Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Perspective:

Artificial intelligence and wireless communications

Jun WANG[1], Rong LI[‡1], Jian WANG[1], Yi-qun GE[2], Qi-fan ZHANG[2], Wu-xian SHI[2]

[1]*Wireless Technology Laboratory, Huawei Technologies Co., Ltd., Hangzhou 310051, China*

[2]*Wireless Technology Laboratory, Huawei Technologies Co., Ltd., Ottawa K0A3M0, Canada*

E-mail: {justin.wangjun, lirongone.li, wangjian23, yiqun.ge, qifan.zhang, wuxian.shi}@huawei.com

Received Sept. 27, 2019; Revision accepted Mar. 27, 2020; Crosschecked May 18, 2020; Published online June 22, 2020

Abstract: The applications of artificial intelligence (AI) and machine learning (ML) technologies in wireless communications have drawn significant attention recently. AI has demonstrated real success in speech understanding, image identification, and natural language processing domains, thus exhibiting its great potential in solving problems that cannot be easily modeled. AI techniques have become an enabler in wireless communications to fulfill the increasing and diverse requirements across a large range of application scenarios. In this paper, we elaborate on several typical wireless scenarios, such as channel modeling, channel decoding and signal detection, and channel coding design, in which AI plays an important role in wireless communications. Then, AI and information theory are discussed from the viewpoint of the information bottleneck. Finally, we discuss some ideas about how AI techniques can be deeply integrated with wireless communication systems.

Key words: Wireless communications; Artificial intelligence; Machine learning
https://doi.org/10.1631/FITEE.1900527 **CLC number:** TN92

1 Introduction

Due to the development of advanced computing capabilities, progress in algorithms, and accessibility to big data, artificial intelligence (AI) is ushering in a new wave of technical revolution in human society (Russell and Norvig, 2002). Machine learning (ML), comprising a more specific subset of AI technologies, helps computers process and learn from data on their own. To date, machines aided by ML technologies have already been performing comparably or even better compared with human beings in many fields, such as voice and image recognition, game playing, semantic analysis, and drug discovery. Most of the aforementioned success has been brought about by the introduction of neural networks (NNs) inspired by the structure of the human brain. In the learning process, neurons are combined into certain structures to form a network, i.e., a training model; experience (data) is fed into the network to help determine the weight values between neurons. A target (either more reward or less penalty) is set up to facilitate the training. By doing so, features are extracted from data in a way similar to how human beings learn. With the help of advanced computing capabilities, the number of layers in NN can be large enough to form a powerful deep neural network (DNN). Generally, according to the different tasks and inputs/outputs, ML technologies can be categorized as follows:

1. Supervised learning builds a model upon a set of training data including inputs and their desired outputs (labels). After training, the model would try to predict the labels of the inputs that are not included in the training data set.

2. Unsupervised learning does not need explicit labels in the training data set. The trained model can extract the structure and identify commonalities

‡ Corresponding author
ORCID: Jun WANG, https://orcid.org/0000-0002-8127-9124; Rong LI, https://orcid.org/0000-0003-1040-1484

© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

in the data.

3. Reinforcement learning (RL) trains an agent to react to an environment to maximize long-term rewards. The agent interacts with the environment by setting action decisions based on the observed state and obtaining the new state and reward from the environment after the decision is executed.

All of these types of ML technologies can be combined with DNNs, which are called deep learning (DL) (Patterson and Gibson, 2017) and deep reinforcement learning (DRL) (Li YX, 2017).

As ML technologies continue to boom, wireless communication technologies are undergoing rapid development. Huge volumes of traffic with different quality-of-service (QoS) requirements are supposed to be handled by wireless communication networks. For instance, fifth-generation wireless communication networks (5G) are designed to support (1) enhanced mobile broadband (eMBB), which provides high per user data rates, (2) ultra-reliable low latency communications (uRLLC), which guarantees reliability within a constrained latency, and (3) massive machine-type communications (mMTC), which allows massive Internet of Things (IoT) equipment to access the network.

Along with the increasing requirements for diverse vertical services, there is an urgent need for an evolution in wireless communication networks. Many researchers have introduced DL and DRL in wireless networks from the upper layers down to the physical layer. For example, routing optimization in the network layer (Stampa et al., 2017), radio resource scheduling in the link layer (Xu ZY et al., 2017), and signal processing in the physical layer (O'Shea and Hoydis, 2017) have all shown the power of ML in wireless networks. In addition, ML is used in several special mobile network application scenarios, such as vehicle to vehicle (V2V), mobile edge computing (MEC), IoT, and eMBB. The mining, classification, recognition, prediction, and learning capabilities of ML will make wireless networks more intelligent, thereby introducing a new research domain called wireless AI.

Despite all of the ML technologies that are combined with wireless communications, there are a range of opinions about whether they are being carried out properly. Until now, nearly all the successful ML cases have been based on big data, e.g., natural language processing (NLP), face recognition, and game playing. DNNs can definitively extract the useful information and structures from a massive amount of data. Without mathematically formulating problems based on a thorough understanding of their models and inherent architectures, one can still obtain the solutions with DNNs. Missing the interpretative nature of DNNs, people regard DL and DRL as "alchemy" more than "chemistry." Therefore, the interpretative nature of DNNs is an on-going research topic.

Thorough summaries in the field of wireless AI can be found in Mao et al. (2018), Luong et al. (2019), Zappone et al. (2019), and Zhang CY et al. (2019). In this paper, we are trying to touch on some of the key points that are critical for making wireless AI useful in realistic systems. These key technical points correspond to the following fundamental open questions:

1. How to handle and use the wireless data, especially the physical layer data?

2. Are DNNs capable of replacing some traditional modules in wireless systems?

3. Can data-driven ML help with the design of some parts of wireless systems?

4. How can expert knowledge in communication theory help ML?

5. Can ML help with the design of a post-Shannon structure of communication systems?

To answer these questions, we introduce some examples in this paper. In Section 2, we introduce a realistic channel model built upon real channel data gathered from a field test. The channel model is an NN containing features learned from the channel data. It can be pre-trained and employed in different channel related downstream tasks, such as channel compression, channel fingerprinting, and mitigating pilot contamination. In Section 3, DNNs are used to carry out traditional signal processing tasks. Two examples are provided, one a DNN receiver for multiple access and the other a DNN channel decoder. After training, DNN can replace its counterparts in wireless systems. In Section 4, instead of replacing the traditional modules, ML is used to help with the design of a channel coding scheme. The design process can be offline. Once the design is finished, the coding scheme can be directly used in wireless systems without changing the encoder or decoder. In Section 5, a DRL-based scheduler is employed to make decisions about resource allocation. Several training

methods are tried, and it is found that using expert knowledge in training can speed up the convergence of training and help avoid local optima. In Section 6, AI and information theory are discussed from the viewpoint of using the information bottleneck to interpret DNNs. In Section 7, we discuss some ideas about how AI techniques can be deeply integrated with wireless communication systems.

2 AI-based channel modeling

A true and realistic wireless channel consists of both random and deterministic factors, and varies from one environment to another and from this moment to the next. It is a stochastic process. Thereby, the state of the art for a channel model was to build a stochastic model (Kermoal et al., 2002) to benchmark different transceiving algorithms rather than to predict a channel in a given environment. Such a stochastic channel model is an abstraction resulting from a number of statistical measurements over different environments, and is inevitably too narrow to predict any realistically true wireless channels.

If we had long before accurately predicted a time-varying channel in a realistic environment, we could have dramatically improved the system performance. We have long been puzzled by how to learn as much deterministic knowledge about a targeted channel as possible. The deterministic knowledge about a realistic true channel is related mainly to the geometry of the radio propagations such as reflections, refractions, diffractions, shadowing, and wave-guiding phenomena in a given environment. Therefore, the geometry of an environment would help infer a deterministic prediction of a channel in this environment (Hur et al., 2016).

Beyond 5G, our ambition is to resort to AI technologies to learn the full deterministic knowledge of a realistic channel and then make an accurate prediction of this time-varying channel. First, we capture and store a large amount of true channel data from true field tests. Second, we use a DL technique to train an NN model with the true data. Third, we infer some information such as channel feedback (Wen et al., 2018), localization (Decurninge et al., 2018), and channel prediction (Arnold et al., 2019) from this trained model to help with such tasks as scheduling, beamforming, and power control.

Huangfu et al. (2019b) proposed a self-supervised pre-training deep learning approach to model a wireless channel for various tasks, while protecting personal privacy not using labels on users' personal data. It is an autonomous supervised learning system in which NN learns some of the inherent deterministic properties of a wireless channel from an infinite number of labeled samples generated automatically. After that, the channel model represented by NN continues to be adjusted and tuned for a channel-dependent task, e.g., network optimization for downlink channels rather than learning from scratch, which leads to significant savings in the training effort.

Such a pre-trained channel model (NN) is a starter for a wide variety of channel-dependent tasks. It fulfills two basic functionalities. The first is to score the likelihood that a channel would occur in a given environment, or even to yield the most likely version in terms of the given environment. The second is to predict (interpolate or extrapolate) the channel in terms of the varying environment.

The performance of a pre-trained channel model depends largely on the neural network architecture. Previous studies were conducted with various architectures, such as the auto-regressive (AR) model (Sternad and Aronsson, 2003), complex-valued NNs (Ding and Hirose, 2014), and DNNs (Luo et al., 2020). Considering a realistic channel as a natural time series, we believe a recurrent neural network (RNN) to be a suitable neural network for the channel model. However, a single RNN can train an input sequence in only one direction, because the future elements are invisible in a time series. To solve this problem, we resort to sequence-to-sequence (seq2seq) models (Sutskever et al., 2014) for bi-directional training by introducing double RNNs, one for the encoder and the other for the decoder. The seq2seq models are successfully trained to translate one sequence from one domain (e.g., one sentence in English) to one sequence in another domain (e.g., voices in Mandarin). When adopting it to channel data training, we design it so that the RNN encoder is trained in the direction from the past to the future and the RNN decoder is trained in the direction from the future to the past to form a bi-directional way (Huangfu et al., 2019a).

In addition, the self-attention mechanism is more attractive than RNN in terms of computation performance for its higher parallelism, as proposed

along with the transformer architecture (Vaswani et al., 2017). It assumes that the inputs have equal influence on the outputs at start. Researchers should try for the most suitable architecture to pre-train a unified channel model.

However, as in any other neural network, a pre-trained channel model does not always guarantee the generalization. To improve its generalization ability for all kinds of realistic scenarios, more than one pre-trained channel model should be trained and deployed at different geographic levels. Fig. 1 illustrates multiple geographic levels: tracking area (TA), base station, and cellular levels for outdoor scenarios; office, corridor, elevator, and other levels for indoor scenarios. Although there are not two identical realistic environments, it is well observed that there is a strong correlation between channel models and geographic levels. For instance, different offices with similar geometric layouts may possess common channel features. Therefore, a pre-trained channel model built for office A may be directly used for office B. Or, only a modest training effort needs to be made to tune the neural network from offices A to B.

By training with the losses between the labeled data and the predicted outputs, the weights of the pre-trained model can be updated to maximize the log probabilities of the correct labels. This pre-trained channel model is capable of extracting the features of the input channel, which is also known as channel fingerprinting. With these features and some additional data, a task-specific channel model can be trained. Any channel-related task could be a downstream task, including (but not limited to) localization, beamforming, user pairing, and network optimization.

Let us take channel fingerprinting as an example to show the procedure for turning raw data into a final solution. In this experiment, we simulate the scenario of a man walking along the Bund, Shanghai, China and carrying a smart edge device. We show how the ML algorithms help this man in communication. Notice that this experiment is performed offline, but online edge training and prediction would be realistic in the future. The first step is data collection. The channel data is collected in the outdoor area of the Bund at walking speed, where the downlink cell reference signal (CRS) is measured for a commercial long-term evolution (LTE) system with a 2×2 antenna. The system bandwidth is 20 MHz, i.e., 100 resource blocks (RB). In one subframe, 200 subcarriers and four symbols are used for pilot allocation for each port. The carrier frequency is 1.9 GHz and the crest factor reduction (CFR) is estimated from pilot symbols. The channel data is used for decoding and discarded in the absence of ML algorithms; therefore, it is easy to collect. The second step is training, whereby a self-attention-based architecture is used for self-supervised pre-training, and channel data in the next subframe is used as labels for this subframe. With training, the channel model can have a good understanding of the current channel. The third step is prediction, where we can input any channel data into the pre-trained channel model and obtain the extracted fingerprints of the channel. The fingerprints are ready for further prediction tasks, which can help the wireless device with random access, localization, and network optimization. The environmental information would be learned as latent information within the high-dimensional vector of these channel fingerprints. If we draw the fingerprints in 20 ms every five seconds eight times, it can be observed that when the user is moving, the fingerprints are also slowly changing. Low-dimensional representations of channel fingerprints using t-distributed stochastic neighbor embedding (t-SNE) are shown in Fig. 2. We can observe

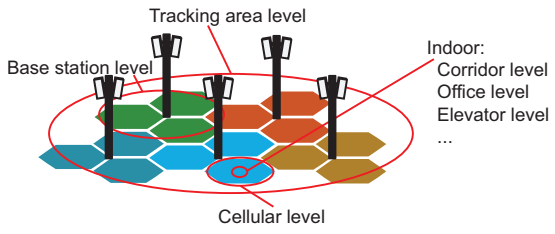


Fig. 1 Schematic of realistic channel models at different levels

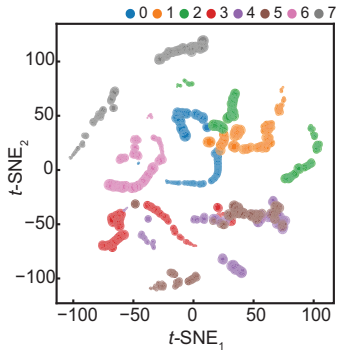


Fig. 2 Channel charting of low-dimensional representations of channel fingerprints using t-SNE

contiguous curves in the 20 ms duration, and the curves are separated for eight sets, which is consistent with the change of the channel features. Following the time flows, the sizes of labels and markers are increasing. It shows that the geometrical information is learned well, and for realistic channel tasks, these fingerprints should be helpful in predicting a final solution.

3 AI-based channel decoding and signal detection

The most intuitive idea in applying DNNs in wireless networks is to replace the traditional building blocks of the systems. In this section, we provide two examples, i.e., channel decoding and signal detection. Both examples belong to the jointly optimized end-to-end system that can exploit the effectiveness and low latency of DNNs with the ability of modeling imperfection/non-linearity in systems.

3.1 Channel decoding

Error correction code (ECC) is used in wireless communications against a noisy channel. A decoder tries to find the correct codeword from all legal ones according to the corrupted transmitted signal. In this sense, a decoder can be regarded as a classifier, where deep learning may reveal its advantages.

Gruber et al. (2017) proposed a multi-layer perceptron (MLP) based one-shot channel decoder. In this approach, an MLP is trained as a black-box and it does not exploit expert knowledge as in traditional coding theory. In the decoding stage, the corrupted codeword is fed into the well-trained MLP-based decoder, which then outputs one classification result among $2K$ candidate codewords. The evaluation results show that a well-trained neural network decoder can reach the maximum a posteriori (MAP) performance. For this approach, there are still some challenges that need further study. On one hand, the "generalization" enables a neural network to accurately infer a larger set from samples on the subset. In case of forward error correction (FEC) code, assume that a neural network decoder has already been well trained with codeword-(n) and codeword-$(n+2)$ while never with codeword-$(n+1)$. If codeword-$(n+1)$ is a natural transition from codeword-(n) to codeword-$(n+2)$, it may be possible for a neural network to interpolate and then learn codeword-$(n+1)$. However, this is generally not the case for FEC since one of the FEC design goals is to increase the code distance; that is, any two codewords should be separated away as far as possible. This observation explains why the MLP-based decoder needs to be trained with the whole set of all possible codewords (size $2K$) (Gruber et al., 2017). For this reason, it is challenging to apply the approach to longer codes. On the other hand, a typical FEC needs to support thousands of combinations of coding rates and lengths. Even for the same code, it may work in various channel conditions, i.e., with different signal noise ratios (SNR). Consider a case in which an MLP network is trained in one scenario (given coding rate, length, and SNR) and then used as the decoder in another scenario. In such a case, the decoding performance will degrade since the statistical characteristics of the input symbols change. This also complicates the training of the MLP network to fit all the required scenarios and needs further study.

A "soft" Tanner graph based decoder proposed in Nachmani et al. (2016) is a new approach for exploiting AI for channel decoding. The network is constructed based on the parity check matrix (expert knowledge) that describes the code. A well-trained network may help improve the belief propagation (BP) algorithm performance for high density parity codes (HDPC). NN has the same structure as the unfolding Tanner graph of an L-iteration BP algorithm. The neural nodes represent the edges in the Tanner graph, and the connections between neural nodes correspond to the message-passing process between variable and check nodes. By properly weighting the reliability of the message, the small cycle effect can be mitigated and the approximation error can be compensated for in the mean time by adopting the min-sum algorithm (Nachmani et al., 2018). Furthermore, this expert knowledge based structure maintains the property of the BP algorithm, so that NN can be trained using only a zero codeword. The decoding performance is improved due to the aforementioned reasons; however, there exists a performance gap between NN and the maximum likelihood algorithm.

For sequential codes like convolutional and turbo codes, an RNN architecture was designed in Kim et al. (2018). Since the encoder of sequential codes can be represented as a hidden Markov model (HMM), traditional decoder algorithms such

as Viterbi and BCJR can efficiently solve the HMM problem using dynamic programming with linear time complexity. Due to the similarity between RNN and HMM, a two-layer gated recurrent unit (GRU) is adopted to solve HMM. Intuitively, the expressive capacity of RNN outperforms that of HMM, so the neural decoder for convolutional codes shows strong generalization ability in code length. However, neither the neural convolutional decoder nor the turbo decoder in Kim et al. (2018) improves the state-of-the-art ECC performance.

3.2 Signal detection

Similar to decoding, a receiver can use the deep learning approach to detect a signal. Farsad and Goldsmith (2017) focused on the detection algorithms for molecular communication, in which the mathematical channel model was difficult to obtain. They evaluated several deep learning based algorithms and traditional methods with experimental data. The results revealed that the sequence detection algorithm outperformed the traditional symbol-by-symbol detection algorithms due to inter-symbol interference (ISI).

Herein, we explore another example of a deep learning based algorithm that deals with multiple-user interferences (MUI). In a typical multiple access communication system over a multi-path fading channel, a receiver can use a bi-directional long short-term memory (bi-LSTM) based neural network to mitigate ISI and MUI (Fig. 3).

In Fig. 3, the correlators discriminate the signals from multiple users. By sliding the processing window over the received chip samples, we have the matched filter outputs for each user, which are then equally sliced into M segments, where M is the number of symbols. All users' slices in each segment are concatenated before being sent to a bi-LSTM layer with corresponding M cells. After multiplication with each user's estimated channel impulse response, another bi-LSTM layer is used, finally outputting the probability distribution of all users' reconstructed bits.

The correlator layer is a kind of convolutional neural network layer whose kernel weights are either trained or predefined by exploiting the spatial correlation of the received signal. Two-layer bi-LSTM structures are adopted to deal with residual ISI and MUI. The estimated channel information between LSTM layers is an indispensable part for good performance of the receiver. In Fig. 4, the average bit error rate (BER) performances of a conventional rake receiver and neural receiver for different numbers of users are compared ($K = 4$ and $K = 8$). As expected, the DL-based receiver successfully achieves interference mitigation capability, and thus the performance is significantly enhanced.

4 AI-based channel coding design

AI technologies can aid the design of a communication scheme. In this section, we take channel coding design as an example.

ECC has been widely used in communication systems for data transmission over unreliable or noisy channels. A classic channel coding design is

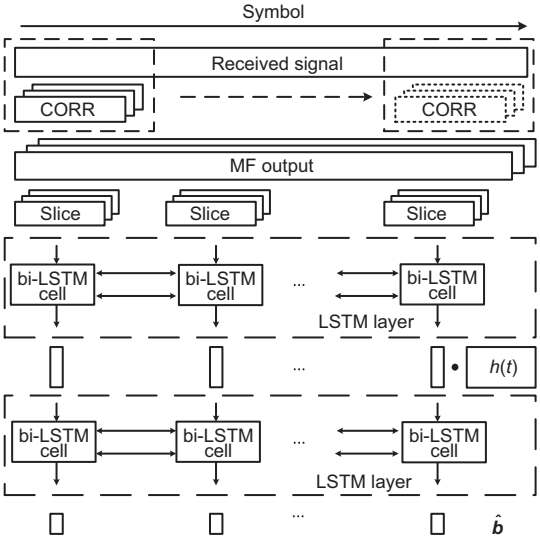


Fig. 3 Structure of the proposed multi-user neural receiver

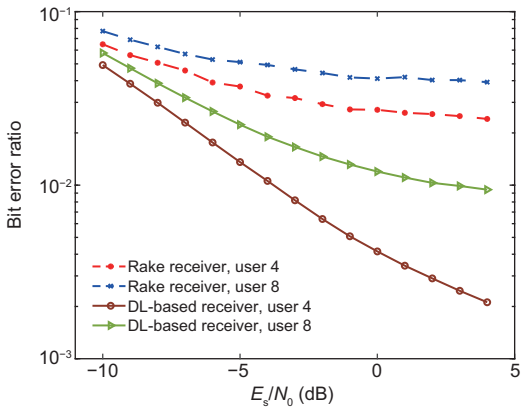


Fig. 4 Performance comparison with the rake receiver

based upon coding theory, in which the code performance is analytically derived in terms of various types of code properties, such as the Hamming distance (maximum distance separable (MDS) codes), free distance (conventional codes), and decoding threshold (low-density parity check (LDPC) codes). As shown in Fig. 5 (left branch), these methods rely on the coding theory in predicting the code performance in terms of code construction. To optimize these code properties, the code is constructed and designed.

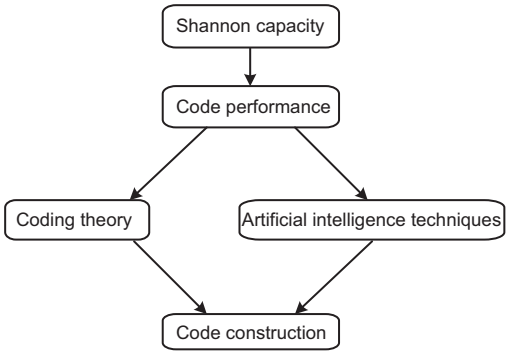


Fig. 5 Error correction code (ECC) design logic

The first AI attempt to "learn" to design code is the "constructor-evaluator" framework (Huang et al., 2019) (Fig. 6). It provides an alternative solution when expert knowledge fails to achieve optimal performance. The framework consists of two parts, i.e., a code constructor and a code evaluator. During the learning process, the code constructor iteratively learns a series of valid code constructions based on the performance metric feedback from the code evaluator. More precisely, the code constructor knows neither the internal mechanism nor the channel condition adopted by the code evaluator. It requests that the code evaluator feeds back an accurate performance metric of its current code construction under the evaluator-defined environment. In this way, the exploration of possible code constructions opens up a wide range of decoding algorithms and channel conditions.

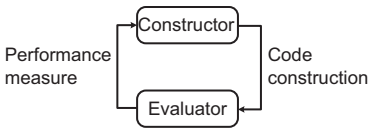


Fig. 6 Constructor-evaluator framework

4.1 Polar codes with a fixed length and rate

Polar codes can be defined by $\boldsymbol{c}=\boldsymbol{u}\boldsymbol{G}$ (Arikan, 2011), where $\boldsymbol{u}$ is the information bit sequence and $\boldsymbol{c}$ is the codeword. The polar transformation matrix is represented by $\boldsymbol{G}=\boldsymbol{F}^{\otimes n}$, where $\boldsymbol{F}$ denotes the kernel and $\otimes$ denotes the Kronecker power.

One of the key aspects to construct polar codes is to determine a binary vector $\boldsymbol{s}$ of length N, in which 1 denotes an information sub-channel and 0 denotes a frozen sub-channel. The optimal vectors are distinct for different decoders, among which successive cancellation (SC) decoders are intensively studied. Even though the successive cancellation list (SCL) decoders have been widely deployed and various heuristic construction methods proposed (Li B et al., 2014; Trifonov and Miloslavskaya, 2015; Wang T et al., 2016; Qin et al., 2017; Zhang HZ et al., 2018), a rigorous performance analysis is still lacking.

A genetic algorithm may be applied to construct polar codes for such decoders. The information sub-channels in code construction play the same role as chromosomes in a genetic algorithm, because they both individually and collaboratively contribute to the fitness of a candidate solution. Since good code constructions are constituted by good sub-channels, a pair of good parent code constructions is likely to produce good offspring code constructions. This drives the genetic algorithm to ultimately converge to a good code construction.

With this algorithm, good polar codes can be learned for SC, SCL path metrics (PM), and SCL genie decoders. For SCL decoders, the learned codes may even outperform existing ones. One interesting observation is that the learned construction may violate the universal partial order (UPO), which shows that UPO only applies in theory to the SC decoder, rather than to the SCL decoder.

4.2 Polar codes within a range of code rates

Due to description and implementation simplicity, polar code constructions with a nested property bear practical significance. Polar codes for code length N and dimension range $[K_{\mathrm{l}}, K_{\mathrm{h}}]$ may be defined as a reliability ordered sequence of length N. The corresponding polar code can be determined by reading out the first (or last) K entries from the ordered sequence to form the information position set.

Compared with the fixed (N, K) case, the nested code constraint makes the optimization more complicated for classical coding theory. The optimal constructions for different (N, K) polar codes do not necessarily constitute a nested sequence. In other words, the performance of some (N, K) codes needs to be compromised for the nested property. Therefore, during the construction of the reliability ordered sequence, a trade-off exists between the short-term reward (from the next state construction) and long-term reward (from the construction of a state that is a few steps away).

With AI techniques, the procedure is to design a nested code that may be modeled by a multi-step Markov decision process (MDP). Specifically, for each design step, a new sub-channel (action) is selected to obtain the $(N, K+1)$ polar code (an updated state) with a given (N, K) polar code (current state). The reliability ordered sequence is constructed by sequentially appending the actions to the end of the initial polar code construction. The problem of maximizing the total reward, which consists of both short-term and long-term rewards, can be naturally solved by the advantage actor-critic (A2C) algorithm. Both the actor and critic functions are implemented by neural networks, and then trained by the mini-batch-based stochastic gradient descent method according to the A2C algorithm.

Fig. 7 compares the relative SNR to achieve a target block error rate (BLER) level of 10^{-2}. A better overall performance can be observed for the learned polar codes, with the largest gain over 0.5 dB over density evolution/Gaussian approximation (DE/GA).

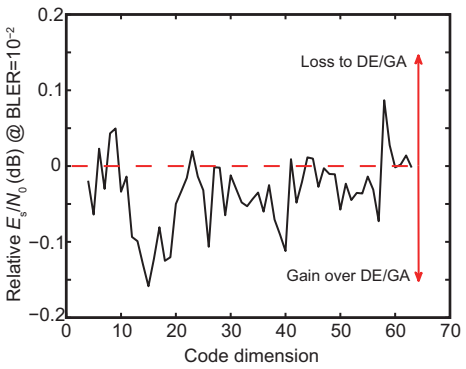


Fig. 7 Relative performance between polar codes constructed by reinforcement learning and DE/GA under SCL decoding

5 AI-based scheduling

In this section, we introduce a DRL-based scheduler that allocates resources among user equipment (UE) in cellular networks. Three methodologies are tried during the training, showing that involving expert knowledge in NN training improves convergence and avoids the local optima (Wang J et al., 2019; Xu C et al., 2019).

5.1 Network scenario and problem formulation

In a single cell cellular network, the base station (BS) is equipped with a DRL-based scheduler to allocate a single resource block group (RBG) among multiple UEs. Its link adaption, feedback, and scheduling mechanisms follow the LTE standard. Hence, the minimum resource allocation unit is 1 ms time interval and one RBG in a frequency interval. For analytical simplicity, we narrow it down to one RBG and one spatial layer, in which the famous proportional fair (PF) scheduling algorithm is proved an optimal trade-off between throughput and fairness (Kelly, 1997; Tse, 2001). Thus, the PF scheduler is the baseline for DRL.

A scheduling problem for a set of UEs $\mathcal{N}$ is formulated into an MDP, which can be solved by DRL. MDP is elaborated as follows:

1. The instantaneous rate $I_n(t)$ and average rate $T_n(t)$ for UE n ($n \in \mathcal{N}$) are included in the state S^t. $T_n(t)$ is updated by

$$T_n(t) = \frac{W-1}{W} T_n(t-1) + \frac{1}{W} I_n(t), \qquad (1)$$

where W is the averaged window size. Eq. (1) indicates the influence of the previous actions by $I_n(t)$; thus, the transition from states S^t to S^{t+1} is Markovian.

2. In the case of a single RBG scheduling scenario, the action is to decide which UE unit to occupy that unique RBG in each scheduling period (i.e., transmission time interval (TTI) in LTE). Similar to the PF algorithm, the action output A^t of a DRL scheduler has the metrics for each piece of UE. Naturally, the DRL scheduler would choose the one with the largest metric in each TTI.

3. The metric of an action is the reward design in DRL, because it has profound impacts on both system performance and convergence. In this paper, the total throughput and UE fairness are jointly

considered into the reward. The UE fairness is Jain's fairness index (JFI) (Jain et al., 1984) calculated by

$$\text{fairness} = \left(\sum_{n=1}^{N} V_n\right)^2 \Bigg/ \left(N \sum_{n=1}^{N} V_n^2\right), \quad (2)$$

where V_n ($n \in \mathcal{N}$) represents the received average throughput of the n^{th} UE, and N is the total number of UEs. Details of the reward calculation will be explained and elaborated on later from a learning methodology perspective.

5.2 Learning methodology

We have tried three learning methodologies for DRL, as shown in Fig. 8.

In a direct-learning methodology, the DRL agent directly learns from the interactions with the environment, and the reward function is set as a linear weighted sum of throughput and fairness as

$$\text{reward} = \alpha \cdot \text{throughput} + \beta \cdot \text{fairness}, \quad (3)$$

where α and β are used to adjust the weights of throughput and fairness in the reward, respectively. Hence, if throughput is considered more important than fairness, we can use a larger α/β value, and conversely, a small α/β value means fairness is the factor with which we are more concerned. Its convergence speed is relatively slow, and the agent may quickly fall into the local optima.

In a dual-learning methodology, two DRL agents are trained iteratively and competitively. During the training, one agent is fixed and the other is trained to outperform the fixed one. The reward is obtained through the comparison between the two agents (Table 1). Although the two DRL agents can compete with each other to reduce the risk of local optima, the convergence speed is still low.

In an expert-learning methodology, the DRL agent is under the supervision of a PF coacher. The agent obtains a positive reward when beating the coacher, and otherwise a zero reward. The reward design is shown in Table 2. With the help of the coacher (expert knowledge), the DRL agent can quickly converge without falling into local optima.

The parameters α and β introduced in this section should be carefully tuned during the training of the NNs, so that a desired trade-off between throughput and fairness can be achieved. From the trials on these three methodologies, we find that expert knowledge from the original communication theory contributes to the efficiency when adopting ML into wireless networks.

6 AI and information theory

Although AI and deep learning based approaches have exceeded human performance in many particular tasks, such as image classification, speech understanding, and strategic game playing, there are still some drawbacks preventing the wide success of AI. DNN is data hungry for good generalization, the

Table 1 Dual-learning reward look-up table

Throughput	Fairness	Reward
Agent 1 > Agent 0	Agent 1 > Agent 0	$\alpha+\beta$
Agent 1 > Agent 0	Agent 1 ≤ Agent 0	α
Agent 1 ≤ Agent 0	Agent 1 > Agent 0	β
Agent 1 ≤ Agent 0	Agent 1 ≤ Agent 0	0

Table 2 Expert-learning reward look-up table

Throughput	Fairness	Reward
DRL > PF	DRL > PF	$\alpha+\beta$
DRL > PF	DRL < PF	α
DRL < PF	DRL > PF	β
DRL < PF	DRL < PF	0
DRL > PF	DRL = PF	$\alpha+0.5\beta$
DRL = PF	DRL > PF	$0.5\alpha+\beta$
DRL = PF	DRL = PF	$0.5(\alpha+\beta)$
DRL = PF	DRL < PF	0.5α
DRL < PF	DRL = PF	0.5β

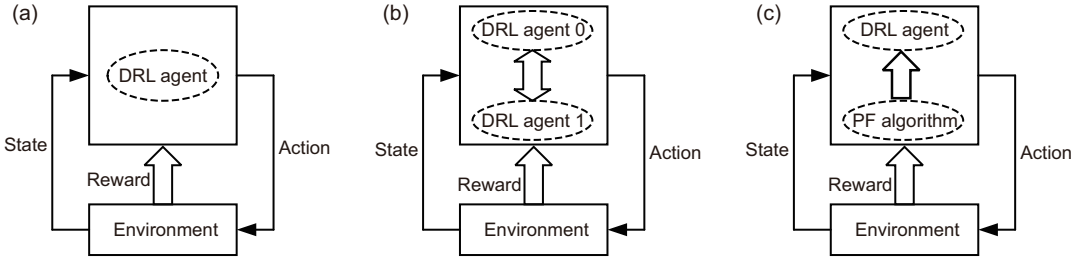


Fig. 8 Learning methodologies: (a) direct learning; (b) dual learning; (c) expert learning

training is not efficient due to the huge neural network, and most importantly, such AI approaches are usually applied in a "black box" manner. A strong theoretical foundation is needed for AI, especially when used for essential tasks, based on which all the above envisioned technologies can become a reality. Information theory has great potential in offering a solid mathematical foundation for the problems of machine learning and in allowing establishment of the fundamental limits of learning in terms of optimal trade-offs between accuracy and complexity.

To accelerate the development of AI, intensive efforts have been made to explain how AI works, namely, Bayesian reasoning (Barber, 2012), visual interpretation (Zhang QS and Zhu, 2018), and the transparency and post-hoc interpretation (Mittelstadt et al., 2019). Except for these approaches, information-theoretical techniques have been tried to explain or improve AI for autoencoder training (Zhao et al., 2018), interpreting representation learning in the generative adversarial network (GAN) (Chen et al., 2016), and Q-learning (Grau-Moya et al., 2018). This direction may be more promising because the learning itself tries to extract information from the data.

Recently, the information bottleneck (IB) method has attracted considerable attention again from both the information theory and computer science communities (Tishby et al., 2000). As shown in Fig. 9, the network uses the latent variable T to represent the noisy input data X, retaining only the features most relevant to general concepts Y. IB theory lies at the frontier between machine learning and prediction, statistics, and information theory. Normally, its function is $\min\left(I(X,T)-\beta I(Y,T)\right)$, where the trade-off parameter β balances the compression level $I(X,T)$ and the amount of captured relevant information $I(Y,T)$. Based on the mutual information plane, Shwartz-Ziv and Tishby (2017) demonstrated that a fitting-compression phase transition exists when training the network. Amjad and Geiger (2019) discussed the main issues of using the IB

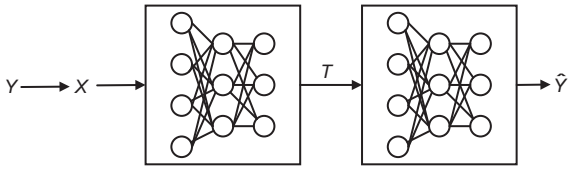


Fig. 9 Information bottleneck in a deep neural network

function, and presented potential solutions, such as including the decision rules and introducing noise and quantization. On the other hand, there is extensive research that focuses on using IB as the optimization objective or regularization to improve the stability and generalization. For instance, Alemi et al. (2016) used variational inference to construct a lower bound on the IB objective. It demonstrated that models trained with such an objective outperform other methods in terms of generalization performance and robustness to adversarial attack. Estella-Aguerri and Zaidi (2019) generalized the centralized IB methods to the setting of multiple distributed encoders, while Achille and Soatto (2018) considered the IB principle and minimized its associated Lagrangian for non-Gaussian high-dimensional continuous random variables.

7 Future of wireless AI

For the next-generation wireless communication systems, a wide range of services with various devices should be supported. For instance, uRLLC and mMTC are newly introduced in 5G; one of the main driving forces behind 5G is the need for massive machine-type communications. Meanwhile, with the rapid progress of AI technologies, machine learning capability will be everywhere, including in the devices, at the edge, and on the cloud, which translates the communication from connected ion-of-things to connection-of-intelligence. A deep integration between AI and communications will surely emerge which contains two aspects, AI-enabled wireless communications and wireless communications for AI.

The first aspect means that AI works as an indispensable tool to facilitate the intelligence in wireless communications. In classical Shannon's communication theory, the communication objective is to reproduce at the destination the message generated at the source, which is the so-called communication at level A (Shannon and Weaver, 1949). Keeping the objective in mind, Shannon defined the metric "entropy" to measure the information, developed source coding and channel coding theorems, and proved that separation of source and channel coding is as good as joint processing. These conclusions and assumptions provided high-level guidelines for practical implementations for communication system designs in the past decades. With AI technologies

involved, wireless communication systems can evolve at least in three directions. First, wireless big data analytics can be applied to future wireless communication systems based on the AI technologies. Useful features can be mined and future states can be predicted from historical data, which can help with system design, fault detection, and performance tuning. Second, AI technologies can be used to optimize many physical layer modules for wireless link. Recent advances in AI provide many useful tools to handle network optimization problems with non-convexity and large scale, which are hard to solve through the traditional methods. Third, AI technologies play a critical role in refining the end-to-end chain for wireless communications. It is possible that traditional building blocks can be fully replaced by different types of NNs, which are trained together to achieve a global optimum.

For the second aspect, most of the AI tasks are computationally intensive, and wireless communications can help handle AI applications in a distributed way with low latency and fulfill the requirements of critical privacy and security. In the future, an increasing number of intelligent applications will be deployed at the edge of wireless networks. Due to cost limits, smart mobile terminals or IoT devices usually cannot provide enough computational capability to handle intelligent applications. In this situation, wireless communications between terminals and central/edge servers are needed; then the capacity and latency of these wireless links become the key bottlenecks. Future wireless communication systems should be designed with the consideration that distributed computation and data storage with privacy protection are supported to enable distributed AI applications. For instance, federated learning (Konečný et al., 2016) is introduced to allow distributed training among multiple mobile devices with the data kept on each device to protect privacy. The bandwidth allocation and scheduling among all the mobile devices should be carefully designed. Another example is that for distributed inference, ultra-low latency communication capability is required, so that the decisions can be obtained in time on each distributed terminal.

The key performance indicators (KPIs) for future wireless communication systems will keep on advancing in diverse directions. For example, the peak data rate will increase to 1 Tb/s and the latency will decrease to 0.1 ms. More kinds of devices will be connected to participate in more emerging applications, including smart cities, self-driving cars, connected infrastructure, seamless virtual and augmented reality, and space-air-ground integrated networks. A deep integration between wireless communications and AI technologies will play an indispensable role in building this intelligent world.

8 Conclusions

In this paper, we provided some examples about possible usage of AI in wireless networks, such as using wireless data to build a realistic channel model, replacing channel coding and signal detectors with DNNs, assisting channel coding design, and helping with resource scheduling. Then, IB theory was introduced, where DNNs were interpreted through the point of view of information theory . Through these examples, it can be seen that AI technologies are indispensable tools for not only improving the performance of existing wireless systems but also defining future wireless networks. As shown in Section 7, the deep integration of AI and wireless communications is very important for the future intelligent world; it is still at the starting phase and greater efforts are needed.

Contributors

Jun WANG and Rong LI guided the research. Yi-qun GE, Qi-fan ZHANG, and Wu-xian SHI collected the references. Rong LI, Jian WANG, and Yi-qun GE drafted the manuscript. Rong LI, Jian WANG, and Jun WANG revised and finalized the paper.

Compliance with ethics guidelines

Jun WANG, Rong LI, Jian WANG, Yi-qun GE, Qi-fan ZHANG, and Wu-xian SHI declare that they have no conflict of interest.

References

Achille A, Soatto S, 2018. Information dropout: learning optimal representations through noisy computation. *IEEE Trans Patt Anal Mach Intell*, 40(12):2897-2905. https://doi.org/10.1109/TPAMI.2017.2784440

Alemi AA, Fischer I, Dillon JV, et al., 2016. Deep variational information bottleneck. https://arxiv.org/abs/1612.00410

Amjad RA, Geiger BC, 2019. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Trans Patt Anal Mach*

Intell, in press.
https://doi.org/10.1109/TPAMI.2019.2909031

Arikan E, 2011. Systematic polar coding. *IEEE Commun Lett*, 15(8):860-862.
https://doi.org/10.1109/LCOMM.2011.061611.110862

Arnold M, Dörner S, Cammerer S, et al., 2019. Enabling FDD massive MIMO through deep learning-based channel prediction. https://arxiv.org/abs/1901.03664v1

Barber D, 2012. Bayesian Reasoning and Machine Learning. Cambridge University Press, Cambridge, UK.

Chen X, Duan Y, Houthooft R, et al., 2016. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. Proc 30th Int Conf on Neural Information Processing Systems, p.2180-2188.

Decurninge A, Ordóñez LG, Ferrand P, et al., 2018. CSI-based outdoor localization for massive MIMO: experiments with a learning approach. Proc 15th Int Symp on Wireless Communication Systems, p.1-6.
https://doi.org/10.1109/ISWCS.2018.8491210

Ding TB, Hirose A, 2014. Fading channel prediction based on combination of complex-valued neural networks and chirp Z-transform. *IEEE Trans Neur Netw Learn Syst*, 25(9):1686-1695.
https://doi.org/10.1109/TNNLS.2014.2306420

Estella-Aguerri I, Zaidi A, 2019. Distributed variational representation learning. *IEEE Trans Patt Anal Mach Intell*, in press.
https://doi.org/10.1109/TPAMI.2019.2928806

Farsad N, Goldsmith A, 2017. Detection algorithms for communication systems using deep learning.
https://arxiv.org/abs/1705.08044

Grau-Moya J, Leibfried F, Vrancx P, 2018. Soft Q-learning with mutual-information regularization. 7th Int Conf on Learning Representations, p.1-19.

Gruber T, Cammerer S, Hoydis J, et al., 2017. On deep learning-based channel decoding. 51st Annual Conf on Information Sciences and Systems, p.1-6.
https://doi.org/10.1109/CISS.2017.7926071

Huang LC, Zhang HZ, Li R, et al., 2019. AI coding: learning to construct error correction codes. *IEEE Trans Commun*, 68(1):26-39.
https://doi.org/10.1109/TCOMM.2019.2951403

Huangfu YR, Wang J, Li R, et al., 2019a. Predicting the mumble of wireless channel with sequence-to-sequence models. Proc 30th Annual Int Symp on Personal, Indoor and Mobile Radio Communications, p.1-7.
https://doi.org/10.1109/PIMRC.2019.8904286

Huangfu YR, Wang J, Xu C, et al., 2019b. Realistic channel models pre-training. https://arxiv.org/abs/1907.09117

Hur S, Baek S, Kim B, et al., 2016. Proposal on millimeter-wave channel modeling for 5G cellular system. *IEEE J Sel Top Signal Process*, 10(3):454-469.
https://doi.org/10.1109/JSTSP.2016.2527364

Jain RK, Chiu DMW, Hawe WR, 1984. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. Technical Report No. DEC-TR-301, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, USA.

Kelly F, 1997. Charging and rate control for elastic traffic. *Eur Trans Telecommun*, 8(1):33-37.
https://doi.org/10.1002/ett.4460080106

Kermoal JP, Schumacher L, Pedersen KI, et al., 2002. A stochastic MIMO radio channel model with experimental validation. *IEEE J Sel Areas Commun*, 20(6):1211-1226. https://doi.org/10.1109/JSAC.2002.801223

Kim H, Jiang YH, Rana R, et al., 2018. Communication algorithms via deep learning.
https://arxiv.org/abs/1805.09317

Konečný J, McMahan HB, Yu FX, et al., 2016. Federated learning: strategies for improving communication efficiency. https://arxiv.org/abs/1610.05492

Li B, Shen H, Tse D, 2014. A RM-polar codes.
https://arxiv.org/abs/1407.5483

Li YX, 2017. Deep reinforcement learning: an overview.
https://arxiv.org/abs/1701.07274

Luo CQ, Ji JL, Wang QL, et al., 2020. Channel state information prediction for 5G wireless communications: a deep learning approach. *IEEE Trans Netw Sci Eng*, 7(1):227-236.
https://doi.org/10.1109/TNSE.2018.2848960

Luong NC, Hoang DT, Gong SM, et al., 2019. Applications of deep reinforcement learning in communications and networking: a survey. *IEEE Commun Surv Tutor*, 21(4):3133-3174.
https://doi.org/10.1109/COMST.2019.2916583

Mao Q, Hu F, Hao Q, 2018. Deep learning for intelligent wireless networks: a comprehensive survey. *IEEE Commun Surv Tutor*, 20(4):2595-2621.
https://doi.org/10.1109/COMST.2018.2846401

Mittelstadt B, Russell C, Wachter S, 2019. Explaining explanations in AI. Proc Conf on Fairness, Accountability, and Transparency, p.279-288.
https://doi.org/10.1145/3287560.3287574

Nachmani E, Be'ery Y, Burshtein D, 2016. Learning to decode linear codes using deep learning. 54th Annual Allerton Conf on Communication, Control, and Computing, p.341-346.
https://doi.org/10.1109/ALLERTON.2016.7852251

Nachmani E, Marciano E, Lugosch L, et al., 2018. Deep learning methods for improved decoding of linear codes. *IEEE J Sel Top Signal Process*, 12(1):119-131.
https://doi.org/10.1109/JSTSP.2017.2788405

O'Shea T, Hoydis J, 2017. An introduction to deep learning for the physical layer. *IEEE Trans Cogn Commun Netw*, 3(4):563-575.
https://doi.org/10.1109/TCCN.2017.2758370

Patterson J, Gibson A, 2017. Deep Learning: a Practitioner's Approach. O'Reilly Media, Inc., Sebastopol, USA.

Qin MH, Guo J, Bhatia A, et al., 2017. Polar code constructions based on LLR evolution. *IEEE Commun Lett*, 21(6):1221-1224.
https://doi.org/10.1109/LCOMM.2017.2656126

Russell S, Norvig P, 2002. Artificial Intelligence: a Modern Approach. Prentice Hall, Upper Saddle River, NJ, USA.

Shannon CE, Weaver W, 1949. The Mathematical Theory of Communication. University of Illinois Press, Urbana, USA.

Shwartz-Ziv R, Tishby N, 2017. Opening the black box of deep neural networks via information.
https://arxiv.org/abs/1703.00810

Stampa G, Arias M, Sánchez-Charles D, et al., 2017. A deep-reinforcement learning approach for software-defined networking routing optimization. https://arxiv.org/abs/1709.07080

Sternad M, Aronsson D, 2003. Channel estimation and prediction for adaptive OFDM downlinks [vehicular applications]. Proc 58th Vehicular Technology Conf, p.1283-1287. https://doi.org/10.1109/VETECF.2003.1285229

Sutskever I, Vinyals O, Le QV, 2014. Sequence to sequence learning with neural networks. Proc 27th Int Conf on Neural Information Processing Systems, p.3104-3112.

Tishby N, Pereira FC, Bialek W, 2000. The information bottleneck method. https://arxiv.org/abs/physics/0004057

Trifonov P, Miloslavskaya V, 2015. Polar subcodes. *IEEE J Sel Areas Commun*, 34(2):254-266. https://doi.org/10.1109/JSAC.2015.2504269

Tse D, 2001. Multiuser Diversity in Wireless Networks. Wireless Communications Seminar, Standford University, Stanford, CA, USA.

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wang J, Xu C, Huangfu YR, et al., 2019. Deep reinforcement learning for scheduling in cellular networks. Proc 11th Int Conf on Wireless Communications and Signal Processing, p.1-6. https://doi.org/10.1109/WCSP.2019.8927868

Wang T, Qu DM, Jiang T, 2016. Parity-check-concatenated polar codes. *IEEE Commun Lett*, 20(12):2342-2345. https://doi.org/10.1109/LCOMM.2016.2607169

Wen CK, Shih WT, Jin S, 2018. Deep learning for massive MIMO CSI feedback. *IEEE Wirel Commun Lett*, 7(5):748-751. https://doi.org/10.1109/LWC.2018.2818160

Xu C, Wang J, Yu TH, et al., 2019. Buffer-aware wireless scheduling based on deep reinforcement learning. https://arxiv.org/abs/1911.05281

Xu ZY, Wang YZ, Tang J, et al., 2017. A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs. IEEE Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC.2017.7997286

Zappone A, di Renzo M, Debbah M, 2019. Wireless networks design in the era of deep learning: model-based, AI-based, or both? *IEEE Trans Commun*, 67(10):7331-7376. https://doi.org/10.1109/TCOMM.2019.2924010

Zhang CY, Patras P, Haddadi H, 2019. Deep learning in mobile and wireless networking: a survey. *IEEE Commun Surv Tutor*, 21(3):2224-2287. https://doi.org/10.1109/COMST.2019.2904897

Zhang HZ, Li R, Wang J, et al., 2018. Parity-check polar coding for 5G and beyond. IEEE Int Conf on Communications, p.1-7. https://doi.org/10.1109/ICC.2018.8422462

Zhang QS, Zhu SC, 2018. Visual interpretability for deep learning: a survey. *Front Inform Technol Electron Eng*, 19(1):27-39. https://doi.org/10.1631/FITEE.1700808

Zhao SJ, Song JM, Ermon S, 2018. The information autoencoding family: a Lagrangian perspective on latent variable generative models. https://arxiv.org/abs/1806.06514