ECE 6604 : Project 3
DOA Estimation using DNN

Author : Parisha Joshi- parishamaheshj18@vt.edu

Introduction

Estimation of direction of arrival (DOA) from data collected by sensor arrays is of fundamental importance to a variety of applications such as radar, sonar, wireless communications, geophysics and biomedical engineering[1]. The conventional methods to derive DOA are mainly of two types, Delay-and-sum and sub-space based approach which were taught in course lectures. The novel technique to use DNN takes lesser computation and can be competitive to the conventional techniques in applications like radar and sonar where time can be a constrained. In this project we built a deep neural network model for estimating direction of arrival(DOA) of 2,3 and 4 signals. For Machine learning purpose the system used is macOS 2.6Ghz intel core i7 with Radeon Pro 555X 4 GB Intel UHD Graphics 630. The estimator was compared with MUSIC estimator and checked with CRLB values for both in Noiseless and SNR range from 0 to 10 dbs.

1 Problem 1

The first task was to develop a neural network with 0.01 degree of RMSE with full range(180 degrees) in the absence of noise. The neural network was trained using keras API with parameters given below in Table 1. The given matlab file generates a dataset for two different methods. 1) Covariance Matrix and 2) Rx samples.The first step to approach the solution is to build a naive network with two layers with 20 and 60 neurons accordingly. The The accuracy was 84 percent and RMSE value approximately 0.5 after 70 epochs. For the purpose of converging to the result better ADAM optimiser is used with Beta1 and Beta2 values set as 0.9 and 0.99. The Architecture then was improved as shown in Table 1 with different configurations of neurons in each layer.

Layer 1	Layer 2	Layer 3	Observation
64	-	-	Slow convergence,High RMSE,Distorted results
10	66	3	Slow Convergence , High RMSE
20	15	3	Overshoot
20	64	55	Fast convergence,high RMSE,Batchsize changed
20	64	55	Fast convergence,low RMSE,Learning rate changed,1000 epochs

The Tensorboard Graph of the final architecture is given in Figure 1.

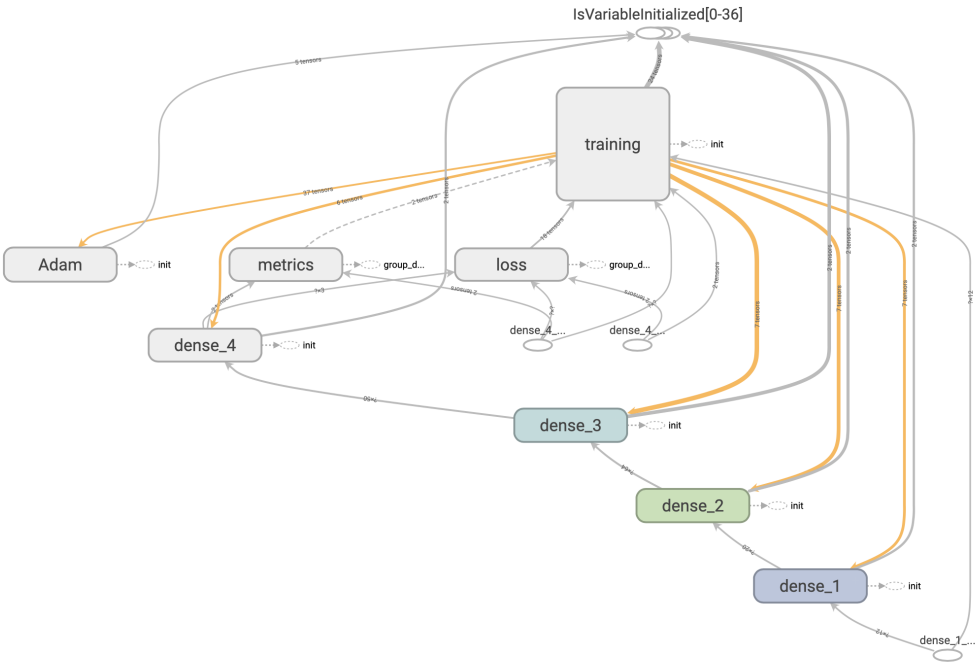


Figure 1: Final Architecture

1.1 3 Signals – CovMat – Range 1 – Noiseless – 20000 Samples – 1000 epochs

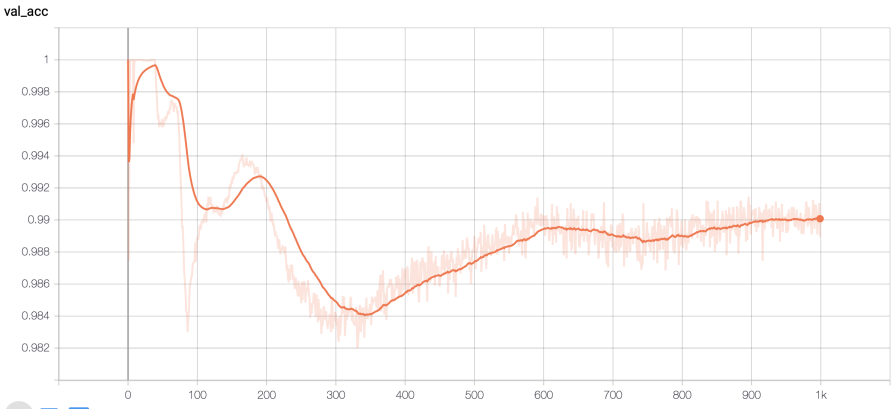


Figure 2: Validation Accuracy for CovMat, Range 1,1k epochs

From Figure 2 it is apparent that the accuracy value becomes unstable during the first couple hundred epochs. The reason might be the training phase being in progress. It glides down to stability

after 500 epochs at 99 percent. The model gives pretty good results with RMSE values in as shown in Figure 3. lowest RMSE value is 0.05 degrees. The curve shown in the figures are smoothed to show the direction of convergence of the average values. The inference from the first graph would be to increase training and samples in next iterations.

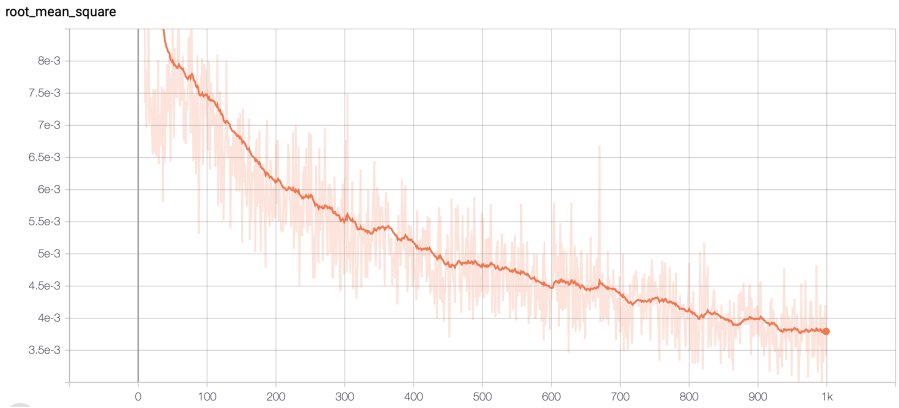


Figure 3: RMSE for CovMat, Range 1,1k epochs

1.2 3 Signals – RxSamp – Range 1 – Noiseless – 20000 Samples – 5000 epochs – Batchsize 1000

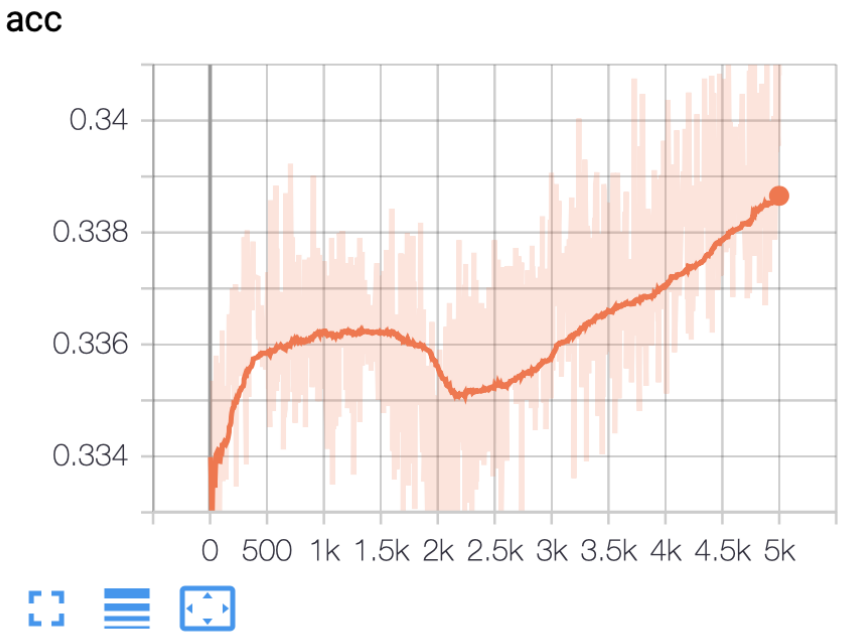


Figure 4: Accuracy for Rx Samples, Range 1,5k epochs

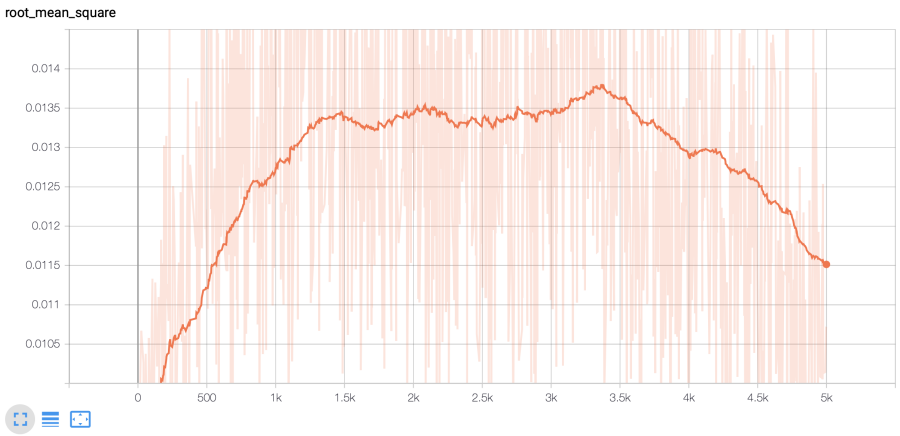


Figure 5: RMSE for Rx Samples, Range 1,5k epochs

Observing the Figure 4, The Training samples generated by RxSamp method needs a lot of training to perform as good as Covariance Matrix Samples. The batchsize was set to 1000 and the epochs to 5000. The accuracy value is lowest in the beginning at 33 percent and Root mean Squared error is 0.58 degrees. It is clear that CovMat Samples work much better with given Neural Network.

1.3 3 Signals – CovMat – Range 0.9 – Noiseless – 500000 Samples – 1000 epochs

For 0.9 range some changes in model has to be done. The model is changed so that it has 3 layers with 20, 64 and 50 neurons respectively. The Sample size is increased in order to experiment with data and the batch size is increased to 1000.The convergence to the root-mean-squared error is faster in the beginning for 300 epochs and slows down for the epochs after that. Which means that the model is working steadily towards the optimum value.The graph for this configuration is given in Figure 2. It is quite intuitive that increasing the number of samples surely helped for accurate training but the steadiness is not proportionate to the increase in number of samples. After a threshold the model convergence no more depends on the size of the dataset.The error graph shows that increasing the number of epochs might help converge to even smaller value of the RMSE.The current RMSE at the end of the training is around 0.0023 radians(0.1318 degrees). The graph for Accuracy seems to fluctuate over a range of 0.99 to 1 in the first few epochs but becomes steady after first 100 epochs.

1.4 3 Signals – CovMat – Range 0.9 – Noiseless – 500000 Samples – 5000 epochs

The increase in Training epochs helps to reach the optimum goal of 0.01 degree. The validation RMSE value glides down to minimum of 0.000552 Radians which is 0.03 Degrees. The final training accuracy is 99.39 percent and the final validation accuracy is 99.5 percent. The Figure 9 shows root mean squared error in radians with respect to the number of epochs. Final average RMSE value for training is 6.23e-4 equivalent to 0.03 degrees. The validation accuracy shown in Figure 8 given even lower value of error. Although does not change in degrees much. Resulting value being 0.03 degrees.

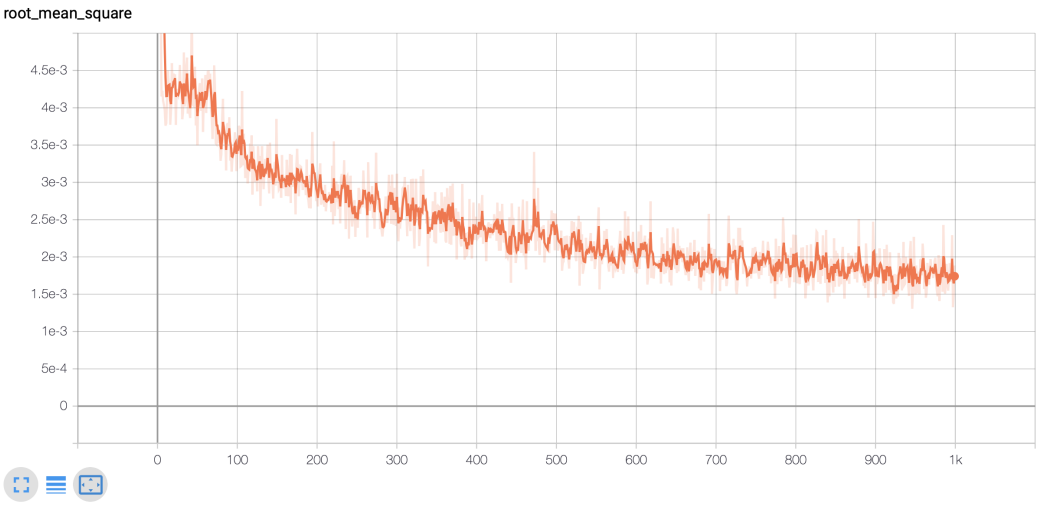


Figure 6: RMSE for CovMat 0.9 range

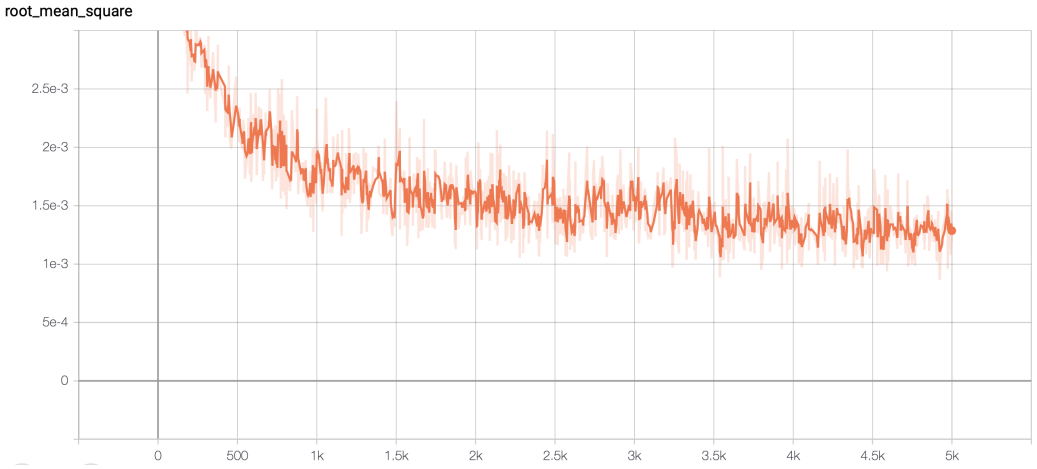


Figure 7: Training RMSE

Validation Accuracy value fluctuates for first 500 epochs, increases during next 1500 and becomes steady after 2k epochs.Which shows the change of weights during training phase.

1.5 3 Signals – RxSamp – Range 0.9 – Noiseless – 500000 Samples – 1000 epochs

For samples generated by RxSamp work poorly with given network architecture.It might need more training or more examples for better results and for that purpose the next subsection shows results with more epochs in training.

Conclusively, It takes approximately 3 hours to train the 5000 epochs with 1000 batchsize. The Covariance matrix method data worked best with the training given 99 percent accuracies and lowest RMSE values.Reason being CovMat method makes use of covariance matrix for generating feature

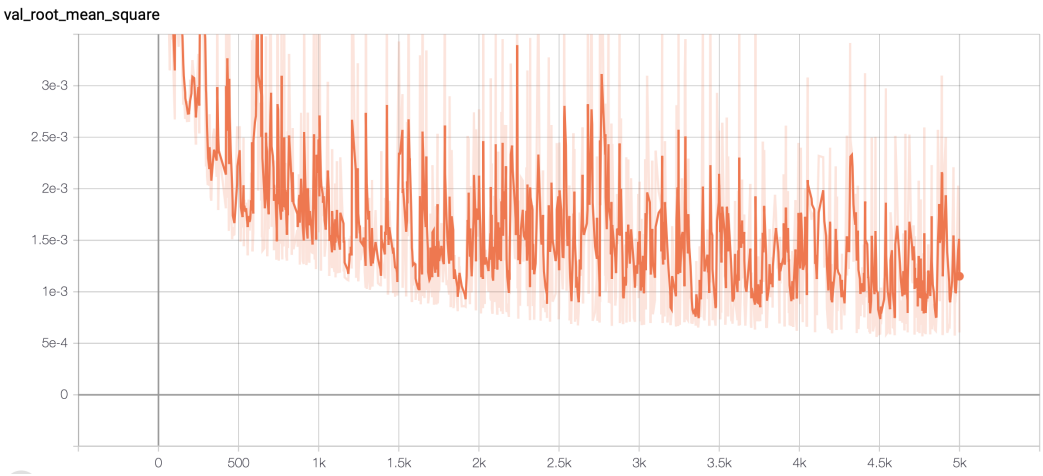


Figure 8: Validation RMSE

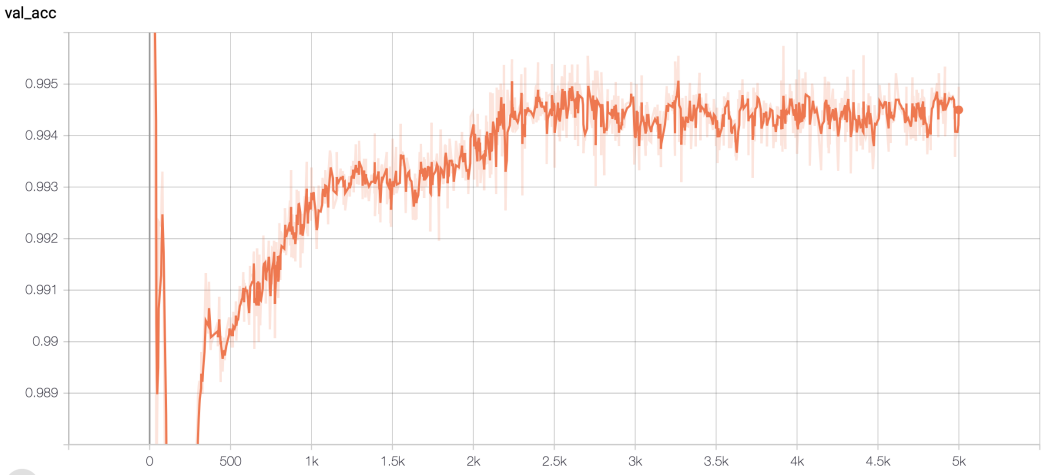


Figure 9: Accuracy

maps taking both real and imaginary parts in different columns for each map. For example, for 3 signals, the covariance matrix is 4-by-3. To generate feature map each element of the matrix maps to each element of the same matrix for generating Training set.It amounts to a covering larger subset of data than RxSamp method.

2 Problem 2

The Model was Trained for the range of SNR from 0db to 10db in the interval of 2. The CRLB was calculated from the formula given in Handouts which makes use of gaussian noise in signal to find the lower bound for figuring out if the estimator is performing well. The method can be used to first draw the boundary taking SNR 0-10db and generating data for them,Calculating CRLB values and plotting

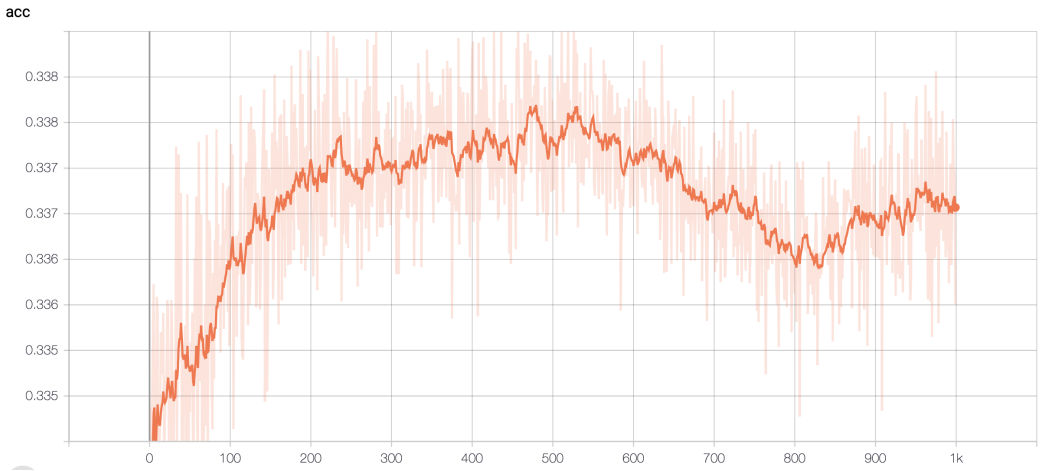


Figure 10: RxSamp 0.9 Range Accuracy

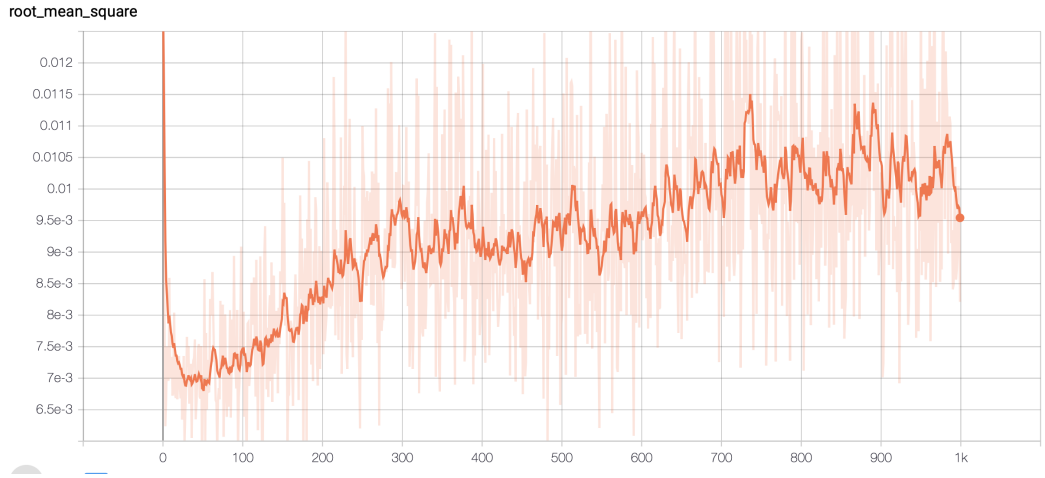
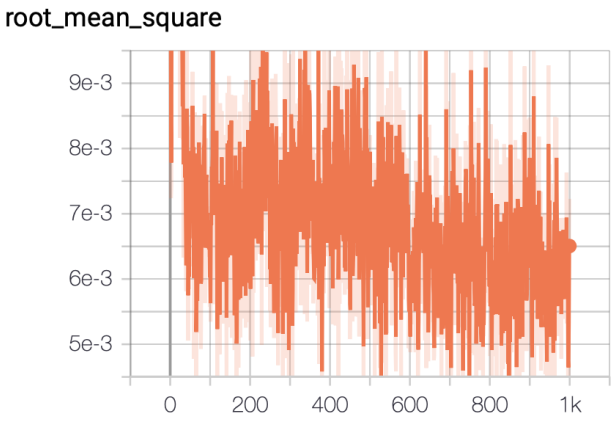


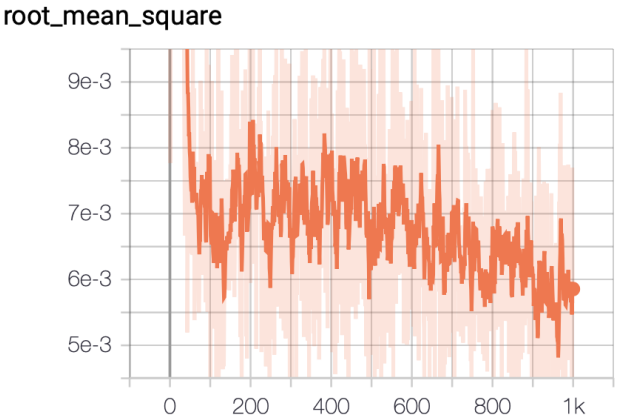
Figure 11: RxSamp RMSE 0.9 Range

them. Here neural network and MUSIC estimator can be taken as non-biased estimators.Variance of the estimation for non-biased estimators is equal to the inverse of Fisher Information Matrix, Which is nothing but the Mean Squared Error from the estimations. For the optimum performance the variance of theta, Here theta being Direction of Arrival, Should be greater than the CRLB value for that SNR. The RMSE graphs for SNR 0 db ,4db and 10 db over 1000 epochs and 20000 Training Samples generated with CovMat can be seen from Figures 12a,12b series.

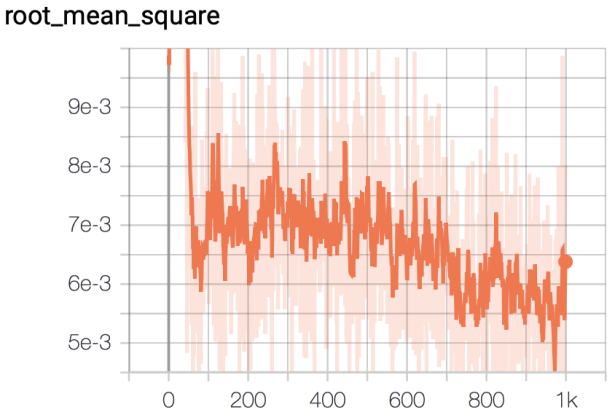
Observing from the RMSE values , the lowest value of error is reached when the noise is lowest (SNR 10 db). The training is model is not changed for training but it is however trained over all the datasets separately. The Figure 12 shows the comparison between Variance of thera for MUSIC and Neural Network Estimator for SNR ranging from 0-10db. It is worth noticing that with highest noise the estimators do not perform well. As SNR 10 shows the lowest noise, Where the Estimators'



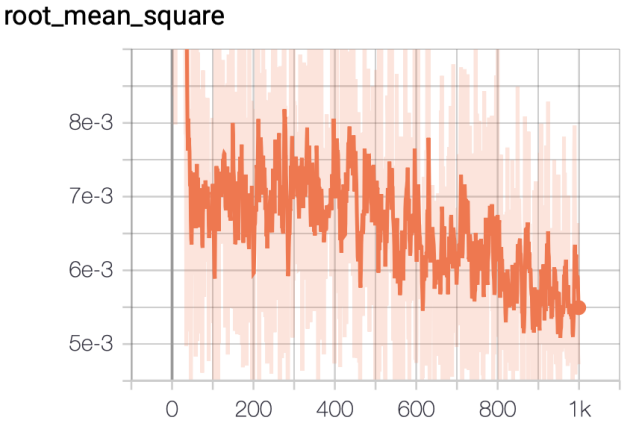
(a) 0 db SNR



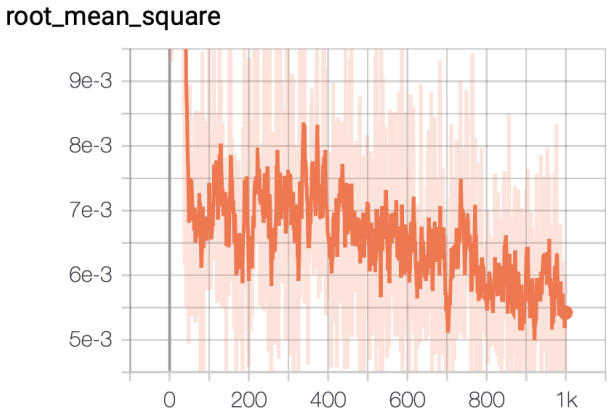
(b) 2 db SNR



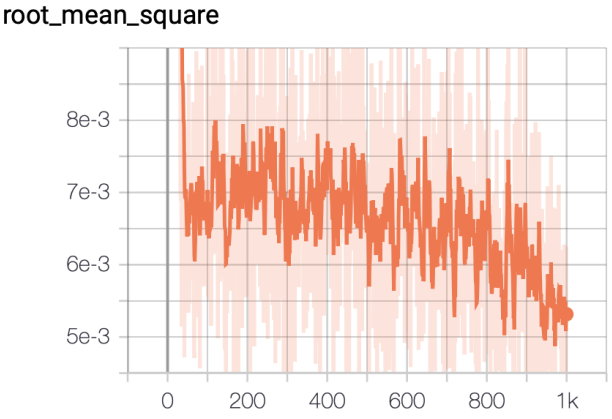
(a) 4 db SNR



(b) 6 db SNR



(a) 8 db SNR



(b) 10 db SNR

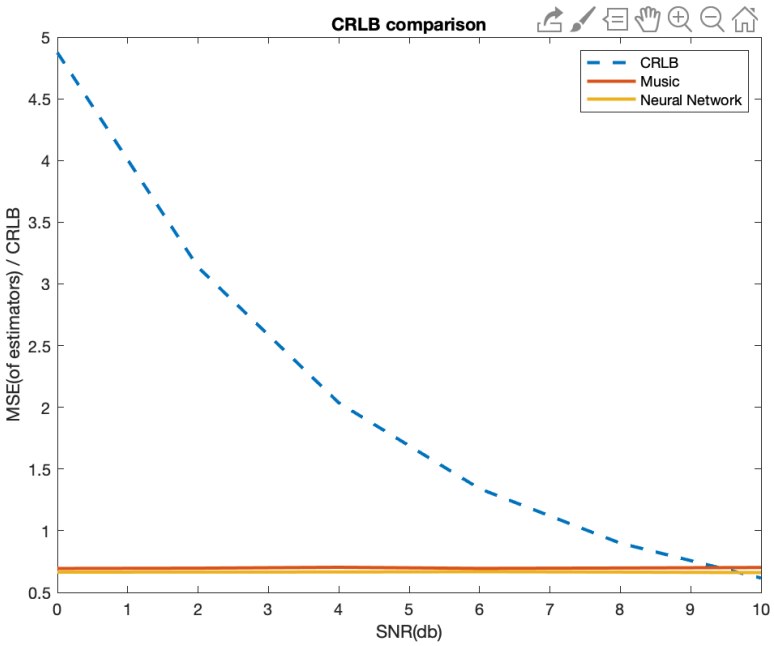


Figure 12: CRLB

MSE value exceeds the CRLB threshold, At the highest noise(0db), The estimator performance is not quite optimum.

The given MSE values are taken in the beginning of the Training of neural network as can be seen from Figure 13. The Mean squared value for neural networks is lower than MUSIC estimator.The network was needed to be trained for all the SNRs.Which is practical as the performance of model for the data with SNR different than what it was originally trained on, can seem deprecated. Most of the practical cases the organisation already have the dataset and noise ratios from the previous data collected. So training the model for different applications and on different SNRs is more practical.

3 Problem 3

To compare the performance of the created Deep neural network with conventional DOA estimator I made use of Phase array system toolbox of matlab which includes function using MUSIC algorithm to estimate the angle of arrival. The MUSIC algorithm is used for frequency estimation or direction of signal estimation.The approaches with maximum likelihood(ML) and Maximum Entropy (ME) are other conventional methods which have certain fundamental limitations[3] MUSIC estimates the frequency content of a signal or autocorrelation matrix using an eigenspace method[3].The observation is subdivided into two subspaces : Orthogonal and Noise subspace. The biggest eigen vectors corresponding to their eigen values form the orthogonal signal subspace whereas the smallest eigenvectors form a noise subspace. The algorithm makes use of grid of arrival angles to look for signals composed

with noise subspace. The resulting angle is the direction of arrival[2].
The accuracy for 100000 samples with SNR infinity(noiseless),3 Signals and 0.9 range the RMSE for MUSIC estimator was 39.96 percent. Where as the constructed Neural Network have 0.004 radians final RMSE,99.06 percent training accuracy and 98.89 percent validation accuracy. For comparing the two methods GenerateData.m file was modified with functions to generate covariance matrix given by sensorcov(d-lambda,targetangle,db2pow(-SNR)) and to generate DOA estimation with function musicdoa(covmat,numSignals). The performance of the MUSIC estimator can be improved by changing Standard Deviation of the data. As MATLAB sets default parameters for estimations, The neural network seems to be working better than MUSIC with Default Parameters. However, It is possible that MUSIC might exceed the performance of DNN if Standard deviation parameters are changed. if

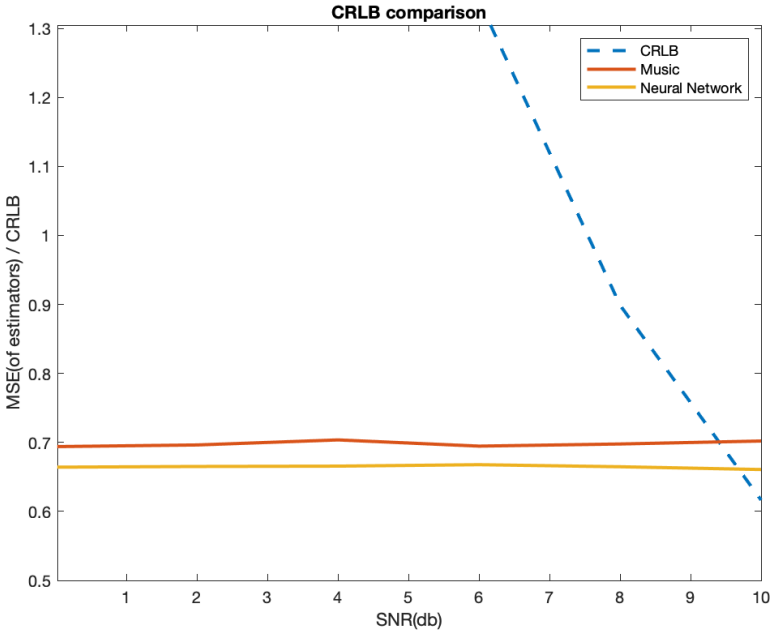


Figure 13: CRLB Comparison

4 Problem 4

Repeating the above procedure with only 2 signals this time turns out to be very informative. The same model cannot be used for 2 signals because of two main reasons. The technical and more logical reason is that the last layer in keras api contains 3 neurons in the previous model. Whereas training for two signals will require 2 output targets to train on. The second reason is that with two signals the the error space becomes much more broader. In simple terms, the model is more prone to making wrong estimation with limited subspace for making decision. This time the datasets are only generated with CovMat method and for 0.9 range as we have observed that the model works better

with those parameters.The training time taken was 14320 seconds(approx 4 hours) for 5000 epochs.

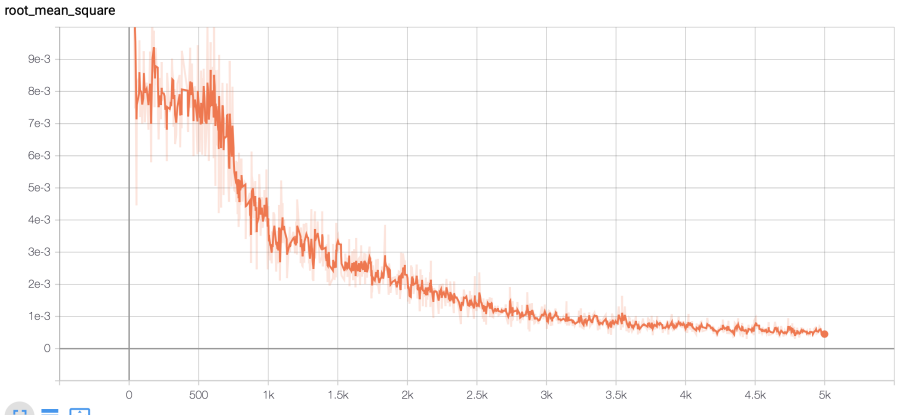


Figure 14: RMSE for 2 signals 0.9 range

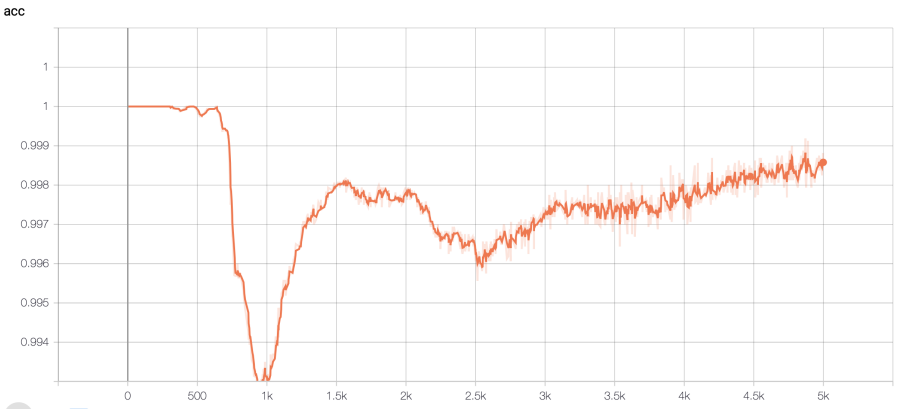


Figure 15: accuracy for 2 signals 0.9 range

Inferring from figure 14 and 15, As expected the model works best for RMSE and accuracy values. The model was not changed as everything except the output signals in the datasets are unchanged. The minimum RMSE is 0.01 Degrees. The resulting estimator is not able to reach the CRLB. Although comparing Mean Squared Error surely gives better picture neural network competence.

5 Problem 5

Running the model for 4 Signals took a lot of changes in the architecture. If we use the same architecture the results are as shown in Figure 17 18.The Stability of model is questionable even thought the accuracy value is much steadier than for 3 signals.

For improving the model for 4 signals the training size is increased to 100000 Samples. After taking more sample and changing values of layers to Layer-1 – 24 neurons, Layer-2 – 68 neurons and

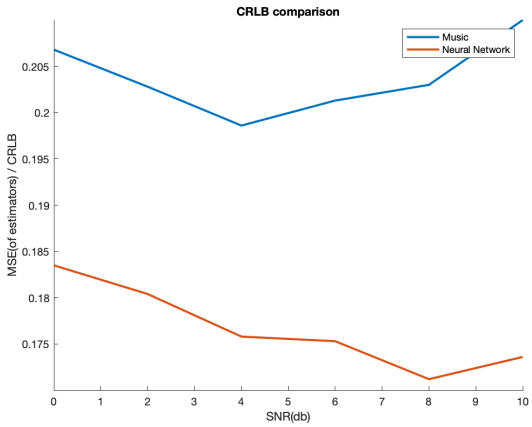


Figure 16: MUSIC vs DNN for 2 signals, Difference is 0.015 radians, proving the competence of neural network with MUSIC estimator

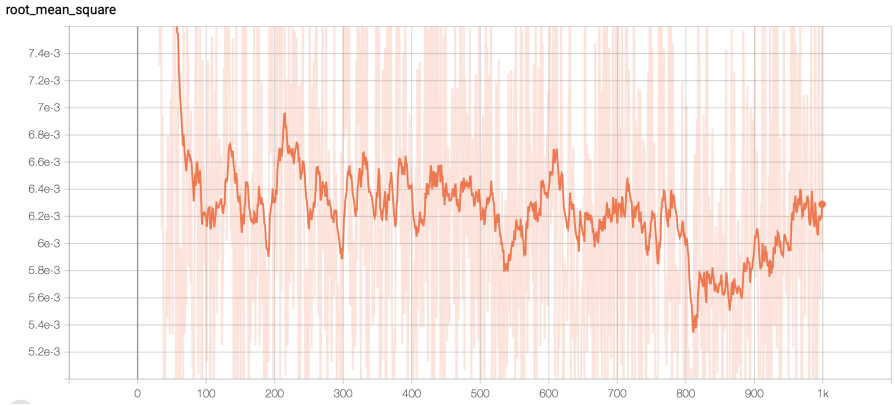


Figure 17: DNN RMSE for 4 signals

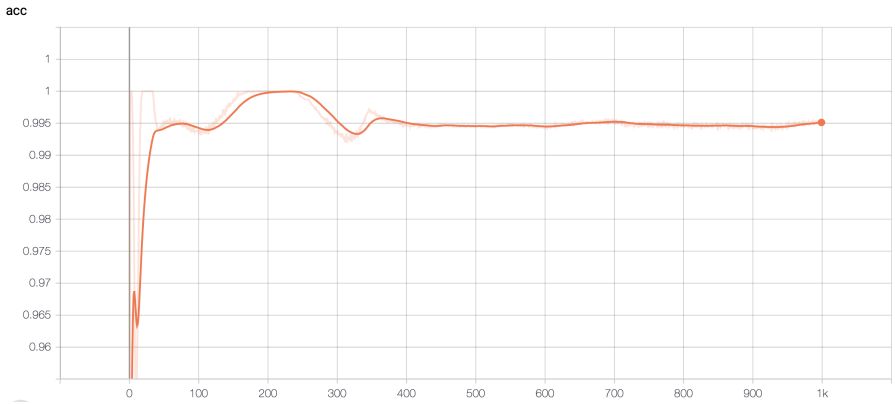


Figure 18: DNN Accuracy for 4 signals

Layer-3 – 59 neurons, The results are given in figures 19 and 20. Here increasing the Samples makes sense as numbers of signals are increased. So the broad range of features are mapped in Covariance Matrix in width. Neural Network and Deep learning by Michael Nielsen states that the number of sample should be increased in squared proportion to the increased number of features [6]. 5000

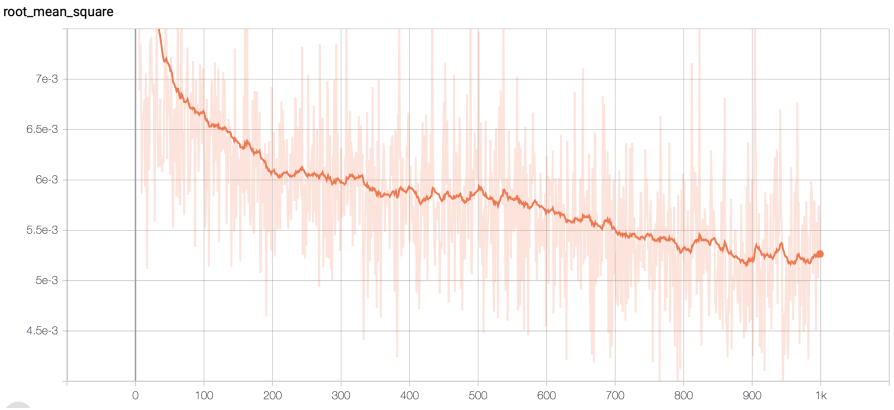


Figure 19: DNN Training RMSE for 4 signals, More Samples

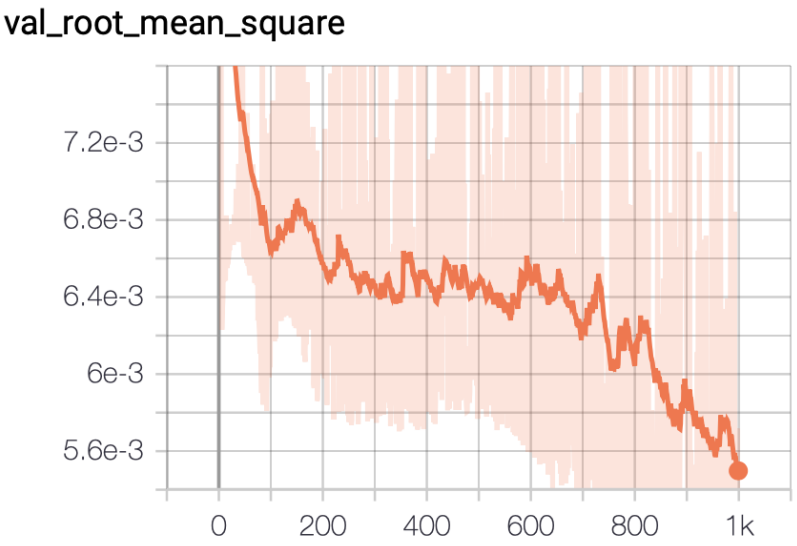


Figure 20: DNN Validation RMSE for 4 signals,More Samples

epochs and 100000 samples works best with 4 signals.However Accuracy value seems to drop to 98 percent, The neural network has much steeper slope for RMSE.

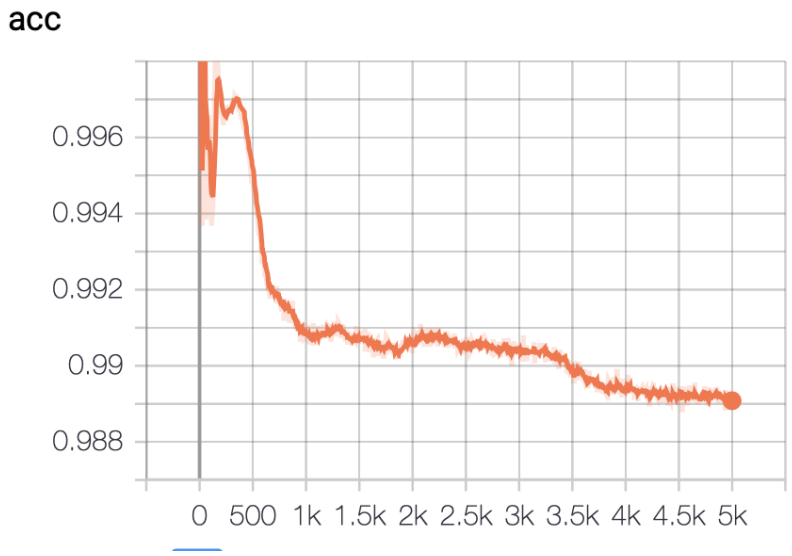


Figure 21: DNN Accuracy for 4 signals,More Samples,More Training

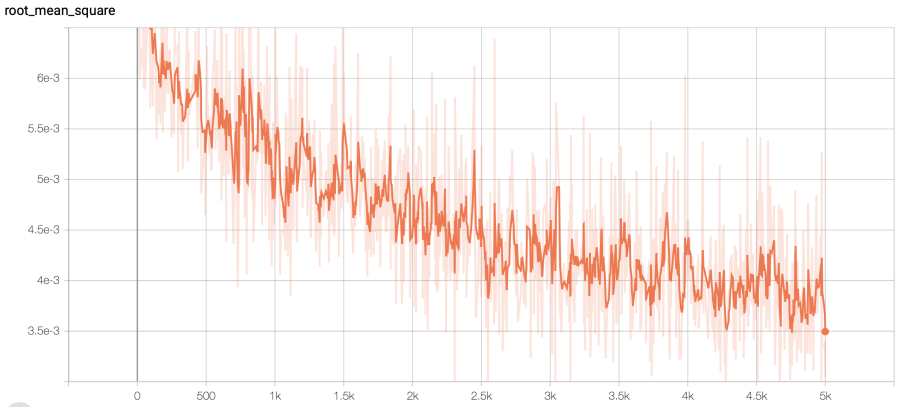


Figure 22: DNN RMSE for 4 signals,More Samples,More Training

6 Conclusion

Conclusively, The neural network for 2, 3 and 4 signals were architectured and tested. The estimator works very competitively with MUSIC estimator. However is not able to exceed the CRLB. Applying autoencoders in the application might have helped in improving the RMSE performance which shall be taken in application for future work. The SNR values from 0 to 10 db were tested on Datasets of RxSamp and CovMat both, tried with various combinations of epochs and batchsize and Samplespace. The best model turned out to be one with moderate amount of samples, 5000 epochs and 1000 batchsize. The 0.03 degrees RMSE was reached with 500k Samples, 5000 epochs and 0.9 range with CovMat and RxSamp datasets.

7 MATLAB files

1)CRLB.m file contains the values calculated and documented manually after every run for Mean-CRLB.
2)Musicold.m file contains the code for implementing MUSIC estimator for signals. The number of Signals need to be changed manually for determining performances with different number of signals.
3)Folders with CovMat and RxSamp datasets contains RunThis.ipynb files to run in jupyter with anaconda environment.

References

[1] Chapter 14 - DOA Estimation Methods and Algorithms
`https://www.sciencedirect.com/science/article/pii/B978012411597200014X`

[2] MUSIC and the Cramer-Rao lower bound
`https://aas.aanda.org/articles/aas/full/1999/11/ds1489/node6.html`

[3] What is MUSIC Estimator
`https://en.wikipedia.org/wiki/MUSIC(algorithm)`

[4] Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network
`Sharath Adavanne, Archontis Politis, Tuomas Virtanen`

[5] Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections
`Zhang-Meng Liu ; Chenwei Zhang ; Philip S. Yu`

[6] Neural Network and Deep Learning
`Michael Nielsen, June 2019`