M.S. THESIS

Sparse Vector Decoding using Deep Neural Network for Ultra Reliable Short Packet Transmission

고신뢰 짧은 패킷 전송을 위한 깊은 신경망을 이용한 희소 벡터 복호에 관한 연구

BY

LEE SEUNG-HWAN

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Sparse Vector Decoding using Deep Neural Network for Ultra Reliable Short Packet Transmission

고신뢰 짧은 패킷 전송을 위한 깊은 신경망을 이용한 희소 벡터 복호에 관한 연구

BY

LEE SEUNG-HWAN

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Sparse Vector Decoding using Deep Neural Network for Ultra Reliable Short Packet Transmission

고신뢰 짧은 패킷 전송을 위한 깊은 신경망을 이용한 희소 벡터 복호에 관한 연구

지도교수 심 병 효

이 논문을 공학석사 학위논문으로 제출함

2019년 2월

서울대학교 대학원

전기 컴퓨터 공학부

이 승 환

이승환의 공학석사 학위 논문을 인준함

2019년 2월

위 원 장: ____________________________

부위원장: ____________________________

위　　원: ____________________________

Abstract

Ultra-reliable and low latency communication (URLLC) is one of the prospective service categories in 5G to be useful in the future hyper-connective industrial field. To support its requirements, 3rd Generation Partnership Project (3GPP) sets an aggressive standard that a packet should be delivered within 1 ms transmission period with an accuracy of 99.999%. Since the current 4G systems designed to maximize the coding gain by transmitting capacity achieving long codeblock resulting in an increase of the latency. A recently proposed approach for the short packet transmission is sparse vector coding (SVC). In SVC, encoding is done by simple sparse mapping and spreading to formulate the system model into an underdetermined system and replaces the decoding process with a simple sparse recovery algorithm. In this paper, we propose a deep neural network-based approach, referred to as deep sparse vector decoding (deep-SVD), to enhance the performance of SVC to better meet the URLLC's extreme requirements. To this end, we reformulate the SVC-decoding process as a multi-label classification and build the network to learn the highly correlated relationship within codebook. Numerical results demonstrate that the proposed deep-SVD outperforms the conventional SVC decoding in both reliability and latency.

keywords: 5G, URLLC, Short packet transmission, SVC, Deep neural network, Compressed sensing

student number: 2017-22314

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

In preparation of the upcoming fourth industrial revolution, hyper-connectivity is emphasized as a core value. Information exchange is spreading beyond the daily life of individuals and is being extended to the whole area of the industry. In this change, the next-generation infrastructures require forms of communication services and applications that are different from traditional human-centric communication in terms of latency, reliability, energy efficiency, and connection density. Wireless systems from 2G to today's 4G have been focused on increasing higher data rates. While this trend is expected to continue in the fifth generation (5G) wireless systems, there are strong indications that 5G will not only be faster than 4G but will also provide services focused on specific requirements. In response to these needs, International Telecommunication Union(ITU) has classified 5G services into three categories: ultra-reliable and low latency communication (URLLC), massive machine-type communication (MMTC), and enhanced mobile broadband (eMBB) [1]. Each of these services requires extreme performance with respect to lower latency & higher reliability, massive connectivity, and better energy efficiency, respectively. Since the current radio access mechanism cannot support these changes, 3rd Generation Partnership Project (3GPP) introduced a new air interface referred to as $NewRadio$ (NR) [2]. The primary goal of NR is to bring entirely new features and technologies that are not compatible with current 4G systems.

Among the three services mentioned, URLLC has attracted much attention in industries with its potential in applications where super real-time and reliable connections are required, e.g. remote medical surgery, factory automation, smart cities, and autonomous vehicle [3]. This is because the main challenging requirements in URLLC is ultra-low latency and ultra high reliability. In order to support this new service category, according to the 3GPP standards, the desired performance value for URLLC are the low latency of less than 5ms and accuracy of 99.999$\%(= 10^{-5}$ error rate$)$ [4]-[5]. To support the reliability in current 4G systems, complex channel coding scheme (e.g. convolutional coding and turbo coding) is done to maximize the coding gain by transmitting capacity achieving long codeblock resulting in an increase of the latency while targeting BLER performance is 10^{-2} to 10^{-3} [6]. One thing to note is that the information in the service areas that URLLC expects to be utilized is a short packet unit of information such as control type information (e.g., move up/down, speed up/down, and start/stop) or sensing information (e.g. temperature, moisture, and pressure) [7]. Therefore, the coding schemes currently applied in 4G systems are not efficient in the context of URLLC.

Recently, an approach to support a short packet transmission based on the principle of compressed sensing, called sparse vector coding (SVC), has been proposed [8]. The main idea of SVC is that the data information is mapped into the position of a sparse vector and then transmitted after the spreading to formulate the system model into an underdetermined system. Also, the decoding process is done by finding the non-zero positions of the sparse vector using any sparse recovery algorithms also known as compressed sensing [9]-[10]. It has been shown that the BLER of SVC outperforms the conventional channel coding schemes. While there exist numerous compressed sensing algorithms, a greedy-based approach is widely used. One potential problem of the conventional greedy algorithms is that the incorrect non-zero index would be selected when the internal correlation of the sensing matrix is high, and consequently leads to the failure of the packet decoding.

In recent studies, the deep learning technique has shown tremendous performance in classification problems [11]. An aim of this paper is to enhance the performance of SVC to better meet URLLC's extremely high performance requirements by applying deep neural network (DNN). To this end, we reformulate the SVC decoding process as a multi-label classification where the non-zero positions of the sparse vector are labels to be classified from received SVC encoded vector. The proposed scheme, referred to as deep sparse vector decoding (deep-SVD), is a supervised learning of which its input is the received signal and the output is the SVC encoded sparse vector. In other words, the proposed scheme utilizes the powerful ability of DNN as a function approximator to approximates the sparse recovery algorithm [12].

The main structure of deep-SVD is to utilize the residual learning framework which was first introduced in Resnet [13]. Due to the residual learning framework, the entire networks are similar to unfolding the conventional iterative sparse recovery algorithms. The differences and also the advantage of the proposed deep-SVD is to put the learnable parameters which give more flexibility than conventional sparse recovery algorithms that use a fixed sensing matrix at every iteration. This implies that the optimal parameters achieved during the training phase of networks might help to alleviate the disruptive correlation of the sensing matrix. In other words, distinguishing between highly correlated codewords in the codebook is more efficient. Therefore, Deep-SVD not only just approximates the sparse vector recovery functions but also performs codebook adaptive support detection by learning those correlations compared to the conventional linearly operating compressed sensing algorithms. In a realistic scenario, deep-SVD is pre-trained at the receiver by transmitting the virtual data signals for training before the data transmission is conducted. At the actual data transmission, pre-trained deep-SVD consists of simple matrix multiplication, making it more suitable for low latency requirements. From the numerical evaluations, we demonstrate that the proposed deep-SVC technique outperforms the conventional decoding scheme based on a greedy algorithm by a large margin in terms of high reliability.

Chapter 2

Short Packet Transmission Using Sparse Vector Coding

In this section, we briefly overview SVC for short packet transmission. We consider the single-user OFDM system model. In the conventional 4G systems, the transmit vector $\mathbf{x} \in \mathbb{C}^{m\times 1}$ is generated via the channel coding and symbol mapping of data information. After passing the channel, the received vector $\mathbf{y} \in \mathbb{C}^{m\times 1}$ is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \tag{2.1}$$

where $\mathbf{H} = \text{diag}(\bar{\mathbf{h}})$ is the diagonal matrix where $\bar{h}_i$ is the channel frequency response at the i-th resource, and $\mathbf{v} \sim \mathcal{CN}(\mathbf{0}, \sigma_v^2\mathbf{I})$ is the additive Gaussian noise.

2.1 SVC encoding

The first step of sparse vector encoding process is to map the information into the positions of a sparse vector $\mathbf{s}$. Let $\mathbf{w}$ be the b-bit data information, then the sparse vector mapping $\alpha : \mathbb{B}^b \rightarrow \mathbb{B}^N$ maps $\mathbf{w}$ to k-sparse binary vector $\mathbf{s} \in \mathbb{B}^N$ which has k non-zero position among N. In this mechanism, when we choose k out of N positions, we can encode $\lfloor \log_2 \begin{pmatrix} N \\ k \end{pmatrix} \rfloor$. For example, if $\mathbf{s}$ is 9-dimentional binary vector with the sparsity $k = 2$, we can encode 5-bit data information, then (see exmple in Table. 2.1)

$$\begin{aligned}
\mathbf{w} = [0\ 0\ 0\ 0\ 0] \quad &\overset{\alpha}{\rightarrow} \quad \mathbf{s} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1] \\
\mathbf{w} = [0\ 0\ 0\ 0\ 1] \quad &\overset{\alpha}{\rightarrow} \quad \mathbf{s} = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0] \\
&\vdots \\
\mathbf{w} = [1\ 1\ 1\ 1\ 1] \quad &\overset{\alpha}{\rightarrow} \quad \mathbf{s} = [1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0].
\end{aligned}$$

After the sparse mapping, the next step is to spread the sparse vector $\mathbf{s}$ into m resources using the spreading codebook $\mathbf{C}$. As proposed in [3], we allocate the resources along the frequency axis, therefore, the transmission latency is efficiently minimized (see Fig. 2.1). As a result of the spreading process, the transmit vector $\mathbf{x}$ takes the distinctive form since the vector $\mathbf{s}$ is sparse. For example, if $k = 2$ and its non-zero position is first and third, the transmit vector $\mathbf{x}$ is given by

$$\begin{aligned}
\mathbf{x} &= \mathbf{C}\mathbf{s} \\
&= s_1\mathbf{c}_1 + s_3\mathbf{c}_3, \qquad (2.2)
\end{aligned}$$

where $\mathbf{c}_i$ is the spreading codeword from the codebook matrix $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_N]$. It is worth mentioning that since the positions of non-zero elements are chosen randomly, the codebook matrix $\mathbf{C}$ should be designed such that the transmit vector $\mathbf{x}$ contains enough information to recover the sparse vector $\mathbf{s}$ irrespective of the selection of the non-zero positions. In this work, we consider random Bernoulli sequences for the codebook design for simplicity. Additional advantages for considering Bernoulli

Table 2.1: Example of mapping between the information $\mathbf{w}$ and the sparse vector $\mathbf{s}$ [8]

Input

Size of sparse vector N,

information vector $\mathbf{s}$.

Output

Sparse vector $\mathbf{s}$,

```
α = 0
for i = 2 to N do
 for j = 1 to i − 1 do
  if α = (w)_(10)
   s = (2^i + 2^j)_(2)
  end if
  α = α + 1
 end for
end for
```

Note: $(\mathbf{w})_{(10)}$ is a decimalexpression of binary vector $\mathbf{w}$ and $(\mathbf{w})_{(e)}$ is binary expression of integer $\mathbf{w}$.

sequences is that the modulation can be performed simultaneously with spreading. For example, let's consider QPSK modulation for high reliability in a practical URLLC scenario. For QPSK modulation, we set $k = 2$ for the sparse mapping and put one of the non-zero elements into 1 and the other into $1j$. From (2.2), we can easily see that the elements of the transmit vector $\mathbf{x}$ are mapped to the QPSK symbol (i.e., $x_i \in \{1 + 1j, 1 - 1j, -1 + 1j, -1 - 1j\}$).

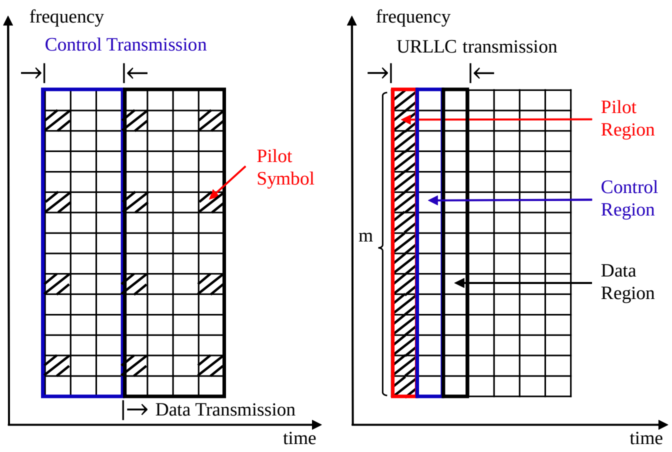

Figure 2.1: packet structure of 4G (left) and the URLLC packet (right)

2.2 SVC decoding

For the short packet transmission, it is natural to consider resource size m smaller than information size N. As a result, the overall system can be modeled as an underdetermined sparse system. Therefore, the compressed sensing algorithm, a popular scheme in sparse vector recovery problem in the underdetermined system, can be utilized for the decoding process. After transmitting SVC encoded vector, the received signal $\mathbf{y}$ is given by

$$\begin{aligned}\mathbf{y} &= \mathbf{HCs} + \mathbf{v} \\ &= \begin{bmatrix} \bar{h}_{11} & & \\ & \ddots & \\ & & \bar{h}_{mm} \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{c}_1 & \cdots & \mathbf{c}_N \\ | & & | \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}.\end{aligned} \tag{2.3}$$

The benefit of SVC is that the decoding process of the information vector is done by the identification of non-zero positions. This also implies that the transmission power or information is concentrated on the non-zero elements of an information vector. Thus, effective power per symbol is much higher compared to the conventional system in which the transmission power is uniformly distributed across entire symbols.

For convenience, (2.3) can be expressed as

$$\begin{aligned}\mathbf{y} &= \mathbf{HCs} + \mathbf{v} \\ &= \mathbf{As} + \mathbf{v}\end{aligned} \tag{2.4}$$

where $\mathbf{A} \in \mathbb{C}^{m \times N}$ is the sensing matrix of the underdetermined system. The corresponding SVC decoding process can be formulated as the support identification problem as

$$\hat{\Gamma} = \arg \min_{|\Gamma|=k} \frac{1}{2} \parallel \mathbf{y} - \mathbf{A}_\Gamma \mathbf{s}_\Gamma \parallel_2^2, \tag{2.5}$$

where $\hat{\Gamma}$ is the set of estimated support. For given $\mathbf{A}$ and k, any greedy sparse recovery algorithms can be used.

2.3 Basic Compressed Sensing

In this section, we briefly overview the basic of compressed sensing. Compressed sensing (CS) technique attract much attention as the importance of restoring the original signal with only a small number of observation are increasingly emphasized [9]. These problems have been studied and applied to various applications such as wireless mobile communication [10], image processing, machine learning, and radar signal.

The basic principle of compressed sensing is to restore the original signal to a small number of measurement values when the original signal is a sparse signal or when it is possible to convert it into a sparse signal at a specific basis. We measure the sparsity by counting the non-zero elements in the signal. For example, we say sparsity k of $\mathbf{s} = [3\ 0\ 0\ 1\ 0\ 0\ 0]$ is 2. Also, we call the set of non-zero element position as support Γ ($\Gamma_{\mathbf{s}} = \{1, 4\}$)

We begin with a linear system having m equations and N unknowns given by

$$\mathbf{y} = \mathbf{A}\mathbf{s}, \tag{2.6}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurement vector, $\mathbf{s} \in \mathbb{R}^N$ is the desired signal vector to be recovered, and $\mathbf{A} \in \mathbb{R}^{m \times N}$. In the overdetermined system $(m \geq N)$, least squares (LS) is the well known solution that is closest to the original signal:

$$\text{LS} \ : \ \mathbf{s}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \tag{2.7}$$

While finding the solution in an overdetermined scenario is straightforward and fairly accurate, in the underdetermined system $(m \leq N)$, the problem is challenging since one cannot find out the unique solution in general. However, the fact that the original signal is sparse gives us a lot of hints. For example, if $\Gamma_{\mathbf{s}} = \{1, 4\}$, (2.6) can be expressed as

$$\mathbf{y} = s_1\mathbf{a}_1 + s_4\mathbf{a}_4, , \tag{2.8}$$

where $\mathbf{a}_i$ is the i-th column vector of $\mathbf{A}$. Thus, the problem turns to overdetermined system and the recovery of $\mathbf{s}$ is straightforward.

Table 2.2: The OMP Algorithm

Input	$\mathbf{A}$, $\mathbf{y}$, and sparsity level k.
Initialize	iteration counter $t = 0$,
	estimated support $\Gamma^0 = \emptyset$,
	and residual vector $\mathbf{r}^0 = \mathbf{y}$.
While	$t < K$ **do**
	$t = t + 1$;
	(Identify) $\gamma^t = \arg\max_{i \in \{1, \cdots, N\}} \lvert\langle \mathbf{r^{t-1}}, \mathbf{a}_i \rangle\rvert$
	(Augment) $\mathbf{\Gamma}^t = \Gamma^{t-1} \cup \{\gamma^t\}$;
	(Estimate) $\hat{\mathbf{s}}^t = \arg\min_{\mathbf{u}:\text{supp}(\mathbf{u}) = \mathbf{\Gamma}^t} \|\mathbf{y} - \mathbf{A}\mathbf{u}\|_2$;
	(Update) $\mathbf{r}^t = \mathbf{y} - \mathbf{A}\hat{\mathbf{s}}^t$;
End	
Output	$\mathbf{\Gamma}$ and $\hat{\mathrm{s}}$.

While there are numerous techniques to solve the underdetermined sparse system, the greedy algorithm is widely used for its simplicity and straightforwardness. By the greedy algorithm, we mean an algorithm to make a local optimal selection at each time with a hope to find the global optimum solution in the end. The most popular and widely used algorithm is perhaps the orthogonal matching pursuit (OMP) [15]. The principle of the OMP is that we hope to find the one most probable support at every iteration. Thus, the iteration time is only k. At t-th iteration, we check the correlation between the residual $\mathbf{r}^t$ ($\mathbf{r}^0 = \mathbf{y}$) and $\mathbf{A}$ and pick the support that corresponds to the largest correlation. After that, we estimate the signal $\hat{\mathbf{s}}^t$ using supports we have found so far. Then, we update the residual by removing the contribution of $\hat{\mathbf{s}}^t$. A more specific process is summarized in Table. 2.2.

One potential problem of the conventional greedy algorithms is that the incorrect index (an index not in the true support) would be selected when the internal correlation of $\mathbf{A}$ is high. To be specific, if two column vectors of $\mathbf{A}$ are highly correlated, inner products between the residual $\mathbf{r}$ and column vectors are similar. In the worst case, for example in a noisy scenario, distinguishing between two columns is challenging. One of the frameworks that quantify the coherence structure of the sensing matrix is the restricted isometry property (RIP) constant [14]. Given the sensing matrix $\mathbf{A}$, the RIP constant $\delta_k[\mathbf{A}] \in [0, 1)$ is the smallest value that satisfies

$$(1 - \delta_k[\mathbf{A}])||\mathbf{x}||_2^2 \leq ||\mathbf{A}\mathbf{x}||_2^2 \leq (1 + \delta_k[\mathbf{A}])||\mathbf{x}||_2^2, \tag{2.9}$$

for all $\mathbf{x} \in \{\mathbf{x} : ||\mathbf{x}||_0 \leq k\}$. $\delta_k[\mathbf{A}]$ indicates how well the system preserves the energy of the original sparse signal. If $\delta_k[\mathbf{A}] \approx 0$, the system matrix is close to orthonormal. Clearly, the smaller the value of the RIP constant $\delta_k[\mathbf{A}]$, the closer any sub-matrix of $\mathbf{A}$ with k columns is being orthogonal. Therefore, the sensing matrix with smaller $\delta_k[\mathbf{A}]$ achieves the better sparse vector estimation.

Chapter 3

Sparse Vector Decoding via DNN

As mentioned, the main task of the SVC decoder is to identify the support of the sparse vector. Considering the decoding process from another point of view, we can reformulate the problem as a multi-label classification that classifies k classes out of N total. Recently, deep learning has achieved tremendous success in classification problems. In this work, we propose the deep neural network architecture, henceforth referred to as deep-SVD, which performs the sparse vector recovery by classifying supports. For a supervised learning, we set the input data as received signal and the non-zero positions of the sparse vector as labels to be classified. The overall process of the deep-SVD-based short packet transmission is depicted in Fig. 3.1. In this section, we briefly review the basic deep learning and explains the detailed architecture of deep-SVD.

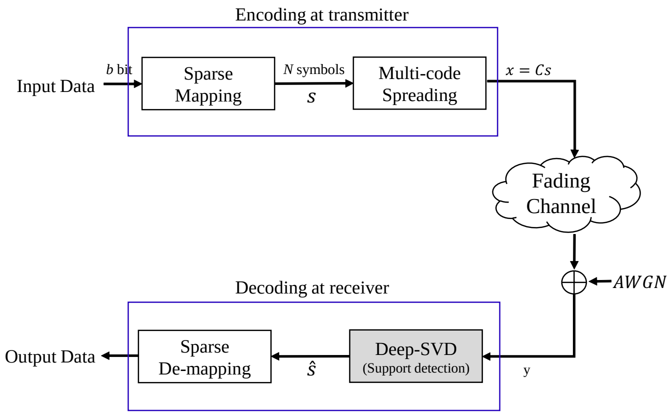

Figure 3.1: packet structure of 4G (left) and the URLLC packet (right)

3.1 Basic Deep Learning

DNN is a neural network that stacks multiple hidden layers deeply to perform the specific functional operation for example classification, clustering, regression, and decision. Each hidden layer consists of multiple neurons.

Fully-connected (FC) layer, a basic unit of the DNN, is given by [16]

$$\mathbf{r}_l = \sigma_l(\mathbf{W}_l \mathbf{r_{l-1}} + \mathbf{b}_l) \tag{3.1}$$

where (l-1)-th layer $\mathbf{r}_{l-1}$ is mapped to l-th layer $\mathbf{r}_l$ with training parameters $\mathbf{W}_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\mathbf{b}_l \in \mathbb{R}^{N_l}$ which are weight and bias, respectively. σ is an activation function to introduce a non-linearity to the layer. Some commonly used activation functions are sigmoid and rectified linear unit (ReLU) function given by [17]

$$\text{sigmoid} \quad : \quad \sigma(r_i) = \frac{1}{1+e^{-r_i}} \tag{3.2}$$

$$\text{ReLU} \quad : \quad \sigma(r_i) = \max(0, r_i). \tag{3.3}$$

Training is a process to obtain the optimal parameters $\boldsymbol{\Theta} = \{\mathbf{W}, \mathbf{b}\}$ from the training data set by minimizing the loss function. Commonly used loss functions include the mean-squared error (MSE) and cross-entropy (CE). Given input as $\mathbf{y}$ and output as $\mathbf{x}$, MSE and CE are expressed as

$$\text{MSE} \quad : \quad L(\mathbf{x}, \mathbf{y}, \boldsymbol{\Theta}) = \| \mathbf{x} - \zeta(\mathbf{y}, \boldsymbol{\Theta}) \|_2^2 \tag{3.4}$$

$$\text{CE} \quad : \quad L(\mathbf{x}, \mathbf{y}, \boldsymbol{\Theta}) = -\sum_{k=1}^{K} x_k \log \zeta(\mathbf{y}, \boldsymbol{\Theta})_k. \tag{3.5}$$

where ζ is the entire process of DNN as a function and K is the number of class in classification problem. Also, parameters are updated by the stochastic gradient descent (SGD) and backpropagation.

Batch normalization is one of the most powerful methods or layer that alleviate the difficulty of training very deep models [18]. Normalization is a well known preprocessing technique in data pre-processing which helps network training convergence

Table 3.1: Batch Normalization

Input	Values of x over a mini-batch: $\mathcal{B} = \{x_{1,...,m}\}$.
	Parameters to be trained: $\gamma,\ \beta$.
Output	y_i.
	$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$
	$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$
	$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$
	$y_i = \gamma\hat{x}_i + \beta$

faster. Taking advantage of these benefits, batch normalization applies a feature-wise normalization at each layer, especially before activation function, thus, it reduces the internal covariate shift problem.:

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathbf{E}[x^{(k)}]}{\sqrt{\mathbf{Var}[x^{(k)}]}} \tag{3.6}$$

where k denotes the feature index. Also, in order to address the representation capacity decrease problem by simple normalization, authors in [18] introduce a pair of trainable parameters γ and β, which scales and shift the normalized value:

$$y^{(k)} = \gamma\hat{x}^{(k)} + \beta \tag{3.7}$$

Table. 3.1 summarizes the procedure for batch normalization.

Resnet is a network based on the residual learning framework which showed a tremendous improvement in classification performance in the ImageNet Challenge [13]. The main idea of residual learning framework is to put the direct identity connection between few stacked hidden layers. Instead of hoping every few stacked layers directly fit the desired function, the authors explicitly let these layers fit a residual mapping function, which is assumed to be easier for optimization (See Fig. 3.2). To be specific, denoting the input as $\mathbf{y}$ and the underlying desired function as $\mathcal{H}(\mathbf{y})$, the residual mapping is defined as $\mathcal{F}(\mathbf{y}) = \mathcal{H}(\mathbf{y}) - \mathbf{y}$, and a residual block structure is thus:

$$\mathcal{H}(\mathbf{y}) = h(\mathbf{y}) + \mathcal{F}(\mathbf{y}, \boldsymbol{\Theta}), \tag{3.8}$$

where $h(\mathbf{y})$ is an identity mapping: $h(\mathbf{y}) = \mathbf{y}$. Due to this identity mapping, it is easy to optimize the residual mapping function $\mathcal{F}(\mathbf{y})$ than to optimize the original underlying desired function $\mathcal{H}(\mathbf{y})$. Therefore, it is shown in [resnet] that identity mapping connection eases the training of very deep models.

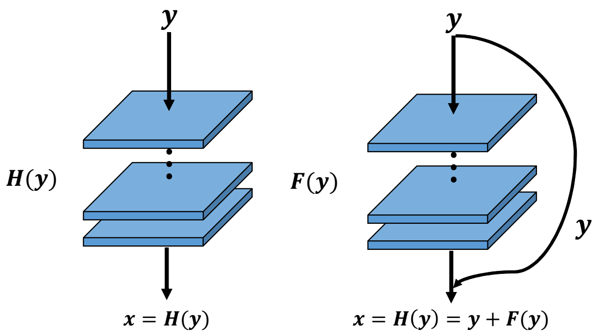

Figure 3.2: Residual learning framework: Direct fitting of desired function (left) and fitting of residual mapping function (right)

3.2 Deep Sparse Vector Decoding (Deep-SVD)

Our goal is to train the deep-SVD as a powerful function approximator to estimate the sparse vector. In this regard, the support identification problem in (2.5) can be expressed as

$$\tilde{\Gamma} = \zeta(\mathbf{y}, \mathbf{A}, \mathbf{\Theta}), \tag{3.9}$$

where $\mathbf{\Theta}$ is the desired optimal parameter to be trained for support identification process. As a basic unit, we use a residual block (RB) to facilitate the residual learning framework to ease the training of deep networks and stack them to construct the overall networks.

Residual Block

In the RB, we put two FC layers with batch normalization and ReLu for activation function. Before operation of RB, we concatenate the input of the current RB with the sensing matrix. For the residual learning framework, before passing the activation function of the last layer, we add the output of the previous RB (also the input of the current RB). With the help of the identity mapping connection for the deeper network, another approach to make the network to have the lager representation capacity is to increase the network width (or dimension) [19]. We simply increase the network width by $10N$ in each layer of RB to increase the layer-wise representation capacity. The detailed structure of RB is depicted in Fig. 3.3.

One interesting point to find is that due to the residual learning framework in RB, the entire structure is similar to that of unfolding the conventional iterative sparse recovery algorithm. To be specific, one single RB operation in the deep-SVD is similar to a single iteration in OMP. In OMP as described in Table 2.2, the residual calculated in the previous iteration is used in the next iteration for the support identification. In deep-SVD, let $\mathbf{y}$ be the input of the current RB and $\mathcal{H}(\mathbf{y}, \mathbf{A})$ be the underlying desired function. From (3.8), our underlying desired function $\mathcal{H}(\mathbf{y}, \mathbf{A})$ is given by

$$\begin{aligned}\mathcal{H}(\mathbf{y}, \mathbf{A}) &= h(\mathbf{y}) + \mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}) \\ &= h(\mathbf{y}) - \mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}') \\ &= \mathbf{y} - \mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}'), \end{aligned} \tag{3.10}$$

where $\mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}') = -\mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}) = \mathbf{y} - \mathcal{H}(\mathbf{y}, \mathbf{A})$ is the residual mapping using learning parameters $\boldsymbol{\Theta}'$. This implies that the underlying desired function $\mathcal{H}(\mathbf{y}, \mathbf{A})$ is the residual of RB and also the input of the next RB.

The difference between OMP and the proposed deep-SVD comes from the learnable parameters. As mentioned in 3.1, the fundamental weakness of the greedy-based algorithms like OMP is that their performance is affected by the inner correlation of the sensing matrix. Thus, using the sensing matrix with the bad property for every iteration is likely to pick the wrong support. This motivates us to design a network structure that can alleviate the bad property of the given sensing matrix by taking learnable parameters. From (3.10), we fit the residual mapping $\mathcal{F}(\mathbf{y}, \mathbf{A}, \boldsymbol{\Theta}')$ which is parameterized by $\boldsymbol{\Theta}'$ at each RB. By taking $\boldsymbol{\Theta}'$, deep-SVD gives more flexibility than conventional greedy algorithms that use a fixed sensing matrix at every iteration. Let $\mathbf{A}' = \mathcal{Q}(\mathbf{A}, \boldsymbol{\Theta}')$ be a abstract concept of learned sensing matrix obtained from learning process of RB. We call it abstract because the RB is composed of multiple layers and non-linear activation functions, making it impossible to represent a single matrix. Using the RIP constnat, if we can find the appropriate parameters $\boldsymbol{\Theta}'$ that satisfies

$$\delta_k[\mathbf{A}'] \leq \delta_k[\mathbf{A}], \tag{3.11}$$

we can estimate the sparse vector more accurately than the conventional. From the empirical results in 4, we can observe that the proposed deep-SVD outperforms the conventional greedy algorithms which imply that the learned parameters actually help to alleviate the inner coherency of the sensing matrix. Thus, in most cases, the learned sensing matrix $\mathbf{A}'$ satisfies (3.11).

To construct the overall network, we stack multiple RBs and put an another FC layer at last which outputs N-dimentional vector. For multi-class and multi-label classification, the softmax layer generates N values representing the probability of being the support element,

$$\text{softmax} \ : \ \sigma(\mathbf{r})_n \ = \frac{e^{r_n}}{\sum_{n=1}^{N} e^{r_n}} \tag{3.12}$$

For support identification at the actual packet decoding process, we take positions of k largest probable elements among N values. Fig. 3.3 depicts the entire structure of deep-SVD.

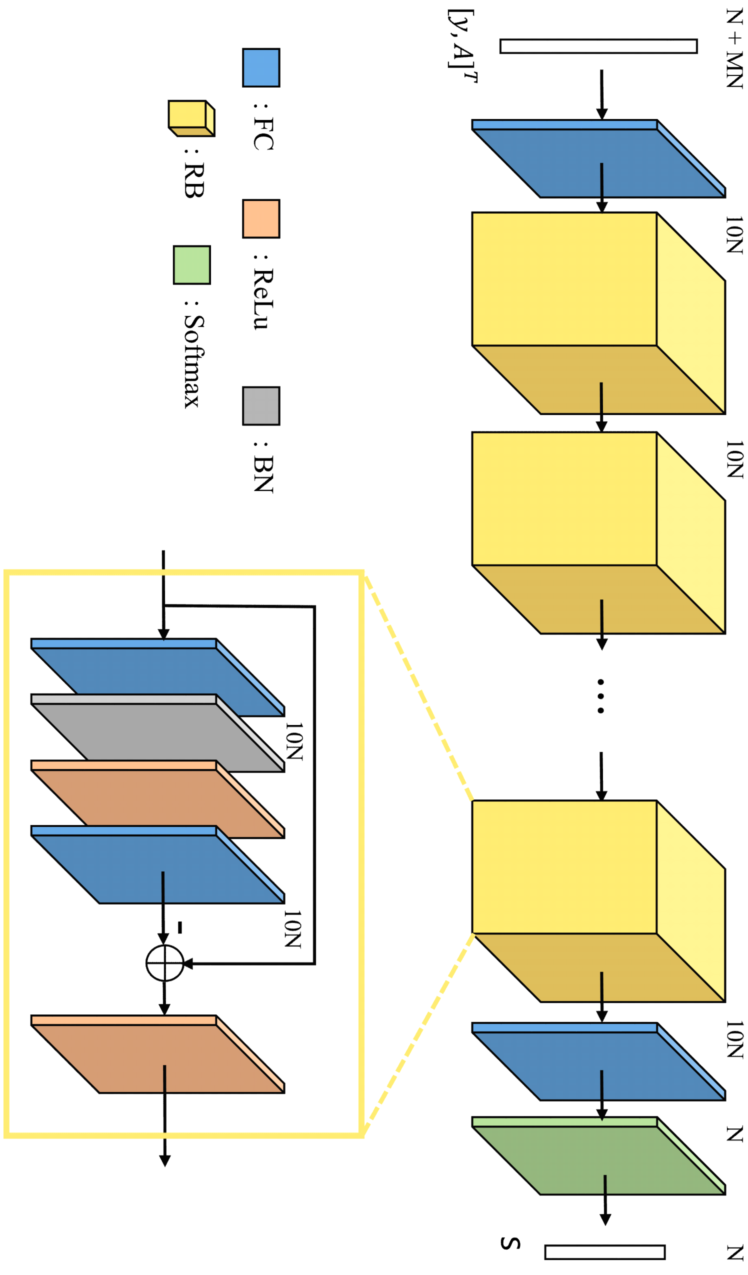

Figure 3.3: Network structure of deep-SVD for short packet decoding

Chapter 4

SIMULATION

4.1 Dataset and Simulation Setup

In this section, we examine the performance of the proposed deep-SVD scheme. Our simulation is based on the single user orthogonal frequency division multiplexing (OFDM) systems. As a channel model, we use AWGN and the fading channel with l channel taps. For comparison, we used conventional SVC with OMP for the decoding process. We also investigate the performance of the PDCCH of LTE-Advanced system, and AWGN lower bound [8]. In the simulation, we generate a single random binary spreading codebook that has specific mutual coherence and fixed it for both training and testing. Mutual coherence $\mu[\mathbf{A}]$, the largest off-diagonal elements in the Gram matrix $\mathbf{A}^T\mathbf{A}$, is an another popularly used measurements that quantify the coherence properties of the sensing matrix[10]. Similar to the RIP constant, the smaller the $\mu[\mathbf{A}]$, the smaller the coherency. For modulation, we use a QPSK signal, thus, the number of supports k is 2. We apply $L(=8)$ times repetition which is commonly used in URLLC for guaranteed reliability. We use BLER of the codeblocks as a performance measure.

For the dataset, we generate numerous sparse pattern based on (2.4). We used $\{\mathbf{y}, \mathbf{A}\}$ for input data and $\mathbf{s}$ for label. To process the sensing matrix $\mathbf{A}$, we simply reshape it to a single vector. We set the learning rate to 0.0001, the size of a batch to 10000, and stack 6 RBs. The simulation environment is implemented using the tensorflow, a popular deep learning library [20]. In order to process the complex value $c \in \mathbb{C}^N$, we merge the real part Re$\{c\} \in \mathbb{R}^N$ and imaginary part Im$\{c\} \in \mathbb{R}^N$ into $\begin{pmatrix} \mathbb{R}^N \\ \mathbb{I}^N \end{pmatrix} \in \mathbb{R}^{2N}$.

4.2 Simulation result

In Fig. 4.1, we investigate the BLER performance of the proposed deep-SVD methods and competing schemes under AWGN channel condition. We set m to 42, N to 96, $\mu[\mathbf{C}]$ to 0.619, and L to 8. We observe that the proposed deep-SVD technique outperforms the conventional SVD decoding and the PDCCH schemes, achieving 1dB gain over the conventional SVD decoding and about 4dB gain over the conventional PDCCH at 10^{-5} BLER point. From this results, the learned parameters actually help to alleviate the inner coherency of the given sensing matrix.

The training issue in the proposed deep-SVD is that in the fading channel model the sensing matrix $\mathbf{A} = \mathbf{HC}$ changes due to channel statistics. When the randomness of the received signal is high, learning parameters might not converge and thus the test performance of the learned network would be poor. To reduce the degree of randomness, we assume the number of channel tap is small. This is reasonable assumption for the short packet transmission [21]. In this paper, we set the number of channel tap to 2. In Fig. 4.2, we investigate the BLER performance under the fading channel condition. We set N to 40, m to 20, $\mu[\mathbf{C}]$ to 0.73, and L to 8. Even in the fading channel scenario, we observe that the proposed deep-SVD technique achieves about 4dB gain over the conventional SVC decoding at 10^{-5} BLER point.

In Fig. 4.3, we evaluate the BLER performance of the SVC decoding and the deep-SVD as a function of SNR for various measurement size $(m = 30,$ and $42)$. We can observe that as the measurement size decreases, the degree of performance degradation becomes smaller in the proposed deep-SVD at 10^{-5} BLER point. This demonstrates that the proposed deep-SVD method is more robust in terms of resource limitation.

In Fig. 4.4, we evaluate the BLER performance for more smaller data information with $m = 20$ and $N = 40$, 50. We observe that the proposed deep-SVD technique outperforms the conventional SVD decoding achieving more than 8dB gain over the conventional SVD decoding at 10^{-5} BLER point.

One another advantage of using a learning-based scheme over a linearly operating conventional scheme is that we can utilize the ensemble technique. The ensemble technique uses multiple learning models to obtain better predictive performance than could be obtained from any of the constituent learning models alone [16]. In this simulation, we separately trained three different deep-SVD for the ensemble. In Fig. 4.5, we can observe that through the ensemble technique we could achieve 3dB gain over single deep-SVD at 10^{-5} BLER point.

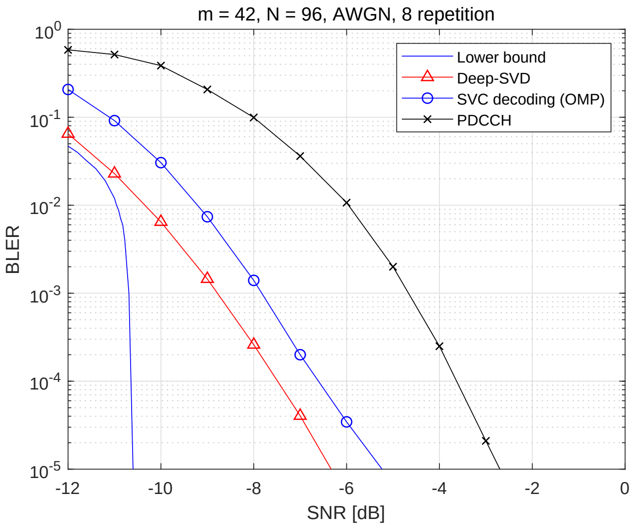

Figure 4.1: BLER performance as a function of SNR for AWGN channel

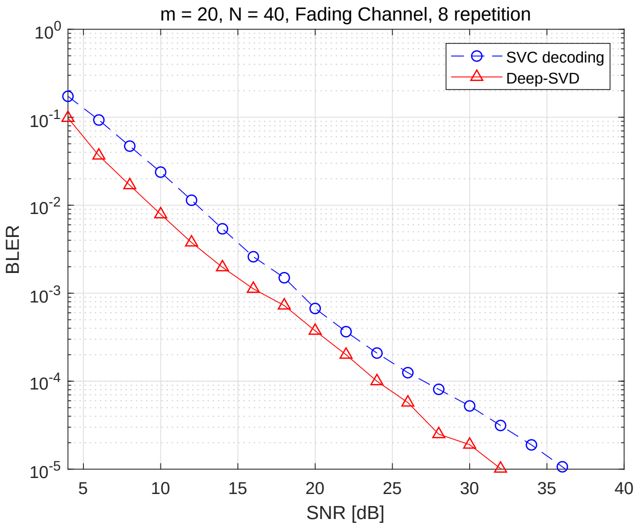

Figure 4.2: BLER performance as a function of SNR for the fading channel

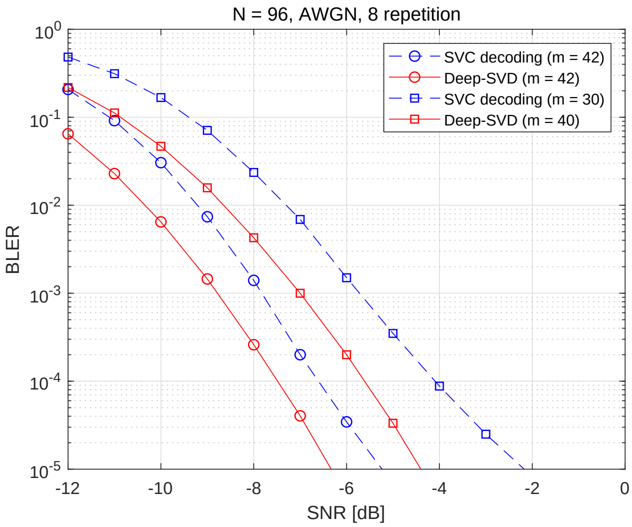

Figure 4.3: BLER performance for different measurement size for AWGN channel

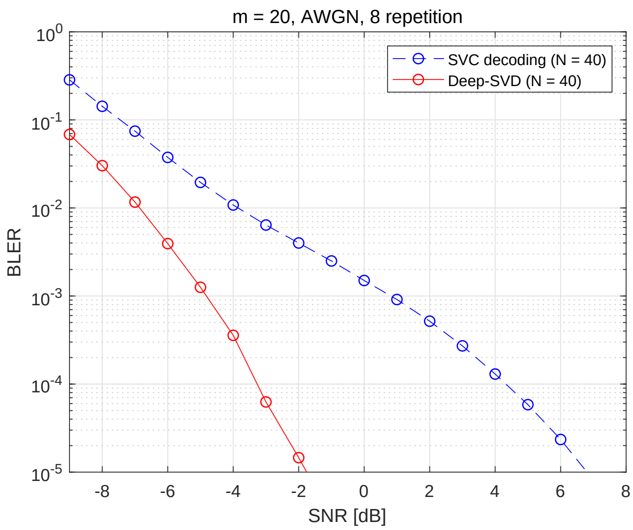

Figure 4.4: BLER performance as a function of SNR for AWGN channel

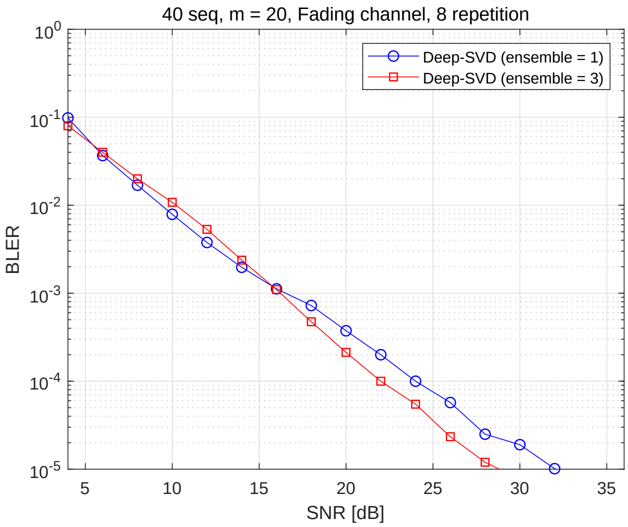

Figure 4.5: BLER performance with ensemble technique for the fading channel

Chapter 5

CONCLUSION

In this paper, we have proposed the DNN-based decoding schemes for URLLC. The key idea behind the proposed deep-SVD is to reformulate the SVC decoding process as a multi-label classification where the non-zero positions of the sparse vector are labels to be classified from received SVC encoded vector. To perform supervised learning, we construct the dataset which is composed of the received signal as the input and the SVC encoded sparse vector as the output. We built the entire model using residual mapping framework for stable training of deep networks. We demonstrated from the numerical evaluations that the proposed deep-SVD scheme is very efficient in the URLLC scenario in terms of high reliability. In this paper, we restricted our attention to the wireless communication scenario but we expect that there are many other applications that can be applied where the principle of CS is used.

Bibliography

[1] ITU-R Rec. M.2083-0, "IMT vision - Framework and overall objectives of the future development of IMT for 2020 and beyond," Sept. 2015.

[2] 3GPP Technical Report 38.913, "Study on scenarios and requirements for next generation access technologies (Release 14)," v14.2.0, 2017.

[3] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra reliable and low latency communications in 5G: Physical layer aspects," *IEEE Wireless Commun.*, vol. 26, no.2, pp.100-107, June, 2018.

[4] 3GPP Technical Report 38.913, "Study on scenarios and requirements for next generation access technologies (Release 14)," v14.2.0, 2017.

[5] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59-65, Sep 2016.

[6] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*, John Wiley & Sons., 2012.

[7] 3GPP Technical Report 38.802, "Study on new radio access technology physical layer aspects (Release 14)," v14.1.0, 2017.

[8] H. Ji, S. Park, and B. Shim, “Sparse vector coding for ultra reliable and low latency communications,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6693-6706, Oct 2018.

[9] D. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory,* vol. 52, no. 4, pp. 1289-1306, April 2006

[10] 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, Feb 2017.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] K.Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal nets,” *Neural networks*, vol. 2, no. 5, pp.359- 366, 1989.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Computer Vision and Pattern Recognition (CVPR), 2016.

[14] E. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Comm. Pure Appl. Math.,* vol. 59, no. 8, pp. 1207-1223, Aug. 2006.

[15] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* Cambridge, MA: MIT Press, 2016.

[17] 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.

[18] S. Loffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 [cs], Mar 2015.

[19] S. Zagoruyko, N. Komodakis, "Wide residual networks" in , Nov. 2016, [online] Available: https://arxiv.org/abs/1605.07146.

[20] M. abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[21] D. Tse and P. Viswanath, *Fundamentals of wireless communication,* Cambridge University Press, 2005.

초 록

URLLC (Ultra Reliability and Low Latency Communication)는 미래의 초연결 산업 분야에서 주목받는 5G 서비스 카테고리 중 하나이다. 3GPP (Third Generation Partnership Project)는 URLLC을 실현하기 위한 요구조건으로 1ms 전송 시간 내에 99.999%의 정확도로 패킷을 전송해야 한다는 다소 까다로운 기준을 설정하였다. 현재의 4G 무선통신 시스템에서는 복잡하고 긴 코드 블록을 전송함으로써 코딩 이득을 최대화하도록 설계하여 정확성을 높이지만 그로 인하여 지연 시간이 길어진다는 단점이 있다. 최근 URLLC를 대상으로 짧은 패킷 전송을 위한 SVC (Sparse Vector Coding) 기법이 제안되었다. SVC에서의 인코딩은 단순한 희소 신호 매핑 및 확산을 하여 패킷을 전송하며 디코딩은 간단한 희소 벡터 복원 알고리즘으로 대체한다. 이 논문에서는 URLLC의 높은 요구조건을 만족하기 위해서 깊은 신경망 기반의 deep sparse vector decoding (Deep-SVC) 기법을 제안한다. 이를 위해서, 우리는 SVC의 디코딩 과정을 다중 레이블 분류 (multi-label classification)으로 재구성한다. 그리고 깊은 신경망을 구성하여 코드북 내의 높은 상관관계를 학습하여 SVC 디코딩 과정의 성능을 끌어올린다. 실험을 통하여 우리는 제안하는 Deep-SVD 기법이 기존의 SVC 디코딩 기법보다 더 좋은 정확성을 갖음을 보인다.

주요어: 5G, URLLC, Short packet transmission, SVC, Deep neural network, Compressed sensing

학번: 2017-22314

ACKNOWLEGEMENT

2년이라는 짧다면 짧고 길다면 길다 할 수 있는 석사과정을 마치게 되었습니다. 그 과정에서 많은 사람들에게 도움을 얻어 이렇게 감사의 글을 남깁니다.

먼저 저의 지도교수이신 심병효 교수님께 감사 드립니다. 부족한 저를 받아주시고 2년간 세심한 지도와 끝없는 지원으로 제가 이렇게 무사히 학위과정을 마치게 되었습니다. 연구 이외에도 항상 좋은 말씀으로 격려해주시고 사회를 살아가는 방법에 대한 주옥 같은 말씀을 많이 해주셔서 앞으로의 대학원 이후의 인생에서 정말 큰 도움이 될 것 같습니다. 다시 한번 감사 드립니다.

2년간 연구실에서 많은 사람들을 만났습니다. 제 2의 지도교수님처럼 많은 것을 알려주신 형주형, 연구실의 엄마 같은 존재인 선호형, 301동 주민 상태형, 가장 부러운 원준이형, 유일한 동갑내기 술친구 구영이, 세상에서 제일 착한 진홍이, 동기이지만 연구에 있어서는 선배처럼 알려주고 같이 고민해주는 우수 연구자 준한이, 몸도 마음도 듬직한 현규, 갓승년이, 범접할 수 없는 5차원 매력 광진이, 그냥 이상한 용준이, 알쓸신잡 준원이, 뉴욕커 선우, 힘내라 루옹, 훠궈 좋아하는 탕 박사까지 좋은 인연으로 2년간 재미있게 잘 지내서 행복했고 감사합니다.

마지막으로 항상 저를 응원해주시고 울타리처럼 지켜주는 사랑하는 가족들에게 감사합니다. 어떤 일이든 제 편에 서서 함께 해주었기에 2년간 든든하게 생활 할 수 있었습니다. 그리고 사랑하는 여자친구 이수현에게도 감사합니다.