M.S. Thesis

Active User Detection for Massive Machine-type Communications via Dimension Spreading Deep Neural Network

대용량 사물 통신에서 차원 확장 심층 신경망을 이용한 활성 단말 검출에 관한 연구

BY

LIM GU-YOUNG

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. Thesis

Active User Detection for Massive Machine-type Communications via Dimension Spreading Deep Neural Network

대용량 사물 통신에서 차원 확장 심층 신경망을 이용한 활성 단말 검출에 관한 연구

BY

LIM GU-YOUNG

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

Massive machine-type communication (mMTC) concerns the access of massive machine-type communication devices to the basestation. To support the massive connectivity, grant-free access and non-orthogonal multiple access (NOMA) have been recently introduced. In the grant-free transmission, each device transmits information without the granting process so that the basestation needs to identify the active devices among all potential devices. This process, called an active user detection (AUD), is a challenging problem in the NOMA-based systems since it is difficult to find out the active devices from the superimposed received signal. An aim of this paper is to propose a new type of AUD scheme suitable for the highly overloaded mMTC, referred to as dimension spreading deep neural network-based AUD (DSDNN-AUD). The key feature of DSDNN-AUD is to set the dimension of hidden layers being larger than the size of a transmit vector to improve the representation quality of the support. Numerical results demonstrate that the proposed AUD scheme outperforms the conventional approaches in both AUD success probability and throughput performance.

keywords: Massive machine-type communication (mMTC), grant-free, non-orthogonal multiple access (NOMA), active user detection (AUD), compressed sensing (CS), deep neural network (DNN)

student number: 2016-20965

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

In recent years, massive machine-type communication (mMTC) has received much attention due to the variety of applications such as autonomous driving, factory automation, public safety and monitoring, smart metering, to name just a few. As the term speaks for itself, mMTC concerns the access of massive machine-type communication (MTC) devices to the basestation [1]. Typically, mMTC focuses on supporting the massive connectivity in the uplink-dominated communication [2]. However, to support massive uplink access is very difficult in the conventional wireless systems (i.e., 4G LTE systems) for the heavy signaling overhead caused by the complicated handshaking-based scheduling [3] and the lack of time/frequency resources caused by the orthogonal resource allocation to a large number of MTC devices.

As a solution to support the massive connectivity, *grant-free access* and *non-orthogonal multiple access* (NOMA) have been introduced in recent years [4]. Grant-free access allows the transmission of MTC device to the basestation without the granting process [5]. Since each device transmits information without scheduling, identification of active devices (i.e., devices transmitting information) among all potential devices in a cell is indispensable. This process, often called an *active user detection* (AUD), is an important problem in the grant-free mMTC since without this process the basestation cannot figure out devices transmitting information. In order to support

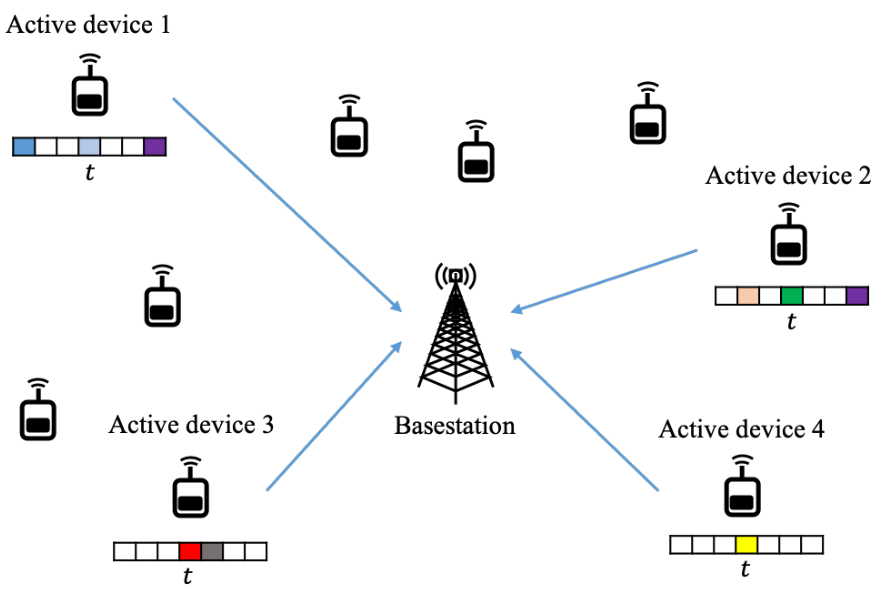

Figure 1.1: mMTC uplink scenario in which only a few active MTC devices transmit the packet (data and pilot symbols) to a basestation.

the massive connectivity using limited amount of resources, an approach to use non-orthogonal sequences, called non-orthogonal multiple access (NOMA), has been proposed [3]. In this scheme, by the superposition of multiple devices' signals, orthogonality of transmit signals is intentionally violated. To control the interuser interference caused by the orthogonality violation, NOMA employs device specific non-orthogonal sequences and deliberately designed nonlinear detector (e.g., message passing algorithm (MPA) [4]).

By exploiting the fact that only a few active devices transmit the information concurrently (see Fig. 1.1), the AUD problem can be formulated as a sparse recovery problem. In solving the problem, compressed sensing (CS) technique has been popularly used [6]-[10]. In [7] and [8], data symbols are used to detect the active users. In [9], pilot signals are used for AUD. While the pilot-based AUD is conceptually simple, its performance depends heavily on the length of pilot sequence. In [10], both pilot and data measurements are exploited and the multiple measurement vector (MMV) based sparse recovery algorithm such as the block orthogonal matching pursuit (BOMP) has been employed. However, performance of the CS-based AUD is not that appealing when the (column-wise) correlation of the system matrix (a.k.a. sensing matrix) increases. Also, performance degradation would be severe when the sparsity (the number of nonzero elements) of the underlying input vector increases. In fact, in many practical mMTC scenarios, correlation among the NOMA sequences and also device activity are high so that the CS-based techniques might not be effective in solving the problem at hand. Therefore, it is of importance to come up with new type of AUD scheme suitable for the highly overloaded access scenarios.

An aim of this paper is to pursue an entirely different approach to solve the AUD problem for the grant-free mMTC scenario. The proposed scheme, referred to as dimension spreading deep neural network-based AUD (DSDNN-AUD), learns the complicated mapping between the received signal and the support[1] using deep neural network (DNN) approach. The key idea of DSDNN-AUD is to set the dimension of hidden layers being larger than the size of a transmit vector to improve the representation quality of the support. In fact, due to the increase in the number of hidden nodes, deep neural network can better discriminate the supports generated from correlated structured environment (e.g., highly overloaded scenario). Numerical simulations demonstrate that the proposed DSDNN-AUD scheme outperforms the conventional CS-based approaches by a large margin, achieving more than 8 dB gain over the pilot-based AUD.

[1]If $\mathbf{s} = [0\ 1\ 0\ 0\ 1\ 0]$, then the support is $\Omega = \{2, 5\}$.

Chapter 2

Grant-free Non-orthogonal Multiple Access

2.1 AUD System Model

We consider the uplink NOMA systems in which the basestation receives information from multiple machine-type devices with a single antenna. In particular, we consider the overloaded scenario where the number of devices N is larger than the number of resources L ($L < N$). Active devices transmit the pilot and data symbols after the spreading with the (device specific) non-orthogonal sequences (see Fig. 2.1). Specifically, the bitstream is mapped to the symbol s_i and then converted into spreading vector $\boldsymbol{q}_i = \boldsymbol{c}_i s_i$ using the device specific codeword $\boldsymbol{c}_i$ [11].

In this work, we employ the low-density signature (LDS) sequence where the codeword of a device has lots of zeros. Due to this sparse nature of a codebook, each symbol is spread into only small number of resources, resulting in the reduction of the interuser interference. Example of the LDS codebook $\mathbf{C}_{(4,6)}$ (6 devices transmit information using 4 resources) is

$$\mathbf{C}_{(4,6)} = \begin{bmatrix} 0 & w_0 & w_1 & 0 & w_2 & 0 \\ w_0 & 0 & w_2 & 0 & 0 & w_1 \\ 0 & w_1 & 0 & w_2 & 0 & w_0 \\ w_2 & 0 & 0 & w_1 & w_0 & 0 \end{bmatrix}, \tag{2.1}$$

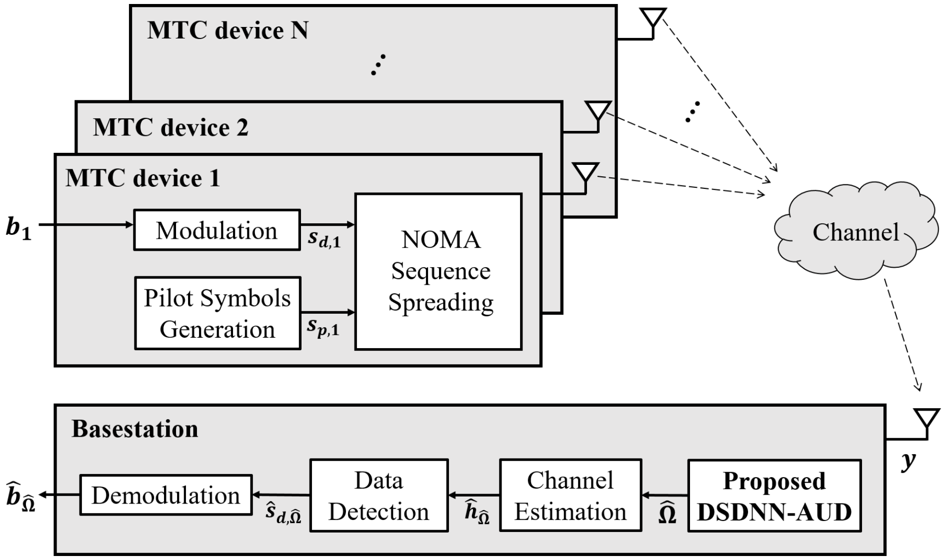

Figure 2.1: Block diagram of the DSDNN-AUD scheme.

where w_j is the non-zero element of the codeword.

Let $s_{p,i}$ and $s_{d,i}$ be the pilot and data symbols for the i-th device, respectively. Then the pilot and data observation vectors $\mathbf{y}_p$ and $\mathbf{y}_d$ at the basestation are given by

$$\begin{aligned} \mathbf{y}_p &= \sum_{i=1}^{N} \operatorname{diag}(\mathbf{c}_{p,i})\mathbf{h}_{p,i}s_{p,i} + \mathbf{v}_p \\ &= \begin{bmatrix} \mathbf{C}_{p,1} & \dots & \mathbf{C}_{p,N} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{p,1}s_{p,1} \\ \vdots \\ \mathbf{h}_{p,N}s_{p,N} \end{bmatrix} + \mathbf{v}_p, \end{aligned} \tag{2.2}$$

$$\begin{aligned} \mathbf{y}_d &= \sum_{i=1}^{N} \operatorname{diag}(\mathbf{c}_{d,i})\mathbf{h}_{d,i}s_{d,i} + \mathbf{v}_d \\ &= \begin{bmatrix} \mathbf{C}_{d,1} & \dots & \mathbf{C}_{d,N} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{d,1}s_{d,1} \\ \vdots \\ \mathbf{h}_{d,N}s_{d,N} \end{bmatrix} + \mathbf{v}_d, \end{aligned} \tag{2.3}$$

where $\mathbf{c}_{p,i}$ and $\mathbf{c}_{d,i}$ are the LDS codeword vectors of the i-th device corresponding to pilot and data, respectively, $\mathbf{h}_{p,i}$ and $\mathbf{h}_{d,i}$ are the channel vectors between the i-

th device and the basestation corresponding to pilot and data, respectively, $\mathbf{v}_p \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ and $\mathbf{v}_d \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ are the complex Gaussian noise vectors, $\mathbf{C}_{p,i} = \text{diag}(\mathbf{c}_{p,i})$ and $\mathbf{C}_{d,i} = \text{diag}(\mathbf{c}_{d,i})$. Since most devices are inactive in mMTC uplink scenarios, the vectors $[(\mathbf{h}_{p,1}s_{p,1})^T \cdots (\mathbf{h}_{p,N}s_{p,N})^T]^T$ and $[(\mathbf{h}_{d,1}s_{d,1})^T \cdots (\mathbf{h}_{d,N}s_{d,N})^T]^T$ are the sparse vectors and further they have the same support.

Since the support information of pilot and data signals in a packet is common, the system model can be readily expressed as the MMV model. Indeed, the transmit packet of active devices contains multiple pilot and data signals in the same resources so that the input of the system can be modeled as a block-sparse vector by stacking the pilot and data observations simultaneously. Let $(\cdot)^{(r)}$ be the vector (or matrix) corresponding to the r-th observation and δ_i be the device activity indicator (i.e., $\delta_i = 1$ for the active device and $\delta_i = 0$ for the inactive device). For notational simplicity, let

$$\begin{aligned}
\mathbf{\Phi}_i &= \text{diag}(\mathbf{C}_{p,i}^{(1)}, \cdots, \mathbf{C}_{p,i}^{(n_p)}, \mathbf{C}_{d,i}^{(1)}, \cdots, \mathbf{C}_{d,i}^{(n_d)}), && (2.4)\\
\mathbf{x}_{p,i} &= \left[(\mathbf{h}_{p,i}^{(1)} s_{p,i}^{(1)})^T \cdots (\mathbf{h}_{p,i}^{(n_p)} s_{p,i}^{(n_p)})^T\right]^T, && (2.5)\\
\mathbf{x}_{d,i} &= \left[(\mathbf{h}_{d,i}^{(1)} s_{d,i}^{(1)})^T \cdots (\mathbf{h}_{d,i}^{(n_d)} s_{d,i}^{(n_d)})^T\right]^T, && (2.6)
\end{aligned}$$

then, the stacked measurement $\mathbf{y}$ can be expressed as

$$\begin{aligned}
\mathbf{y} &= \begin{bmatrix} \mathbf{\Phi}_1 & \ldots & \mathbf{\Phi}_N \end{bmatrix} \begin{bmatrix} \delta_1 \mathbf{x}_1 \\ \delta_2 \mathbf{x}_2 \\ \vdots \\ \delta_N \mathbf{x}_N \end{bmatrix} + \begin{bmatrix} \mathbf{v}_p^{(1)} \\ \vdots \\ \mathbf{v}_p^{(n_p)} \\ \mathbf{v}_d^{(1)} \\ \vdots \\ \mathbf{v}_d^{(n_d)} \end{bmatrix} \\
&= \mathbf{\Phi}\mathbf{x} + \mathbf{v}, && (2.7)
\end{aligned}$$

where $\mathbf{x}_i = [\mathbf{x}_{p,i}^T \ \mathbf{x}_{d,i}^T]^T$, $\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_1 & \ldots & \mathbf{\Phi}_N \end{bmatrix}$, and $\mathbf{x} = [\delta_1 \mathbf{x}_1^T \ \cdots \ \delta_N \mathbf{x}_N^T]^T$.

Since a small number of devices (say k devices) is active, the stacked sparse vector $\mathbf{x}$ has k nonzero blocks. In light of this, main task of the basestation is to identify k submatrices in $\mathbf{\Phi}$ participating in the received vector[1]. The corresponding AUD problem can be formulated as the support identification problem as

$$\tilde{\Omega} = \arg\min_{|\Omega|=k} \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}_{\Omega}\mathbf{x}_{\Omega}\|_2^2 . \tag{2.8}$$

2.2 Conventional AUD

In the AUD process, which is done by the BOMP algorithm, we compute the correlations between the residual vector and each submatrix $\mathbf{\Phi}_i$ (corresponding to the i-th device). Specifically, we choose the index $s^\star$ maximizing the correlation between $\bar{\mathbf{r}}^{k-1}$ and $\mathbf{\Phi}_s$ as

$$s^\star = \arg\max_{s=1,\ldots,N} \|\mathbf{\Phi}_s^H \bar{\mathbf{r}}^{k-1}\|_2^2, \tag{2.9}$$

where $\bar{\mathbf{r}}^{k-1}$ is the residual vector of $\mathbf{y}$ at the k-th iteration (note that $\bar{\mathbf{r}}^0 = \mathbf{y}$). Then we update the support vector $\mathbf{\Omega}^k$ and the subsequent estimate $\mathbf{x}_{\mathbf{\Omega}^k}$ based on LMMSE technique as

$$\mathbf{\Omega}^k = [(\mathbf{\Omega}^{k-1})^T \ \ s^\star]^T, \tag{2.10}$$

$$\hat{\mathbf{x}}_{\mathbf{\Omega}^k} = E[\mathbf{x}_{\mathbf{\Omega}^k}\mathbf{y}^H]E^{-1}[\mathbf{y}\mathbf{y}^H]\mathbf{y}. \tag{2.11}$$

The residual vector $\bar{\mathbf{r}}^k$ used for the next iteration is given by

$$\bar{\mathbf{r}}^k = \mathbf{y} - \sum_{i \in \mathbf{\Omega}^k} \mathbf{\Phi}_i \hat{\mathbf{x}}_i. \tag{2.12}$$

However, this type of greedy sparse recovery algorithms might not be effective in the practical mMTC scenarios for the following reasons. First, correlation of codewords increases with the number of devices. Indeed, when we try to support a large number of devices with relatively small amount of resources (e.g., the number of devices and resource elements are 100 and 20, respectively), column dimension of the codebook $\mathbf{C}$ would be much larger than the size of measurement vector $\mathbf{y}$, increasing

[1] If $\Omega = \{2, 5\}$, then $\mathbf{\Phi}_2$ and $\mathbf{\Phi}_5$ participate in $\mathbf{y}$.

the underdetermined ratio of the system. In this case, clearly, the mutual coherence[2] of $\mathbf{C}$ will increase sharply, causing a severe degradation of the AUD performance. Second, when the activity of devices is high (i.e., k is large), the required number of iterations of the greedy sparse recovery algorithm to identify the support will also increase. In this case, estimation error caused by the residual update in the sparse recovery process is propagated, deteriorating the AUD performance severely. Due to these reasons, design of new type of AUD scheme robust to the increases in the device activity is of great importance for the success of grant-free mMTC.

[2]The mutual coherence $\mu\left(\mathbf{\Phi}\right)$ is defined as the largest magnitude of normalized inner product between two distinct columns of $\mathbf{\Phi}$.

Chapter 3

Support Function Approximation via DNN

3.1 Network Description

As mentioned, main goal of AUD is to identify the nonzero positions of $\mathbf{x}$, not the recovery of nonzero elements. Hence, DSDNN learns the (nonlinear) mapping g between the input (i.e., received signal $\mathbf{y}$) and the desired support of $\mathbf{x}$. The corresponding support identification problem in (2.8) can be expressed as

$$\hat{\Omega} = g(\mathbf{y}; \boldsymbol{\Theta}, \boldsymbol{\Delta}), \tag{3.1}$$

where $\boldsymbol{\Theta}$ and $\boldsymbol{\Delta}$ are sets of the hidden layer weights and biases, respectively. The primary goal of DSDNN is to find g parametrized by $\boldsymbol{\Theta}$ and $\boldsymbol{\Delta}$ given $\mathbf{y}$, closest to the optimal function g^*. To do so, we design the deep neural network in a way to reduce the correlation of sensing matrix $\boldsymbol{\Phi}$. Theoretically, to quantify the level of correlation in a matrix $\boldsymbol{\Phi}$, a restricted isometry property (RIP) has been popularly used [12]. The RIP constant $\delta_k[\boldsymbol{\Phi}]$ is the smallest number satisfying

$$(1 - \delta_k[\boldsymbol{\Phi}])\|\mathbf{x}\|_2^2 \leq \|\boldsymbol{\Phi}\mathbf{x}\|_2^2 \leq (1 + \delta_k[\boldsymbol{\Phi}])\|\mathbf{x}\|_2^2. \tag{3.2}$$

It is now-well known that the sensing matrix with smaller $\delta_k[\boldsymbol{\Phi}]$ achieves better sparse recovery performance, which motivates us to pursue a reduction in $\delta_k[\boldsymbol{\Phi}]$ in the design of DSDNN. Specifically, DSDNN estimates the support $\hat{\Omega}$ via the following optimization problem

$$\hat{\Omega} = \arg \min_{|\Omega|=k} \frac{1}{2} \|\mathbf{\Lambda y} - \mathbf{\Lambda \Phi}_{\Omega} \mathbf{x}_{\Omega}\|_2^2, \tag{3.3}$$

where $\mathbf{\Lambda}$ is the matrix (parametrized by $\mathbf{\Theta}$ and $\mathbf{\Delta}$) representing the multiplicative and additive operations of hidden layers. In the training phase, we train the network such that the composite of $\mathbf{\Lambda}$ and $\mathbf{\Phi}$ has better (lower) correlation structure (i.e., $\delta_k[\mathbf{\Lambda\Phi}] < \delta_k[\mathbf{\Phi}]$).

Fig. 3.1 depicts the structure of the proposed DSDNN. The DSDNN consists of the dimension spreading (DS) modules, fully-connected (FC) layers, pooling layer, and softmax layer. In particular, each DS module is composed of a rectified linear unit (ReLU) and FC layers. The hidden node in the FC layer is modeled as

$$\tilde{\mathbf{q}} = f\left(\mathbf{W}\mathbf{q} + \mathbf{b}\right) \tag{3.4}$$

where $\mathbf{q}$ and $\tilde{\mathbf{q}}$ are the input and output vectors of hidden layer, $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ is the bias vector. f is the element-wise activation function to add the nonlinearity to the network. In DSDNN, we employ the ReLU function to avoid vanishing gradient problem [13]. Since the proposed scheme learns the mapping between $\mathbf{y}$ and the support Ω, the final support $\hat{\Omega}$ will be strongly affected by the activation patterns (presumably on/off patterns) of hidden nodes.

When the coherence of the sensing matrix $\mathbf{\Phi}$ is low, $\mathbf{y}$ can be expressed as a linear combination of *less correlated* columns of $\mathbf{\Phi}$ indexed by Ω which implies that the identification of Ω from $\mathbf{y}$ would be relatively straightforward, even in the presence of noise. However, if the sensing matrix $\mathbf{\Phi}$ is highly correlated, mapping between $\mathbf{y}$ and Ω might not be clear and hence can be easily confused in the presence of randomly distributed perturbations (e.g., channel and noise). Suppose two columns of $\mathbf{\Phi}$ are strongly correlated and only one of these is associated with the support, then it might not be easy to distinguish the correct support element from the wrong one. For example, if $\Omega_1 = \{1, 8\}$ and $\Omega_2 = \{4, 6\}$ and $|\langle \mathbf{\Phi}_1, \mathbf{\Phi}_4 \rangle| \approx 1$ and $|\langle \mathbf{\Phi}_8, \mathbf{\Phi}_6 \rangle| \approx 1$,

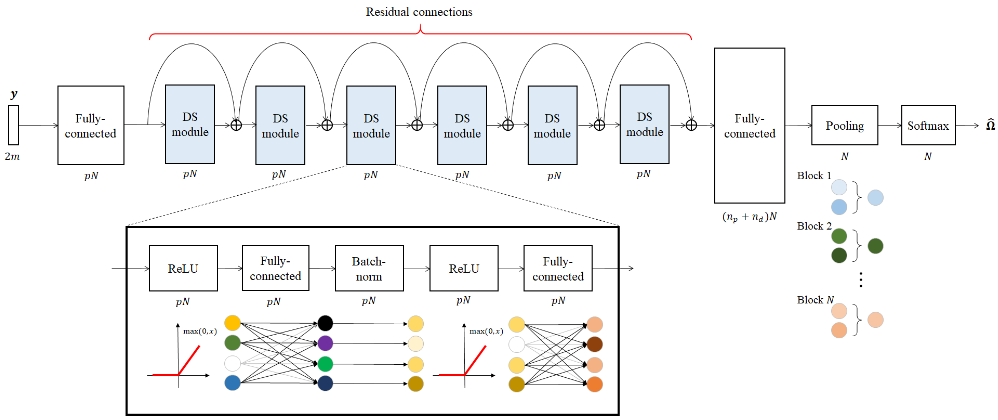

Figure 3.1: Basic network structure of DSDNN for the AUD.

then the activation patterns of hidden nodes for Ω_1 and Ω_2 would be similar, ending up having incorrect support identification by a small perturbation (see Fig. 3.2). In order to mitigate this type of error, we increase the dimension of hidden layers from N to pN ($p > 1$ is a spreading factor). In doing so, the number of hidden nodes increases and thus the capacity to represent the support using the activation patterns can be improved. In other words, by increasing the dimension of hidden layers, the similarity (ambiguity) of the activity patterns among correlated supports can be better resolved, which implies that DSDNN can identify the support accurately even when the sensing matrix is badly correlated.

After the DS modules, the FC layer produces $(n_p + n_d)N$ output values whose dimension is matched with the size of stacked sparse input vector $\mathbf{x}$. In order to exploit the block sparse structure of $\mathbf{x}$ (see (2.7)), we add the pooling layer before the final softmax operator. In the pooling layer, all the elements in each block ($n_p + n_d$ elements) are averaged, generating N block-wise weighted output values. After the pooling operation, the softmax layer generates N soft values representing the probability of being the support element. Finally, estimate of the support is obtained by taking k elements having the largest probabilities.

In the proposed DSDNN, the batch normalization enforcing the distribution of

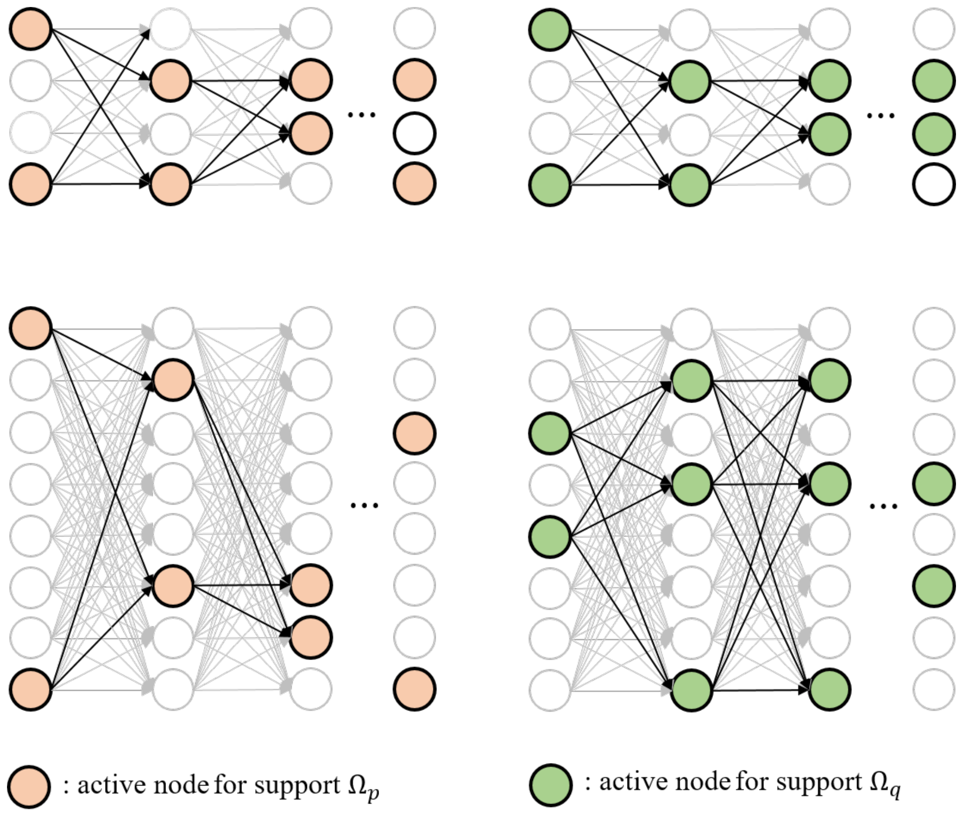

Figure 3.2: Examples of activation patterns of strongly correlated supports Ω_p and Ω_q. By increasing the number of hidden nodes, the similarity of activation patterns among the correlated supports is reduced significantly.

each layer input to have zero mean and unit variance is used to reduce the internal covariate shift caused by the randomly-generated channel and also boost the training speed [14]. Further, residual connections borrowed from ResNet architecture is used to prevent vanishing and exploding gradients caused by deep layer structure [15]. In the proposed scheme, the input of DS module links to the output of DS module directly so that the stability of gradients is guaranteed in the DSDNN. Since the support identification problem is a classification problem using the support Ω as a label, we use the softmax cross entropy $\mathcal{J}(\Omega, \hat{\Omega})$ as a loss function. That is,

$$\mathcal{J}(\Omega, \hat{\Omega}) = -\sum_{i=1}^{k} p(\omega_i) \ln(p(\hat{\omega}_i)), \tag{3.5}$$

where $p(\omega_i) = 1/k$ for $\omega_i \in \Omega$ and $p(\hat{\omega}_i)$ is the probability of the estimated support element $\hat{\omega}_i$ being equivalent to the true support.

3.2 Training Issue in DSDNN

One important issue in the training phase is that the abundant amount of training datasets are needed. Fortunately, in wireless communication systems, a large amount of the training examples $\{\mathbf{y}, \Omega\}$ can be generated using the channel statistics and noise distribution. However, when the randomness of the received signal components (e.g., channel, noise, and codewords) is high, learning parameters might not converge and thus the test performance of the learned network would be poor. Hence, in order to ensure the convergence of DNN, it is important to reduce the degree of randomness in training examples.

To this end, we first generate the flat fading channels whose channel components are independently drawn for each device. This is reasonable for the short packet transmission in the mMTC systems. Recall that the packet transmission time nT_s (n is the number of symbols and T_s is the symbol duration in LTE systems) is typically much smaller than the channel coherence time T_c when the packet size is short. For example, when the carrier frequency is $f_c = 2$ GHz and the device speed is $\nu = 10$ km/h, then $T_c = \frac{9c}{16\pi\nu f_c} = 3.99$ ms is much larger than nT_s ($T_s = 0.07$ ms) for small n [16]. Also, we generate a predefined LDS codebook $\mathbf{C}$ in our training. In this manner, we can easily obtain the massive synthetic training data with reduced randomness.

Another issue of the DSDNN is that the amount of training data depends heavily on the number of active devices. Indeed, the number of different labeled datasets increases quickly as the number of devices increases, resulting in the significant increase in the required amount of training data. To address this issue, we generate the training data and then train the network periodically by fixing the number of active devices. In the mMTC systems, this setup is reasonable for two reasons. First, the number of active devices remains unchanged in several time slots [17]. Second, the number of active devices is much smaller than the number of total devices even in the busy hours [18]. Furthermore, due to the static and sporadic packet transmission, the off-line training of DSDNN can be performed with long periodicity (e.g., once a week) and thus training complexities would not be a burden in practical scenarios.

Chapter 4

Simulations and Discussions

4.1 Simulation Setup

In our simulations, we consider the NOMA-based grant-free transmission in the orthogonal frequency division multiplexing (OFDM) systems. We set the total number of MTC devices to 100 ($N = 100$) and active devices are randomly chosen among them. Each device transmits 3 pilot symbols and 7 data symbols using the quadratic phase shift keying (QPSK). To test highly overloaded scenarios, we use 20×100 LDS codebook ($m = 20$). As a channel model, we use the Rayleigh fading channel whose tap coefficients are independently distributed for each device.

For comparison, we use the pilot-based AUD and hybrid-AUD schemes. When performing AUD, pilot-based AUD scheme uses pilot symbols exclusively and hybrid-AUD scheme uses both the pilot and data symbols. In both schemes, BOMP is used to identify the support. In the DSDNN-AUD, we generate 10^5 different samples for training and testing. In the training phase, we set the learning rate to 0.0004, the size of batch to 500, and the size of DS module to 500 (i.e., $p = 5$). As a performance metrics, we use the AUD success probability and data throughput per device. The AUD success probability is defined as the probability that all active devices are identified accurately. When obtaining the data throughput, we perform the channel estimation

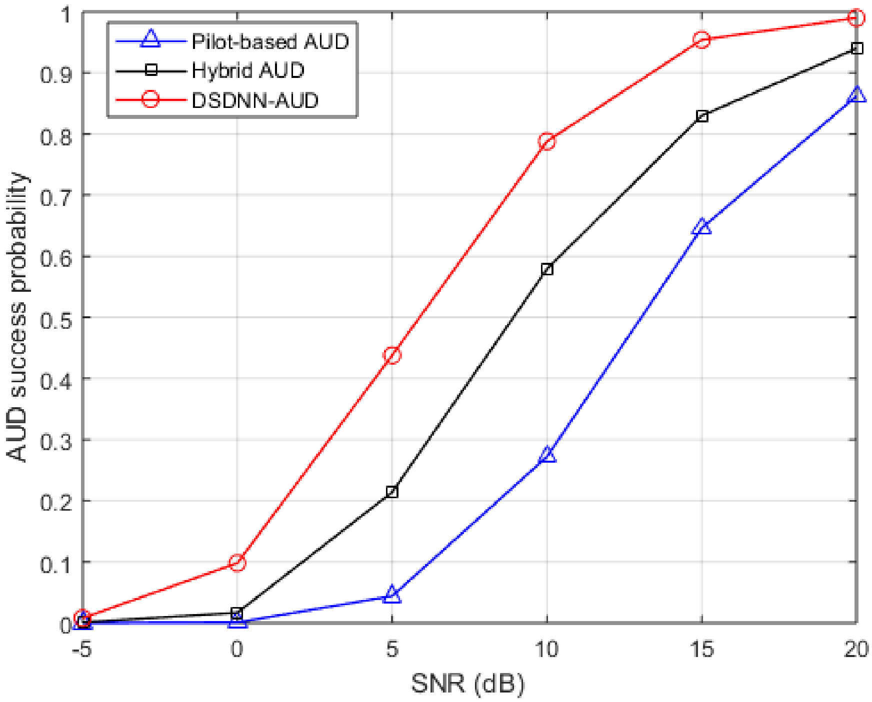

Figure 4.1: AUD success probability as a function of SNR for three different AUD schemes ($k = 4$).

using minimum mean square error (MMSE) and symbol detection using MPA algorithm.

4.2 Simulation Results

In Fig. 4.1, we evaluate the AUD success probability as a function of SNR. We observe that the DSDNN-AUD outperforms both the pilot-based AUD and hybrid AUD in all SNR regime. For example, when the AUD probability is 0.8, the proposed scheme achieves huge gain (around 8 dB) over the pilot-based AUD scheme and 5 dB gain over the hybrid AUD scheme. This is because DSDNN-AUD resolves the similarity of the activation patterns among correlated supports and thus reduces the coherence of

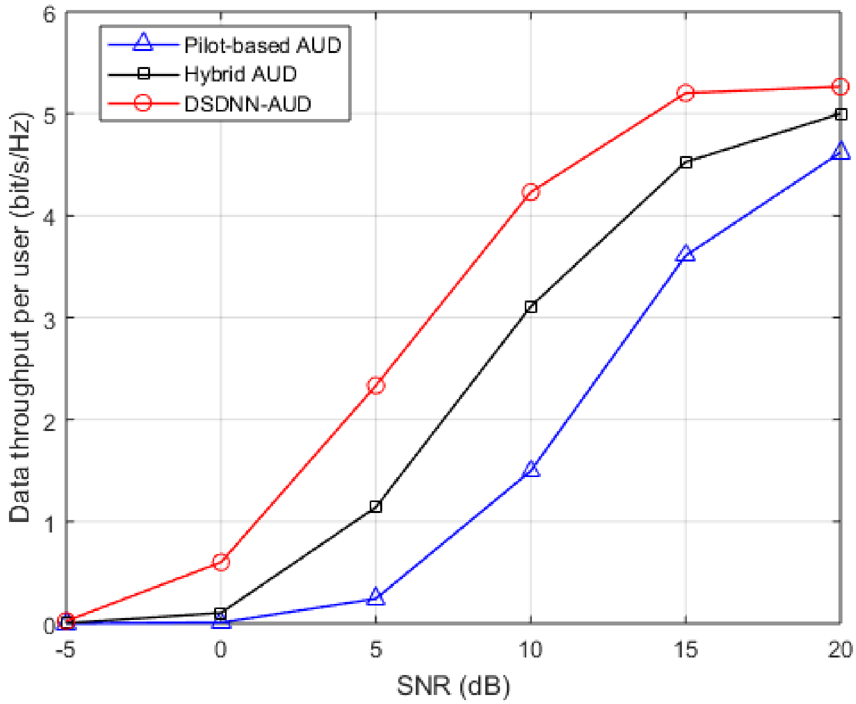

Figure 4.2: Data throughput corresponding to the correctly detected active devices as a function of SNR ($k = 4$).

sensing matrix $\mathbf{\Phi}$.

In Fig. 4.2, we evaluate the data throughput as a function of SNR. We observe that the data throughput of the proposed DSDNN-AUD is higher than that of the pilot-based and hybrid AUD schemes. For example, when the SNR is 10 dB, the proposed DSDNN-AUD scheme achieves about 183% gain over the pilot-based AUD scheme and 36% gain over the hybrid AUD scheme. These results clearly demonstrate that the AUD performance directly affects the data throughput.

In Fig. 4.3, we plot the AUD success probability under the different numbers of active devices. We observe that the AUD performance of proposed DSDNN-AUD scheme is higher than that of conventional AUD schemes in all SNR regime. For example, when the number of active devices is 18 (i.e., $k = 18$), the AUD success

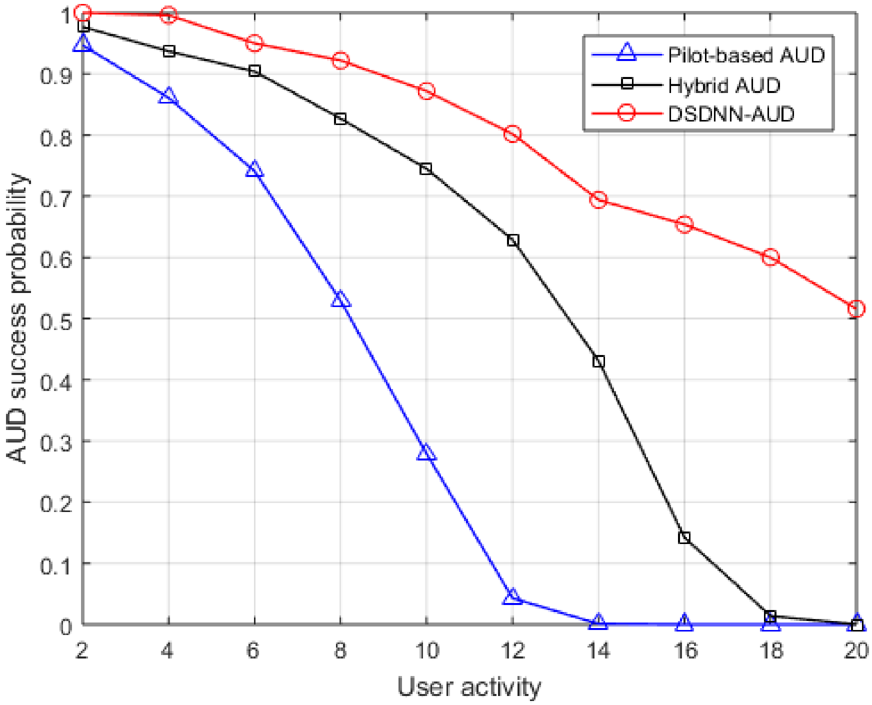

Figure 4.3: AUD success probability as a function of the number of active devices (SNR$=$ 20 dB).

probability of the DSDNN-AUD is 0.6 while that of the conventional AUD schemes are (nearly) zero. Further, when k increases from 4 to 10, the AUD success probability of the DSDNN-AUD scheme decreases from 0.99 to 0.88 but that of the pilot-based AUD and hybrid AUD schemes decreases sharply from 0.87 to 0.28 and from 0.94 to 0.74, respectively. This clearly implies that the DSDNN-AUD scheme is robust to the increase in the number of active devices and hence it is very effective for the highly overloaded mMTC scenarios.

Chapter 5

Conclusion

In this paper, we proposed a novel AUD scheme based on the deep neural network for the mMTC uplink scenario. Our work is motivated by the observation that CS-based AUD could not support the massive number of devices and high device activity cases in the grant-free NOMA systems. In the proposed DSDNN-AUD scheme, we set the dimension of hidden layers being larger than the size of a transmit vector. In doing so, the representation quality of the support is improved and hence DSDNN can identify the support accurately. We showed from the numerical simulations in the massive access scenario that the proposed DSDNN-AUD scheme is very effective in the highly overloaded mMTC.

Bibliography

[1] Rec. ITU-R M.2083-0, “IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond,” Sep 2015.

[2] A. Osserian, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. Uusitalo, B. Timus, and M. Fallgren, “Scenarios for 5G mobile and wireless communications: the vision of the METIS project,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26-35, May 2014.

[3] T. Taleb and A. Kunz, “Machine type communications in 3GPP networks: Potential, challenges, and solutions,” *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 178-184, Mar 2012.

[4] R. Hoshyar, F. P. Wathan, and R. Tafazolli, “Novel low-density signature for synchronous CDMA systems over AWGN channel,” *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1616-1626, Apr 2008.

[5] K. Au et al., “Uplink contention based SCMA for 5G radio access,” *Globecom Workshops (GC Wkshps), 2014*, Austin, TX, 2014, pp. 900-905.

[6] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, “Compressed Sensing for Wireless Communications: Useful Tips and Tricks,” *IEEE Commun. Survey and Tutorials*, vol.19, pp.1527-1550, 2017.

[7] B. C. Wang, L. L. Dai, Y. Zhang, T. Mir, and J. J. Li, "Dynamic compressive sensing based multi-user detection for uplink grant-free NOMA," *IEEE Communications Letters*, vol. 20, no. 5, pp. 2320-2323, Nov 2016.

[8] A. C. Cirik, N. M. Balasubramanya, and L. Lampe, "Multi-User Detection Using ADMM-Based Compressive Sensing for Uplink Grant-Free NOMA," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 46-49, Feb 2018.

[9] S. Park, H. Seo, H. Ji, and B. Shim, "Joint Active User Detection and Channel Estimation for Massive Machine-Type Communications," *Proc. of IEEE workshop on signal processing advances in wireless communications (SPAWC) 2017.*

[10] G. Lim, H. Ji, and B. Shim, "Hybrid Active User Detection for Massive Machine-type Communications in IoT," *The 9th International Conference on Information and Communication Technology Convergence (ICTC 2018).*

[11] B. Shim and B. Song, "Multiuser detection via compressive sensing," *IEEE Communications Letters*, vol. 16, no. 7, pp. 972-974, July 2012.

[12] E.J. Candès, "The restricted isometry property and its implications for compressed sensing," C. R. l'Acad. Sci. 346 (2008) 589–592.

[13] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807-814, 2010.

[14] S. Loffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167 [cs], Mar 2015.

[15] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016.

[16] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[17] A. T. Abebe and C. G. Kang, “Iterative order recursive least square estimation for exploiting frame-wise sparsity in compressive sensing-based MTC,” *IEEE Communications Letters*, vol. 20, no. 5, pp. 1018-1021, May 2016.

[18] J. P. Hong, W. Choi, and B. D. Rao, “Sparsity controlled random multiple access with compressed sensing,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 998-1010, Feb 2015.

초 록

대용량 사물 통신 (massive machine type communication, mMTC)은 다수의 사물 통신 기기들이 기지국에 접속하는 상황과 관계가 있다. 대규모 연결성을 지원하기 위하여 최근에 비승인 접속과 비직교 다중 접속 (non-orthogonal multiple access, NOMA)이 고려되었다. 비승인 기반 전송 시, 각 기기는 승인 절차 없이 정보를 전송하기 때문에 기지국은 모든 기기들 중 활성 상태에 있는 기기들만을 검출하는 과정을 수행해야 한다. 이러한 절차를 활성 기기 검출 (active user detection, AUD)이라고 하며, 비직교 다중 접속 기반의 시스템에서는 수신 신호에 활성 기기들의 신호들이 중첩되어 있기 때문에 활성 기기를 검출하는 것은 어려운 문제이다. 본 논문에서는 전체 기기의 수가 매우 많은 대용량 사물 통신에 적합한 새로운 방식의 활성 기기 검출 기술을 제안한다. 이 기술을 차원 확장 심층 신경망 기반의 활성 기기 검출 (dimension spreading deep neural network based active user detection, DSDNN-AUD)이라고 명명하며, 본 기술의 핵심적인 특징은 은닉층의 차원을 송신 정보 벡터의 크기보다 크게 설정함으로써 서포트 검출 능력을 향상시키는 것이다. 모의 실험 결과를 통해 제안하는 활성 기기 검출 기술이 기존의 기법들보다 활성 기기 검출 성공 확률과 스루풋 성능 관점에서 우수함을 확인했다.

주요어: 대용량 사물 통신, 비승인 접속, 비직교 다중 접속, 활성 기기 검출, 압축 센싱, 심층 신경망

학번: 2016-20965