Machine Learning Based Link-to-System Mapping for System-Level Simulation of Cellular Networks

Eunmi Chu
Chungnam National University
Daejeon, Republic of Korea, 34134
E-mail: emchu@cnu.ac.kr

Hyuk Ju Jang
Chungnam National University
Daejeon, Republic of Korea, 34134
E-mail: entaline@gmail.com

Bang Chul Jung
Chungnam National University
Daejeon, Republic of Korea, 34134
E-mail: bcjung@cnu.ac.kr

***Abstract*—This paper proposes a machine learning (ML)-based exponential effective signal-to-noise ratio (SNR) mapping (EESM) method for simulating the system-level performance of cellular networks, which utilizes a deep neural network (DNN) regression algorithm. We first explain overall procedure of the link-to-system (L2S) mapping algorithm which has been used in commercial standardization organizations such as IEEE 802.16 and 3GPP LTE. Then, we apply the proposed ML-based EESM method to the existing L2S mapping procedure. The processing time of the L2S mapping becomes significantly reduced through the proposed method while the mean squared errors (MSE) between the actual block-error rate (BLER) from the link-level simulator and the estimated BLER from the L2S mapping technique is also decreased, compared with the conventional L2S mapping method.**

***Keywords*—Link-to-system mapping, exponential effective SNR mapping (EESM), physical-layer abstraction, system-level simulation, machine learning (ML), deep neural network (DNN).**

I. Introduction

Future wireless mobile networks will be mainly operated on a wide bandwidth to provide high data rate service. In a wide band channel, a transport block (TB) is allocated into N narrow band channels and each narrow band channel goes through a different fading condition on its own subcarrier. Therefore, user equipment (UE) experience different post-processing signal to interference plus noise ratio (SINR) over every subcarrier.

In a traditional narrow band channel, block error rate (BLER) is estimated from a curve of mean SINR and mean BER. On the contrary, in the wide band channel, different N post-processing SINRs are mapped to the averaged post-processing SINR. Since the concept of the averaged post-processing SNR is defined as an effective SNR, this many-to-one mapping is called an effective SNR mapping (ESM) technique. Besides, ESM technique is used for the purpose of physical layer abstraction when evaluating system-level simulator (SLS). A simplified link-level simulator (LLS) helps SLS reduce complexity of computation and it can help improve simulator performance. Since the concept of physical level abstraction for SLS is reflected, this is also called link-to-system (L2S) mapping technique. Accordingly, L2S mapping is that post-processing SINRs extracted from LLS are mapped to an effective SNR and BLER is predicted by the effective SNR.

In the prior studies, many researchers analyzed exponential effective SINR mapping (EESM) [1], [2] and mutual information based effective SINR mapping (MIESM) [3] as representative L2S mapping. In [4], effective SNR is analyzed on the side of uplink. In [5], impact of L2S is analyzed on the side of system level. However, there are too many data extracted from LLS as well as too much processing time is need to find EESM mapping parameters for various cases. Moreover, loss incurs due to an inaccuracy from AWGN curve of SNR and BLER.

Therefore, recent researchers has studied ML-based link abstraction models. In [6], support vector machine (SVM) is used to enable ML classification for fast adaptive modulation coding. This scheme exploits measurement of single TB success or failure to train the classifier. In [7], a ML method based on a logistic regression is proposed. To predict a TB success or failure, their basic model uses mean and standard deviation of the SINR set, modulation rate, and TB size as input variables. To improve the estimation accuracy, adding terms of higher order or combinations of input variables are used in an enhanced model.

In order to utilize ML-based link abstraction models that have been studied so far, ML algorithms should be applied on both of eNB and UE sides. However, since the number of UEs is too large, it is difficult to embed ML algorithms in all UEs. Some UEs can directly apply ML algorithms while other UEs should take the existing EESM method. Therefore, eNBs still need the existing EESM method. In this paper, we propose a ML-based EESM method where training data are learned by deep neural network (DNN) regression and L2S mapping based on EESM is executed by optimizer algorithms. From training DNN, we can dramatically reduce processing time and accurately yield an AWGN curve form DNN regression. From optimizer algorithms, we can speedily find EESM mapping parameters compared to existing search algorithms.

II. Effective SNR Mapping Procedure

The overall procedure of the *exponential* effective SNR mapping (EESM) is shown in Fig. 1, which basically receives BLER from the LLS and then passes over two parameters (α_1, α_2) to the SLS. The details are as follows:

1) First of all, the BLER results according to the channel type (AWGN, fading channel), the number of used

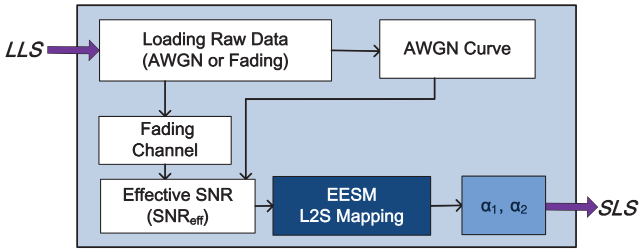

Fig. 1. Overall procedure of effective SNR mapping.

subcarriers, and the number of realizations are received from the LLS. The number of post-processing SNRs is determined by the number of used subcarriers and the number of realizations denotes the number of transport blocks (TBs) under the same channel condition. The range of SNR values needs to be carefully adjusted at the LLS so that a similar number of BLER performances are collected, which covers from 0.01 to 0.9 in general. If the BLER performances do not appear evenly, then an exact L2S mapping can not be obtained. As a reference, the BLER performance according to the SNR values is needed in the AWGN channel for each modulation and coding schemes (MCS) with channel quality indication (CQI).

2) AWGN curve corresponding SNRs and BLERs is generated from the fitting curve. Since AWGN curve for each CQI is varied, we find best fitting curve for each CQI.
3) Then, the BLER and post-processing SNRs are jointly obtained over various fading channels through the LLS. In particular, N different post-processing SNRs over subcarriers are denoted by $\{\gamma_1, \gamma_2, \cdots, \gamma_N\}$, where γ_k denotes the k-th post-processing SNR. Let e denote the BLER.
4) When $\{\gamma_1, \gamma_2, \cdots, \gamma_N\}$, α_1, and α_2 are given, the effective SNR based on the EESM is given by

$$\gamma_{\text{eff}}\left(\alpha_1, \alpha_2\right) = -\alpha_1 \ln\left(\frac{1}{N}\sum_{k=1}^{N} \exp\left(-\frac{\gamma_k}{\alpha_2}\right)\right), \quad (1)$$

where α_1 and α_2 are determined later. At the SLS, the BLER with the post-processing SNR values, $\{\gamma_1, \gamma_2, \cdots, \gamma_N\}$, will be determined by $f(\gamma_{\text{eff}}(\alpha_1; \alpha_2))$, where f indicates the BLER in the AWGN channel when the SNR value is equal to $\gamma_{\text{eff}}\left(\alpha_1, \alpha_2\right)$.
5) For a given MCS, the optimal parameters, (α_1^*, α_2^*), are determined by

$$(\alpha_1^*, \alpha_2^*) = \underset{(\alpha_1,\alpha_2)}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M} \left[\log_{10}(e^i) - \log_{10} f(\gamma_{\text{eff}}^i(\alpha_1, \alpha_2))\right]^2 \right\}, \quad (2)$$

where M denotes the total number of independent LLS simulations with different post-processing SNR values and e^i denotes the BLER of the i-th post-processing SNR values. In addition, $\gamma_{\text{eff}}^i(\alpha_1, \alpha_2)$ is obtained by Eq. (1).

III. Machine Learning-Based EESM

Obtaining the optimal parameters with Eq. (2) is burdensome since the simulation results from the LLS is large in general. Thus, it is necessary to reduce the computation complexity as well as to improve the accuracy. In this section, we first apply the DNN regression method to make the BLER curve according to SNR in AWGN channel, which is summarized in Algorithm 1.

Algorithm 1: DNN regression

```
/* Configure DNN regression */
regressor = learn.DNNRegressor(feature_columns,
  hidden_units=[100, 200,100],
  optimizer=tf.train.ProximalAdagradOptimizer(
  learning_rate=0.1, l1_regularization_strength=0.001),
  activation_fn=tf.nn.sigmoid)
/* Train measured data up to 4000
   times */
input_training_fn ← (awgn_snr, awgn_bler)
regressor.fit(input_fn=input_training_fn, steps=4000)
/* Predict of BLERs for test SNRs */
input_reff_fn ← snr range
predictions = list(regressor.predict_scores(input_fn =
input_reff_fn))
regressed_bler = np.asarray(predictions)
```

1) We utilize the DNN regression method instead of the best fitting curve to obtain the BLER curve in AWGN channels. The DNN consists of several hidden layers between the input and output layers. Hidden layers of (100, 200, 100) layers are used in Adagrad optimizer [8]. Learning rate is set to 0.1 in this paper, which implies how quickly tune to the target SNR value. The regularization strength to prevent overfitting is set to 0.001. The sigmoid function $1/(1+e^x)$ is used as an activation function in hidden layers.
2) DNN regression continues training for SNRs and BLERs on AWGN channel with the learning rate at each epoch. The number of training is 4,000.
3) After training data, BLERs are predicted for test set of SNRs. Finally, we can get an enhanced AWGN curve of SNRs and BLERs.

Next, we apply the optimization algorithm to efficiently find (α_1^*, α_2^*), which is summarized in Algorithm 2.

1) To find the optimal parameters (α_1^*, α_2^*), we load the simulation results from the LLS in fading channels as described in Fig. 1.
2) In the ML scheme, the loss function is defined as the difference between the calculated effective SNR value from algorithm 2 and AWGN SNR obtained from algorighm 1 at the same BLER. We calculate loss as the expectation of loss function over BLERs

Algorithm 2: Find optimal α_1 and α_2

```
/* Load data on Fading channel */
snr_k ← post-processing SNRs, bler ← BLER
/* Calculate γ_eff with α_1 and α_2 */
snr_eff = -1*alpha1*tf.log(tf.reduce_mean
  (tf.exp(-1*snr_k/alpha2), axis=1))
/* Decide target SNR by regression */
target_snr ← predicted snr corresponding to BLER
/* Calculate loss function */
loss = tf.reduce_sum(tf.abs(tf.subtract
  (target_snr,snr_eff)))
/* Select training algorithms */
train=tf.train.AdagradOptimizer(0.1).minimize(loss)
train=tf.train.RMSPropOptimizer(0.1).minimize(loss)
/* Training data 4,000 times */
with tf.Session() as sess:
    sess.run(init)
    for i in range(4000):
        sess.run(train)
/* Calculate MSE in test data set */
regressed_bler ← estimated BLER, y_data ← BLER
mse =
  np.mean(np.square(np.subtract(np.asarray(y_data),
  np.asarray(regressed_bler))))
```

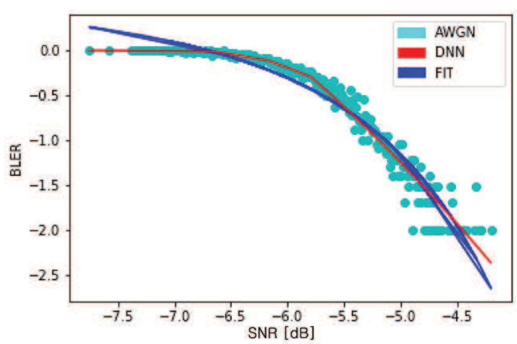

Fig. 2. BLER according to SNR in AWGN channels when the first CQI of the 3GPP LTE system is used.

3) We apply optimization algorithms, Adagrad and RMSProp, to find the optimal parameters that minimize the loss function.
4) With the optimal parameters, the mean squared error (MSE) is calculated by

$$\mathsf{MSE} = \frac{1}{M}\sum_{i=1}^{M}\{\log_{10} e^{i} - \log_{10} f(\gamma_{\text{eff}}^{i}(\alpha_1^*, \alpha_2^*))\}^2. \quad (3)$$

IV. Simulation Results

Fig. 2 shows the BLER curve in the AWGN chnnel when the first CQI of the commercial 3GPP LTE system is used. Legend *'AWGN'* (sky blue dot) presents the measured SNRs and the measured BLERs from the LLS and legend *'FIT'* (blue

TABLE I
Loss function comparison

CQI	FIT	DNN	CQI	FIT	DNN
CQI1	0.033	0.018	CQI9	0.019	0.011
CQI2	0.027	0.014	CQI10	0.021	0.008
CQI3	0.033	0.016	CQI11	0.011	0.013
CQI4	0.038	0.013	CQI12	0.025	0.013
CQI5	0.032	0.014	CQI13	0.013	0.010
CQI6	0.017	0.011	CQI14	0.015	0.013
CQI7	0.020	0.015	CQI15	0.030	0.010
CQI8	0.012	0.016	-	-	-

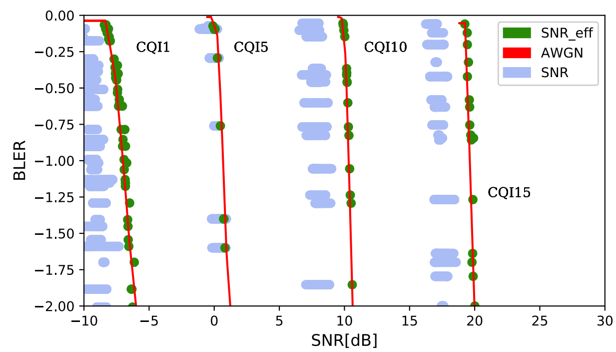

Fig. 3. Effective SNR mapping results of the proposed ML-based EESM method in case of CQI={1, 5, 10, 15}.

line) presents the BLER curve in the AWGN chnnel, which is obtained by the best fitting in Section 2. Legend *'DNN'* (red line) presents the BLER curve by the DNN regression via Algorithm 1 in Section 3. We can show that *'DNN'* yields a more accurate curve compared to *'FIT'*. Table I shows the results of the loss function of *'DNN'* and *'FIT'* for various CQIs of 3GPP LTE systems.

Table 2 shows the optimal EESM parameters (α_1^*, α_2^*) for various CQIs of 3GPP LTE systems by AdaGrad optimizer and RMSProp optimizer. The MSE performance of RMSProp is better than that of AdaGrad. Fig. 3 shows the effective SNR

TABLE II
Optimal parameters (α_1^*, α_2^*)

CQI	AdaGrad			RMSProp		
	α_1	α_2	MSE	α_1	α_2	MSE
1	2.78	1.64	0.076	3.80	2.25	0.075
2	3.44	3.26	0.078	2.19	2.08	0.078
3	3.47	3.23	0.026	3.95	3.68	0.026
4	3.14	2.95	0.060	3.12	2.93	0.060
5	3.64	3.32	0.016	2.24	2.05	0.018
6	4.09	2.43	0.044	2.67	1.58	0.038
7	1.89	1.72	0.018	3.11	2.83	0.027
8	3.78	3.50	0.013	3.35	3.10	0.012
9	2.35	2.19	0.131	4.21	3.94	0.098
10	3.08	1.80	0.092	6.77	4.00	0.037
11	2.69	1.57	0.077	6.62	3.90	0.030
12	3.34	1.94	0.158	9.99	5.89	0.067
13	3.67	2.13	0.153	13.92	8.19	0.070
14	3.89	2.24	0.308	15.17	8.92	0.099
15	4.70	2.68	0.312	12.51	7.31	0.172

mapping results of the proposed ML-based EESM method in case of CQI={1, 5, 10, 15}, where the RMSProp optimizer algorithm is used. With this figure, we obverse that the proposed method predicts the BLER quite well.

V. Conclusions

In this paper, we proposed a ML-based effective SNR mapping method to reduce the computational complexity and improve the accuracy of BLER prediction for system-level simulation of cellular networks. As a further study, we will apply the proposed method for link-to-system mapping of 5G wireless networks.

Acknowledgement

This work was supported by “The Cross-Ministry Giga KOREA Project” grant from the Ministry of Science, ICT and Future Planning, Korea, [GK 18S0400, Research and Development of Open 5G Reference Model] and this work was supported by “The Basic Science Research Program through the NRF” funded by the Ministry of Science and ICT, [NRF-2016R1A2B4014834].

References

[1] J. Olmos, S. Ruiz, M. Gareia-Lozano, and D. Martin-Sacristan, “Link abstraction models based on mutual information for LTE downlink,” *COST 2100 TD(10) 11052 Aalborg, Denmark*, June 2010.

[2] E. Tuomaala and H. Wang, “Effective SINR approach of link to system mapping in OFDM/multi-carrier mobile network,” *IEE Mobility Conference 2005. The Second International Conference on Mobile Technology, Applications and Systems*, pp. 140–144.

[3] *IEEE C802.16m-07097*, “Link performance abstraction based on mean mutual information per bit (MMIB) of the LLR channel,” 2007.

[4] M. B. Hcine and R. Bouallegue, “Analysis of uplink effective SINR in LTE networks,” *Proc. IWCMC 2015*, Aug. 2015.

[5] Z. Hanzaz, H. D. Schotten, “Impact of L2S interface on system level evaluation for LTE system,” *Proc. IEEE MICC 2013*, Nov. 2013.

[6] R. Daniels and R. W. Heath, Jr., “Online adaptive modulation and coding with support vector machines,” (invited) *Proc. of the IEEE European Wireless Conference*, Lucca, Italy, pp. 718-724, April 12-15, 2010.

[7] A. C. Mesa, M. C. AguayoTorres, F. J. MartinVega, G. Gmez, F. BlanquezCasado, I. DelgadoLuque, J. Entrambasaguas, “Link abstraction models for multicarrier systems: a logistic regression approach,” *International Journal of Communication Systems*, 31(1), 2018.

[8] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv preprint arXiv:1609.04747 2016.