Channel matrix forecasting with deep learning methods

Ilya Selnitskiy [1] **Ivan Oseledets** [1]

Abstract

The report describes various methods and approaches aimed on forecasting the complex matrix of the wireless channel. Predictions of channel state is crucial for determining further data rate which particular user can receive from Base Station. Many classical methods from statistics are used for such purposes or some algorithms which will predict underestimated state. Nowadays more deep learning algorithms are used for solving tome series forecasting problem. In this project we will evaluate them with classical algorithms and suggest some methods of improvements.

1. Inroduction

Time series forecasting is an important area of machine learning that is often neglected. However there exist lots of application that involve a time component. The telecommunications is on of the areas where time series forecasting takes place. The problem arise when we want to predict the further state of the channel having some measures obtained from pilot signals. The aim of such prediction is to choose a appropriate modulation and coding scheme which (MCS) will determine the data rate of a wireless connection. The channel state matrix is a complex matrix $X \in C^{m,n}$, where m is a number of users and n is a number of antennas. We should admit that in our context user can be a particular device with antenna. Nowadays, the base stations determine the channel state by doing some measurements with special pilots signal. However is too computationally expensive. Therefore, to reduce the computation costs on base stations we study the d regression approaches to forecast channel matrices $X_{K+1}, \ldots, X_{K+T}$ based on the already observed matrices $X_1, \ldots, X_K$. This problem is a classical time series forecasting problem, but here we need to predict not one time series bu a set of them. Also, one of the problem is the complex numbers which are the elements of out matrix, the complex nature of such matrix is due to fact what the signal is characterised with some module and phase. Also, we should take into account that the signal from several antennas are dependent and we should take into account this dependence.

2. Related works

There exists lots of works aimed on prediction channel state. Some works like (Arnold et al., 2019) predict channel state information (CSI), which is not precise values of channel, but some information which describes it. The authors suggest use normalise mean square error which better describes the ability of model. The Block Error Rate metric is ASL evaluated, such metric describes how precise in terms of channel error we can predict the future state. Also, Gaussian white noise (AWGN) as regularisation to the training samples to prevent over fitting. However predicted CSI do not fully characterise the channel. The authors of (Yang et al., 2019)suggest to use a simple feed forward neural net. They found out that 3-layer neural net can describe the channel and based on such idea thy constructed a new SCNet.Authors suggest to work with complex data which better describes channel. However as in a previous work authors predict only a CSI which does not totally describe the channel. In (Jiang & Schotten, 2019) authors suggested to use RNN to predict the matrix,however the simple conversion of matrix in vector is not a good approach due to loss of spatial sub-carrier information.

Th main assumption that we are going to investigate is that channel matrices in the considered time interval $[t0, t1]$ are not arbitrary but lie in some matrix manifold. So the essential part of time was aimed on studying the matrix manifolds. In (Zhang et al., 2018) discussed the properties of Grassmannian manifolds total approach of learning on it and some extra literature to study. In (Huang et al., 2018) was suggest deep network architecture in the context of Grassmann manifolds and generalised backpropagation rule to train the proposed network with deriving an connection weight update rule on a specific Riemannian manifold. Authors of (Masci et al., 2015) suggest to use new geodesic convolutions to achieve higher results. In (Chen et al., 2017) were propossed some new layers that are able to learn the hidden

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Ilya Selnitskiy <i.selnitskiy@skoltech.ru>.

Final Projects of the Machine Learning 2020 Course, Skoltech, Moscow, Russian Federation, 2020.

manifold, however they solved a classification problem and thesee layers could not be applied to regression task. As in previous work (Li et al., 2020) authors solve classification problem and used graph embedding on constructed manifold. There also existed not deep learning techniques to analyse time series hidden manifold. In (Erem et al., 2016) authors analysed bioelectric signals with help of . Laplacian eigenmaps combined with time embedding. But the authors did not solve the regression problem which is studied in this project.

The aim of the research is forecast the channel matrix. One of the possible ways of doing it is to consider the channel matrix as a multiple time series. Time series prediction is a well-studied area of statistics with a set of methods, which successfully solve problems from different domains. The most classical methods are AR, MA, ARMA(Wilson, 2016) One of the simplest methods to forecast multivariate time series is $AR(p)$ model, which refers to the auto-regressive model of order p. The $AR(p)$ model is written

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t,$$

where $\varphi_1, \ldots, \varphi_p$ are parameters, c is a constant, and the random variable $\varepsilon_t \varepsilon_{t+1} \ldots$is white noise.

The notation $MA(q)$ refers to the moving average model of order q:

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i},$$

where the $\theta_1, \ldots, \theta_q$ are the parameters of the model, μ is the expectation of X_t, and the $\varepsilon_t, \varepsilon_{t-1}, \ldots$ are again, white noise error terms.

The notation $ARMA(p, q)$ refers to the model with p autoregressive terms and q moving-average terms. This model contains the AR(p) and MA(q) models,

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

Since time series can be considered as a sequence of values, the neural network methods for sequence analysis and prediction can be used here, too. We will discuss the most famous classcal model LSTM(Gers et al., 1999).

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ \sigma_h(c_t)$$

In the evaluation on real-world data with complex mixtures of repetitive patterns, LSTNet achieved significant performance improvements over that of several state-of- the-art baseline methods. However the usu of classical model is limited of use it one time service, in next section we will describe the algorithm. wich does not have such problem.

Also a good approach is to assosiate a the channel state matrix at current timestamp with some kind of picture. It means that predicting of timeserice is the same to a problem of predicting the next video frame. For solution of such problem can be used such networks like Convolution LSTM(Xingjian et al., 2015). Also in (Li et al., 2019) authors suggest to learn some manifold structure for video, however they aimed on solving classification task and their approach is not applicable in regression because they learned some distinguishing of classes.

3. Models and Algorithms

In this section we will describe the algorithms we used to evaluate. First of all we decided to consider the channel matrix as a multiple time series, and try to evaluate ARMA model(Wilson, 2016). Also nowadays is quite popular library for predicting is Prophet model from Facebook. In contrast of classical algorithms we. took two deep learning ones. The first is a classical LSTM(Gers et al., 1999) and its extension (Xingjian et al., 2015). We should note that the last model has benefits that we do not need to train it on a particular sub carrier instead we can train it on whole set of data matrices, which is quite beneficial, also it can take inter account some impact of neighbour sub carrier. The geral model of Convolutian LSTM coold be written as

$$\begin{aligned}
\mathrm{i}_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o\right)
\end{aligned}$$

Here $*$ represents convolution and $\circ$ denotes the Hadamard product:

4. Experiment and results

First of all we should understand how the data looks like. The input we had is a complex matrix of the shape 4x312x25600. Here the first dimension describes the number of receiver antennas, the second nmber of base station antennas and the last dimension is time dimension. If we want to look at the shape of data

We can see that the signal is has periodicity but very noisy. First we will start with the classical ARMA(Wilson, 2016) model. Varying different models we obtained such results Table4

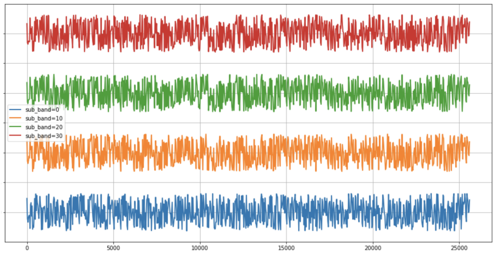

Figure 1. Phase. of the signal

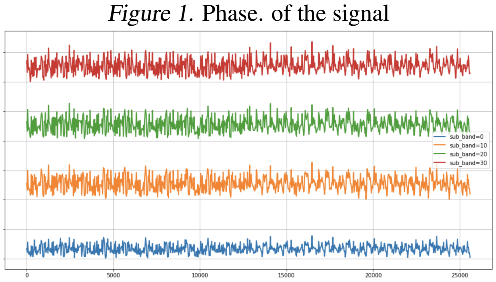

Figure 2. Modulus of the signal

heightModel(p,q)	P-value	Loss Train	Loss Val
Arma(3,1)	6.6e-50	0.008	0.022
Arma(3,2)	4.0e-50	0.008	0.022
Arma(5,2)	6.5e-43	0.007	0.024
Arma(5,3)	6.0e-42	0.007	0.024

Here we used MSE loss. However, we can see that the validation loss 10 times higher than the train loss. Also such low loss should not misunderstand , because here we have everywhere the values qite low. If we look at result predictions we will see that the validation prediction is constant, which means that model found just mean of the signal Fig3 and Fig4

To improve such bad results form classical model the Prophet model from Facebook was taken, however we achived the save results, so we will not post them.

We found out that for some classical models is qutie difficult to predict the signal, so we can try to use LSTM net. To train the simple net we used, Adam optimiser with learning rate 1e-5 and run 500 epochs, weight decay was 1e-6, also for more robust results we predict just single measurement and basing on such prediction predict the other. The learning curves are presented on Fig7 and Fig**??**

The predictions we can see on Fig7 and Fig6

However, such results do not show the power of deep learning models. First of all we want to take into account the impact of neighbours it means that we need some convolutions, also we should be also able to predict time series so Convolution LSTM is good solution, of we accurately place all dimensions and choose appropriate kernel size we would

Figure 3. Train Arma(3,2)

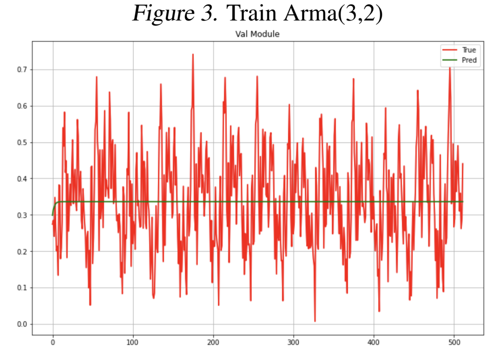

Figure 4. Validation Arma(3,2)

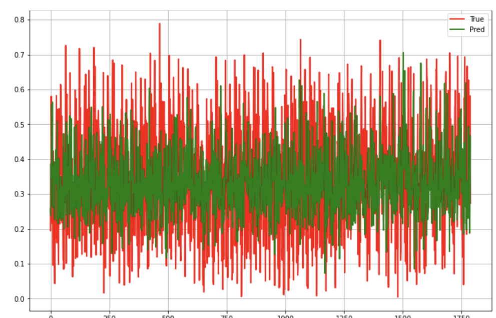

Figure 5. Train LSTM

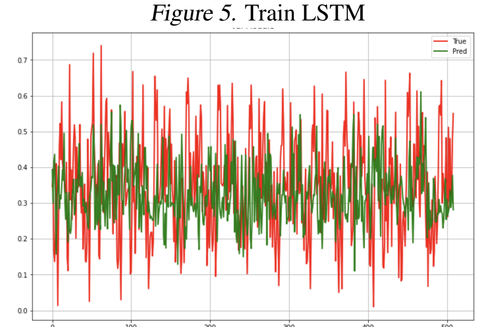

Figure 6. Validation LSTM

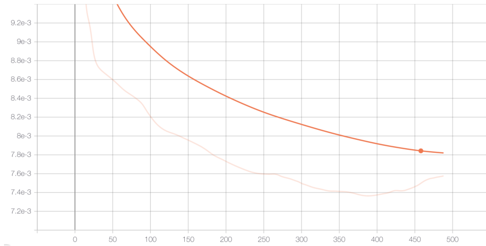

Figure 7. Train LSTM loss

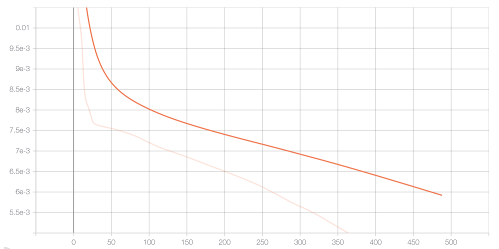

Figure 8. Validation LSTM loss

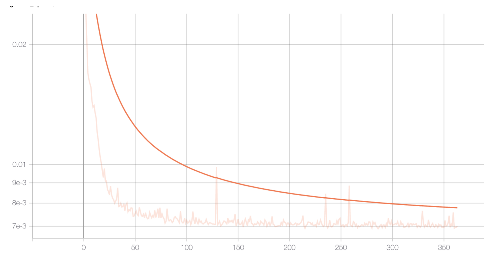

Figure 9. Train Convolution LSTM

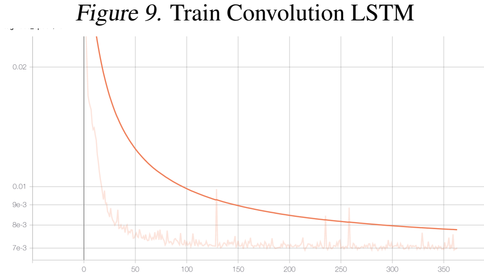

Figure 10. Validation Convolution LSTM

be able to predict the while channel matrix. Again, like in vanilla LSTM we will predict channel state for on moment of future. To train such model we used 300 epoch, AdamW optimiser with learning rate 1e-4, batch size equal to 50, and weight decay 1e-6, the results of loss curves are presented on Fig9 and Fig10. The result predictions are shown on Fig11 and Fig12. Note, here we decided to show the result of the model on unseen data. The total results in Table4

Model	Loss Train	Loss Val	Loss Test
LSTM	0.004	0.007	0.01
Conv LSTM	0.004	0.007	0.01
Conv LSTM one channel	0.002	0.004	0.005
Arma(3,2)	0.007	0.024	0.024

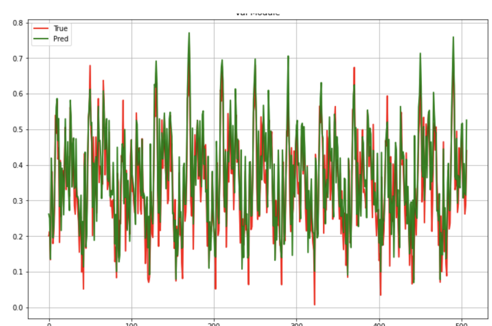

Figure 11. Convolution LSTM validation

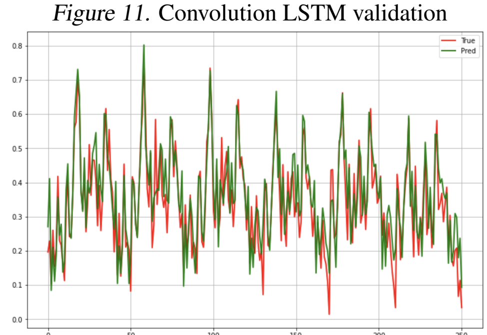

Figure 12. Convolution LSTM test

5. Further work

In previous section some good baseline Convolution LSTM was found. We should improve it. First we used the data which consist of channel matrix of one user which is not true in general case, it means that to our data one extra dimension will be added so we should be able to learn on tensor data. The second improvement is adding phase in part in out model, which means we should be able to learn complex matrix from out network. The next improvements can be taken to replace the LSTM part of the network and is more popular in our days Transformer network. Finally we can replace the existing convolutions with the special one which take the information from some manifold structure, for example geodesic one.

6. Conclusion

In this short research project, were evaluated some classical and deep learning methods on the task of predictions the channel state matrix of wireless network. We found out that the classical models can poorly predict the complicated dependence. Also if we take the classical LSTM we will face with the problem of loss of the spatial information from neighbour sub-carries, it brings us to worse results. The better solution is to use the Convolution LSTM which saves the spatial and temporal information. However such approach has lots to improve. First of all, the recurrent part can be replaced with more powerful networks like Transformer. Also the we can replace the classical convolutions with special convolution which can obtain some information from learned manifold structure. Finally we should learn two models for module and phase, one of the possible way is to train one model which works with complex data.

References

Arnold, M., Dörner, S., Cammerer, S., Yan, S., Hoydis, J., and Brink, S. t. Enabling fdd massive mimo through deep learning-based channel prediction. *arXiv preprint arXiv:1901.03664*, 2019.

Chen, X., Weng, J., Lu, W., Xu, J., and Weng, J. Deep manifold learning combined with convolutional neural networks for action recognition. *IEEE transactions on neural networks and learning systems*, 29(9):3938–3952, 2017.

Erem, B., Orellana, R. M., Hyde, D. E., Peters, J. M., Duffy, F. H., Stovicek, P., Warfield, S. K., MacLeod, R. S., Tadmor, G., and Brooks, D. H. Extensions to a manifold learning framework for time-series analysis on dynamic manifolds in bioelectric signals. *Physical Review E*, 93 (4):042218, 2016.

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. 1999.

Huang, Z., Wu, J., and Van Gool, L. Building deep networks on grassmann manifolds. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Jiang, W. and Schotten, H. D. Recurrent neural network-based frequency-domain channel prediction for wideband communications. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–6. IEEE, 2019.

Li, C., Zhang, B., Chen, C., Ye, Q., Han, J., Guo, G., and Ji, R. Deep manifold structure transfer for action recognition. *IEEE Transactions on Image Processing*, 28(9):4646–4658, 2019.

Li, Z., Huang, H., Duan, Y., and Shi, G. Dlpnet: A deep manifold network for feature extraction of hyperspectral imagery. *Neural Networks*, 2020.

Masci, J., Boscaini, D., Bronstein, M., and Vandergheynst, P. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 37–45, 2015.

Wilson, G. T. Time series analysis: Forecasting and control, by george ep box, gwilym m. jenkins, gregory c. reinsel and greta m. ljung, 2015. published by john wiley and sons inc., hoboken, new jersey, pp. 712. isbn: 978-1-118-67502-1. *Journal of Time Series Analysis*, 37(5):709–711, 2016.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Yang, Y., Gao, F., Li, G. Y., and Jian, M. Deep learning-based downlink channel prediction for fdd massive mimo system. *IEEE Communications Letters*, 23(11):1994–1998, 2019.

Zhang, J., Zhu, G., Heath Jr, R. W., and Huang, K. Grassmannian learning: Embedding geometry awareness in shallow and deep learning. *arXiv preprint arXiv:1808.02229*, 2018.