# Optimal Cache Resource Allocation Based on Deep Neural Networks for Fog Radio Access Networks

Sovit Bhandari, Hoon Kim, Navin Ranjan, Hong Ping Zhao, Pervez Khan

IoT and Big-data Research Center, Department of Electronics Engineering, Incheon National University, South Korea
sovit198@gmail.com, hoon@inu.ac.kr, ranjannavin07@gmail.com, hongpieng614@inu.ac.kr,
pervaizkanju@hotmail.com

## Abstract

Cache resource allocation is of great significance for the advanced cellular networks, especially for fog radio access networks (F-RANs). Many cache resource allocation schemes have been proposed to increase the performance of F-RANs optimally. However, it is still challenging to apply these schemes and attain live performance in F-RAN systems since most of them need accurate and real-time data which shows radio link information or other network information. This paper presents a cache resource allocation strategy based on deep neural network (DNN) along with the training method required to train the neural networks. Simulation results in terms of DNN accuracy are shown to validate that the performance of proposed method approaches to that of the conventional iterative method in most cases.

**Keywords:** Cache resource allocation, Fog access point, Optimal, Deep neural network

## 1 Introduction

The phenomenon of the increasing number of user equipments (UEs) and the rapidly growing demand on broadband communication services has caused an exponential growth of requirements for data processing, storage and communications [1-2]. To alleviate the problem associated with the increasing demand for mobile data traffic much research has been focused on the design of next generation cellular communication systems. Among many research activities related to the development of advanced cellular architecture, F-RANs have been considered as a promising paradigm for improving the spectral efficiency (SE) [2-3].

F-RAN compensates the drawbacks of conventional cellular architecture such as cloud radio access networks (C-RANs) as it pays more attention towards the improvement of user experience by minimizing the latency occurring in the backhaul link [4-5]. In F-RANs, the fog access points (F-APs) and fog user equipments (F-UEs) are equipped with caching. Besides, the F-APs can execute radio signal processing locally and can manage their caching memories flexibly [6-7]. With the increasing demand for the mobile data traffic, the limited nature of caching and signal processing capabilities of F-APs cannot tackle the heavy burden occurring on the fronthaul of the F-RAN system. Recently, much attention has been addressed for achieving ultra-low latency and maximize delivery rate by allocating proper cache resource of the F-APs [8] from different aspects.

### 1.1 Related Works

There are some related works worth mentioning. The work in [8-9] focused on maximizing the delivery rate by assuming that the requested content is already available in the cache memory of the fog nodes. The authors in [10] studied a joint resource allocation and content caching problem which targeted at minimizing the maximum content request rejection rate. The work in [11] studied a problem of joint caching, channel assignment, and interference management of small-cell cellular networks to maximize the system throughput. The paper in [12] presented a two-step iterative method to determine caching placement for two network slices. Similarly, the work in [13] focused on designing a dynamic resource allocation strategy to balance the load in fog environment. Likewise, in [14] joint optimization framework for low power fog nodes, data service operators, and data service subscribers is presented to achieve optimal resource allocation schemes in a distributed fashion. However, an efficient optimal cache resource allocation strategy in fog computing is still absent. The high computational requirement for the iterative algorithms used in above works makes their real-time implementation challenging because they are typically executed in a time frame of milliseconds.

### 1.2 Contributions and Organizations

Inspired by the fact that the use of machine learning approach to design resource allocation schemes has tremendous potential to lessen the system complexity and obtain real-time performance [15-17], we have

*Corresponding Author: Pervez Khan; E-mail: pervaizkanju@hotmail.com

DOI: 10.3966/160792642020072104007

applied the deep neural network (DNN) to develop optimal cache resource allocation strategies for maximizing total delivered data contents. The DNN model with moderate size can allocate resource in almost real time as the passing of the inputs through DNN layers only requires a small number of simple operations [15, 18]. In this paper, the training method is provided to obtain the parameters of DNN.

The rest of this paper is organized as follows. The system model is described in section 2. Section 3 formulates the cache resource allocation problem. Section 4 presents the simulation results. Finally, Section 5 concludes the paper and provides future direction of this work.

## 2 System Model

In this section, a downlink $N \times M$ F-RAN system composed of N multi-antenna (UEs) devices, M multi-antenna F-APs, one base-band unit (BBU), and one centralized cloud is modeled which is shown in Figure 1. We have $U = (1, 2, ..., N)$ UEs requesting data from $F = (1, 2, ..., M)$ F-APs. In the system model diagram, the solid lines denotes fronthaul links whereas dashed lines indicate air interface links. The every UEs are served by the every other F-APs that are connected to a BBU pool in the cloud via common public radio interface (CPRI) cables.

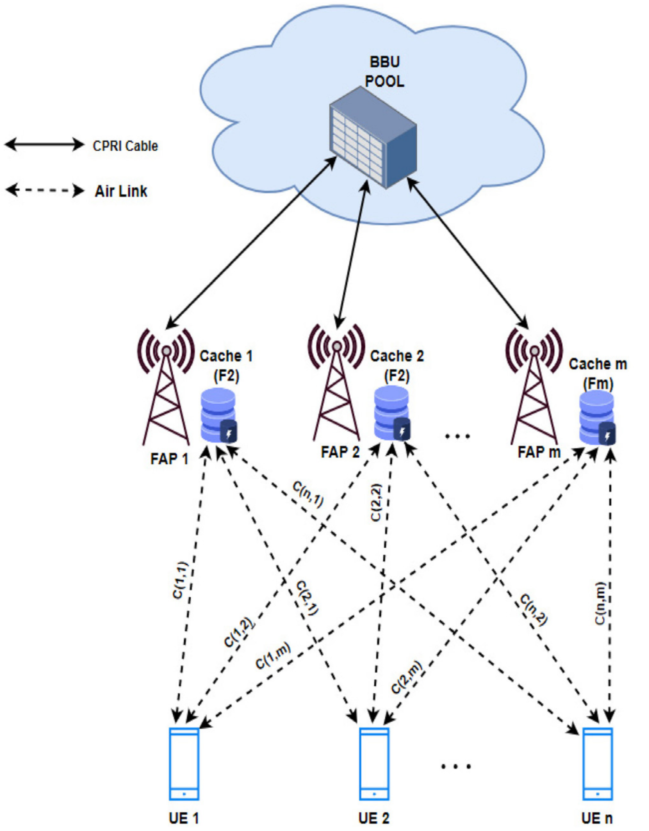

**Figure 1.** A system model for fog radio access networks

The model deals only with F-RAN cache delivery phase, so for a full caching case, all the requested files are assumed to be stored in the local cache such that it can be retrieved directly from the F-AP without downloading it from BBU. Since the single F-AP with edge cache is not enough to store all the information, the model assumes that the portion of the information requested by the users is present in many F-APs [9]. The single UE can be connected to multiple F-APs at the same time based on the concept of global cloud radio access network mode [6]. The link capacity between UEs and F-APs is determined by Shannon capacity limit i.e., $B \times \log_2(1 + SIR)$, where B is the channel bandwidth. The channel distribution of the air interface link is assumed to be a Rayleigh distribution, while the fronthaul link capacity is fixed.

Let $C_{i,j}^{Ai}$ be the air interface link capacity between UE $i(i \in U)$ and access point $j(j \in F)$, $C_{j,1}^{FH}$ be the fronthaul link capacity between access point j and cloud 1, and $F_j$ be the cache capacity of F-AP *j*. Similarly, let $R_i$ be the data to be received by the UE *i*, $V_i$ be the minimum data volume requested by UE *i*, and $\phi_{i,j}$ be the optimal cache resource allocation parameter and can be defined as the portion of user *i* requested content in F-AP *j*.

## 3 Resource Allocation Problem

In this section, the total data volume maximization and cache resource allocation problems are studied. The cache resource allocation framework based on DNN is presented along with the training method.

### 3.1 Problem Formulation

The maximum total data volume that can be delivered in a multi F-APs F-RAN system can be realized by solving the cache resource allocation problem presented in section 2 with the arbitrary number of users, N. The total data obtained by all the users at any timeslot is given by $R_{total} = \sum_{i=1}^{N} R_i$, where $R_i$ is $\sum_{j=1}^{M} \phi_{i,j} F_j$. For cache-level transmission, to maximize the total sum of receivable data contents of all users can be formulated as (P1) which is given as [19]:

$$\text{(P1)}\ \max \left[ \sum_{i=1}^{N} \sum_{j=1}^{M} \phi_{i,j} F_j \right] \quad \textbf{(1)}$$

$$\text{s.t.}\ \frac{\phi_{i,j} F_j}{\tau} \le C_{i,j}^{Ai} \qquad \forall i, j \quad \textbf{(2)}$$

$$\sum_{j=1}^{M}\phi_{i,j}F_j \ge V_i \qquad \forall i \tag{3}$$

$$\sum_{i=1}^{N}\phi_{i,j} \le 1 \qquad \forall j \tag{4}$$

$$\phi_{i,j} \ge 1 \qquad \forall i,j \tag{5}$$

where the transmission delay $\tau$ is the sum of the signal processing and information exchange time between F-AP $j$ and the BBU pool. For simplicity, $\tau$ is assumed to be same for all F-APs. In (P1), constraint (2) shows that the portion of user data delivery rate $(\phi_{i,j}F_j/\tau)$ obtained from specific F-AP is bounded by the air interface link capacity connecting user to that F-AP i.e., $C_{i,j}^{Ai}$. Likewise, constraint (3) denotes that the total data volume obtained by the user must be greater than or equal to the volume of the data requested by them. Similarly, constraint (4) means that the sum of the portion of contents that can be accessed by all the users from any particular F-AP cannot be more than 100% of its total capacity. Furthermore, constraint (5) ensures that the portion of content that any user $i$ can access from F-AP $j$ is non-negative.

For the feasible solution of (P1) all the constraints representing it need to be satisfied. The complexity of the iterative problem (P1) increases with the increase of UEs and F-APs. Furthermore, above formulated cache resource allocation strategy is based on the perfect air link channel between UEs and F-APs. It is still challenging to carry out this strategy and attain real-time performance in practice when there exist immense UEs and F-APs.

### 3.2 Cache Resource Allocation Based on DNN

In this part, in order to achieve the optimal cache resource allocation strategy and obtain live performance, DNN based cache resource allocation framework is presented.

Based on the problem statement (P1), we propose a cache resource allocation framework based on DNN to optimize the cache resource of F-RANs. For simplicity, we model a DNN based framework for $3\times 2$ F-RAN system as shown in Figure 2. The modeled framework consists of three layers, they are; the input layer, two hidden layers, and the output layer. The inputs are the random air interface link capacity between UEs and F-APs, F-AP cache capacity, user data volume requirement, and transmission delay, i.e., $C$(1, 1), $C$(1, 2), $C$(2, 1), $C$(2, 2), $C$(3, 1), $C$(3, 2), and $F_j$ respectively. These inputs have continuous probability density functions (PDF). The outputs are the optimal values of the cache resource, i.e., $\phi_{1,1}$, $\phi_{1,2}$, $\phi_{2,1}$, $\phi_{2,2}$, $\phi_{3,1}$, and $\phi_{3,2}$, respectively. In this paper, the optimal values of the F-APs cache resource are used to maximize the receivable total data volume i.e. $R_{total}$. The activation function for the hidden layers is rectified linear unit (ReLU), namely, $y=\max(0,x)$, where $x$ and $y$ indicate the input and output of the neurons (neural unit), respectively. Similarly, the activation for output layer is sigmoid, namely, $a=1/(1+e^{-b})$, where $a$ and $b$ denote the input and output of the neural unit, respectively. We have used sigmoid activation function in the output layer to obtain outputs between 0 and 1 [20-21]. The bias inputs to the neurons is considered at every layer. The other parameters used for the proposed neural network is given in the simulation results.

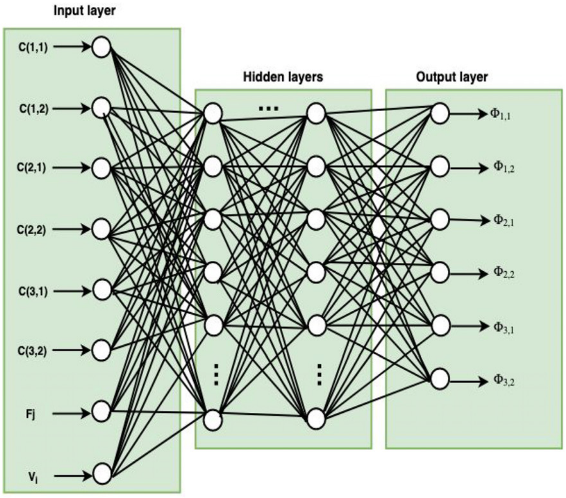

**Figure 2.** Cache resource allocation framework for $3\times 2$ F-RAN system on DNN

The DNN needs to be trained to get the weights of each neurons. Figure 3 illustrates the training process for the proposed cache resource allocation framework based on DNN. In the training process, the mean squared error minimization criterion is implemented as it is a multi-variant regression problem [22]. Adam optimizer is used to update the weight and learning rate values since it is straightforward to implement, computationally efficient and has little memory requirements [23].

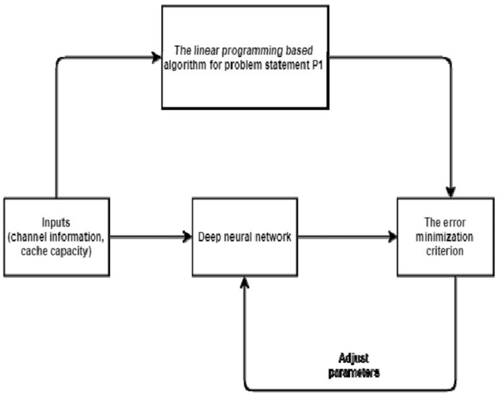
The linear programming based algorithm for problem statement P1
Inputs (channel information, cache capacity)
Deep neural network
The error minimization criterion
Adjust parameters

**Figure 3.** Cache resource allocation training process based on DNN

The fog nodes in the F-RAN system are responsible for computation, communication, and storage. Each F-APs consists of two layers of fog nodes. The fog nodes in fog layer 1 are responsible as a gateway to edge devices, as they are well equipped with the routers. Similarly, fog nodes in layer 2 cover larger area as they are connected to the cloud and fog nodes in the first layer via fiber. These nodes are well equipped with the faster CPUs, more cores, dedicated Graphic Processing Units (GPUs), and larger storage [24]. Since the fog nodes in layer 2 are computationally powerful, they can be used for training the datasets for prediction.

### 3.3 Data Generation

For training the DNN model, we first generate a large set of optimal cache portion ($\phi_{i,j}$) for $3\times 2$ F-RAN system by solving the iterative problem presented in (P1), using the MATLAB simulation platform. For simplicity, the value of $F_j$, $C_{i,j}^{Ai}$, $V_i$, and $\tau$ are fixed, for all $i, j$. The average value of SIR to calculate the $C_{i,j}^{Ai}$ is assumed to be 5 dB, and channel bandwidth is considered to be 20 MHz, respectively. For each $F_j$, training and validation data samples are obtained, considering the value of $F_j$ between 1 to 6 Mb. The different data samples are then combined to get the entire training data set, as well as the validation data set. The optimal values of cache portions are realized over the $1\times 10^2$ different communication channels for the fixed values of other system parameters. The zero and negative values of the cache portions representing the non-optimal solutions were removed from the datasets. The size of the validation dataset is assumed to be smaller than that of the training dataset as per the ongoing trend. The test datasets are taken as the validation datasets.

### 3.4 Algorithm for Training DNN

In this paper, the training method for the obtained dataset is based on Adam optimizer to lessen the time required to train the DNN model with a very large data samples. The weight for datasets are updated by computing moment estimates, which is presented in [23]. Since the training the DNN with a very large data samples is resource consuming, mini-batch gradient descent (MBGD) algorithm is used to train the proposed DNN based model. The total datasets are divided into *K* batches of size *B*. For each batch, the datasets are trained, and the loss function for each batch is calculated using mean square error (MSE). The loss function for each batch of size *B* is calculated using, $J(\theta)=\frac{1}{6B}\sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{M}(\phi_{i,j,b}-\phi_{i,j,b}^{*})^2$, where $\phi_{i,j,b}$, and $\phi_{i,j,b}^{*}$ is the actual and predicted values of the cache portions, respectively. The DNN weights ($\theta$) is updated at each batch by minimizing the loss function. The weight is updated using, $\theta_k=\theta_{k-1}-a\cdot\hat{m}_k/(\hat{v}_k+\in)$, where, $a$, $\hat{m}_k$, and $\hat{v}_k$ is the learning rate, time varying first moment estimate, and time varying second moment estimate, respectively.

The detail procedure is shown in Algorithm 1.

**Algorithm 1.** DNN based framework

```
i.   Input: link capacity, user volume requirement,
     transmission delay, cache capacity;
ii.  Output: optimal values of cache portion for each
     users are obtained by the conventional scheme
     proposed in (P1);
iii. Training Data: Divide the training data into $I$
     batches of size $B$;
iv.  Initialize: learning rate ($a$), exponential decay
     rate ($\beta_1,\beta_2\in[0,1]$) for moment estimates
v.   For each batch $b_k (k=1,2,...,K)$ do
       a. Calculate the DNN accuracy using MSE
          $J(\theta)=\frac{1}{6B}\sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{M}(\phi_{i,j,b}-\phi_{i,j,b}^{*})^2$
       b. Compute the updated DNN weight $\theta$ θ by
          minimum loss function and moment estimates
          $(\hat{m}_k,\hat{v}_k)$;
          $\theta_k=\theta_{k-1}-a\cdot\hat{m}_k/(\hat{v}_k+\in)$
     endfor
vi.  Save the better accuracy producing DNN model
```

## 4 Performance Evaluation

In this section, simulation results are shown to evaluate the performance of our proposed cache resource allocation based framework on the DNN model. We have used Keras library on top of TensorFlow framework in Python 3.6 as a programming platform. The training process for our datasets is performed by using a computation server with one Intel core i7 CPU, four Intel Xeon E7-1680 processors, and 128 GB random access memory. The results are obtained by using a computer with 16 GB random access memory and Intel Core i7-8700 processor.

### 4.1 Simulation Parameters

Table 1 given below summarizes the simulation parameters and their values which are taken into account. The number of neurons used in each hidden layer is 100. The initial value of learning rate ($a$) for the 32 batch size ($B$) is taken as 0.0001. Similarly, the values of exponential decay rate i.e., $\beta_1$, and $\beta_2$ are taken as 0.9 and 0.999, respectively. The training process is based on the data obtained by using the scheme proposed in (P1). The total size of input dataset is 22,402 in which 16,801 entries are used for training the model, whereas remaining 5,601 entries are used as the validation data sets to predict the optimal values of the cache portion.

**Table 1.** Simulation parameters

| Parameters | Values |
|---|---|
| Training Size | 16,801 |
| Test Size | 5,601 |
| Number of UEs | 3 |
| Number of F-APs | 2 |
| Number of Hidden Layers | 3 |
| Neurons used in each Hidden Layer | 100 |
| Number of Inputs | 8 |
| Number of Outputs | 6 |
| Epoch | 1-1,000 |

Detailed configuration of the parameters for proposed model is listed in Table 2.

**Table 2.** Configuration of parameters for DNN based model

| Layer | Name | Shape | Trainable Parameters |
|---|---|---|---|
| 0 | Inputs | (1,8) | —— |
| 1 | Dense (ReLU) | (1,100) | 9,00 |
| 2 | Dense (ReLU) | (1,100) | 10,100 |
| 3 | Dropout (0.2) | ——— | ——— |
| 4 | Dense (ReLU) | (1,100) | 10,100 |
| 5 | Dense (Sigmoid) | (1,6) | 6,06 |

## 4.2 Simulation Results

In this sub-section, performance of test data set, the prediction accuracy of optimal cache portions $(\phi_{i,j})$ and the objective function $(R_{total})$ is shown for different values of training epoch. Here, the values of optimal cache portions, and objective function obtained from (P1) are referred to as actual values of optimal cache portions and objective function, and are represented as $\phi_{i,j}$, and $R_{total}$, respectively. Similarly, the values of optimal cache portions, and objective function predicted from the DNN model are referred to as predicted values of optimal cache portions and objective function, and are represented as $\phi^*_{i,j}$, and $R^*_{total}$, respectively. The MSE is used for loss function, and normalized root mean square error (NRMSE) method is used for calculating the overall accuracy of the trained model. As the values of $\phi_{i,j}$ lies within 0 and 1, and the values of $R_{total}$ is normally greater than one, the normalization method is taken into account for calculating accuracy.

### 4.2.1 Prediction of Test Data Set

Figure 4 shows the performance of the DNN model for different training epochs. In Figure 4, MSE performance of test data sets against training data set is shown, provided that the test data set is not exclusively used during the training phase. The 'cyan curve' in the graph shows the MSE during training the DNN model, while the 'magenta curve' denotes the testing error. From the graph we can conclude that, there is no over-fitting problem in the trained model as the gap between the training and the testing error is very less. Moreover, we can also say that the direction of convergence on the target data set is desirable, as there is minimal fluctuation on testing error.

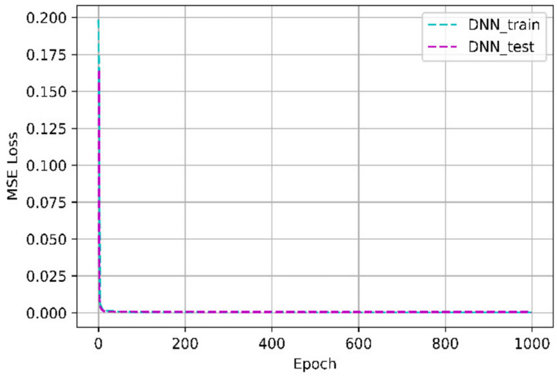

**Figure 4.** Performance of the model on the test data set which is exclusively generated after the training phase

### 4.2.2 Prediction of Optimal Values $(\phi^*_{i,j})$

Figure 5 to Figure 8 show how the predicted optimal values $(\phi^*_{i,j})$ converge around the regression line. The values of $\phi^*_{i,j}$ illustrated in the figures below are from 5,601 entries of test data sets. The total number of $\phi^*_{i,j}$ plotted in the figures given below are 6×5601. The solid blue straight line going from the origin, as shown in the figures, is the regression line. Regression line helps in the graphical interpretation of the errors obtained in every training epoch. The distance between each predicted point and the regression line is referred to as a prediction error. The cumulative sum of every deviation gives the overall error of the model. If the predicted points are below the regression line, then the error obtained is negative, else it is positive. If the points are along the regression line then there is no/minimal error in the system.

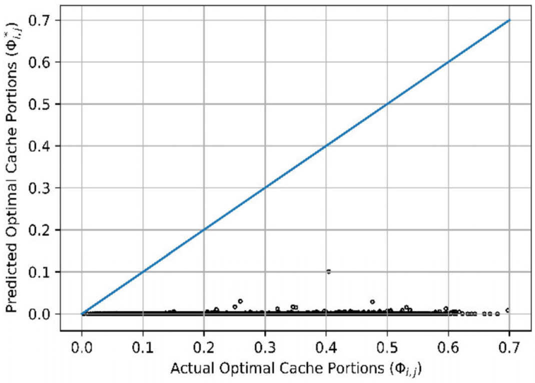

**Figure 5.** Actual vs. Predicted optimal cache portions, when the training Epochs = 1

Figure 5 shows that the trained model for only one training epoch doesn't predict the optimal values of the cache portion at all, as we can see most almost all of the predicted data are away from the regression line.

Similarly, Figure 6 to Figure 8 show that the predicted optimal values of cache portions lie nearer to the regression line whenever the model training epochs is increased, i.e., for 10, 100, and 1,000 epochs, respectively.

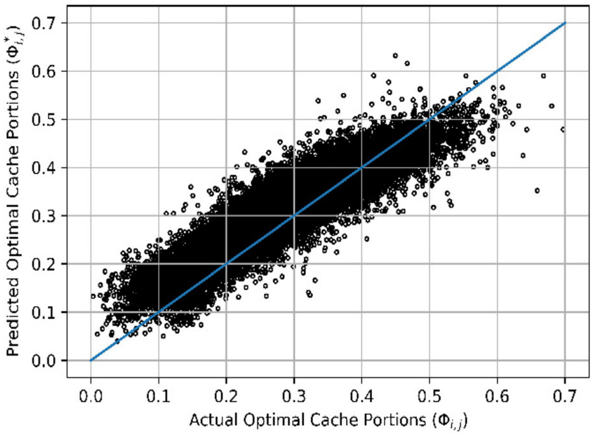

**Figure 6.** Actual vs. Predicted optimal cache portions, when the training Epochs = 10

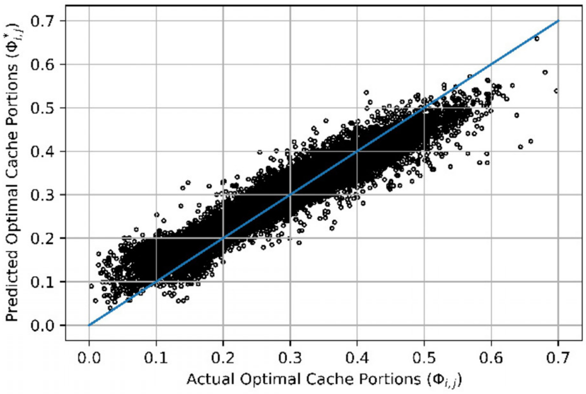

**Figure 7.** Actual vs. Predicted optimal cache portions when the training Epochs = 100

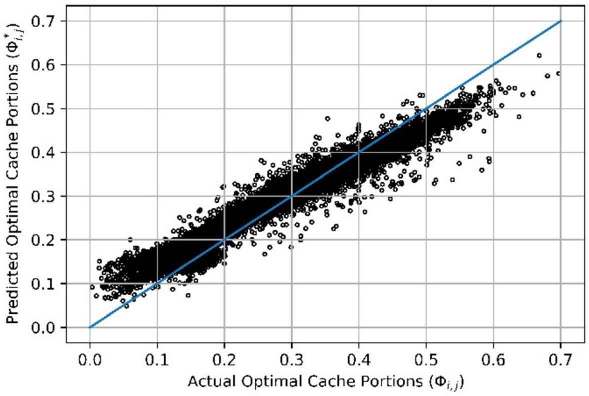

**Figure 8.** Actual vs. Predicted optimal cache portions, when the training Epochs = 1000

From Figure 7 and Figure 8, we can see that the predicted outputs are almost similar irrespective of the training epochs. Thus, we can conclude that the larger training epoch saturates the prediction.

### 4.2.3 Prediction of Objective Values ($R^*_{total}$)

Figure 9 to Figure 12 shows how the predicted values of objective function fit along the regression line for the training epochs, 1, 10, 100, and 1,000, respectively. The predicted values of the objective function ($R^*_{total}$) is obtained by multiplying the predicted results of the $\phi^*_{i,j}$ (from portion 4.2.2) with the corresponding $F_j$, as defined in the problem statement (P1). The values of $R^*_{total}$ illustrated in the Figure 9 to Figure 12 are from 5,601 entries of test data sets. The total number of $R^*_{total}$ plotted in the figures given in this part are 1×5601.

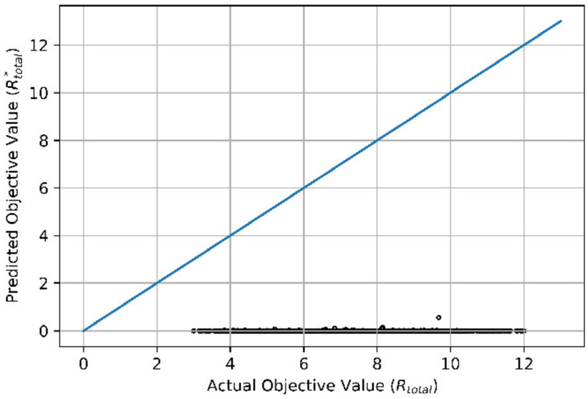

**Figure 9** Actual vs. Predicted objective value, when the training Epochs = 1

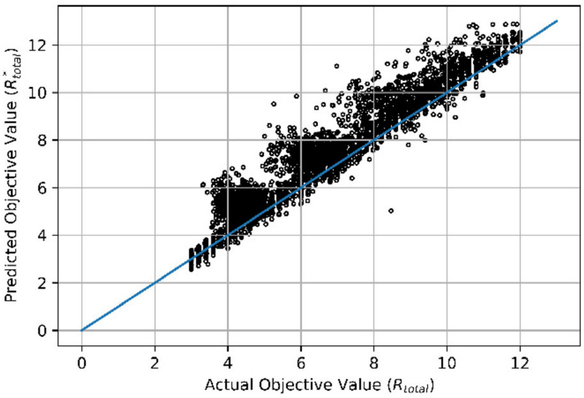

**Figure 10.** Actual vs. Predicted objective value when the training Epochs = 10

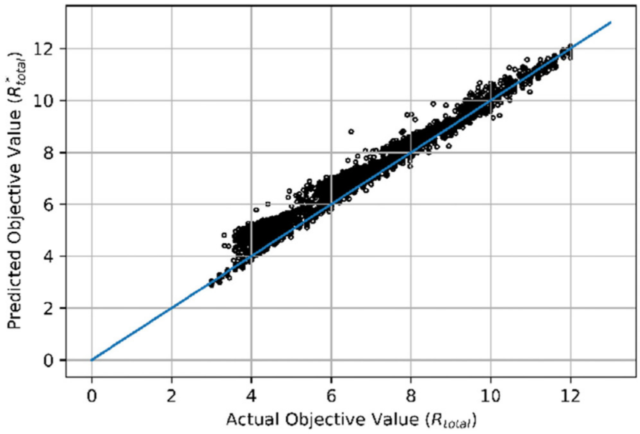

**Figure 11.** Actual vs. Predicted objective value, when the training Epochs = 100

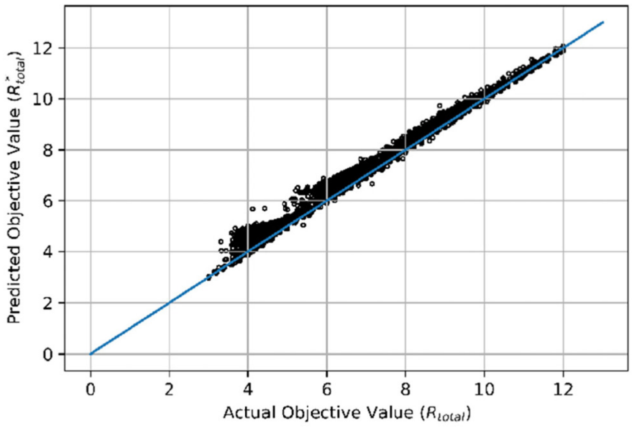

**Figure 12.** Actual vs. Predicted objective value when the training Epochs = 1000

Figure 9 shows that for the lower value of the training epoch, the predicted values of $R_{total}$ i.e., $R^*_{total}$ are sparsely distributed along the regression line and densely distributed in the region away from the regression line. Whereas, Figure 12 shows that $R^*_{total}$ almost approaches to $R_{total}$ (fitting along the regression line) when the value of training epoch is very high.

If we compare the results obtained in the part 4.2.2 and 4.2.3, we can say that the predicted data points of $R_{total}$ are sixth times lesser than that of $\phi_{i,j}$, as we have considered 3×2 F-RAN system in our paper.

### 4.2.4 Overall Performance

Figure 13 shows the NRMSE (normalized root mean square error) while predicting the values of $\phi_{i,j}$, and $R_{total}$ for different values of the training epoch. From the figure, it can be seen that, for the lower value of training epoch, the NRMSE for predicting the values of $\phi_{i,j}$, and $R_{total}$, almost equals unity. However, for the higher value of the training epoch, the NRMSE is lower than 0.2 for the prediction of both entities. As per the simulation results, we can see that the NRMSE is slightly less while predicting the value of objective function than optimal cache portion, independent of the training epoch.

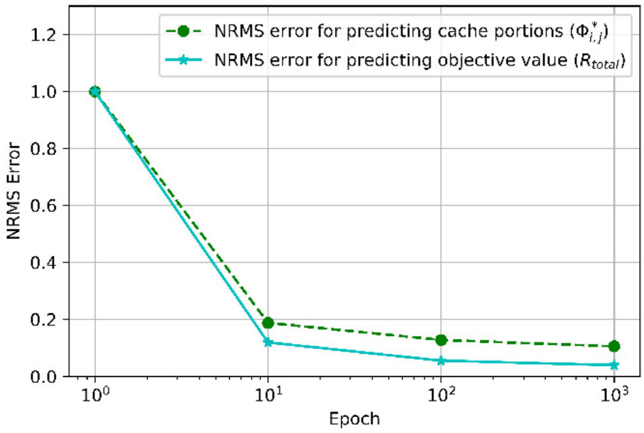

**Figure 13.** NRMSE while predicting $\phi_{i,j}$ and $R_{total}$ for different values of training epochs

Finally, Figure 14 plots the accuracy of DNN model for predicting the value of $\phi_{i,j}$, and $R_{total}$ for different training epochs. The DNN accuracy for predicting both the entities is almost similar for lowest value of training epoch, whereas it is found that the accuracy is marginally more for predicting the value of $R_{total}$ than $\phi_{i,j}$ for the higher value of training epoch. It is mathematically verified in the portion 4.3. From the figure, it can also be seen that the accuracy for predicting data points is greater for high value (1,00 epoch) of training epoch, but it almost saturates when the epoch is very high (1,000 epoch). The DNN accuracy of the predicted values of $R_{total}$, and $\phi_{i,j}$ are almost similar to the results obtained from the conventional iterative method for higher value of the training epoch i.e., 0.96 and 0.89 times accurate for 1,000 epoch, respectively.

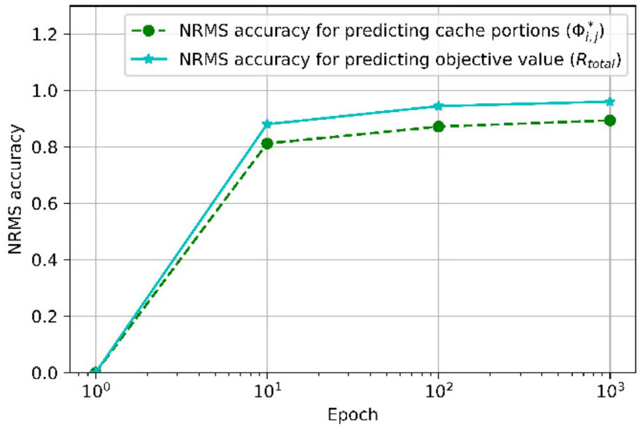

**Figure 14.** DNN model accuracy while predicting $\phi_{i,j}$ and $R_{total}$ for different values of the training epochs

### 4.3 Mathematical Quantification

For simplicity, let us assume that there is 2×1 F-RAN system. Referring to (P1), we can say, $\phi_{1,1}$, and $\phi_{2,1}$ are the possible optimal cache portion. Similarly $F_1$ is available F-AP capacity of the F-RAN system. The total value of $R_{total}$ for this scenario can be written as:

$$R_{total} = (\phi_{1,1} + \phi_{2,1})F_1 \tag{6}$$

Let $\phi^*_{1,1}$, and $\phi^*_{2,1}$, be the predicted value of cache portion using DNN method. The predicted value of $R_{total}$ can be written as:

$$R^*_{total} = (\phi^*_{1,1} + \phi^*_{2,1})F_1 \tag{7}$$

The NRMSE for predicting optimal cache portions can be written as [25]:

$$\sqrt{\frac{(\phi_{1,1} - \phi^*_{1,1})^2 + (\phi_{2,1} - \phi^*_{2,1})^2}{{\phi_{1,1}}^2 + {\phi_{2,1}}^2}} \tag{8}$$

Similarly, NRMSE for $R_{total}$ can be written as:

$$\sqrt{\frac{(R_{total} - R^*_{total})^2}{(R_{total})^2}} \tag{9}$$

Replacing the values of variables in (9) by (6) and (7), and mathematically simplifying it, we can write (9), as:

$$\sqrt{\frac{(\phi^e_{1,1})^2 + (\phi^e_{2,1})^2 + 2(\phi^e_{1,1})^2(\phi^e_{2,1})^2}{{\phi_{1,1}}^2 + {\phi_{2,1}}^2 + 2\phi_{1,1}\phi_{2,1}}} \tag{10}$$

where, $\phi^e_{1,1} = (\phi_{1,1} - \phi^e_{1,1})$, and $\phi^e_{2,1} = (\phi_{2,1} - \phi^*_{2,1})$, and are referred to as erroneous values.

If we compare (8) and (10), we can say that $NRMSE(\phi)$ is greater than $NRMSE(R_{total})$, as the addition of extra part in the numerator of (10) is lesser than the denominator of (10), i.e., $2(\phi^e_{1,1})^2(\phi^e_{2,1})^2 < 2\phi_{1,1}\phi_{2,1}$.

Conversely, DNN accuracy for predicting $\phi_{i,j}$ is lesser than predicting $R_{total}$, as DNN accuracy is $1 - NRMSE$.

## 5 Conclusion

This paper studied the optimal cache resource allocation process in F-RAN system using a machine learning technique. To achieve real-time performance, and realize low implementation complexity, DNN based framework has been proposed to predict the optimal portion of cache resource of F-RAN system such that it is fully utilized. Simulation results showed that our proposed DNN based cache resource allocation framework almost approaches to that of the conventional iterative method of cache resource allocation scheme. For future works, our DNN based framework should be used for the massive number of UEs and F-APs case using different DNN based algorithms.

## Acknowledgements

This work was supported by Post-Doctoral Research Program of Incheon National University in 2016.

## References

[1] X. Liu, Z. Qin, Y. Gao, Resource Allocation for Edge Computing in IoT Networks via Reinforcement Learning, https://arxiv.org/abs/03.01856.

[2] S. Bhandari, H. P. Zhao, H. Kim, An Efficient Scheduling Scheme for Fronthaul Load Reduction in Fog Radio Access Networks, *China Communications*, Vol. 16, No. 11, pp. 146-153, November, 2019.

[3] X. Huang, G. Xue, R. Yu, S. Leng, Joint Scheduling and Beamforming Coordination in Cloud Radio Access Networks With QoS Guarantees, *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 7, pp. 5449-5460, July, 2016.

[4] H. Dahrouj, A. Douik, O. Dhifallah, T. Al-Naffouri, M. Alouini, Resource Allocation in Heterogeneous Cloud Radio Access Networks: Advances and Challenges, *IEEE Wireless Communications*, Vol. 22, No. 3, pp. 66-73, June, 2015.

[5] M. Peng, Y. Li, Z. Zhao, C. Wang, System Architecture and Key Technologies for 5G Heterogeneous Cloud Radio Access Networks, *IEEE Network*, Vol. 29, No. 2, pp. 6-14, March-April, 2015.

[6] M. Peng, S. Yan, K. Zhang, C. Wang, Fog-computing-based Radio Access Networks: Issues and Challenges, *IEEE Network*, Vol. 30, No. 4, pp. 46-53, July-August, 2016.

[7] H. Zhang, Y. Qiu, K. Long, G. Karagiannidis, X. Wang, A. Nallanathan, Resource Allocation in NOMA-Based Fog Radio Access Networks, *IEEE Wireless Communications*, Vol. 25, No. 3, pp. 110-115, June, 2018.

[8] G. Rahman, M. Peng, K. Zhang, S. Chen, Radio Resource Allocation for Achieving Ultra-Low Latency in Fog Radio Access Networks, *IEEE Access*, Vol. 6, pp. 17442-17454, February, 2018.

[9] S. Park, O. Simeone, S. Shamai Shitz, Joint Optimization of Cloud and Edge Processing for Fog Radio Access Networks, *IEEE Transactions on Wireless Communications*, Vol. 15, No. 11, pp. 7621-7632, November, 2016.

[10] T. D. Tran, L. B. Le, Joint Resource Allocation and Content Caching in Virtualized Multi-cell Wireless Networks, *IEEE Global communications Conference*, Marina Bay, Singapore, 2017, pp. 1-6.

[11] A. Khreishah, J. Chakareski, A. Gharaibeh, Joint Caching, Routing, and Channel Assignment for Collaborative Small-cell Cellular Networks, *IEEE Journal on Selected Areas in*

*Communications*, Vol. 34, No. 8, pp. 2275-2284, August, 2016.

[12] L. Tang, X. Zhang, H. Xiang, Y. Sun, M. Peng, Joint Resource Allocation and Caching Placement for Network Slicing in Fog Radio Access Networks, *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, 2017, pp. 1-6.

[13] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, A. X. Liu, Dynamic Resource Allocation for Load Balancing in Fog Environment, *Wireless Communications and Mobile Computing*, Vol. 2018, pp. 1-15, April, 2018.

[14] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, Z. Han, Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching, *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1204-1215, October, 2017.

[15] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, N. D. Sidiropoulos, Learning to Optimize: Training Deep Neural Networks for Wireless Resource Management, *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, 2017, pp. 1-6.

[16] C. Zhang, P. Patras, H. Haddadi, Deep Learning in Mobile and Wireless Networking: A Survey, *IEEE Communications Surveys & Tutorials*, Vol. 21, No. 3, pp. 2224-2287, Third Quarter, 2019.

[17] M. Yan, G. Feng, S. Qin, Multi-rat Access Based on Multi-agent Reinforcement Learning, *GLOBECOM 2017-2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-6.

[18] J. He, L. Li, J. Xu, C. Zheng, ReLU Deep Neural Networks and Linear Finite Elements, http://arxiv.org/abs/1807.03973.

[19] S. Bhandari, H. P. Zhao, H. Kim, J. M. Cioffi, An Optimal Cache Resource Allocation in Fog Radio Access Networks, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2063-2069, December, 2019.

[20] C. E. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*, http://arxiv.org/abs/1811.03378.

[21] A. Farzad, H. Mashayekhi, H. Hassanpour, A Comparative Performance Analysis of Different Activation Functions in LSTM Networks for CLassification, *Neural Computing and Applications*, Vol. 31, No. 7, pp. 2507-2521, July, 2019.

[22] S. Ruder, An Overview of Gradient Descent Optimization Algorithms, http://arxiv.org/abs/1609.04747.

[23] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, http://arxiv.org/abs/1412.6980.

[24] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Quek, H. Shin, Enabling Intelligence in Fog Computing to Achieve Energy and Latency Reduction, *Digital Communications and Networks*, Vol. 5, No. 1, pp. 3-9, February, 2019.

[25] A. Botchkarev, *Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology*, http://arxiv.org/abs/1809.03006.

## Biographies

**Sovit Bhandari** received his Bachelor's degree from Kathmandu University, Nepal, in 2016. He worked as a Core Network Engineer at Huawei Technologies Nepal Pvt. Ltd. from Jan 2018 to Aug 2018. He has been working as a Graduate Research Assistant at Incheon National University, South Korea since Sept 2018. His research interests include radio resource management, 5G and beyond mobile communication, machine learning, and internet of things.

**Hoon Kim** has been working as an Associate Professor at Department of Electronics Engineering, Incheon National University, South Korea, since 2008. His research interests include radio resource management, optimization techniques, 5G mobile communication systems, machine learning, and internet of things. He is a Member of KICS, IEIE, IEEE, and IEICE.

**Navin Ranjan** received his Bachelor's degree from Kathmandu University, Nepal, in 2016. He is currently working as a Graduate Research Assistant at Incheon National University, South Korea. His research interest include wireless communications, computer vision and reinforcement learning.

**Hong Ping Zhao** received his Bachelor's degree from Jilin Jianzhu University, China, in 2016. He has been working as a Graduate Research Assistant at Incheon National University, South Korea since Mar 2018. His research interests include machine learning, big data and networking.

**Pervez Khan** worked as a Post-Doctoral Researcher at Incheon National University, South Korea from 2016-2018. His research interests include wireless communications, wireless sensor networks, wireless ad-hoc networks, wireless body area networks, 5G networks, machine learning, fuzzy logic and MAC protocol design.