Deep learning-based massive multiple-input multiple-output channel state information feedback with data normalisation using clipping

Sanguk Jo, Jaehee Lee, and Jaewoo So

Department of Electronic Engineering, Sogang University, Seoul, 04107, Republic of Korea

E-mail: jwso@sogang.ac.kr

Massive multiple-input multiple-output (MIMO) can provide real-time high-capacity data transmission service to the user equipment (UE). However, the use of massive MIMO exponentially increases the channel state information (CSI) feedback overhead. Deep learning-based approaches have been proposed to reduce CSI feedback overhead with significant CSI accuracy. Data normalisation is a very important part of deep learning, because the performance of the deep learning highly depends on the data normalisation. In this letter, an efficient data normalisation method for deep learning-based CSI feedback in a massive MIMO system is proposed, where the proposed method uses a clipping technique based on the received signal strength. Simulation results show that the proposed normalisation with the clipping decreases the CSI feedback error and increases the accuracy of the beamforming vector in deep learning-based CSI feedback.

Introduction: Massive multiple-input multiple-output (MIMO) is a key technique in future wireless communications. Wireless communications can take the advantage of excellent spectral efficiency and superior energy efficiency by using massive MIMO, which uses large antenna arrays at the base station (BS) [1]. To maximize data rate in the massive MIMO system, the BS adapts to a transmission with CSI reported by each user equipment (UE). However, large antenna arrays exponentially increase the feedback overhead; there have been techniques such as codebook-based quantization [2] and spatial correlation adjustment of CSI [3] to reduce the feedback overhead. Moreover, recent techniques based on deep learning, such as a convolutional neural network (CNN) and long short-term memory models (LSTM), showed both the reduction of feedback overhead and the high accuracy of CSI feedback [4, 5]. The authors of [6] proposed a neural network architecture including CNN, LSTM, and attention mechanism in order to achieve a compromise between performance and complexity, where they utilized a single-stage network to reduce the number of training parameters. The authors of [7] designed a separate deep learning-based denoising network to overcome the influence of various interference and non-linear effects in practical feedback channels. The authors of [8] proposed a multiple-rate compressive sensing neural network to compress and quantise the CSI. Here, they improved the original CsiNet of [4]; and developed a new quantization framework and training strategy considering that the feedback is in the form of the bitstream in practical feedback systems. Most previous studies for CSI feedback using deep learning focused on the deep learning architectures. However, data preprocessing such as normalisation is also a very important factor in deep learning. In this letter, we propose a normalisation method with clipping in CSI feedback using CNN-based deep learning.

System model: We consider a massive multiple-input multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) system with N subcarriers. The base station (BS) is equipped with M antennas and the user equipment (UE) is equipped with a single antenna. The downlink received signal on the rth subcarrier at UE can be expressed as:

$$y_i = h_i^H x_i + n_i,$$  \hspace{1cm} (1)

where $x_i \in \mathbb{C}^{d_t \times 1}$ is the transmit signal, $h_i \in \mathbb{C}^{d_r \times 1}$ is the frequency domain channel vector, $n_i \in \mathbb{C}^{d_t \times 1}$ is the additive white Gaussian noise (AWGN), thus the channel matrix in the spatial-frequency domain vector can be represented as $H = [h_1 \ldots h_N] \in \mathbb{C}^{N \times M}$. H can be sparsified in angular-delay domain using a 2D discrete Fourier transform (DFT) as:

$$\tilde{H} = W_c H W_d^H,$$  \hspace{1cm} (2)

where $W_c$ and $W_d$ are $N \times N$ and $M \times M$ DFT matrices, respectively. To reduce feedback overhead, we retain only the first $N_r$ rows of $\tilde{H}$ due to limited multipath time delay in the delay domain [4]. Finally, the number of feedback parameters is reduced from $2NM$ to $2N_r M$ ($\tilde{H}$ are separated to the real and imaginary parts, used as inputs), and the truncated matrix becomes $\hat{H}$.

After generating $\hat{H}$, normalisation is performed before training the CNN. The normalisation proceeds in two stages: First, the clipping $N_c(\cdot)$ is applied and then the scaling $N_s(\cdot)$ is applied sequentially. The normalisation channel matrix, $\tilde{H}$, is given by:

$$\tilde{H} = N_c (N_s (\hat{H})).$$  \hspace{1cm} (3)

The encoder of CNN is trained to make the compressed representation $L$ (codewords) from the CSI data, trained encoder $T_{enc}(\cdot)$ can be expressed as:

$$L = T_{enc} (\tilde{H}),$$  \hspace{1cm} (4)

and decoder $T_{dec}(\cdot)$ is trained to reconstruct the original CSI information as follows:

$$\hat{H} = T_{dec} (L).$$  \hspace{1cm} (5)

The overall CSI feedback architecture including data normalisation is shown in Figure 1.

Proposed normalisation with clipping: Data normalisation before training the network is crucial to improve the accuracy and reduce the time complexity [9]. Another advantage of normalisation speeds up the gradient descent convergence [10]. In many machine learning algorithms, feature scaling is needed so that some numbers (outliers) do not impact the model just because of their large magnitude. In particular, the feature scaling is essential for deep learning that calculates distance between data. If not scaled, the feature with a high value starts dominating when calculating distances. Moreover, the outliers can skew the distribution artificially and gradient descent methods can have trouble due to outliers [10].

Geometry-based stochastic channel models (GSCMs) are widely used to generate the massive MIMO channel matrix. The channel elements of GSCMs are generated by summing the rays with delay, power, and angle. Therefore, some channel elements can have very high power. These elements have a great influence on the statistical operation of deep learning. One approach to fix outliers for linear models is to keep va-
Algorithm 1 Scaling with clipping threshold

1: Input:
Angular-delay domain channel matrix \( \hat{H} = [\hat{h}_{1,1}, \ldots, \hat{h}_{1,n}] \)
Determine the value of \( \xi \) specifying the clipping threshold.
2: Output:
Normalized channel matrix \( \tilde{H} = [\tilde{h}_{1,1}, \ldots, \tilde{h}_{1,n}] \).
3: /* Transform the channel matrix from a complex number to a polar form */
4: \( r_{clipping} = A \) polar form of the channel element.
5: for all channel elements \( h_{i,m} \in \hat{H} \) do
6: Compute the magnitude \( r \) and the phase angle \( \theta \) for \( h_{i,m} \).
7: Update \( h_{i,m} \leftarrow r_{clipping} \).
8: end for
9: \( A \) = Clipping threshold, the ith largest magnitude in \( \hat{H} \); \( f = \) Multiplication factor for scaling for \( f \in \mathbb{C} \).
10: Update \( f = 0.5,f \).
11: /* Clipping and scaling */
12: for all channel elements \( h_{i,m} \in \hat{H} \) do
13: if \( |h_{i,m}| > f \) then
14: Update \( d_{clipping} = |h_{i,m}| \).
15: Update \( h_{i,m} \leftarrow d_{clipping} \).
16: end if
17: /* Transform the channel matrix to a complex number */
18: for all channel elements \( h_{i,m} \in \hat{H} \) do
19: Compute the real part \( x \) and the imaginary part \( y \) for \( h_{i,m} \).
20: Update \( h_{i,m} \leftarrow x + yj \).
21: end for
22: /* Shift the range from \([-0.5, 0.5]\) to \([0, 1]\) */
23: Update \( \tilde{H} \leftarrow \hat{H} + 0.5 \).
24: return \( \tilde{H} \).

Clipping: To reduce the negative impact of very high power elements during the normalisation process, we limit the maximum and minimum values on the basis of a specific clipping threshold. First, we choose a clipping threshold, \( A \), where \( A \) is the magnitude of the channel element at \( \xi \% \) of the entire dataset. Next, the clipping is performed based on the clipping threshold as follows:

\[
\tilde{h}_{i,m} = \begin{cases} 
\hat{h}_{i,m}, & |\hat{h}_{i,m}| \leq A \\
Ae^{\theta}, & |\hat{h}_{i,m}| > A.
\end{cases}
\]  

where \( \tilde{h}_{i,m} \) is the \( i \)-th sample in the channel dataset and it represents the channel element on the \( i \)-th delay and \( m \)-th angle, and \( \theta \) is the phase of the channel element \( h_{i,m} \).

Scaling: Most of the real and imaginary parts of the element are distributed around zero. The output value of CNN has a range of \([0, 1]\). Hence, for the accuracy of the CSI feedback, the input value needs to be scaled in \([0, 1]\). A general min-max scaling method can be applied. However, the min-max scaling is sensitive to high power elements and it also increases the complexity of inverse scaling. Therefore, we first scale the data to \([-0.5, 0.5]\) and then rescale to \([0, 1]\), by using simple multiplication and shifting. We present the scaling procedure with clipping in Algorithm 1. Figure 2 compares the probability density function (PDF) of the scaling without clipping and the scaling with clipping.

We use a CNN-based model called CsiNet for CSI feedback [4]. When \( \omega = \{\omega_{max}, \omega_{dec}\} \) is the set of parameters after CNN training, the reconstructed channel matrix is given by:

\[
\tilde{H} = T_{dec}(\{T_{enc}(\hat{H}; \omega_{max}); \omega_{dec}\}).
\]  

Simulation results: We generate two types of channel matrices at a centre frequency of 3.65 GHz and a bandwidth of 50 MHz. First, we assume the Berlin Urban Macro (UMa) non-line-of-sight (NLOS) scenario in the QuaDRiGa channel model [11], where the number of transmit antennas is \( M = 32 \) and the UE moves along a circle. Second, we assume the semiurban NLOS scenario in the COST 2100 channel model [12], where the number of transmit antennas in \( M = 32 \) and the UE moves in several directions. We retain the first 32 rows of the channel matrix in the angular-delay domain by limiting the time period due to multipath arrivals. Therefore, the size of \( \hat{H} \) is \( 2 \times 32 \times 32 \). The clipping threshold \( \xi \) is set as 3.0\%. The size of channel vector dataset for training, validation, and testing is 60,000, 20,000, and 20,000, respectively. The other parameters are as follows: the learning rate is 0.001, the epoch is 1000, and the batch size is 100. Additionally, Adam optimizer is used in the simulation.

The difference between the reconstructed channel \( \tilde{H} \) and the input \( \hat{H} \) is defined as the normalised MSE (NMSE),

\[
\text{NMSE} = \mathbb{E}\left\{ \| \tilde{H} - \hat{H} \|_2^2 / \| \hat{H} \|_2^2 \right\}.
\]  

The cosine similarity can be used to evaluate the accuracy of the beamforming vector. The cosine similarity is given by,

\[
\rho = \mathbb{E}\left\{ 1 / N \sum_{n=1}^{N} |\tilde{h}_n^H \hat{h}_n| / (\| \tilde{h}_n \|_2 \| \hat{h}_n \|_2) \right\}.
\]  

Figures 3 and 4 show the simulation results. Figure 3 shows the NMSE as the size of codewords increases from 64 to 512. The proposed data normalisation with clipping has a lower NMSE than the data normalisation without clipping for all codewords sizes. The scaling with clipping significantly reduces NMSE, especially at high codewords. If we clip the outliers (actually, we transform the outlier channels to a specific upper bound of the magnitude without losing the phase information), we can focus more on the distribution of the normal elements, and therefore improve the accuracy of the machine learning processing linear models. However, if we regard too many elements as outliers, the original distribution of all elements may be distorted and the performance will no longer improve.

Figure 4 shows the accuracy of \( \rho \) as the size of codewords increases from 64 to 512. The proposed data normalisation using clipping improves the performance of cosine similarity, \( \rho \), in the QuaDRiGa channel model. Notice, because we clip the only magnitude of the channel information, the phase of all the channel information is maintained. Small
Fig. 4 The cosine similarity according to the size of codewords

Fig. 5 Validation loss at codewords 512

codewords result in poor performance regardless of how the data is normalised with or without clipping due to limitations in the CNN structure. In the QuaDRiGa channel model, however, as the size of codewords increases, the proposed data normalisation with clipping significantly improves the cosine similarity. Because the COST 2100 channel model does not fade the clusters in and out over time, the cosine similarity is higher in the COST 2100 channel than in the QuaDRiGa channel. Moreover, the cosine similarity is almost similar regardless of clipping in the COST 2100 channel.

Figure 5 shows the validation loss in the QuaDRiGa channel as the number of epochs increases, when the size of codewords is 512. The data normalisation without clipping shows large fluctuations as the number of epochs increases, which makes it difficult for CNN to learn CSI feedback behaviour. However, the proposed data normalisation with clipping converges quickly without fluctuations. In massive MIMO systems, the channel state may change rapidly and significantly due to the fast fading. Hence, fast convergence is one of important factors in the learning-based CSI feedback. The proposed data normalisation using clipping reaches the desired MSE in a very fast time. Because the complexity of the neural network depends on the network architecture (e.g. number of parameters and number of layers), network pruning has been widely used to reduce the network complexity and over-fitting by reducing the number of parameters or connections [13]. Hence, the data preprocessing may be less relevant to the network complexity. However, the proper data preprocessing can accelerate the training as well as it may reduce the complexity of the neural network because the reduction of training time due to the data preprocessing makes the possibility to compress the network.

Conclusion: The proposed data normalisation using clipping improves the performance of the deep learning-based CSI feedback in massive MIMO channels. The data normalisation with clipping reduces the NMSE and increases the cosine similarity as the size of codewords increases. The proposed data normalisation with clipping significantly reduces the learning time, making it suitable for use in a practical massive MIMO system where the channel environment is dynamically changed. Additionally, the performance of the proposed data normalisation using clipping is robust regardless of the channel model, QuaDRiGa and COST 2100.

In a practical system, the feedback is in the form of the bitstream. Hence, the UE quantises the feedback information, where a non-uniform quantizer can be used in order to reduce the quantization error [8]. The proposed data normalisation using clipping method can simplify the quantization process by reducing the negative influence of high power channel elements in the quantization process, so the proposed normalisation would be useful for digital CSI feedback.

Acknowledgments: This work was partly supported by the Technology development Program of MSS (No. S2950045) and the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2020R1F1A1065109).

Author Contributions: Sanguk Jo: Conceptualization; formal analysis; investigation; methodology; software; validation; visualization; writing—original draft. Jaehee Lee: Resources; validation; visualization; writing-review and editing. Jaewoo So: Conceptualization; funding acquisition; project administration; resources; supervision; writing-review and editing.

© 2021 The Authors. Electronics Letters published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
doi: 10.1049/el2.12080

References
1 Gampala, G., Reddy, C. J.: Massive MIMO—Beyond 4G and a basis for 5G. In: Proceedings of the International Applied Computational Electromagnetics Society Symposium (ACES). Denver, CO, pp. 1–2 (2018)
2 Krishnamachari, R., Varanasi M.: Interference alignment under limited feedback for MIMO interference channels. In: Proceedings of IEEE International Symposium on Information Theory (ISIT). Austin, TX, pp. 619–623 (2010)
3 Rao, X., Ruan, L., Lau, V.K.N.: Limited feedback design for interference alignment on MIMO interference networks with heterogeneous path loss and spatial correlations. IEEE Trans. Signal Process. 61(10), 2598–2607 (2013)
4 Wen, C.-K., Shih, W.-T., Jin, S.: Deep learning for massive MIMO CSI feedback. IEEE Wireless Commun. Lett. 7(5), 748–751 (2018)
5 Wang, T., et al.: Deep learning-based CSI feedback approach for time-varying massive MIMO channels. IEEE Wireless Commun. Lett. 8(2), 416–419 (2018)
6 Zhang, Z., et al.: Massive MIMO CSI reconstruction using CNN-LSTM and attention mechanism. IET Commun. 14(18), 3089–3094 (2020)
7 Ye, H., et al.: Deep learning-based denoise network for CSI feedback in FDD massive MIMO systems. IEEE Commun. Lett. 24(8), 1742–1746 (2020)
8 Guo, J., et al.: Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis. IEEE Trans. Wireless Commun. 19(4), 2827–2840 (2020)
9 Sola, J., Sevilla, J.: Importance of input data normalisation for the application of neural networks to complex industrial problems. IEEE Trans. Neural Netw. Learn. Syst. 44(3), 1464–1468 (2017)
10 Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv, arXiv:1502.03167, pp. 1–11 (2015)
11 Jaeckel, S., et al.: QuaDRiGa: A 3-D multicell channel model with time evolution for enabling virtual field trials. *IEEE Trans. Antennas Propag.* 62(6), 3242–3256 (2016)

12 Liu, L., et al.: The COST 2100 MIMO channel model. *Trans. Wireless Commun.* 19(6), 92–99 (2012)

13 Han, S., Mao, H., Dally, W. J.: Deep compression compressing deep neural networks with pruning, trained quantization and huffman coding. In: Proceedings of the International Conference on Learning Representations (ICLR). San Juan, PR, pp. 1–14 (2016)