Application of Deep Transfer Learning for Optimal Wireless Beam Selection in a Distributed RAN

Chitwan Arora and Abheek Saha
Hughes Systique Corporation,
Gurgaon, India,
email:*chitwan.arora@hsc.com,abheek.saha@hsc.com*

***Abstract*—This paper continues previous explorations in the area of deep learning applications in the field of cellular wireless networks, specifically the problem of identifying optimal beams in a highly directional urban environment, using topographical data. In our previous work, we have studied the problem and demonstrated how deep-learning can be used on static topographical data for prediction of optimal beams. In this paper, we show a potential architecture for realization of the same for a network of nodes in a given area, taking into account challenges of computational complexity, response time and the inherent architecture of the next generation RAN. This is achieved by using *deep transfer learning* as a way of translating between a global feature space inherent to the coverage area and local variations thereof, specific to the location of each radio-unit.**

Keywords*—*Transfer Learning; Deep Learning; Beam prediction; Distributed/Cloud RAN

I. INTRODUCTION

It is well recognized that Deep Learning (DL) is one of the foundational technologies for 5th generation cellular networks, especially in the problem of beam selection and channel estimation in higher frequency bands (mmwave) for urban environments where the radio-environment is highly directional. The problem of urban canyons and shadowing due to buildings is well known [1]. One of the most promising technologies to deal with this problem is the use of machine learning; in this approach, we use Light Detection and Ranging (LIDAR) or Global Positioning System (GPS) maps of a given urban topology to determine the wireless propagation capabilities of the coverage area. It is premised that using deep-learning, we can radically speeden up the process of optimal beam selection for any given User Terminal (UT), if we know its position. To this end, the International Telecommunications Union (ITU) organized a competition in 2020 [2] to explore deep-learning approaches on a multitude of real-world data. The authors participated in this competition and our approach was recognized as achieving 70% accurate prediction of the top-5 beams for a UT in any position in the coverage region. Other competitors showcased solutions, which yielded more than 90% accuracy.

Given that we are already achieving good results using deep learning, it is time to consider the next step of practical deployment of these technologies in the field. It is here that we come up against the biggest engineering challenges. Deep learning algorithms are well known to be prodigious consumers of both computing power and energy; further vast amounts of training data are required to adequately “train” the neural networks (NN). Running a multi-layer neural network in each individual radio unit (RU) for an urban geometry with multiple RHs per sq.km. of coverage area is clearly wasteful (both in terms of computing power as well as energy consumption) and furthermore, very expensive. What is required is to use the combined resources of multiple nodes operating in a common environment, in order to maximally utilize the expensive computing resources in the radio front-end. This is what we shall explore further in this article.

The rest of this paper is organized as follows. In Section II we review the problem in further detail, with a survey of the relevant literature. In Section III, we review the technologies of *transfer learning* and *multiview learning* as modifications introduced in the standard deep-learning methodologies and show how they are relevant to our environment. In Section IV, we present our analysis of the ITU-R dataset and show how it is relevant to the problem at hand. The simulations and corresponding results are work-in-progress and we hope to report our results in a subsequent revision of this paper.

II. PROBLEM DESCRIPTION

In Figure 1, we show the conceptual layout of a 5G cellular network in an urban environment. As we know, the 5G network architecture utilizes the *cloud Radio Access Network (RAN)* concept, where the RAN is disaggregated into the Radio Unit (RU), the Distributed Unit (DU) and the Core Unit (CU). The RUs are placed in diverse locations within the coverage region and are configured to create multiple radio-beams, focusing on specific hotspots. The RUs are connected to a smaller number of DUs, which provide the baseband processing. Finally, the CUs are deployed as a cloud and are designed to provide core signaling and control functions, including the radio-resource management and beam processing functions. ML algorithms can be hosted in various ways within the architecture, most notably within the RAN intelligent controllers (RIC). Some of these schemes have been explored in [3]. There are many possible configurations of this basic architecture, each pertaining to a different use case. A good overview is given in [4].

A. Network Operation

This system works as follows. When a user terminal enters the system, it detects a common signaling channel (low bandwidth, blind detectable) and then signals its position

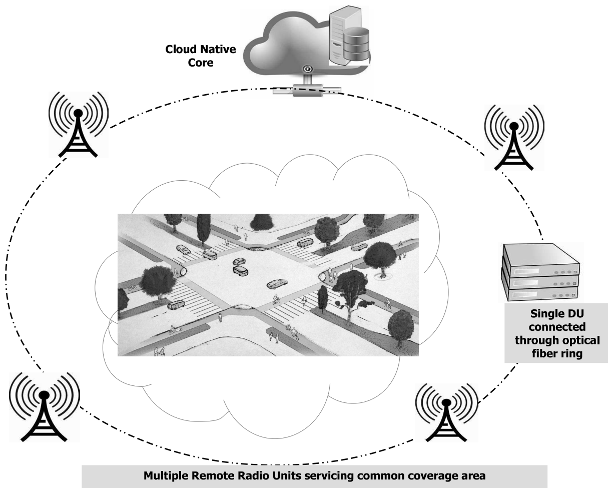

Figure 1. Conceptual View of Distributed RAN covering an urban location

to the network. The network responds to it by identifying a list of predicted top-N beams for it to use. The UT then successively attempts to setup a high-bandwidth data connection with the RU servicing each beam in the list till it achieves success. A beam corresponds to a precoding filter f on the transmitter side and a post-coding vector w on the receiver side. For a given channel matrix $W(p,i)$ corresponding to the channel experienced between the UT at position p and the base-station/RU i the received signal is given by (1).

$$r = \|w^T W(p,i) f\| \tag{1}$$

Obviously, the optimum beam is the one which maximizes the signal strength. We assume a large number of fixed beams, each identified by a tuple of $B \rightarrow \langle b, w, f \rangle$, where b is the beam-id. Each beam is serviced by a given RU (this is invisible to the UT, but important for the beam allocation problem, as we shall see later). The creation and configuration of the individual beams is done externally and available to the network as a database.

Clearly, our algorithm for predicting beams based on UT position has a local (RU specific) as well as a global element to it. Each RU sees an individual view of the environment based on the static topographical features relative to its position, as well as the position of the UT. These static features include high buildings, wide streets, overpasses and other similar features which could potentially either obstruct the signal or provide new reflective paths for it. On the other hand, the system as a whole has to take into account the alignment for all the RUs relative to a given position to determine the optimal beam list.

Matching the tiered nature of the problem, within the network as well, there are tiered layers of control. The *near realtime RAN intelligent controller (rt-RIC)* is typically placed in the DU and the *Non-Realtime RAN Intelligent Controller (nrt-RIC)* is typically placed in the core (Figure 2). The rt-RIC provides closed loop control at very tight latencies, typically focusing on local, high-speed control. The rt-RIC algorithms operate within tight constraints of compute

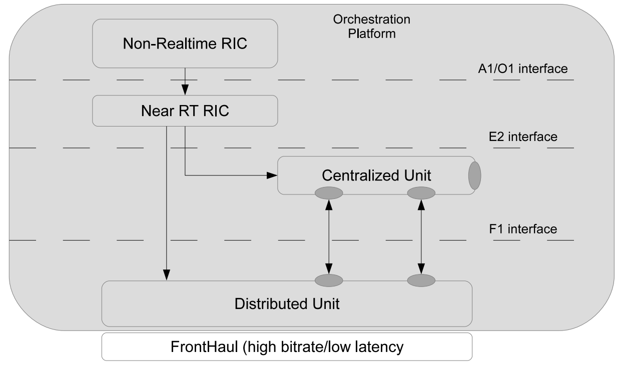

Figure 2. Conceptual View of GnodeB in ORAN

power and latency, in order to fit within the constraints of the DU environment. The nrt-RIC, on the other hand, provides slower control to the DUs using a relatively higher latency link. It has substantially larger compute and memory resources at its disposal, and can afford to take a global view of the network, due to its ability to store and process data from multiple DUs and RUs. This will subsequently play a role in the actual deployment of our ML based beam prediction solution, as we shall discuss in Section IV.

We now consider the ML algorithm. The input to the ML is the topographical information about the coverage region and labelled data corresponding to specific locations within the region and the beam/RU to which it maps. The format of the topographical data can take many forms, such as LIDAR scans [5] from the perspective of individual positions within the coverage area, with the reflections identifying local obstacles, along with GPS topographical data and images taken by wide-angle cameras. In other literature, topographical data is in the form of 3-d maps (for example, as provided by OpenStreetMaps) or in the form of GPS contour data [6][7]. The labelled data comprises of actual measurements from specific UTs at specific positions identifying the UT location and the empirically measured optimal beam id (or top N beams). This will be used to train the DL model.

The problem thus can be summarized as follows. Assuming that we have topographical information for the network coverage area, how do we build an RU specific view, as well as a global view of the propagation characteristics, and subsequently map this to optimal beam positions.

B. Literature Survey

There is a lot of recent literature in beam identification for mmwave communication. In [5], the problem is presented from the perspective of the UT attempting to compute the optimal beam list, based on LIDAR data. In [8], the authors present the problem in a vehicular perspective, using realtime LIDAR measurements to fingerprint a position relative to other vehicles in a given highway. In [9], the authors present a network oriented approach using coordinated beams and a centralized deep-learning model, similar to the problem we are addressing. However, the authors use directly measured signal strengths as the input. Each BS individually *learns* the system and the coordination is purely on the basis of selection, not in the model itself. In [10], the authors focus on the beam sweeping pattern itself as the output to the ML, as opposed to the beam prediction itself.

For our particular problem, we shall use the technology of *transfer learning* (TL). The area of transfer learning is an active field in DL theory; comprehensive surveys are given in [11][12]. The success of transfer learning is predicated on the ability to extract features in the preliminary part of the DL model; this problem is surveyed in [13][14]. The authors in [15][16] analyze the transferability of the extracted features, by selectively migrating some layers of a pre-trained DL model and comparing it to the performance of the same with randomized starting weights. This is extended in [17] into a concept of a *Joint Adaptation Network*, which will be used in the rest of our paper.

The problem of *multiview learning* is also an area of active research; see the surveys in [18]-[20]. The advantage of multi-view learning is that it enables significant simplification of the input data to be processed at individual nodes, by using commonality to remove redundancies and noise. Multi-view learning seems to be peculiarly applicable to a network node scenario as we have presented in Section II. However, there doesn't seem to be much published research in this domain.

III. Adaptations of Deep Learning to a distributed/hierarchical environment

If we analyze our problem from the TL angle, we see that we have a large number of independently operating nodes, each of which has to learn variations of the same data, i.e., the topography of the coverage region independently. It has been pointed out that we can make substantial savings by coordinating the learning procedure in some way. The two major technologies that we have considered are *transfer learning* and *multiview learning*, which are summarized in the following Subsections.

A. Transfer Learning

TL is a method whereby the information acquired by particular DL model can be *transferred* in suitably adapted form to another DL model. The transfer can be cross-domain or (as in our case), intra-domain. In our particular situation, we can have a central system which *learns* about the topology by processing all the path specific data available to the system and then transfers the learned model to individual RUs for their use. To implement the transfer scheme, we need to decide two things. First is what exactly to transfer and the second is how to accomplish it.

While there are many variants of transfer learning, one of the most appealing is that of *feature based* transfer learning. In this mode, the *features* of the data are extracted and learnt by the main ML and then transferred to the subsequent MLs; these MLs take this feature knowledge and further refine it. Features are fairly intuitive (especially when geographical data is involved) and it is possible to extract them efficiently from raw data. In our case, a feature could be a large building or other artefact that significantly impacts the propagation characterestics within the environment. It is well known that a DL based learning engine learns features in all its layers, starting with the most generic and moving towards the more specific; the problem then becomes selecting the layer within which the features are learnt at the optimal level of specificity. A second problem is the applicability of the features and how to use them in the target inference engine. In our particular environment, it is not just a matter of weighting the feature set, but rather of determining the applicability of a feature and its impact on the inference problem as a whole.

B. Multiview Learning

When we have multiple data sets from a single common environment (for example, RSSI readings for different UT positions from the perspective of multiple base-stations/RUs), a primary problem is the risk of over-fitting, especially if the data is simply concatenated together and fed into a single DL engine. This is the problem that multiview learning tries to avoid. On the other hand, simply separating out the data and treating them completely independent data-sets leads to insufficient training, especially if individual data-sets are small, or uneven. There are many different ways to implement multi-view training, each of which focusses on a different aspect of the problem. Co-training looks at maximizing the agreement between different views, whereas multi-kernel learning and subspace learning operate by implementing a certain structure on the underlying data-space.

IV. Architecture for Deep Adaptation Learning for the RAN beam selection problem

We now come to the realization of the beam-selection algorithm. In our earlier work [21], we described a generic realization as a single centralized inference engine as a Deep Neural Network (DNN) of 11 layers, using UT position as the index, in conjunction with the angles of arrival and departure and signal strength as labels to match optimal beams with UTs in other, unlabelled positions within the coverage area. As shown in the ITU-R challenge referenced above, it is possible to augment the data set with other parametric information. For example, LIDAR/image data is highly perspectival; by providing LIDAR based ranging data from individual BS locations, we can augment the empirical wireless information and get better training of individual inference engines.

In Figure 3, we show conceptually how the beam selection algorithm works. The algorithm is broken up into two tiers. The central algorithm learns the common features of the urban environment and transfers the DNN with pre-trained layers to the RU specific tier. This tier then augments the DNN with local data and computes the final inference engine. For global data, we use the GPS data indexed by position with labelled information about UTs which were able to acquire beams (with associated signal quality). Based on this, we can form a top level view of the predicted coverage for beams which is learned by the engine. In the local tier, we augment this information by using signal strength

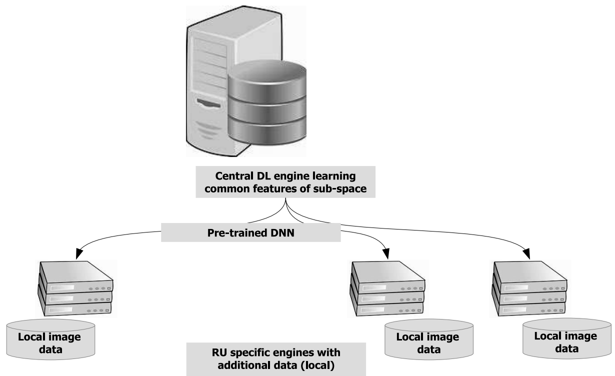

Figure 3. Hierarchical implementation of beam prediction DL engine

measurements (and LOS/NLOS computation) for individual UTs with respect to the position of the associated RU. This allows the RUs to create shortlists of predicted beams, which are then consolidated to form an overall list for advertising to the UTs. To improve the performance of the DNN at the RU, we can augment the central model by using local data specific to the RU. In our case, we use images of the horizon from the RU position. These images can highlight the presence of tall buildings or other obstructions in the surrounding area, which can be utilized to predict the possibility of LOS paths from different UT positions. Using self-supervised auto-encoders, we can identify the key feature-sets of each image and then match the endoded version to beam directions. By adding this information to the feature level data derived from the top level model, we hope to build accurate, but computationally simple local DNNs, which can be implemented relatively cheaply at the RU.

V. Conclusions

We have taken the baseline of the ITU-R data-set as described in [5] as the starting point as one of the few available empirical data-sets available in the field of wireless. The data-set provides GPS, LIDAR and imagery based data. As described above, we must start with the GPS based data as the global data-base. Primary analysis at the global level will be targetted at learning the features of the data-set. Once we have a good understanding of where these features are captured, we will consider the problem of moving the pre-trained DNNs to the RU and adding image data analysis to the same. This shall be explored in the final version of this article.

References

[1] M. K. Samimi and T. S. Rappaport, “3-d millimeter-wave statistical channel model for 5g wireless system design,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp. 2207–2225, 2016.

[2] “ITU Artificial Intelligence/Machine Learning in 5G Challenge,” https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx last accessed March,2021, 2020.

[3] H. Lee, J. Cha, D. Kwon, M. Jeong, and I. Park, “Hosting AI/ML workflows on O-RAN RIC platform,” in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.

[4] S. K. Singh, R. Singh, and B. Kumbhani, “The evolution of radio access network towards open-ran: Challenges and opportunities,” in *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6.

[5] A. Klautau, N. Gonzalez-Prelcic, and R. W. Heath, “LIDAR Data for Deep Learning based mmwave Beam-Selection,” *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[6] A. Klautau, P. Batista, N. Gonzalez-Prelcic, Y. Wang, and R. W. H. Jr, “5G MIMO Data for Machine Learning: Application to Beam-Selection using Deep Learning,” in *Proc of the information theory and application workshop, February*, 2018, pp. 1–9.

[7] M. Y. Takeda, A. Klautau, A. Mezghani, and R. W. Heath, “MIMO Channel Estimation with Non-Ideal ADCs: Deep Learning versus GAMP,” in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

[8] Y. Wang, M. Narasimha, and R. W. Heath, “Mmwave beam prediction with situational awareness: A machine learning approach,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[9] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,,” *IEEE Access, , –37 348, June*, vol. 6, pp. 37–328, 2018.

[10] A. Mazin, M. Elkourdi, and R. D. Gitlin, “Accelerating Beam Sweeping in mmwave Standalone 5G New Radios using Recurrent Neural Networks,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–4.

[11] F. Zhuang *et al.*, “A Comprehensive Survey on Transfer Learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[12] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on Deep Transfer Learning,” in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[13] S. Dara and P. Tumma, “Feature Extraction by using Deep Learning: A Survey,” in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1795–1801.

[14] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in Machine Learning,” in *2014 Science and Information Conference*, 2014, pp. 372–378.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in Deep Neural Networks?” *arXiv preprint*, 2014. Online:https://arxiv.org/abs/1411.1792 last accessed March,2021

[16] M. Long, Y. Cao, J. Wang, and M. Jordan, “Learning Transferable Features with Deep Adaptation Networks,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 97–105.

[17] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep Transfer Learning with Joint Adaptation Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 2208–2217.

[18] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” *arXiv preprint*, 2013. Online:https://arxiv.org/abs/1304.5634, last accessed: March,2021

[19] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: Recent progress and new challenges,” *Information Fusion*, vol. 38, pp. 43–54, 2017.

[20] S. Sun, “A survey of multi-view Machine Learning,” *Neural computing and applications*, vol. 23, no. 7, pp. 2031–2038, 2013.

[21] C. Arora and A. Saha, “AI based beam management for 5G (mmwave) at wireless Edge,” *Advances in Telecommunication, International Journal on*, vol. 13, no. 1&2, pp. 1 – 9, 2020.