Cognitive NETWORKS

Artificial Intelligence for Dynamic Spectrum Management

Lector Materials

gabriel.bugar@tuke.sk – BN 32, 515

Lessons content

Key Topics

Week	Topic description
7.	Cognitive Radio for Dynamic Spectrum Management
8.	OSA and Spectrum Sensing Theories and Methods
9.	Assignments for elaboration - performance and discussions
10.	Blockchain for Dynamic Spectrum Management
11.	Artificial Intelligence for Dynamic Spectrum Management
12.	ML for Spectrum Sharing, ML for Signal Classification, Deep Reinforcement Learning for Dynamic Spectrum Access
13.	-

Cognitive NETWORKS

Week 10. Lector Content

This chapter covers the following content:

- Artificial Intelligence & DSM
    - Challenges in applying AI techniques
- Overview of Machine Learning Techniques
    - Statistical Machine Learning – SVM, KNN, K-means
    - Deep Learning – CNN, RNN
    - Deep Reinforcement Learning – DQN,
- Machine Learning for Spectrum Sensing and Signal Classification
- Deep Reinforcement Learning for Dynamic Spectrum Access

1.1 Artificial Intelligence & DSM

- In the past decade, a significant advancement has been made in artificial intelligence (AI) research from both theoretical and application perspectives.
- Researchers have also applied AI techniques, particularly machine learning (ML) algorithms, to DSM, the results of which have shown superior performance as compared to traditional ones.

Artificial intelligence (AI)

- AI, also known as machine intelligence, has been seen as the key power to drive the development of future information industry [1].
- The term AI was coined by John McCarthy in a workshop at Dartmouth College in 1956, and he defined AI as “the science and engineering of making machines, especially intelligent computers” [2].
- Generally, AI is defined as the study of the **intelligent agent**, which is able to judge and execute actions by observing the surrounding environment so as to complete certain tasks.
- The intelligent agent can be a computer program or a whole system.

Artificial intelligence and DSM

- With the significant advancement in the computational capability of *computer hardware*, *various theories*, especially **machine learning techniques**, and applications of AI have been developed in the past two decades.

- With the surging demand for wireless services and the increasing connections of wireless devices, the network environments are becoming more and more complex and dynamic, which imposes stringent **requirements on DSM**.

- In the age of 5G, AI has been seen as an effective tool to support DSM in order to tackle the transmission challenges, such as high rate, massive connections and low latency [3, 4].

Cognitive NETWORKS

Artificial intelligence (AI)

- By adopting ML techniques, the traditional model-based DSM schemes would be transformed to the data-driven DSM schemes, in which the **controller** in the network can adjust itself adaptively and intelligently to improve the efficiency and robustness of DSM.
- AI-based DSM schemes have thus attracted more and more attention in recent years, and have shown great potentials in practical scenarios.

Artificial Intelligence & DSM

AI techniques benefits

- The applications of AI techniques would bring significant benefits to DSM:

1. Firstly, the AI-based DSM schemes normally do not need environmental information as the prior knowledge, and they can extract useful features from the surroundings automatically.
2. Secondly, AI-based DSM schemes can be re-trained periodically and thus they are more robust to the changing environment.
3. Additionally, by applying AI techniques, DSM can be done in a decentralized and distributed manner, leading to a significant reduction of signal overheads, especially for large-scale systems.
4. Finally, once trained, AI-based DSM schemes are low in complexity for processing newly arrived data and thus they are more suitable for practical implementation.

Cognitive NETWORKS

Challenges in applying AI techniques

- While it is believed that machine learning techniques are effective methods for developing and optimizing the next generation networks [5], there also exist some challenges in applying AI techniques in DSM.
- For example, different from images, the received signal and its higher order statistics in wireless networks are normally complex numbers, which are hard to process directly by neural networks.
- Additionally, in a typical wireless communication system, accurate network data such as channel information, is hard to obtain in practice
- Hence, there are many remaining challenges and problems to be addressed for achieving wireless intelligence.

1.2 Overview of Machine Learning Techniques

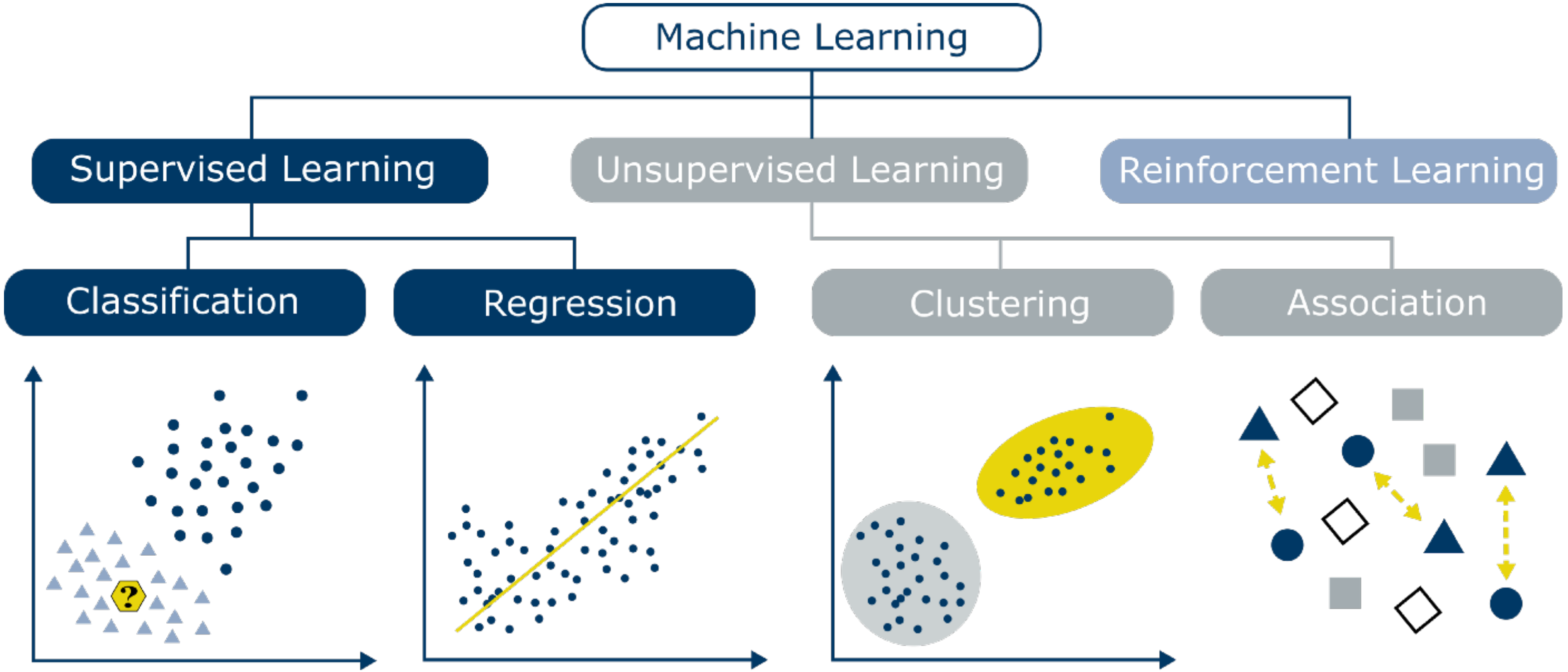
- As the core technique of AI, **machine learning** (ML) is a multidisciplinary subject involving multiple disciplines such as:
  - probability theory,
  - statistics,
  - information theory,
  - computational theory,
  - optimization theory,
  - and computer science.

Cognitive NETWORKS

Machine Learning Techniques

- T. Mitchell provided a brief definition of machine learning in 1997 as follows: "*machine learning is the study of computer algorithms that* ***improve automatically through experience***" [6].
- Hence, the main objective of ML is to make agents simulate or implement human learning behaviors.
- Based on the type of training data used, ML can be divided into three branches, namely, ***supervised learning***, ***unsupervised learning*** and ***reinforcement learning***.
- The former requires labeled training data, while the latter only uses unlabeled training data.

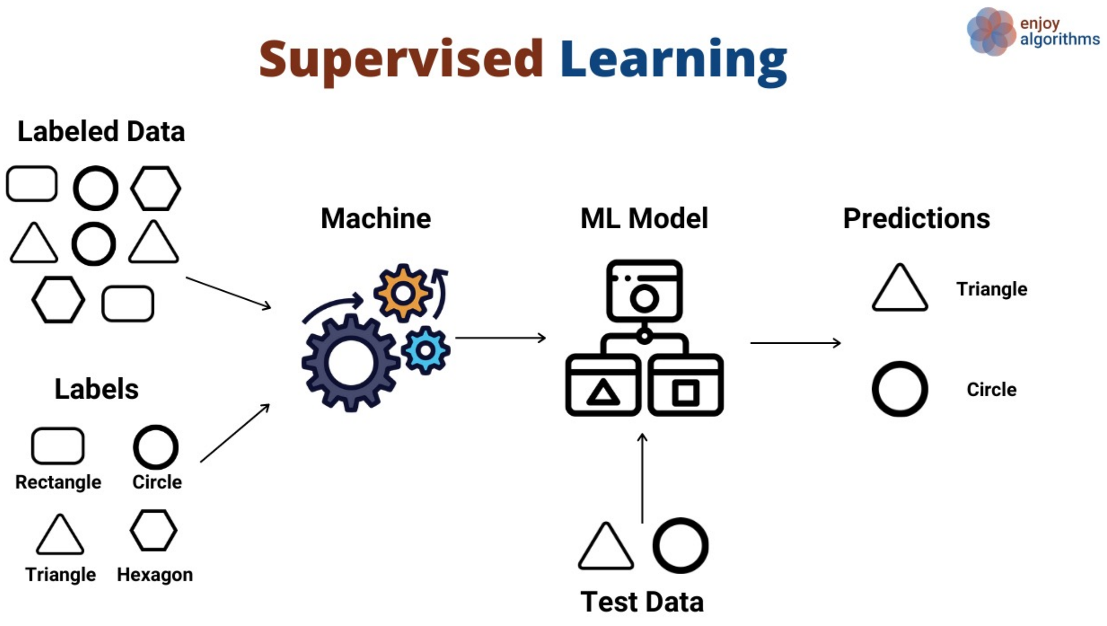
Machine Learning Techniques



Cognitive NETWORKS

Machine Learning Techniques - **Supervised Learning**

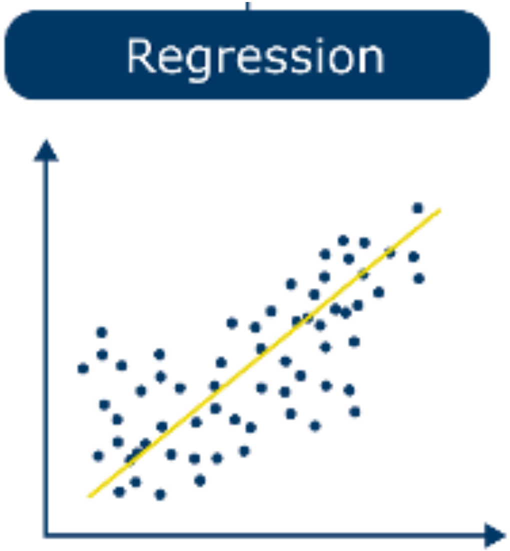
- In supervised learning, the objective for an agent is to learn a parameterized function from the given labeled training dataset and then based on the function learnt to predict the result directly while new data arrives.



Machine Learning Techniques - **Supervised Learning**

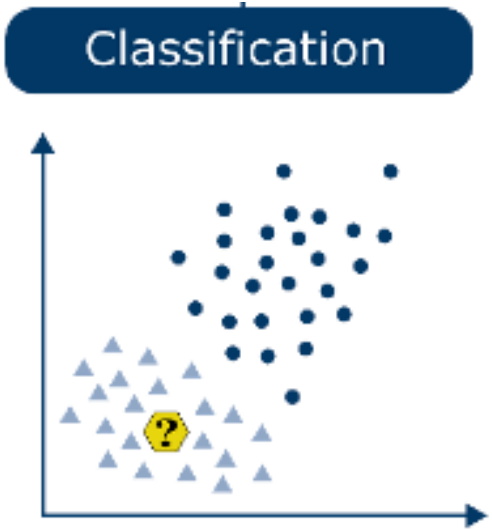
- The tasks in supervised learning are regression and classification.

- Specifically, **regression** is to determine the quantitative relationship between certain variables based on a set of training data. It learns a model based on a training dataset to make predictions about unknown or future data.



Machine Learning Techniques - **Supervised Learning**

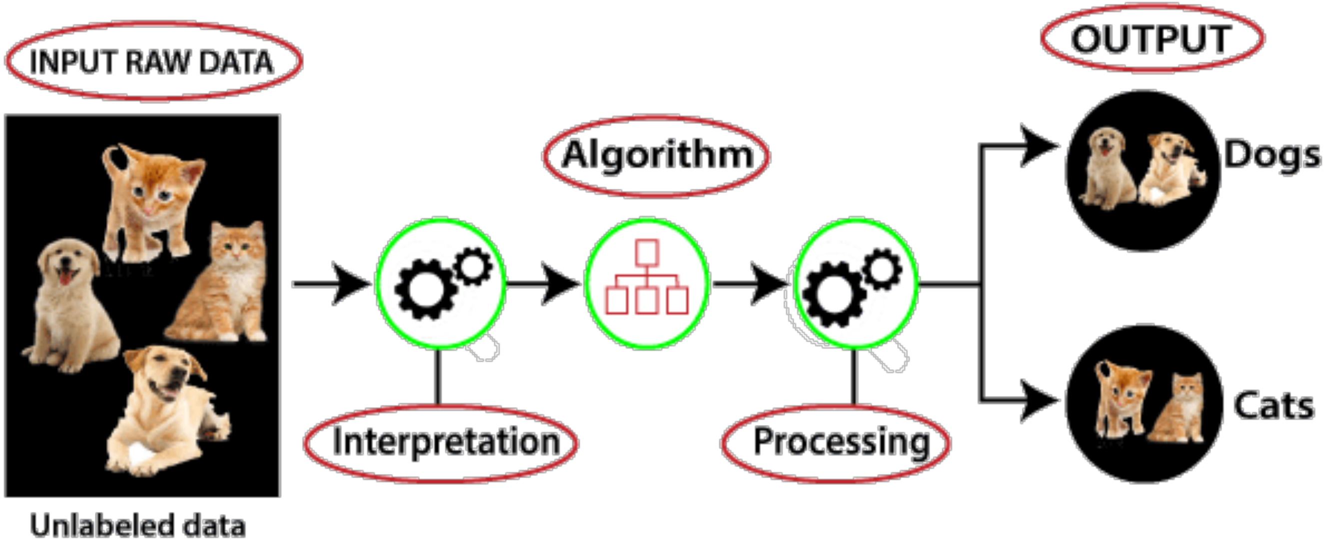
- The tasks in supervised learning are regression and classification.
- and **classification** is to find a function to determine the category to which the input data belongs.
- The difference between the subcategories Regression and Classification is only due to the output value. While Classification divides the dataset into classes, Regression is used to output continuous values.



Cognitive NETWORKS

Machine Learning Techniques - **Unsupervised learning**

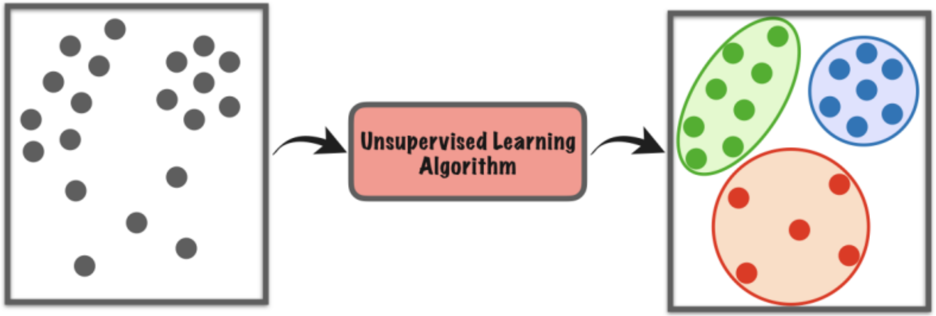
- In unsupervised learning, since the training data is unlabeled, the agent needs to adopt clustering methods to obtain the relationship.



Cognitive NETWORKS

Machine Learning Techniques - **Unsupervised learning**

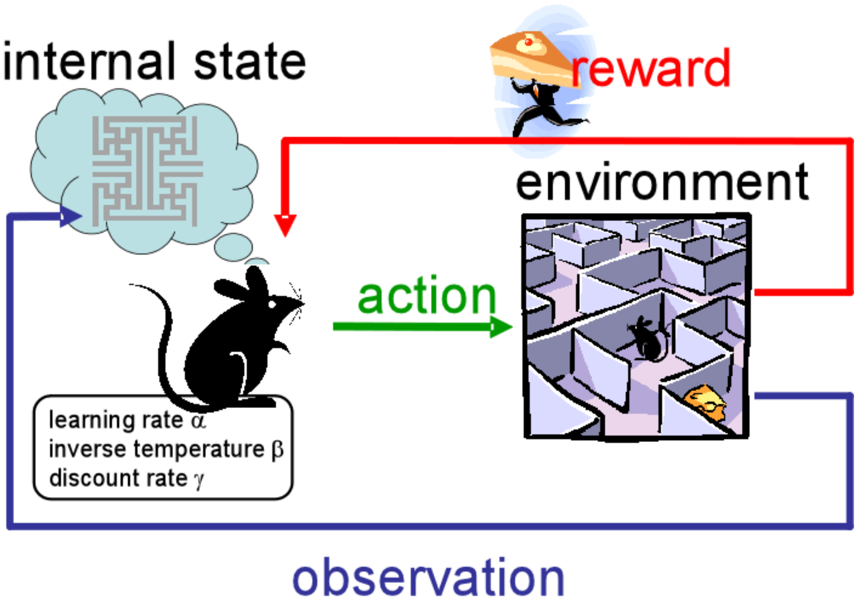
- A clustering method aims to divide the training data into several classes based on the similarity of the data.
- The objective of clustering is to minimize intra-class distance while maximizing inter-class distance.
- Compared to supervised learning, unsupervised learning is more like self-study.



Cognitive NETWORKS

Machine Learning Techniques - **Reinforcement Learning**

- Labeled data can also be generated through online learning such as reinforcement learning (RL).



Machine Learning Techniques - **Reinforcement Learning**

- In particular, RL **produces labeled experiences to train itself** from continuous interactions with the environment, it is developed to solve a *Markov decision process* (MDP) ***M*** = {***S***, ***A***, ***P***, ***R*** }, where:
    - ***S*** is the state space,
    - ***A*** is the action space,
    - ***P*** is the transition probability space and
    - ***R*** is the reward function [7].

Cognitive NETWORKS

Machine Learning Techniques – **SML & DL**

- ML techniques can also be grouped into two categories, namely, Statistical Machine Learning (SML) and Deep Learning (DL).
- Using statistics and optimization theory, SML constructs proper *probabilistic* and *statistical models* with training data.
- DL, on the other hand, makes use of Artificial Neural Network (ANN), also known as Deep Neural Network (DNN), to perform supervised learning tasks.
- In recent years, neural network techniques have also been applied to RL, leading to the birth of Deep Reinforcement Learning (DRL).

Statistical Machine Learning (SML)

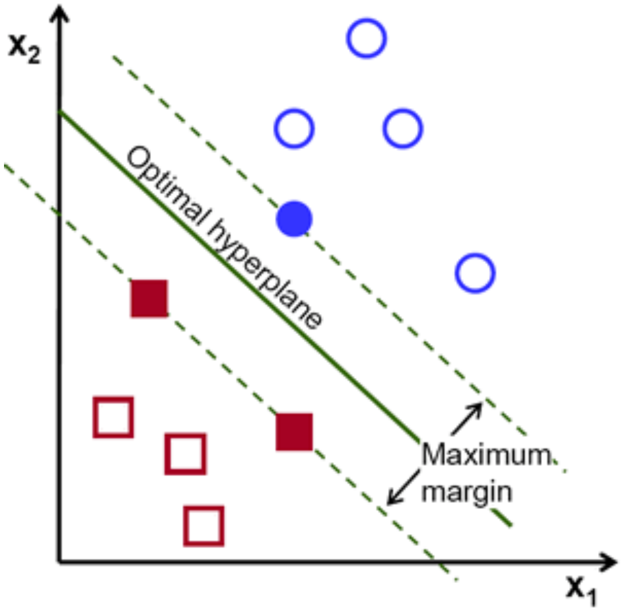
- The objective of SML is to construct a probabilistic and statistical model using the training data, then, based on the constructed model, to make inferences with new data [8].
- SML can be applied in both **supervised** and **unsupervised learning**.
- The commonly used **supervised** learning methods with SML are:
  - Support Vector Machine (SVM) and K-Nearest Neighbor (KNN),
- the commonly used **unsupervised** learning methods with SML are:
  - K-Means and Gaussian Mixture Model (GMM).

Cognitive NETWORKS

SML - **Support Vector Machine** (SVM)

- SVM algorithm is a typical binary classification algorithm.

- The basic idea of the SVM algorithm is to find a decision (optimal) hyperplane to maximize the margin between different classes.



SML - **Support Vector Machine** (SVM)

- Specifically, for a given training data set:

$$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}, \text{ where } y_i \in \{-1,1\},$$

- the objective of the SVM algorithm is to find the hyperplane denoted by $\mathbf{w} \cdot \mathbf{x} + b = 0$ to make the data sets linearly separable,

- where: $\mathbf{w}$ are the normal vector
  and b the intercept of the plane

SML - **Support Vector Machine**

- If the decision hyperplane is obtained, the corresponding classification decision function is given as

$$f(\mathbf{x}) = \operatorname{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

- The hyperplane can be learnt by solving the following convex quadratic programming problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & y_i\,(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i\,, \\ & \mathbf{x}_i \geq 0,\ i = 1, 2, \ldots, N \end{aligned}$$

- Generally, SVM is used to solve a linear classification problem, but it can also be used as a nonlinear classifier by introducing different kernel functions such as *Gaussian kernel function* and *radial basis function*.

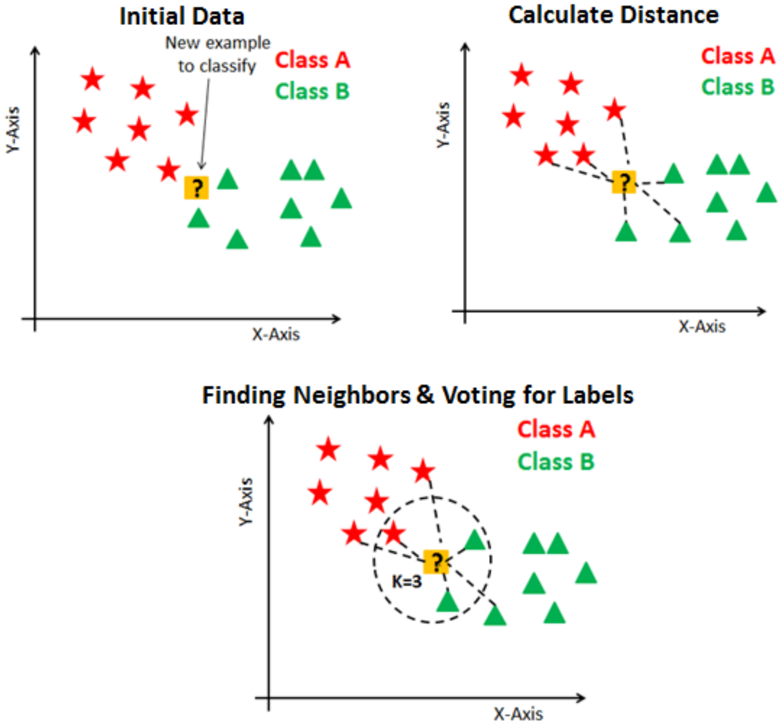
where *C* is a punishment parameter and ξ_i is the soft constant for *i*-th data set.

SML – **K-Nearest Neighbor** (KNN)

- KNN algorithm is a basic supervised learning algorithm for classification.

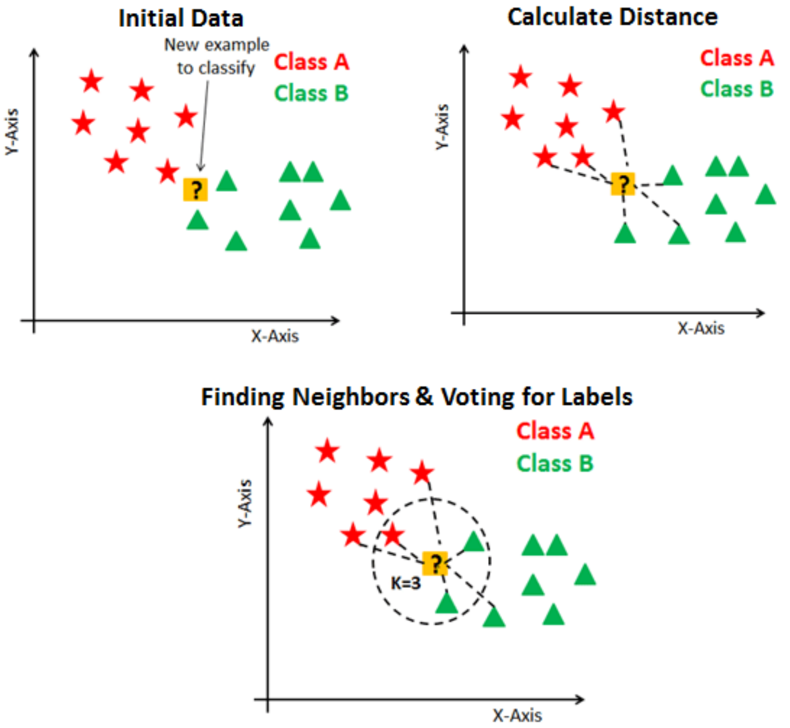
- Let $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ denote a given training dataset,

- where $\mathbf{x}_i$ is the i-th data set
- and y_i is the corresponding label



SML – **K-Nearest Neighbor** (KNN)

- Assume that all data sets come from *J* classes.
- For a newly arrived data set **x**, its label, i.e., class, is determined by its nearest labeled neighbors based on the adopted classification decision rules.
- Hence, the basic elements in KNN ai the number of neighbors *K*, the distance measure and the classification decision rule.



SML – **K-Nearest Neighbor** (KNN)

- Specifically, the classification process consists of two steps:

1. the first step is to search K labeled data sets which are closest to the newly arrived data set **x** according to the given distance measure. Denote the region covering these K data sets as $N_K(\mathbf{x})$.

2. The second step is to determine its label y by using the chosen classification decision rule based on $N_K(\mathbf{x})$. The commonly used classification decision rule is the majority voting rule, which is given as:

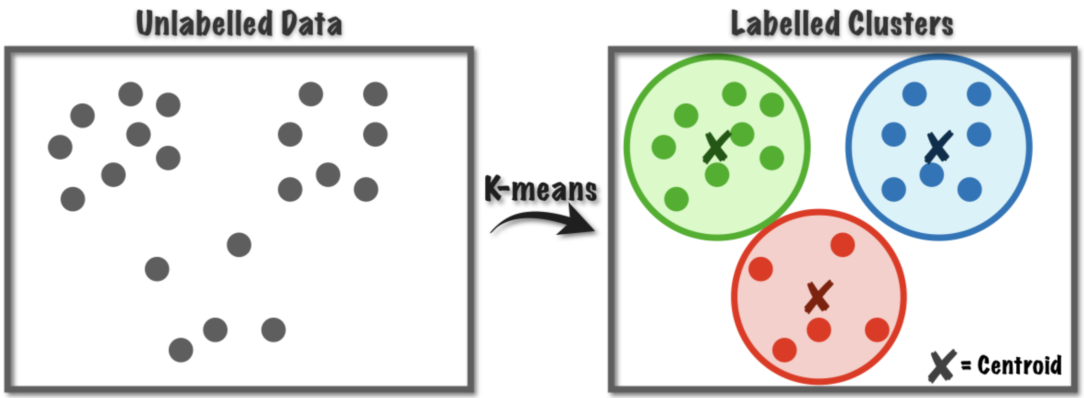
$$y = \arg \max_{c_j, j=1,\ldots,J} \sum_{\mathbf{x}_i \in N_K(\mathbf{x})} I\left(y_i = c_j\right)$$

where I (·) is the indicator function that indicates if a label belongs to class c_j

Cognitive NETWORKS

SML – K-means

- K-means algorithm is a clustering algorithm, in which the unlabeled data sets are processed iteratively to form *K* clusters.
- Specifically, at the beginning, *K* data sets are chosen to form the initial centroids of the *K* clusters.



Cognitive NETWORKS

SML – **K-means**

- Then, the *K*-means algorithm alternates the following two steps:

  1. assign each of the remaining data sets to its nearest cluster.

     - this is determined by evaluating the Euclidian distance between the data set and the centroid of each cluster and choosing the cluster with the smallest distance.

SML - **K-means**

2. update the centroid of each cluster, denoted as c_k, based on the newly labeled data sets. Mathematically, this can be expressed as:

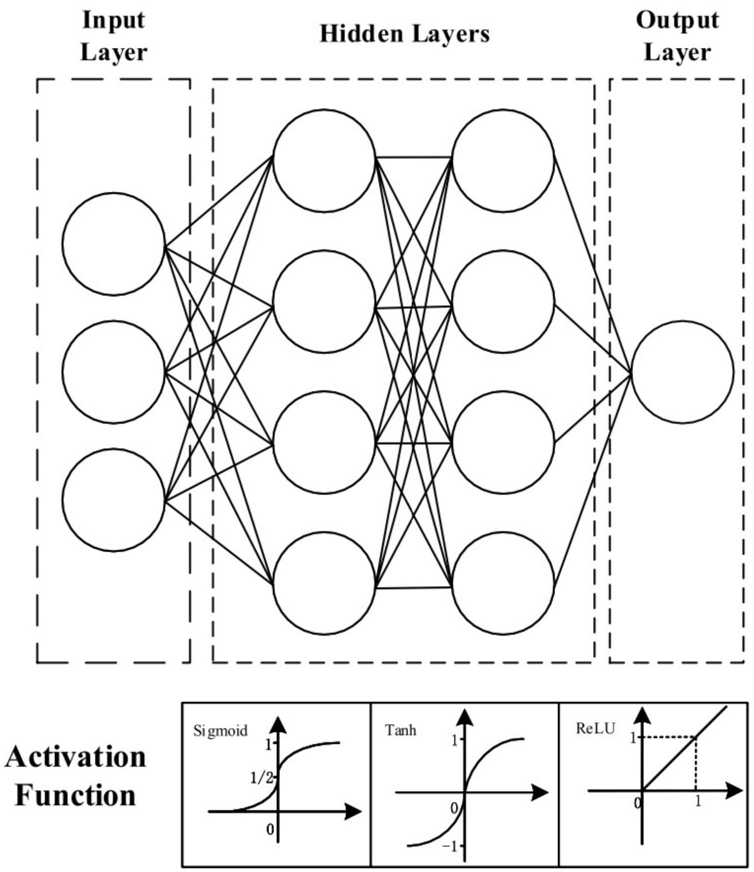
$$c_k = \frac{1}{|N_k|} \sum_{\mathbf{x} \in N_k} \mathbf{x}, \; k = 1, 2, \ldots, K$$

where N_k denote the set of examples assigned to cluster k.

- These two steps will repeat until a termination condition is met.
- Although K-means algorithm can be implemented with low complexity, its performance is influenced significantly by the initialization parameters such as the number of clusters and the cluster centroids.

Deep Learning (DL)

- DL has significantly advanced the development of computer vision and natural language processing recently.
- As the core technique of DL, ANN has been used to approximate the relationship between an input and an output.
- Generally, a typical ANN is composed of three parts, namely, input layer, output layer and hidden layers as shown in Figure:



Deep Learning (DL)

- In each layer, many cells with different activation functions are placed, and the cells in adjacent layers are connected with each other.
- There are different network structures used for different types of data.
- A convolutional neural network (CNN), which consists of convolutional layers, pooling layers and fully connected layers, is suitable for images;
- while a recurrent neural network (RNN), which contains many recurrent cells in the hidden layers, is suitable for time series data.
- Furthermore, in order to improve the generalization and convergence performance of the DL, dropout and other techniques are introduced in the design of neural networks [9].

DL - Convolutional Neural Network (CNN)

- CNN is a special network for processing images, in which the cells adopt convolution operations. A typical CNN is composed of multiple convolutional layers, pooling layers and fully-connected layers [10]:
- *Convolutional Layer*: Different from the fully-connected layers, a convolutional layer contains a set of multiple feature maps, which are obtained by using different convolution kernels to operate on the input image.
- In particular, one feature map is calculated by one convolution kernel operating on the input image, which means all the elements in the same feature map share the same weight and bias with each other.

DL - Convolutional Neural Network (CNN)

- *Pooling Layer*: A pooling layer is usually placed after a convolutional layer in a CNN to capture invariant features.
- Specifically, the pooling operation is to replace the output of one position in the input image with a summary statistic of the neighborhood.
- A commonly used pooling method is the max-pooling function, which gives the maximum output of the rectangular region.
- In fact, the pooling operation can be seen as an action to add a strong prior knowledge.

DL – Recurrent Neural Network (RNN)

- RNN is a powerful tool for time series data, which have shown superior performance on speech recogni- tion [11].
- Different from traditional neural networks, there are many connected cells in each layer in an RNN.
- All cells in the same layer have the same structure and each of them passes its information to its successor.
- The output of an RNN is determined by not only its current input but also the memory recorded in the past time steps.
- However, conventional RNNs cannot learn long-term dependent information and suffer from the gradient vanishing problem easily.

DL – Recurrent Neural Network (RNN)

- Then long short-term memory (LSTM) network, as a kind of gated RNN network, is proposed to mitigate this problem.

- Specifically, in each cell of an LSTM network, there are three gates, namely, the input gate, the forget gate and the output gate, which are given as follows

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_t - 1 + \mathbf{U}_i x_t + b_i)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_t - 1 + \mathbf{U}_f x_t + b_f)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_t - 1 + \mathbf{U}_o x_t + b_o)$$

- where $\mathbf{i}t$, $\mathbf{f}t$, and $\mathbf{o}t$ are the input gate, the forget gate and the output gate, respectively; $\mathbf{W}i$, $\mathbf{U}i$, $\mathbf{b}i$, $\mathbf{W}f$, $\mathbf{U}f$, $\mathbf{b}f$, Wo, Uo, bo are the weight matrices and biases of the corresponding gate, respectively; and σ (·) is the sigmoid function.

DL – Recurrent Neural Network (RNN)

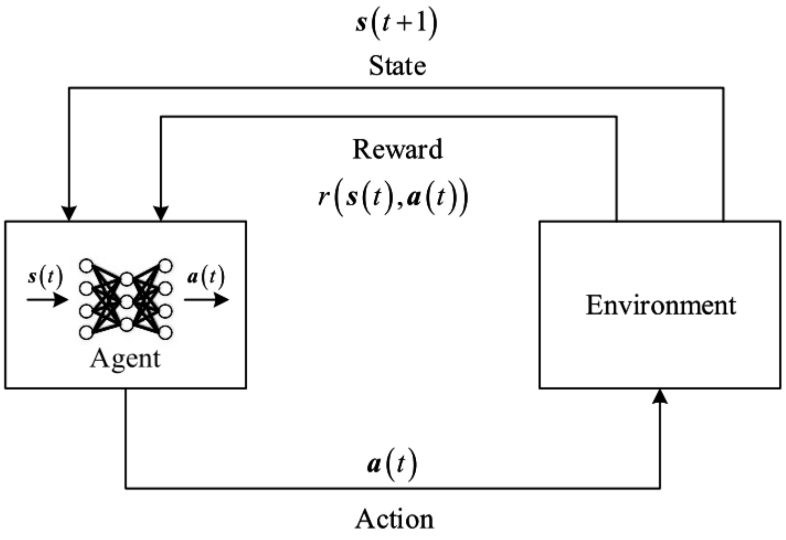
- Additionally, each cell has a self-loop and its cell state is jointly controlled by the forget gate and the input gate.
- Specifically, the forget gate determines what information to remove and the input gate determines what information to add to the next cell state.
- Mathematically, the cell state can be expressed as

$$\mathbf{c}_t = \mathbf{f}_t \cdot \mathbf{c}_t{-}1 + \mathbf{i}_t \cdot \tanh(\mathbf{W}_c\mathbf{h}_t{-}1 + \mathbf{U}_c\mathbf{x}_t + b_c)$$

- where $\mathbf{W}_c$, $\mathbf{U}_c$ and b_c are the weight matrices and the bias of the cell memory, respectively. The gated structure allows the LSTM network to learn the long-term dependent information while avoiding vanishing gradients.

Deep Reinforcement Learning (DRL)

- As the combination of DL and RL, deep reinforcement learning (DRL) has shown superior performance in sequential decision-making tasks.
- In the DRL framework, as shown in Figure, the agent inputs its observation (state) $s(t) \in \boldsymbol{S}$ into the neural network and outputs an action $a(t) \in \boldsymbol{A}$



- Then it obtains a reward *R(s(t)*, a(t)) which is used to evaluate the profit of the selected action by executing it.

Deep Reinforcement Learning (DRL)

- After a period of learning, the agent can learn the optimal strategy, which maps an state to an action, to maximize its long-term accumulative reward from continuous interactions with the environment.
- Similar to RL, the basic elements of DRL are also state space S, action space A and the reward function R.
- Different from the traditional RL, which uses a table to indicate the relationship between the state space and the action space, DRL uses a neural network as the function approximator, and therefore it works more effectively for problems with high dimensional state and action spaces.
- In DRL, the commonly used methods is deep Q-network (DQN)

DRL - Deep Q-network

- Different from the tabular method in traditional RL, a neural network called deep Q-network (DQN) is adopted to approximate the relationship between state space and action space [12].

- Since the DQN is optimized by minimizing the temporal difference error, the loss function of DQN is given as

$$L(\theta) = \mathrm{E}[\, y^{DQN} - Q(s, a; \theta)^2]$$

- where E [·] indicates the expectation operation, Q(s, a; θ) is the Q-function with the parameter θ, and the target valuey DQN is given as

$$y^{DQN} = \mathrm{R}(s,a) + \gamma Q(s',a';\theta)$$

DRL - Deep Q-network

- To improve the performance of the basic DQN, two other techniques, i.e., experience replay and quasi-static target network, are introduced in the design of DQN technique.
- **Experience Replay**: The agent needs to construct a fixed-length memory M, which is based on the first-in-first-out (FIFO) rule. In each training step t, the agent needs to store newly obtained experience into the memory M, and then a mini-batch dt of experiences is sampled randomly from M for training.
- **Quasi-static Target Network**: The agent constructs two DQNs of the same structure, i.e., the target DQN $Q(s,a;\theta')$ and the trained DQN $Q(s,a;\theta)$, where θ' and θ are their respective parameters. In every K steps, the trained network share its parameter with the target network.

DRL - Deep Q-network

- Additionally, in order to balance the relationship between exploration and exploitation, the ε-greedy algorithm is usually adopted in a DRL.
- Specifically, an agent selects the action corresponding to the maximum Q-value of the trained network with a probability $1 - \varepsilon$, and selects an action randomly otherwise.
- After the algorithm converges, the agent just selects the action with the maximum Q-value, and the target network is closed.

1.3 Machine Learning for Spectrum Sensing

- Spectrum sensing is an important task to realize DSM in wireless communication systems and is usually used to assist users to find out the channel status.

Cognitive NETWORKS

Machine Learning for Spectrum Sensing

- In order to increase the accuracy of spectrum sensing, many spectrum sensing algorithms have been developed in the past years, such as estimator-correlator (EC) detector, the semi-blind energy detector and the blindly combined energy detection (BCED).
- Although the EC detector can achieve the optimal performance, it needs the knowledge of PU signals and noise level.
- The semi-blind energy detector is more practical, and it only requires the knowledge of the noise power.
- However, the performance of the semi- blind energy detector depends heavily on the accurate knowledge of noise power, which is usually uncertain.

Machine Learning for Spectrum Sensing

- The BCED does not need any prior knowledge about the PU signals or noise, but the performance is worse than the performance of the semi-blind energy detector.
- It is noticed that most existing algorithms are model-driven, and need the prior knowledge of noise or PU signals to achieve good performance.
- However, this feature makes them unsuitable for practical environment, and the lack of prior knowledge would result in performance degradation.
- To solve the above issues, machine learning techniques have been adopted to develop cooperative spectrum sensing (CSS) framework [14].

Machine Learning for Spectrum Sensing

- Specifically, the work considers a CR network, in which multiple SUs share a frequency channel with multiple PUs.
- The channel is considered to be unavailable for SUs to access if at least one PU is active and it is available if there is no active PU.
- For cooperative sensing, each SU estimates the energy level of the received signals and reports it to another SU who acts as a fusion center.
- After the reports of the energy level from all SUs are collected, the fusion center makes the final classification of the channel availability.

Machine Learning for Spectrum Sensing

- Using the machine learning technique such as K-means algorithm, GMM clustering, SVM algorithm and KNN algorithm, the fusion center can construct a classifier to detect the channel availability.
- With unsupervised machine learning such as K- means and GMM clustering, the detection of the channel availability relies on the cluster that the sensing reports from all the SUs are mapped to.
- On the other hand, with supervised machine learning such as SVM algorithm and KNN algorithm, the classifier is first trained using the labeled sensing reports from all SUs.
- After the clas- sifier is trained, it can be directly used to derive the channel availability.

1.4 Machine Learning for Signal Classification

- Signal classification, usually performed before signal detection, is a fundamental task in cognitive radio networks. Consider the modulation classification as an example.

Machine Learning for Signal Classification

- Traditionally, there are two kinds of modulation classification approaches, namely, the likelihood-based (LB) approach and the feature-based (FB) approach.
- The LB approach is based on computing the likelihood function of received signals under different modulation schemes hypotheses, and the modulation scheme with the maximum likelihood value is validated.
- With perfect knowledge of channel and noise parameters, the LB approach can achieve the optimal performance in a Bayesian sense.
- However, the estimation of these parameters imposes high computation complexity.

Machine Learning for Signal Classification

- In the FB approach, useful features such as higher-order statistics are extracted for decision-making.
- In general, the FB approach has lower compu- tational complexity but it can only achieve sub-optimal performance.
- Therefore, in order to achieve near optimal performance with low computational complexity, ML techniques have been introduced in solving the modulation classification problem, and have shown superior performance recently.

Modulation-Constrained Clustering Approach

- A clustering-based LB classifier is proposed for modulation classification in multiple-input and multiple-output (MIMO) communication systems.
- In that work, a spatial-multiplexed MIMO system with Nt transmit antennas and Nr receive antennas is considered, in which data symbols are transmitted independently from each transmit antenna.
- The signal model of the n-th received signal vector y (n) is given as

$$\mathbf{y}(n)=\mathbf{Hs}(n)+\mathbf{u}(n),\ n=1,\ldots,N$$

- where H ∊ CNr ×Nt is the channel matrix which remains constant within each block of N symbols, and u (n) denotes the AWGN vector.

Deep Learning Approach

- The modulation classifier requires accurate knowledge of the channel model. In addition, the channel model may not be available in practice.
- As a powerful supervised learning framework, DL can also be applied in modulation classification.
- In [16], a low-complexity blind data-driven modulation classifier based on DNN is proposed, which operates under uncertain noise condition modeled by a mixture of white Gaussian noise, white non-Gaussian noise and time-correlated non-Gaussian noise.

Deep Learning Approach

- In [16], a single-input and single-output (SISO) channel is considered, and the n-th received signal sample is given as

$$r(n) = hs(n) + u(n), \; n = 1, 2, \ldots, N$$

- where s (n) is the transmitted symbol from an unknown modulation scheme Mi , N is the number of symbols in a block, h is the channel coefficient and u (n) denotes the additive noise.

- Denote the set of the candidate modulation schemes and the received signal sequence by M={Mi, i =1,2,...,L} and r=[r(1),r(2),...,r(N)], respectively. Let P(Mi|r) denote the a posterior probability of the modulation scheme Mi given the received signal r.

Deep Learning Approach

- The objective of the work is to find the modulation scheme which maximizes the a posterior probability. This is known as the maximum a posterior (MAP) criterion.

$$\hat{M}_i = \arg\max_{M_i \in \mathcal{M}} P(M_i|\mathbf{r})$$

- In order to accurately make classification decisions with low complexity, the DNN is adopted to learn the a posterior probability P(Mi |r), i = 1,… L.
- The DNN is used as an approximation function f mapping the received signal to the a posterior probability.
- The in-phase and quadrature (IQ) components of the received signal samples are chosen as the inputs to the proposed neural network.

Deep Learning Approach

- Motivated by its superior performance for processing time-dependent data, the long short-term memory (LSTM) network is introduced in the design of the proposed neural network.
- There are three main reasons that the LSTM network is suitable for solving a modulation classification problem:

1. The LSTM network is able to learn features effectively from highly time-dependent data. This indicates the neural network with LSTM layer have advantages in learning the a posterior probability from the signal samples which are highly time-dependent over time-correlated non-Gaussian channels.

Deep Learning Approach

2. Different from the fully-connected network which can only receive a one- dimensional as input, LSTM network allows two-dimensional vectors as the input in each time step. Hence, the network can process complex signal samples com- posed of IQ components, and can learn better from the input data.

3. Compared to the conventional fully-connected network, there are fewer parameters in the LSTM network because all time steps share the same weight matrices and biases.

- Additionally, in order to summarize the output form the LSTM network, a temporal attention mechanism is adopted in the final LSTM layer over the outputs from all time steps.

Deep Learning Approach

- In the temporal attention mechanism, each output has a different weight, which indicates the importance of each to the modulation classification results.
- Specifically, the proposed seven layer-neural network is composed of three stacked-LSTM layers and four fully-connected layers.
- In the training phase, the one-hot coding vectors of true modulation schemes of the input signal samples are used as the labels.
- The Adaptive Moment Estimation (Adam) optimizer is used to minimize the loss function to optimize the weights and bias in the network.

Deep Learning Approach

- After the training phase, the modulation classification is made by according to the MAP criterion defined in equation.
- The simulation results show that the classification accuracy of the proposed classifier approaches that of the ML classifier with all the channel and noise parameters known.
- Moreover, under uncertain noise conditions, with lower computational online complexity, the proposed classifier can achieve a better performance than the EM and ECM classifiers.

1.5 Deep Reinforcement Learning for Dynamic Spectrum Access

- In traditional DSA mechanism, there exists a centralized control node responsible for allocating the spectrum resources to users.
- Before making the access decisions, the centralized node needs to collect the global network information, such as the position information of users and base stations as well as the channel state information.

Cognitive NETWORKS

Deep Reinforcement Learning for DSA

- However, such global network information is difficult to obtain in practice, as it imposes significant signal overheads on the system especially when there is a large number of users.
- Additionally, the collected information may be outdated in a highly dynamic network environment, resulting in invalid access strategy and poor performance.
- To solve the above issues, intelligent DSA framework operating with local network information is desirable.
- Recently, researchers introduced DRL techniques for DSA, showing superior performance on sequential decision-making tasks, to enable more flexible and intelligent DSA mechanism [17].

Deep Multi-user Reinforcement Learning for Distributed DSA

- DRL-based DSA framework is proposed to manage dynamic spectrum access in multichannel wireless networks, in which each user acts as an agent to make channel access decisions intelligently and independently to maximize its long-term transmission rate.
- In this work, a wireless network composed of N users and K shared orthogonal channels is considered. Denote the set of users and the set of channels as $N = \{1,2,..., N\}$ and $K = \{1,2,...,K\}$, respectively.
- It is assumed that each user needs to choose a single channel for transmission in each time slot, and it always has packets to transmit.

Cognitive NETWORKS

Deep Multi-user Reinforcement Learning for Distributed DSA

- Additionally, the transmission is successful if there is only one user accessing the channel, and the transmission fails otherwise.
- After each transmission, each user can receive a binary observation on (t) to indicate whether its transmission is successful or not, i.e., on(t) = 1 if the transmission is successful and on(t) = 0 otherwise.
- With the assumption that users don't have message exchange in each time slot, they can only make access decisions by their local observations. In order solve the above problem, a DRL-based distributed framework for DSA is proposed, in which each user acts as an agent and constructs a DQN. The action space, state space and reward function are described as follows.

Deep Multi-user Reinforcement Learning for Distributed DSA

1. **Action Space**: In each timeslot, each user needs to choose whether to transmit or not. If the user chooses to transmit, it needs to select a channel for transmission. The action of user n in time slot t is given as

$$an(t) \in \{0,1,\ldots,K\}$$

where an(t) = 0 indicates that user n chooses not to transmit in time slot t.

Deep Multi-user Reinforcement Learning for Distributed DSA

2. **State Space**: The state of each user is composed of its action and observation up to time slot t, which is given as

$$\mathcal{H}_n(t) = \left(\{a_n(i)\}_{i=1}^{t-1}, \{o_n(i)\}_{i=1}^{t-1}\right)$$

Deep Multi-user Reinforcement Learning for Distributed DSA

2. **Reward Function**: Since the objective is to maximize the long-term rate, the function of achievable rate is chosen as the reward function

$$rn(t) = B \log 2(1 + SNRn(k))$$

where B is the channel bandwidth and SNRn(k) is SNR of user n on chanel k

In the training phase, each user trains the parameters of their respective DQN cooperatively by communicating with a central unit. After updating the parameters, each user uses the trained DQN to make access decisions autonomously and independently. After the DQN is well-trained, the central unit is closed, and users use the converged DQN to obtain efficient access policy directly.

DRL for Joint User Association and Resource Allocation

- In heterogeneous networks (HetNets), all the base stations (BSs) normally provide services to users on shared spectrum bands in order to improve the spectrum efficiency.
- However, most existing methods need accurate global network information, e.g., channel state information, as the prior knowledge, which is difficult to obtain in practice.
- In [19], a distributed DRL-based DSA framework is proposed for user association and resource allocation in the downlink HetNets. Specifically, a three-tier heterogeneous network is considered, which consists of Nm macrocell base stations (MBSs), Np pico base stations (PBSs), Nf femto base stations (FBSs) and N user equipments (UEs).

DRL for Joint User Association and Resource Allocation

- The sets of UEs and BSs are denoted, respectively, by N = {1, . . . , N } and B = {0, 1, . . . , L − 1}, where L = Nm + N p + N f .
- All the BSs share the same K orthogonal channels for downlink transmission, and the set of channels can be denoted as K = {1,..., K}.
- For each UEi, denote bl(t)= b0(t),...bL−1(t),i∈N,l∈B as the binary iii
- user-association vector, where bil (t) = 1 if UE i is associated with the B Sl at time t and bl (t) = 0 otherwise. For each BS, a binary channel-allocation vector is defined asci (t)= ci (t),...ci (t) ,i ∈N,k ∈ K , where ci(t)=1 if UEi uses channel resource Ck at time t and cil(t) = 0 otherwise.

DRL for Joint User Association and Resource Allocation

- It is assumed that each UE can only be connected to one BS and each channel can only be allocated to one UE for each BS in each time slot t.
- Since all the BSs share the common spectrum resource, the co-channel interference should be considered.
- Hence, the signal-to-interference-plus-noise-ratio (SINR) of UE i associated with BS l and allocated with channel Ck is given as

$$\Gamma_{li}^{k}(t) = \frac{b_i^l(t)\, h_l^{i,k}(t)\, c_i^k(t)\, p_{li}^k(t)}{\sum_{j \in B \setminus \{l\}} b_i^j(t)\, h_j^{i,k}(t)\, c_i^k(t)\, p_{ji}^k(t) + W N_0}$$

- where hi,k (t) is the channel gain between the UE i and BS l at time t, W is the bandwidth of each channel and N0 is the noise spectral power.

DRL for Joint User Association and Resource Allocation

- Therefore, the total achievable transmission rate of UE i at time t can be expressed as

$$r_i(t) = \sum_{l=0}^{L-1} b_i^l(t) \sum_{k=1}^{K} W \log_2\left(1 + \Gamma_{li}^k(t)\right)$$

- Considering that the operation cost of the UE i from BS l is determined by the transmit power pk (t), the total operation cost of UE i is given as

$$\varphi_i(t) = \sum_{l=0}^{L-1} \varphi_i^l(t) = \sum_{l=0}^{L-1} \lambda_l b_i^l(t) \sum_{k=1}^{K} c_i^k(t)\, p_{li}^k(t)$$

DRL for Joint User Association and Resource Allocation

- where λl is the price per unit of transmit power from BS l.
- Then we define the utility function of UE i as the total achievable profit minus the operation cost, which is denoted as

$$\omega_i(t) = \rho_i(t)\, r_i(t) - \varphi_i(t)$$

- where ρi > 0 is the profit per unit transmission rate.

DRL for Joint User Association and Resource Allocation

- In this work, the objective of each UE is to maximize its own long-term utility.
- Since the problem is an integer programming problem and the objective of the problem is long-term, it is difficult to adopt the traditional optimization algorithms such as convex optimization to solve it.
- Additionally, the dimension of the decision space increases exponentially. Hence, a distributed DRL-based multi-agent framework for user association and resource allocation is proposed to maximize the long-term utility.

DRL for Joint User Association and Resource Allocation

- The state space, action space and reward function for modeling such a problem are given as follows.

1. **State Space**: In each time slot, the state is composed of the QoS of all the UEs, and we have

$$s(t) = \{s_1(t), s_2(t), \ldots, s_N(t)\}$$

where $si(t)$ is a binary index indicating that whether UE i 's QoS is larger than the minimum threshold i or not, i.e., if the UE i's QoS is larger than i, $si(t) = 1$ and otherwise, $si(t) = 0$.

DRL for Joint User Association and Resource Allocation

2. **Action Space**: In each time t, each UE needs to choose a BS and a channel to access. Hence, the action of UE i consists of two parts, i,e, the user-association vector and the resource-allocation vector

$$a_{li}^{k}(t) = \left\{ b_i^l(t), c_i^k(t) \right\}$$

where $b_{il}(t) \in \{0, 1\}$ and $c_{ik}(t) \in \{0, 1\}$.

DRL for Joint User Association and Resource Allocation

3. **Reward Function**: The reward function of UE i is mainly determined by its achievable rate in time slot t. Besides, to improve the convergence performance of the algorithm, the action-selection cost is also considered in the design of the reward function.

$$R_i\left(t\right) = \begin{cases} \omega_i\left(t\right), & \Gamma_i\left(t\right) \geq \Omega_i \\ -\Psi_i, & \text{otherwise} \end{cases}$$

DRL for Joint User Association and Resource Allocation

- In the proposed framework, each UE is equipped with a DQN to make access decisions independently. In the initialization stage, each UE is first connected to the BS which resulted in the maximum received signal reference power (RSRP) and constructs a DQN, in which the parameter is initialized randomly. At each training time t, each UE has a common state s and selects an action, namely, access request, according to its Q-value Qi (s, ai , θ) obtained from its DDQN. The access request contains the indices of the required BS and the channel. Then, if the BS accepts the request, the BS would send a feedback signal to the UE, which indicates the resource is available, and otherwise, the BS would not reply.

References

1.S.Russell,N.Peter,*ArtificialIntelligence:AModernApproach*,3rdedn.(PrenticeHallPress, Upper Saddle River, 2009)
2.Whatisartificialintelligence,https://www.aisb.org.uk/public-engagement/what-is-ai
3.The 5G future will be powered by AI, https://www.networkcomputing.com/wireless-infrastructure/5g-future-will-be-powered-ai
4.Leveraging machine learning and artificial intelligence for 5G, https://www.cablelabs.com/leveraging- machine- learning- and- artificial- intelligence- for-5g
5.C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, L. Hanzo, Machine learning paradigms for next-generation wireless networks. IEEE Wirel. Commun. **24**(2), 98–105 (2017)
6.T.M.Mitchell,*MachineLearning*(McGraw-Hill,NewYork,1997)
7.R.S. Sutton, A.G. Barto, *Introduction to Reinforcement Learning*, 1st edn. (MIT Press, Cambridge, 1998)

Artificial Intelligence & DSM
References

8. C.M.Bishop,*PatternRecognitionandMachineLearning*(Springer,NewYork,2006)
9. Y.LeCun,Y.Bengio,G.Hinton,Deeplearning.Nature(2015)
10. A.Krizhevsky,I.Sutskever,G.Hinton,Imagenetclassificationwithdeepconvolutionalneural networks, in *Proceedings of the NIPS* (2012)
11. A.Graves,A.Mohamed,G.Hinton,Speechrecognitionwithdeeprecurrentneuralnetworks,in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), BC, Vancouver* (2013), pp. 6645–6649
12. V. Mnih et al., Human-level control through deep reinforcement learning. Nature 518(7540), 529–533 (2015)
13. H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in *Proceedings of the AAAI* (2016), pp. 2094–2100
14. K.M.Thilina,K.W.Choi,N.Saquib,E.Hossain,Machinelearningtechniquesforcooperative spectrum sensing in cognitive radio networks. IEEE J. Sel. Areas Commun. **31**(11), 2209

Cognitive
NETWORKS