A REPORT ON

MACHINE LEARNING APPLICATIONS FOR RESOURCE ALLOCATION IN MASSIVE MIMO SYSTEMS

In fulfilment of the course: EEE F366

Submitted by

PRATEEK MAHAJAN 2017A3PS0317P

Instructor-in-Charge: Dr. Syed Mohammad Zafaruddin

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
RAJASTHAN, 333031
JULY, 2020

ACKNOWLEDGEMENTS

I would like to thank Prof. V.K. Chaubey and the EEE dept at BITS-Pilani for providing the opportunity and resources to work on a project on the ML applications for resource allocation in massive MIMO systems.

I would also like to express my deepest gratitude towards Dr. Syed Mohammad Zafaruddin, who, in spite of his busy schedule, took time out to hear and guide me, and provided his full support in this project in every way possible.

BACKGROUND

Massive multiple-input multiple-output (MIMO), where a base station with many antennas simultaneously serves many users in the same time-frequency resource, is a promising 5G wireless access technology that can provide high throughput, reliability, and energy efficiency with simple signal processing. a distributed Massive MIMO system is one where a large number of service antennas, called access points (APs), serve a much smaller number of autonomous users distributed over a wide area. All APs cooperate phase-coherently via a backhaul network, and serve all users in the same time-frequency resource via time-division duplex (TDD) operation. There are no cells or cell boundaries. Therefore, we call this system "Cell-Free Massive MIMO".

In Cell-Free Massive MIMO, there is a central processing unit (CPU), but the information exchange between the APs (access points) and this CPU is limited to the payload data, and power control coefficients that change slowly. There is no sharing of instantaneous channel state information (CSI) among the APs or the central unit. All channels are estimated at the APs through uplink pilots. The so-obtained channel estimates are used to precode the transmitted data in the downlink and to perform data detection in the uplink.

The transmission of data from the APs to the users is known as downlink transmission and that from the users to the APs is called uplink transmission. Communication in such a system is done in 3 phases- uplink training, downlink transmission and uplink transmission.

In the uplink training phase, the users send pilot sequences to the APs and each AP estimates the channel to all users. The so-obtained channel estimates are used to precode the transmit signals in the downlink, and to detect the signals transmitted from the users in the uplink. Now, further for the aforementioned system to perform well, its resources need to be managed efficiently. The 2 main systems that need to be optimised in a MIMO system are pilot assignment and power control.

Pilot assignment essentially refers to the assignment of orthogonal pilots for the both the uplink and downlink transmission of data according to the channel coefficients between the user and the AP. Generally, greedy pilot assignment is used for the above.

On the other hand, power control essentially refers to assignment of powers/magnitudes to various user equipments chosen to achieve the optimum rate of data transfer (without loss of signal quality).

EXPLANATION OF THE PAPER

[2] essentially proposes an energy efficient power control approach that uses an artificial neural network to estimate the optimal relationship between the system propagation channels and the optimal power allocation rule. The approach also relies on an improved branch-and-bound algorithm that allows the offline generation of a large amount of training data with affordable complexity.

The approach essentially tries to maximise the energy efficiency of the communication systems and uses the following formula to calculate energy efficiency-

$$\mathrm{EE}_i = \frac{B \log\left(1 + \frac{\alpha_i p_i}{1 + \sum_{j \neq i} \beta_{i,j} p_j}\right)}{\mu_i p_i + P_{c,i}}$$

Therefore, effectively, we can say that the ANN seeks to solve the below optimisation problem-

$$\max_{\boldsymbol{p}} \sum_{i=1}^{L} w_i \frac{\log\left(1 + \frac{\alpha_i p_i}{1+\sum_{j \neq i} \beta_{i,j} p_j}\right)}{\mu_i p_i + P_{c,i}}$$
$$\text{s.t.} 0 \leq p_i \leq P_i, \quad \text{for all } i = 1, 2, \ldots, L,$$

Where $\alpha_i = \|h_{a(i),i}\|^2/\sigma_i^2$ $\beta_{i,j} = |h_{a(i),i}^H h_{a(i),j}|^2/\|h_{a(i),i}\|^2 \sigma_i^2$ $h_{a(i),j} \in C^{nR}$ is the channel from UE j to BS a(i).

Further, B is the communication bandwidth, L is the number of UEs, μ_i is the inefficiency of UE i's power amplifier and $P_{c,i}$ is the total static power consumption of UE i and its associated BS.

Now, essentially, the proposed ANN-based power allocation depends on 2 main tasks:

1. computing the power allocation vector using the trained ANN
2. building the training set and processing it to train the ANN.

However, it is important to note that the training set can be generated *offline*. Thus, real-time constraints do not apply and a much higher complexity can be afforded.

The training set can be generated with the help of the below formula-

$$\widetilde{p}_i^{(k)} = \frac{1}{\widetilde{\alpha}_i}\left(\frac{\frac{\widetilde{\alpha}_i}{\mu_i}P_{c,i} - 1}{W_0\left(\left(\frac{\widetilde{\alpha}_i}{\mu_i}P_{c,i} - 1\right)e^{-1}\right)} - 1\right)$$

Where W_O (·) is the principal branch of the Lambert W function.

(other symbols have the same meanings as above)

Once, the training set is generated, the ANN can be brought into the picture.

It essentially takes a vector a = (α_i ,$\beta_{i,j}$, P_{max}) as input and seeks to give an output of the optimal p_avg values of the the UEs. (Note that P_{max} essentially contains the maximum allowed power values of each of the UEs)

However, since the values of the above vector are generally very high, they are normalised according to the below equation-

$$\mathcal{S}_T = \{(\log_{10}\tilde{\boldsymbol{a}}_n, \max\{-20, \log_{10}\tilde{\boldsymbol{p}}_n^*\}) \,|\, n = 1, \ldots, N_T\}$$

Where the vector a is the input to the ANN and p is the output of the ANN.

The proposed ANN-based solution of is implemented through a feedforward ANN with K + 1 fully- connected layers, having K = 5 hidden layers with 128, 64, 32, 16, 8 neurons, respectively. Specifically, the first hidden layer has an exponential linear unit (ELU) activation, to compensate for the logarithmic conversion in the training set. The following hidden layers alternate ReLU and ELU activation functions, while the output layer deploys a linear activation function.

WORK DONE BY ME

In recent years, power control has become an extremely big topic of research in 5G systems. Multiple methods have been proposed to do the above though the application of machine and deep learning to do the above has become especially popular over the past few years.

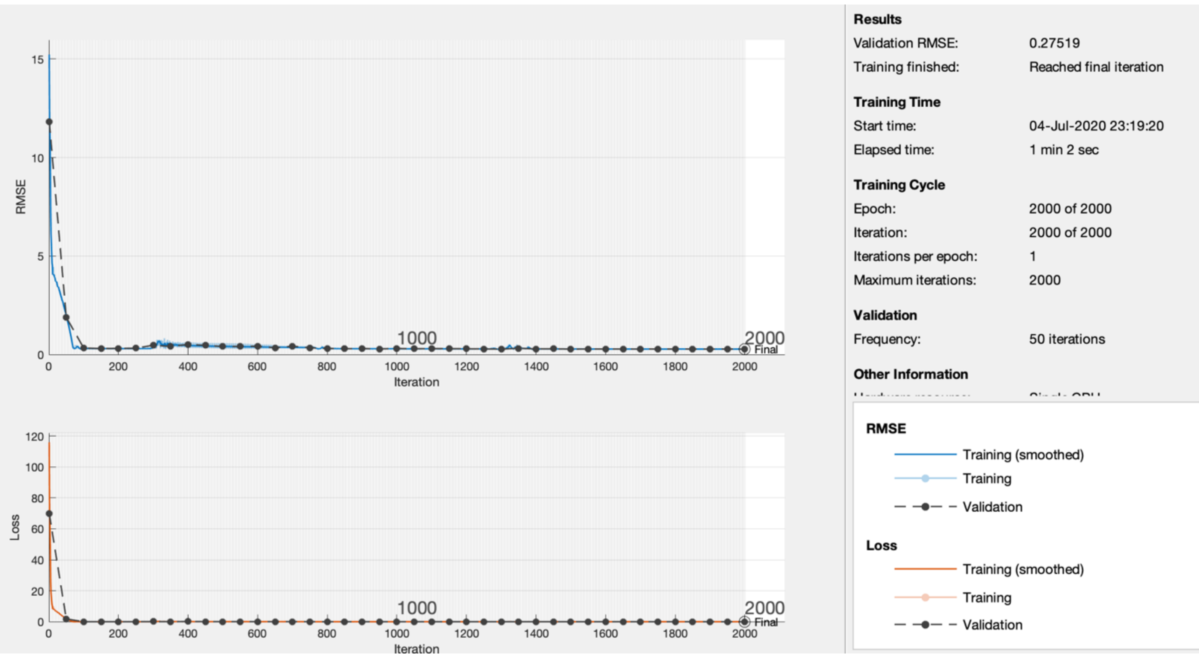
All such algorithms however, firstly require an accurate representation of the channel between the user in question and the AP. However, owing to small-scale and large scale fading, that's a difficult task to accomplish. To estimate our channel, we used the channel representation in [1] on page number 2. Essentially, we used the below equation to calculate path loss.

$$PL^{ABG}(f, d)[dB] = 10\alpha \log_{10}(d) + 10\gamma \log(f) + \beta + \chi^{ABG}$$

The channel coefficients were then calculated by taking the square root of the path loss. Note that d over here is in meters and f is in GHz. Further, the values of α, β and γ taken were 3.3, 17.6 and 2.0. Also, note that the last term is a gaussian noise distribution.

Once the path loss was implemented and the channel coefficients were obtained, I worked on implementing a paper that discussed a implementation of deep learning for power control in 5G communication networks (this paper can be found in [2]).

With respect to what has been explained in the above paper, I have compiled a training set to train the ANN on and have a basic, untested model of the ANN ready using the Deep Learning Toolbox offered on Matlab. The codes associated with the above have been submitted to Dr. Syed Mohammad Zafaruddin, but have not been included in this report. The ann has not been tested completely though the 2 below graphs can be observed to note its training progress.



REFERENCES

[1] S. Sun et al., "Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios," 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, 2016, pp. 1-6, doi: 10.1109/VTCSpring.2016.7504435.

[2] B. Matthiesen, A. Zappone, E. A. Jorswieck and M. Debbah, "Deep Learning for Real-Time Energy-Efficient Power Control in Mobile Networks," 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, 2019, pp. 1-5, doi: 10.1109/SPAWC.2019.8815516.

[3] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," in IEEE Transactions on Wireless Communications, vol. 16, no. 3, pp. 1834-1850, March 2017, doi: 10.1109/TWC.2017.2655515.