exploit the strong local spatial correlation found in natural images [73]. It is therefore of interest to investigate how the distinguishing characteristics of telecommunication network signals can influence the deep architectures for learning-based optimizations. For example, deep network architectures may be designed to better process data with strong spatio-temporal correlation, or to account for the interdependencies among network elements induced by network topology. Moreover, when the deep network is fed with data originated by multiple devices interconnected through a communication network, there may be significant communication delays or even packet losses, thereby posing concrete challenges to a learning system that is usually expected to receive "clean" and reliable training patterns. These problems represent a less studied field of research that can potentially generate new insights and advances also in the machine learning domain.

F. ALTERNATIVE BUILDING BLOCKS FOR UNSUPERVISED LEARNING

In relation with the previous point, we observe that, while in this paper we referred to hierarchical generative models in generic terms, they can actually take different forms, such as autoencoders, RBMs and, more generally, energy-based models. Most of these models obtain similar performance in canonical machine learning experimental evaluations [100]. However, deterministic and probabilistic models have different optimization objectives, which result in implementations with different computational properties.

Moreover, several advanced regularization techniques have been recently proposed to improve generalization in deep networks, for example by imposing sparsity constraints or by exploiting drop-out schemes [101]. Therefore, an interesting research topic will be the systematic study of the strengths and weaknesses of each approach in light of the considered optimization framework, and the investigation of which regularization techniques are more effective with the type of data and tasks required in such scenarios.

G. KNOWLEDGE DISTRIBUTION ACROSS NETWORK ELEMENTS

A centralized management system may become the bottleneck of the optimization framework, in which case it would be preferable to distribute the optimization tasks to different network elements that should nevertheless be able to perform optimizations according to a global view of the networking environment. Some interesting recent studies have shown that the performance level obtained by very-large-scale deep neural networks in supervised classification tasks can be replicated in much smaller learning modules (*model compression*), such as simple networks with only one hidden layer, if we use as training labels the *soft labels* at the output of the large-scale deep network [102]. This intriguing result motivates further research about how to possibly create ''lightweight'' processing nodes that can support efficient optimization in a highly distributed system. Moreover, distributing the generative model over multiple nodes might be a valuable approach to speed up learning and inference tasks via efficient parallelization [70]. This feature is even more appealing considering that modern mobile devices (e.g., smartphones or notebooks) are equipped with powerful computing hardware, as discussed later on.

H. SECURITY ASPECTS

In our cognition-based approach, the network will need to continuously collect large amounts of data, apply a learning process to it, and take actions as a result, which will make the confidentiality of the original data, as well as that of the ''reasoned'' outcome, much more important than in traditional TCP/IP networks. For example, by changing behavior and observing how the network reacts, a user may obtain private information of others [103]. An open issue is to find the proper tradeoff between confidentiality and effectiveness of the proposed solutions, also considering possible de-anonymization techniques and privacy attacks based on machine learning [104]. Another problem is to design solutions to assess the trustworthiness of both peers and data (against attacks to either evade security checks or poison the learning process with fake data), as well as to make the learning process resilient to malicious attacks (e.g., based on Adversarial Machine Learning [105]). These are just a few examples of a number of innovative and challenging research problems concerning the security of cognitive systems, which shall also include data confidentiality, trustworthiness, and resiliency to attacks.

I. IMPLEMENTATION AND PROTOTYPING

As mentioned, we believe that experimental activities shall take a primary role in the design of COBANETS. While some testbeds capable of collecting system-wide cross-layer parameters have been proposed (see [106]), the real-time testing of machine learning algorithms on experimental data has not yet been systematically addressed. Running machine learning algorithms is indeed computationally intensive, because learning might involve the optimization of millions of parameters. Since the introduction of CUDA [69], a new parallel computing framework for common Graphics Processing Units (GPUs) presented by NVIDIA®, many computational tasks (e.g., the matrix manipulations typical of deep learning algorithms) can be efficiently carried out. Our prior experience on optimizing the parallel design of deep learning algorithms [55], [107], and on their implementation on GPU platforms, has shown that impressive performance can be achieved in terms of learning time and datasets size. For example, we found that even an entry-level GPU card (336 cores) yielded a 10-fold speed-up in a benchmark problem compared to a quad-core PC (where the card was mounted) and was twice as fast in learning as a computer cluster with 60 CPU cores. Based on this experimental evidence, it seems possible to exploit GPU parallelization on mobile devices to run complex machine learning algorithms. Indeed, top-level Android-based devices (smart phones and tablets) are built around the NVIDIA® Tegra mobile parallel computing platform endowed with many GPU cores. Using such devices shall make it possible to run even complex algorithms in real-time and at affordable prices. However, the practical implementation of complex GDNNs in mobile devices, and the development of a system-wide cognitive testbed capable of performing learning tasks in real time, are still open research fields.

VI. CONCLUSIONS

In this paper we advocated Generative Deep Neural Networks (GDNNs) as the key building block for a new generation of cognition-empowered networks and systems, called COgnition-BAsed NETworkS (COBANETS), where the ability of GDNNs to extract richer context representations will be combined with different kinds of machine learning techniques to realize specific tasks, and will be integrated with the Software Define Networking and Network Function Virtualization paradigms to enable the flexible actuation and management of complex systems.

We provided some preliminary examples of the potential of such an approach, reporting some results of a study where the size of the encoded data frames was used to train a generative neural network in an unsupervised manner. The model learnt by the network was then used to estimate the quality-rate characteristics of video flows, and this context information was then exploited in QoE-aware resource management schemes. These preliminary results, though encouraging, only scratch the surface of the potential of the proposed approach, which opens a number of interesting interdisciplinary research issues.

A possible way to address these exciting challenges is to approach the problem gradually, progressively widening the scope of the network optimization goal. The first fundamental step shall consist in gaining a deeper understanding of the potential of the generative deep learning approach to model and optimize specific network functionalities, such as resource allocation at the PHY layer, setting of MAC parameters, scheduling, routing, traffic source modeling, and so on. Supported by a solid theoretical and experimental foundation, it will then be possible to develop a generative deep learning approach to system-level optimization. This phase will require the design of GDNNs capable of representing all the relevant functionalities that concur in determining the system performance, and to address the most critical and challenging issues related to the scalability of the approach, the multi-objective optimization of the system parameters, the coordination of the different functionalities and network elements, and the implementation of the planned actions. Such theoretical and simulation studies need to be complemented in a synergistic manner by experimental activities, with the dual objective of validating the proposed innovative techniques and revealing other possible challenges that may arise in practical settings.

Besides leading to novel methods for the optimization of communications systems, this research may stimulate innovation in cognitive science and machine learning as well, leading to the development of new learning techniques that need to obey different constraints and boundary conditions than traditionally found in those areas. Therefore, we believe that the COBANETS concept may pave the way to new research avenues that intersect multiple sectors in cognitive science and information and communication engineering, with the potential of leading to disruptive innovation in these fields and unpredictable effects on other fields that may benefit from the stimuli and the change of perspective brought about by the proposed vision.

REFERENCES

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[2] W. H. Chin, Z. Fan, and R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 106–112, Apr. 2014.

[3] P. Demestichas *et al.*, "5G on the horizon: Key challenges for the radio-access network," *IEEE Veh. Technol. Mag.*, vol. 8, no. 3, pp. 47–53, Sep. 2013.

[4] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digit. Commun. Netw.*, vol. 1, no. 1, pp. 1–19, Feb. 2015.

[5] S.-Y. Lien and K.-C. Chen, "Massive access management for QoS guarantees in 3GPP machine-to-machine communications," *IEEE Commun. Lett.*, vol. 15, no. 3, pp. 311–313, Mar. 2011.

[6] F. Guidolin, I. Pappalardo, A. Zanella, and M. Zorzi, "A Markov-based framework for handover optimization in HetNets," in *Proc. 13th Annu. Medit. Ad Hoc Netw. Workshop (MED-HOC-NET)*, Jun. 2014, pp. 134–139.

[7] D. Lopez-Perez, I. Güvenc, and X. Chu, "Mobility management challenges in 3GPP heterogeneous networks," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 70–78, Dec. 2012.

[8] S. Thakolsri, W. Kellerer, and E. Steinbach, "QoE-based cross-layer optimization of wireless video with unperceivable temporal video quality fluctuation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–6.

[9] A. Khan, L. Sun, and E. Ifeachor, "QoE prediction model and its application in video quality adaptation over UMTS networks," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 431–442, Apr. 2012.

[10] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. New York, NY, USA: Wiley, 2012.

[11] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2*, document 3GPP TR 36.300, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/36_series/36.300/

[12] *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions*, document 3GPP TR 36.902, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/specs/archive/36_series/36.902

[13] B. Fu, Y. Xiao, H. Deng, and H. Zeng, "A survey of cross-layer designs in wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 110–126, Feb. 2014.

[14] V. Srivastava and M. Motani, "Cross-layer design: A survey and the road ahead," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 112–119, Dec. 2005.

[15] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[16] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[17] J. Mitola, III, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, May 2000.

[18] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[19] N. Devroye, M. Vu, and V. Tarokh, "Cognitive radio networks," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 12–23, Nov. 2008.

[20] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[21] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Netw.*, vol. 7, no. 5, pp. 810–836, Jul. 2009.

[22] C. Fortuna and M. Mohorcic, "Trends in the development of communication networks: Cognitive networks," *Comput. Netw.*, vol. 53, no. 9, pp. 1354–1376, Jun. 2009.

[23] F. H. P. Fitzek and M. D. Katz, Eds., *Cognitive Wireless Networks*. Dordrecht, The Netherlands: Springer-Verlag, 2007.

[24] R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks," in *Proc. IEEE Symp. New Frontiers Dyn. Spectr. Access Netw. (DySPAN)*, Baltimore, MD, USA, Nov. 2005, pp. 352–360.

[25] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*, H. Arslan, Ed. Dordrecht, The Netherlands: Springer-Verlag, 2007, pp. 17–41.

[26] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives," *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, Dec. 2006.

[27] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *Proc. ACM SIGCOMM*, 2003, pp. 3–10.

[28] P. Mähönen, M. Petrova, J. Riihijärvi, and M. Wellens, "Cognitive wireless networks: Your network just became a teenager," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 23–29.

[29] M. Petrova and P. Mähönen, "Cognitive resource manager," in *Cognitive Wireless Networks*, F. H. P. Fitzek and M. D. Katz, Eds. Dordrecht, The Netherlands: Springer-Verlag, 2007, pp. 397–422.

[30] M. Levorato, S. Firouzabadi, and A. Goldsmith, "A learning framework for cognitive interference networks with partial and noisy observations," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3101–3111, Sep. 2012.

[31] L. Giupponi, A. Galindo-Serrano, P. Blasco, and M. Dohler, "Docitive networks: An emerging paradigm for dynamic spectrum management [Dynamic Spectrum Management]," *IEEE Wireless Commun.*, vol. 17, no. 4, pp. 47–54, Aug. 2010.

[32] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, Jul. 2013.

[33] C. Clancy, J. Hecker, E. Stuntebeck, and T. O'Shea, "Applications of machine learning to cognitive radio networks," *IEEE Wireless Commun.*, vol. 14, no. 4, pp. 47–52, Aug. 2007.

[34] A. Forster, "Machine learning techniques applied to wireless ad-hoc networks: Guide and survey," in *Proc. IEEE 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inf. (ISSNIP)*, Dec. 2007, pp. 365–370.

[35] S. Whiteson and P. Stone, "Adaptive job routing and scheduling," *Eng. Appl. Artif. Intell.*, vol. 17, no. 7, pp. 855–869, Oct. 2004.

[36] A. X. Zheng, J. Lloyd, and E. Brewer, "Failure diagnosis using decision trees," in *Proc. Int. Conf. Auto. Comput.*, May 2004, pp. 36–43.

[37] M. Mesnier, E. Thereska, G. R. Ganger, D. Ellard, and M. Seltzer, "File classification in self-* storage systems," in *Proc. Int. Conf. Autonomic Comput.*, May 2004, pp. 44–51.

[38] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, Oct. 2008.

[39] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in *Proc. ACM SIGMETRICS*, 2005, pp. 50–60.

[40] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, Oct. 2006.

[41] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM Workshop Mining Netw. Data (MineNet)*, 2006, pp. 281–286.

[42] K. Winstein and H. Balakrishnan, "TCP ex machina: Computer-generated congestion control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 123–134, Oct. 2013.

[43] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to TCP throughput prediction," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1026–1039, Aug. 2010.

[44] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for Internet video," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 339–350, Oct. 2013.

[45] N. Agoulmine, Ed., *Autonomic Network Management Principles: From Concepts to Applications*. New York, NY, USA: Academic, 2010.

[46] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 336–361, Feb. 2013.

[47] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," in *Proc. IEEE 8th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May/Jun. 2010, pp. 170–176.

[48] R. Razavi, S. Klein, and H. Claussen, "A fuzzy reinforcement learning approach for self-optimization of coverage in LTE networks," *Bell Labs Tech. J.*, vol. 15, no. 3, pp. 153–175, Dec. 2010.

[49] R. Razavi, S. Klein, and H. Claussen, "Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach," in *Proc. IEEE 21st Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2010, pp. 1865–1870.

[50] M. N. ul Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 2818–2823.

[51] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.

[52] G. Hinton and T. J. Sejnowski, Eds., *Unsupervised Learning: Foundations of Neural Computation*. Cambridge, MA, USA: MIT Press, 1999.

[53] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: A tutorial overview," *Frontiers Psych.*, vol. 4(515), Aug. 2013.

[54] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 7. 2011, pp. 1–20.

[55] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, "Deep unsupervised learning on a desktop PC: A primer for cognitive scientists," *Frontiers Psych.*, vol. 4(251), May 2013.

[56] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[57] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[58] I. Stoianov and M. Zorzi, "Emergence of a 'visual number sense' in hierarchical generative models," *Nature Neurosci.*, vol. 15, pp. 194–196, Jan. 2012.

[59] G. E. Hinton and Z. Ghahramani, "Generative models for discovering sparse distributed representations," *Philos. Trans. Roy. Soc. B, Biol. Sci.*, vol. 352, no. 1358, pp. 1177–1190, Aug. 1997.

[60] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognit. Sci.*, vol. 9, no. 1, pp. 147–169, Jan. 1985.

[61] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[62] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[63] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2007.

[64] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[65] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. 2007, pp. 153–170.

[66] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[67] G. E. Hinton, "Learning multiple layers of representation," *Trends Cognit. Sci.*, vol. 11, no. 10, pp. 428–434, Oct. 2007.

[68] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, Jan. 1991.

[69] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," *Queue*, vol. 6, no. 2, pp. 40–53, Mar./Apr. 2008.

[70] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24. 2012, pp. 1232–1240.

[71] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 873–880.

[72] Q. V. Le *et al.*, "Building high-level features using large scale unsupervised learning," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, U.K., 2012, pp. 1–8.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24. 2012, pp. 609–616.

[74] A.-R. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[75] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Feb. 2011.

[76] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, Feb. 2015.

[77] H. Y. Xiong *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, pp. 144–154, Jan. 2015.

[78] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[79] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted Boltzmann machine," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21. 2008, pp. 1601–1608.

[80] G. W. Taylor and G. E. Hinton, "Factored conditional restricted Boltzmann machines for modeling motion style," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1025–1032.

[81] A. Testolin, I. Stoianov, A. Sperduti, and M. Zorzi, "Learning orthographic structure with sequential generative neural networks," *Cognit. Sci.*, to be published, doi: 10.1111/cogs.12258.

[82] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[83] A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, and K. J. Friston, "Canonical microcircuits for predictive coding," *Neuron*, vol. 76, no. 4, pp. 695–711, Nov. 2012.

[84] K. Friston, R. A. Adams, L. Perrinet, and M. Breakspear, "Perceptions as hypotheses: Saccades as experiments," *Frontiers Psych.*, vol. 3(151), May 2012.

[85] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, 2010, p. 3.

[86] D. Munaretto, D. Zucchetto, A. Zanella, and M. Zorzi, "Data-driven QoE optimization techniques for multi-user wireless networks," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 653–657.

[87] A. Testolin *et al.*, "A machine learning approach to QoE-based video admission control and resource allocation in wireless systems," in *Proc. IEEE 13th Annu. Medit. Ad Hoc Netw. Workshop (MED-HOC-NET)*, Jun. 2014, pp. 31–38.

[88] A. Osyczka, "Multicriteria optimization for engineering design," in *Design Optimization*, J. Gero, Ed. Cambridge, MA, USA: Academic, 1985, pp. 193–227.

[89] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.

[90] N. Changuel, B. Sayadi, and M. Kieffer, "Control of multiple remote servers for quality-fair delivery of multimedia contents," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 746–759, Apr. 2014.

[91] L. Dal Col, S. Tarbouriech, L. Zaccarian, and M. Kieffer, "A linear consensus approach to quality-fair video delivery," in *Proc. IEEE 53rd Annu. Conf. Decision Control (CDC)*, Dec. 2014, pp. 5296–5301.

[92] Y. Li, Z. Li, M. Chiang, and A. R. Calderbank, "Content-aware distortion-fair video streaming in congested networks," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1182–1193, Oct. 2009.

[93] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern Recognit.*, vol. 35, no. 4, pp. 945–965, Apr. 2002.

[94] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19. 2007, pp. 1345–1352.

[95] L. Badia, D. Munaretto, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "Cognition-based networks: Applying cognitive science to multimedia wireless networking," in *Proc. IEEE 15th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2014, pp. 1–6.

[96] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," in *Proc. 12th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, 2014, pp. 656–661.

[97] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, 2003. [Online]. Available: https://www.itu.int/rec/T-REC-H.264-201402-I/en

[98] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[99] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Proc. 2nd IEEE Int. Workshop Quality Multimedia Exper. (QoMEX)*, Jun. 2010, pp. 29–34.

[100] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence," *Neural Comput.*, vol. 14, no. 6, pp. 1601–1621, Jun. 2009.

[101] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.

[102] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27. 2014, pp. 2654–2662.

[103] G. Ács, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, "Cache privacy in named-data networking," in *Proc. 33rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Philadelphia, PA, USA, Jul. 2013, pp. 41–51.

[104] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Can't you hear me knocking: Identification of user actions on Android apps via traffic analysis," in *Proc. 5th ACM SIGSAC CODASPY*, New York, NY, USA, Mar. 2015, pp. 297–304.

[105] B. Miller *et al.*, "Adversarial active learning," in *Proc. Workshop Artif. Intell. Secur. Workshop*, 2014, pp. 3–14.

[106] M. Danieletto, G. Quer, R. R. Rao, and M. Zorzi, "CARMEN: A cognitive networking testbed on Android OS devices," *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 98–107, Sep. 2014.

[107] M. De Filippo De Grazia, I. Stoianov, and M. Zorzi, "Parallelization of deep networks," in *Proc. ESANN*, 2012, pp. 621–626.

MICHELE ZORZI (S'89–M'95–SM'98-F'07) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padua, in 1990 and 1994, respectively. From 1992 to 1993, he was with the University of California at San Diego (UCSD). In 1993, he joined the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, as a Faculty Member. After spending three years with the Center for Wireless Communications, UCSD, in 1998 he joined the School of Engineering, University of Ferrara, Italy, where he became a Professor in 2000. Since 2003, he has been with the Information Engineering Department, University of Padua, as a Faculty Member. His current research interests include performance evaluation in mobile communications systems, wireless sensor networks and Internet of Things, and underwater communications. He was the Editor-in-Chief of the *IEEE Wireless Communications* from 2003 to 2005, and of the IEEE Transactions on Communications from 2008 to 2011, and is currently the Editor-in-Chief of the IEEE Transactions on Cognitive Communications and Networking. He served as a Member-at-large of the Board of Governors of the IEEE Communications Society from 2009 to 2011, and is currently its Director of Education and Training.

ANDREA ZANELLA (S'98–M'01–SM'13) received the Laurea degree in computer engineering and the Ph.D. degree in electronic and telecommunications engineering from the University of Padua, Italy, in 1998 and 2001, respectively. In 2000, he was a Visiting Scholar with the Department of Computer Science, University of California at Los Angeles. He is currently an Associate Professor with the Department of Information Engineering, University of Padua. He is one of the coordinators of the SIGnals and NETworking Research Laboratory. His long-established research activities are in the fields of protocol design, optimization, and performance evaluation of wired and wireless networks. He is also an Associate Editor of the IEEE Transactions on Cognitive Communications and Networking, the IEEE Internet of Things Journal, and the *Digital Communications and Networks Journal*.

ALBERTO TESTOLIN received the M.Sc. degree in computer science and the Ph.D. degree in cognitive science from the University of Padua, in 2011 and 2015, respectively. He is currently a Post-Doctoral Researcher with the University of Padua, focusing on computational modeling of cognitive processes. His main interests include deep learning, recurrent neural networks, and probabilistic generative models, which are applied to investigate visual processing and attentional mechanisms.

MICHELE DE FILIPPO DE GRAZIA is currently a Researcher with the Department of General Psychology, University of Padua, Italy. He is involved in the development of computational models (such as deep networks) of cognitive processes. His research interests are focused on machine learning techniques like neural networks and SVM algorithms to develop intelligent systems for industrial and biomedical applications.

MARCO ZORZI is currently a Full Professor of Cognitive Psychology and Artificial Intelligence with the Department of General Psychology and the Center for Cognitive Neuroscience, University of Padua, Italy. He is also a Senior Researcher and Consultant with San Camillo Neurorehabilitation Hospital, Venice-Lido. Research in his Laboratory, the Computational Cognitive Neuroscience Laboratory, crosses the frontiers between cognitive science, computer science, and neuroscience, with a focus on the computational bases of cognition, from development to skilled performance and to breakdowns of processing in a typical development or after brain damage. His research on generative models of human cognition has been supported by a Starting Grant for Frontier Research, grant 210922, from the European Research Council. Machine learning techniques, in particular, deep neural networks, are used in his laboratory for modeling human behavior and cognition, and for neuroinformatics and industry-related applications.