Active User Detection of Machine-type Communications via Dimension Spreading Neural Network

Wonjun Kim, Guyoung Lim, Yongjun Ahn and Byonghyo Shim
Institute of New Media and Communications and Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
Email: {wjkim, gylim, yjahn, bshim}@islab.snu.ac.kr

Abstract—Massive machine-type communication (mMTC), key component for internet of things (IoT), 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. In doing so, the proposed scheme can better discriminate the supports generated from correlated structured environment. Numerical results demonstrate that the proposed AUD scheme outperforms the conventional approaches in both AUD success probability and throughput performance.

I. INTRODUCTION

In recent years, massive machine-type communication (mMTC) has received much attention due to the variety of IoT 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]. Main goal of mMTC is to support the massive connectivity in the uplink-dominated communication [2]. However, this task is quite demanding since it is difficult to find out the active devices from the superimposed received signal. This process, 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 the massive connectivity with limited amount of resources, an approach to use non-orthogonal sequences, called non-orthogonal multiple access (NOMA), has been proposed [5]. 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) [5]).

By exploiting the fact that only a few active devices transmit the information concurrently (see Fig. 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]-[8]. In these works, 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 AUD 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 learning approach. 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. 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.

II. 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). Specifically, the bitstream is mapped to the symbol \(s_i\) using the device specific codeword \(c_i\) [9].

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 \(C_{(4,6)}\) (i.e., 6 devices transmit information using 4 resources) is

\[
C_{(4,6)} = \begin{bmatrix}
0 & w_0 & w_1 & 0 & w_2 & 0 \\
0 & w_0 & w_2 & 0 & 0 & w_1 \\
0 & w_1 & 0 & w_2 & 0 & w_0 \\
w_2 & 0 & 0 & w_1 & 0 & w_0
\end{bmatrix},
\]

where \(w_j\) is the nonzero element of the codeword.

\(^1\)If \(s = [0 \ 1 \ 0 \ 0 \ 1 \ 0]\), then the support is \(\Omega = \{2, 5\}\).

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 \(y_p\) and \(y_d\) at the basestation are given by

\[
y_p = \sum_{i=1}^{N} \text{diag}(c_{p,i}) \mathbf{h}_{p,i} s_{p,i} + \mathbf{v}_p
\]

\[
= \begin{bmatrix}
\mathbf{h}_{p,1} s_{p,1} \\
\vdots \\
\mathbf{h}_{p,N} s_{p,N}
\end{bmatrix} + \mathbf{v}_p, \quad (2)
\]

\[
y_d = \sum_{i=1}^{N} \text{diag}(c_{d,i}) \mathbf{h}_{d,i} s_{d,i} + \mathbf{v}_d
\]

\[
= \begin{bmatrix}
\mathbf{h}_{d,1} s_{d,1} \\
\vdots \\
\mathbf{h}_{d,N} s_{d,N}
\end{bmatrix} + \mathbf{v}_d, \quad (3)
\]

where \(c_{p,i}\) and \(c_{d,i}\) 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, \(\mathbf{C}_{p,i} = \text{diag}(c_{p,i})\) and \(\mathbf{C}_{d,i} = \text{diag}(c_{d,i})\). 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 \(\delta_i\) be the device activity indicator (i.e., \(\delta_i = 1\) for the active device and
\[ \delta_i = 0 \text{ for the inactive device}. \] Also, let
\[ \Phi_i = \text{diag}(C_p^{(1)}, \ldots, C_p^{(n_p)}, C_d^{(1)}, \ldots, C_d^{(n_d)}), \tag{4} \]
\[ \mathbf{x}_{p,i} = \begin{bmatrix} (h_p^{(1)})^T \cdot (s^{(1)}_{p,i})^T \ldots (h_p^{(n_p)})^T \cdot (s^{(n_p)}_{p,i})^T \end{bmatrix}^T, \tag{5} \]
\[ \mathbf{x}_{d,i} = \begin{bmatrix} (h_d^{(1)})^T \cdot (s^{(1)}_{d,i})^T \ldots (h_d^{(n_d)})^T \cdot (s^{(n_d)}_{d,i})^T \end{bmatrix}^T, \tag{6} \]
then, the stacked measurement \( \mathbf{y} \) can be expressed as
\[
\mathbf{y} = \begin{bmatrix} \Phi_1 \ldots \Phi_N \end{bmatrix} \begin{bmatrix} \delta_1 \mathbf{x}_1 \\ \delta_2 \mathbf{x}_2 \\ \vdots \\ \delta_N \mathbf{x}_N \end{bmatrix} + \begin{bmatrix} v_1^{(1)} \\ \vdots \\ v_p^{(n_p)} \\ v_i^{(1)} \\ \vdots \\ v_d^{(n_d)} \end{bmatrix} = \Phi \mathbf{x} + \mathbf{v}, \tag{7} \]
where \( \mathbf{x}_i = [\mathbf{x}_{p,i}^T \mathbf{x}_{d,i}^T]^T, \Phi = [\Phi_1 \ldots \Phi_N], \) and \( \mathbf{x} = [\delta_1 \mathbf{x}_1^T \ldots \delta_N \mathbf{x}_N^T]^T. \)
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 \( \Phi \) participating in the received vector\(^2\). The corresponding AUD problem can be formulated as the support identification problem as
\[
\tilde{\Omega} = \arg \min_{|\Omega| = k} \frac{1}{2} \| \mathbf{y} - \Phi \Omega \mathbf{x} \|_2^2. \tag{8} \]
In solving the problem, greedy sparse recovery algorithm (e.g., BOMP [10]) can be employed. However, this type of 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 the underdetermined ratio of the system. In this case, clearly, the mutual coherence\(^3\) of \( \mathbf{C} \) will also 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. Due to these reasons, new type of AUD scheme robust to the increases in the device activity is of great importance for the success of grant-free mMTC.

### III. SUPPORT FUNCTION APPROXIMATION VIA DNN

#### A. 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 (8) can be expressed as
\[
\tilde{\Omega} = g(\mathbf{y}; \Theta, \Delta), \tag{9} \]
where \( \Theta \) and \( \Delta \) are sets of the hidden layer weights and biases, respectively. The primary goal of DSDNN is to find \( g \) parametrized by \( \Theta \) and \( \Delta \) given \( y \), 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 \( \Phi \). Theoretically, to quantify the level of correlation in a matrix\(^3\)

\(^2\)If \( \Omega = \{2, 5\} \), then \( \Phi_2 \) and \( \Phi_5 \) participate in \( \mathbf{y} \)

\(^3\)The mutual coherence \( \mu(\Phi) \) is defined as the largest magnitude of normalized inner product between two distinct columns of \( \Phi \).
the following optimization problem:
\[
\hat{\Omega} = \arg \min_{\Omega} \frac{1}{2} \| \Lambda y - \Lambda \Phi \hat{\Omega} \|_2^2.
\]

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

\[
\hat{\Omega} = \arg \min_{|\Omega|=k} \frac{1}{2} \| \Lambda y - \Lambda \Phi \Omega \|_2^2,
\]

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

Fig. 3 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

\[
\hat{q} = f(Wq + b)
\]

where \( q \) and \( \hat{q} \) are the input and output vectors of hidden layer, \( W \) is the weight matrix, and \( b \) 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 [11]. Since the proposed scheme learns the mapping between \( y \) and the support \( \Omega \), 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 \( \Phi \) is low, \( y \) can be expressed as a linear combination of less correlated columns of \( \Phi \) indexed by \( \Omega \), which implies that the identification of \( \Omega \) from \( y \) would be relatively straightforward even in the presence of noise. However, if the sensing matrix \( \Phi \) is highly correlated, mapping between \( y \) and \( \Omega \) 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 \( \Phi \) 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 \(|\langle \Phi_1, \Phi_4 \rangle| \approx 1 \) and \(|\langle \Phi_8, \Phi_6 \rangle| \approx 1 \), then the activation patterns of hidden nodes for \( \Omega_1 \) and \( \Omega_2 \) would be similar, ending up having incorrect support identification by a small perturbation (see Fig. 4). 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\)). When the number of hidden nodes increases, the capacity to represent the support using the activation patterns can also 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 in turn 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 \( x \). In order to exploit the block sparse structure of \( x \) (see (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 each layer input to be the 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 [12]. Further, residual connections borrowed from ResNet architecture [13] is used to prevent vanishing and exploding gradient problems caused by the deep layer structure. 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 \( \Omega \) as a label, we use the softmax cross entropy \( J(\Omega, \hat{\Omega}) \) as a loss function. That is,

\[
J(\Omega, \hat{\Omega}) = -\sum_{i=1}^{k} p(\omega_i) \ln(p(\hat{\omega}_i)),
\]

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.

B. Training Issue in DSDNN

One well-known problem in the training phase is the abundant amount of training datasets. Fortunately, in wireless communication systems, a large amount of the training examples \( \{y, \Omega\} \) can be generated synthetically 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 one cannot expect the reliable test performance of the learned network. 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 \( T_c = \frac{9c}{4f_c\nu} = 3.99 \) ms is much larger than \( nT_s \) (\( T_s = 0.07 \) ms) for small \( n \) [14]. Also, we generate a predefined LDS codebook \( C \) 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 [2]. Second, the number of active devices is much smaller than the number of total devices even in the busy hours [3]. 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.

IV. SIMULATIONS AND DISCUSSIONS

A. 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 \( 20 \times 100 \) LDS codebook (\( m = 20 \)). As a channel model, we use the Rayleigh fading channel whose tap coefficients are independently distributed for each device. The noise vector \( \nu \) is generated by a zero mean Gaussian vector with variance adjusted to have a desired signal to noise ratio (SNR).

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

V. Conclusion

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

REFERENCES