POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Deep Learning based Approximate Message Passing for MIMO Detection in 5G

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Andrea Pozzoli

Advisor: Prof. Umberto Spagnolini

Academic year: 2020-2021

---

1. Introduction

Massive Multiple Input Multiple Output (MIMO) technology is the key technology in the 5G mobile communication system. This technique enhances a large number of antennas at both the transmitter and the receiver side [6]. With a high number of antennas, trying to recover the transmitted information in a massive MIMO up-link receiver is more computationally complex, because the transmitted signals interfere with each other. The thesis focuses on signal detection in a multi-user communication scenario, in particular at the receiver side.

2. Problem and research question

Promising detectors are iterative detectors based on Approximate Message Passing (AMP) algorithm [2]. Deep Learning (DL) reported promising results in signal detection. Recently two new algorithms based on deep learning and AMP have been proposed. The first one is called LVAMP and it is a deep leaning based neural network that is obtained by unfolding the Vector AMP algorithm (VAMP). The second one instead is a new network obtained by unfolding Orthogonal AMP (OAMP) algorithm, called OAMPNet2. A comparison between these two algorithms in different situations and with a common baseline is missing, therefore the thesis is focused on that.

Research Question: *Can LVAMP and OAMPNet2 be considered as sub-optimal solutions to MIMO detection problem and which one is the best network in terms of complexity and SER on different MIMO scenarios?*

3. MIMO

MIMO system can be represented trough a matrix mathematical approach.

$$\bar{y} := \bar{H}\bar{x} + \bar{n} \tag{1}$$

where $\bar{y}$ is a vector of complex values of length $\bar{N}_r$, with $\bar{N}_r$ that is the number of receiver antennas, and represents the signal received by the BS. $\bar{x}$ represents the signal transmitted by the UEs and it is a vector of complex values, taken from a discrete alphabet $\bar{\mathcal{A}}$, of length $\bar{N}_t$, with $\bar{N}_t$ that is the number of transmitter antennas. $\bar{n}$ represents the noise and it is a vector of complex numbers of length $\bar{N}_r$. Finally, $\bar{H}$ is called channel matrix, it has shape $\bar{N}_r \times \bar{N}_t$ and $h_{ij} \in \mathcal{C}$ is the channel gain that derives from $\bar{x}_j$ and $\bar{x}_i$

antennas. The channel matrix can assume different structures, called channel models.

3.1. i.i.d Gaussian channel model

This channel model is so common because it is simple but it is far from the real scenario. In fact, it assumes that each $\bar{h_{ij}}$ is independent from all the others and $\bar{h_{ij}} \sim \mathcal{CN}(0, 1/\bar{N}_r)$. The columns of $\bar{H}$ are normalized such that $\|\bar{h_i}\|_2 = 1$ in this thesis.

3.2. Kronecker channel model

The Kronecker channel model simulates the spatial correlation among channels. It is modelled as: $\bar{H} = R_R^{1/2} K R_T^{1/2}$ where $\bar{k_{ij}} \sim \mathcal{CN}(0, 1/\bar{N}_r)$ and R_R and R_T are the receiver and transmitter Correlation matrices respectively. These matrices are generated according to an exponential correlation model, depending on a parameter of correlation that assumes values from 0 (iid Gaussian) to 1 when all the channels interfere with each other. The coefficient at receiver side is indicated as ρ_r while the one at transmitter side is ρ_t. Also in this case, the column of $\bar{H}$ are normalized such that $\|\bar{h_i}\|_2 = 1$.

3.3. From Complex to Real MIMO System

Due to the fact that working with complex values is more difficult than working with real numbers, the MIMO model can be convert to a real-valued system. A new vector is defined for each variable of the model, in particular the transmitted symbol vector becomes

$$x = \begin{bmatrix} \Re(\bar{x}^T) & \Im(\bar{x}^T) \end{bmatrix}^T \tag{2}$$

where $\Re(\cdot)$ is the real part of the complex number between brackets and $\Im(\cdot)$ is its imaginary part. Also the other variables are modified and in particular

$$\begin{aligned} y &= \begin{bmatrix} \Re(\bar{y}^T) & \Im(\bar{y}^T) \end{bmatrix}^T \\ n &= \begin{bmatrix} \Re(\bar{n}^T) & \Im(\bar{n}^T) \end{bmatrix}^T \\ H &= \begin{bmatrix} \Re(\bar{H}^T) & -\Im(\bar{H}^T) \\ \Im(\bar{H}) & \Re(\bar{H}) \end{bmatrix} \\ y &= Hx + n \\ N_r &= 2\bar{N}_r \\ N_t &= 2\bar{N}_t \end{aligned} \tag{3}$$

The last modification that must be applied concerns the alphabet $\mathcal{A}$. With these new definition of MIMO system, x can assume values coming from $\mathcal{A}$, where $\mathcal{A} = \Re(\bar{\mathcal{A}}) = \Im(\bar{\mathcal{A}})$ in the case of QAM modulation.

4. MIMO Detection

Definition 1 (MIMO Detection): *Given a MIMO system a MIMO Detection method can be defined as the problem of retrieving the transmitted signal vector $x \in \mathcal{R}^{N_t}$ from a noisy linear measurement that can be expressed as $y = Hx + n \in \mathcal{R}^{N_r}$ where H is a known matrix $\in \mathcal{R}^{N_r \times N_t}$ and n is an unknown unstructured noise vector $\in \mathcal{R}^{N_r}$.*

The optimal method for solving the MIMO detection problem is represented by the ML detector:

$$\hat{x} = \arg\max_{x \in \mathcal{A}^{N_t}} prob(y|x, H) = \arg\min_{x \in \mathcal{A}^{N_t}} \|y - Hx\|^2 \tag{4}$$

ML does not scale, therefore several suboptimal solutions have been tested in order to solve this problem during the years, and the most common is Minimum Mean Squared Error (MMSE). MMSE finds $\hat{x}_{MMSE}$ through

$$\hat{x}_{MMSE} = \arg\min_{x \in \mathcal{A}^{N_t}} \int \|x - \bar{x}\|^2 p(x|y) dx \tag{5}$$

that is equal to compute $\mathbf{E}[x|y]$.

4.1. AMP

The AMP algorithm [1] performs the steps:

$$\begin{aligned} z_t &= \hat{x}_t + H^H(y - H\hat{x}_t) + b_t \\ b_t &= \alpha_t(H^H(y - H\hat{x}_{t-1}) + b_{t-1}) \\ \hat{x}_{t+1} &= \eta_t(z_t; \sigma_t) \end{aligned} \tag{6}$$

AMP is an iterative algorithm that uses a b_t term that is called Onsanger correction term. Both σ_t and α_t can be computed using Signal Noise Ratio (SNR) and system parameters such as the dimension of the system or the constellation. The denoising function is the optimal one for each element of z_t.

Optimal denoiser for Gaussian noise: if the noise at the input of the denoiser $z_t - x$ has an iid Gaussian distribution with a covariance matrix that is diagonal with value $\sigma_t^2 I_{N_t}$, the element-wise thresholding function derived from the previous formula is

$$\eta_t(z; \sigma_t^2) = \frac{1}{Z} \sum_{x_t \in \mathcal{A}} x_t \exp(-\frac{\|z - x_t\|^2}{\sigma_t^2}) \tag{7}$$

where $Z = \sum_{x_t \in \mathcal{A}} \exp(-\frac{\|z-x_t\|^2}{\sigma_t^2})$. In this case σ_t represents the standard deviation of the Gaussian noise.

4.2. OAMP

A variant of AMP that relaxes the iid Gaussian channel assumption is the Orthogonal AMP (OAMP) that works for unitarily invariant channel matrices [3]. OAMP is an optimal estimator with excellent convergence properties [4]. The principle of OAMP is to decouple the posterior probability $p(x|y,H)$ into a series of probabilities $p(x_i|y,H)_{i=1,2,\dots,N_t}$ in an iterative way. The OAMP detector can be written as Algorithm 1

Algorithm 1 OAMP

Require: received signal y, channel matrix H, noise covariance matrix $R_{\hat{n}\hat{n}}$
Output: Estimated signal $\hat{x}_{T+1}$
Initialize: $\tau_0 \leftarrow 1, \hat{x}_0 \leftarrow 0$
for $t = 1,\dots,T-1$ **do**
$r_t = \hat{x}_t + W_t(y - H\hat{x}_t)$
$\hat{x}_{t+1} = \mathbf{E}\{x|r_t,\tau_t\}$
$v_t^2 = \frac{\|y-H\hat{x}_t\|_2^2 - \mathbf{tr}(R_{\hat{n}\hat{n}})}{\mathbf{tr}(H^H H)}$
$\tau_t^2 = \frac{1}{N_t}\mathbf{tr}(B_t B_t^H)v_t^2 + \frac{1}{N_t}\mathbf{tr}(W_t R_{\hat{n}\hat{n}} W_t^H)$
end for
Return $\hat{x_T}$

where

$$\begin{aligned} v_t^2 &= \frac{\mathbf{E}[\|q_t\|_2^2]}{N_t} \\ \tau_t^2 &= \frac{\mathbf{E}[\|p_t\|_2^2]}{N_t} \\ W_t &= \frac{N_t}{\mathbf{tr}(\hat{W}_t H)}\hat{W}_t \\ \hat{W}_t &= v_t^2 H^H (v_t^2 H H^H + R_{\hat{n}\hat{n}})^{-1} \end{aligned} \tag{8}$$

where $R_{\hat{n}\hat{n}}$ is the covariance matrix of the noise in signal detector $\hat{n}$, and $B_t = I - W_t H$. Due to the fact that the values of x are taken from a constellation of symbols and that the estimation is based on MMSE, in order to estimate $\hat{x}$ it is used

$$\hat{x}_{t+1} = \mathbf{E}\{x_i|r_i,\tau_i\} = \frac{\sum_{s_i} s_i \mathcal{N}_{\mathcal{C}}(s_i; r_i, \tau_t^2)p(s_i)}{\sum_{s_i} \mathcal{N}_{\mathcal{C}}(s_i; r_i, \tau_t^2)p(s_i)} \tag{9}$$

where $p(s_i)$ is the prior distribution of the symbol x_t and is defined as $p(x_i) = \sum_{j \in N_r} \frac{1}{\sqrt{N_r}}\delta(x_i - s_j)$.

4.3. VAMP

In [5] the Vector Approximate message passing (VAMP) algorithm is proposed, showing that it holds under a bigger class of channel matrices respect to AMP, those that are right-orthogonally invariant. The VAMP algorithm is based on the “economy” SVD of the channel matrix: $H = \bar{U}\mathbf{diag}(\bar{s})\bar{V}^T$ where $\bar{s} \in \mathcal{R}^R forR := \mathbf{rank}(H) \leq \min(N_r, N_t)$. Algorithm 2 is the SVD form of the VAMP algorithm.

Algorithm 2 VAMP in SVD form

Require: received signal y, channel matrix H, denoiser function $g_1(\cdot,\gamma_t)$, noise precision $\gamma_w \geq 0$, number of iterations T, $r_0 \geq 0$, $\gamma_0 \geq 0$

Output: Estimated signal $\hat{x}_T$
Compute economy SVD $H = \bar{U}\mathbf{diag}(\bar{s})\bar{V}^T$
Compute preconditioned $\tilde{y} = \mathbf{diag}(\bar{s})^{-1}\bar{U}^T y$
for $t = 0,1,\dots,T$**do do**
$\hat{x}_t = g_1(r_t,\gamma_t)$
$\alpha_t = \langle g_1'(r_t,\gamma_t)\rangle$
$\tilde{r}_t = (\hat{x}_t - \alpha_t r_t)/(1-\alpha_t)$
$\tilde{\gamma}_t = \gamma_t(1-\alpha_t)/\alpha_t$
$d_t = \gamma_w \mathbf{diag}(\gamma_w \bar{s}^2 + \tilde{\gamma}_t \mathbf{1})^{-1}\bar{s}^2$
$\gamma_{t+1} = \tilde{\gamma}_t\langle d_t\rangle/(\frac{N_t}{R} - \langle d_t\rangle)$
$r_{t+1} = \tilde{r}_t + \frac{N_t}{R}\bar{V}\mathbf{diag}(d_t/\langle d_t\rangle)(\tilde{y} - \bar{V}^T\tilde{r}_t)$
end for
Return$\hat{x}_T$

In the algorithm, $g_1(r_t,\gamma_t) : \mathcal{R}^{N_t} \to \mathcal{R}^{N_t}$ is defined as $g_1 = \arg\min_{x \in \mathcal{A}}[\frac{\gamma_t}{2}|x - r_t|^2 - \ln p(x)]$ where $p(x)$ is the prior distribution of x, while $g_1'(r_t,\gamma_t)$ as $g_1'(r_t,\gamma_t) = diag[\frac{\partial g_1(r_t,\gamma_t)}{\partial r_t}]$, and $\langle\cdot\rangle$ is the empirical averaging operation $\langle u\rangle := \frac{1}{N_t}\sum_{n=1}^{N_t} u_n$. Moreover, $r_t \in \mathcal{R}^{N_t}$ is called residual term at iteration $t-th$ and γ_t represents the reciprocal of its variance. Finally, R is defined as $R = rank(H)$.

There is also another approach to write the VAMP algorithm without using the SVD form. In this second variant, the linear MMSE and the trace of the covariance matrix must be computed at each iteration, involving the inverse of a $N_t \times N_t$ matrix. The second version of the VAMP algorithm is called LMMSE form and it follows the steps of Algorithm 3.

In the algorithm, $g_2(r_{2t},\gamma_{2t}) = (\gamma_w H^T H + \gamma_{2t} I)^{-1}(\gamma_w H^T y + \gamma_{2t} r_{2t})$ is a MMSE estimate linear in r_{2k} (that's why it is called LMMSE form) of a random vector x_2 under likeli-

Algorithm 3 VAMP in LMMSE form

Require: LMMSE estimator $g_2(r_{2t}, \gamma_{2t})$, denoiser function $g_1(\cdot, \gamma_{1t})$, number of iterations T, r_{10} and $\gamma_{10} \geq 0$
Output: Estimated signal $\hat{x}_T$
for $t = 0, 1, \dots, T$ **do**
  Denoising
  $\hat{x}_{1t} = g_1(r_{1t}, \gamma_{1t})$
  $\alpha_{1t} = \langle g_1'(r_{1t}, \gamma_{1t}) \rangle$
  $\eta_{1t} = \gamma_{1t}/\alpha_{1t}$
  $\gamma_{2t} = \eta_{1t} - \gamma_{1t}$
  $r_{2t} = (\eta_{1t}\hat{x}_{1t} - \gamma_{1t} r_{1t})/\gamma_{2t}$
  LMMSE
  $\hat{x}_{2t} = g_2(r_{2t}, \gamma_{2t})$
  $\alpha_{2t} = \langle g_2'(r_{2t}, \gamma_{2t}) \rangle$
  $\eta_{2t} = \gamma_{2t}/\alpha_{2t}$
  $\gamma_{1,t+1} = \eta_{2t} - \gamma_{2t}$
  $r_{1,t+1} = (\eta_{2t}\hat{x}_{2t} - \gamma_{2t} r_{2t})/\gamma_{1,t+1}$
end for
Return$\hat{x}_{1T}$

hood $\mathcal{N}(y; Hx_2, \gamma_w^{-1} I)$ and prior $\mathcal{N}(r_{2k}, \gamma_{2k}^{-1} I)$. Instead, $\langle g_2'(r_{2k}, \gamma_{2k}) \rangle$ can be defined as $\langle g_2'(r_{2k}, \gamma_{2k}) \rangle = \frac{\gamma_{2t}}{N_r} \mathbf{tr}[(\gamma_w H^T H + \gamma_{2t} I)^{-1}]$. For what concerns g_1 and g_1', they are defined as in Algorithm 2.

4.4. OAMP-Net2

In [2] the authors proposed a model driven deep learning network based on OAMP, named OAMPNet2. OAMPNet2 performs considerably better than OAMP and is more robust with respect to SNR, channel correlation, modulation symbol and MIMO configuration mismatches. OAMPNet2 is a deep learning network composed by T cascade layers with the same architecture but different parameters. The inputs of the network are y and $\hat{H}$, while the output is $\hat{x}_{T+1}$. For each layer, the input is the $\hat{x}_t$ estimation of x computed in the previous layer. The OAMPNet2 detector follows these steps at each layer:

$$
\begin{aligned}
r_t &= \hat{x}_t + \gamma_t W_t (y - \hat{H}\hat{x}_t) \\
\hat{x}_{t+1} &= \eta_t(r_t, \tau_t^2; \phi_t, \xi_t) \\
v_t^2 &= \frac{\|y - \hat{H}\hat{x}_t\|_2^2 - \mathbf{tr}(R_{\hat{n}\hat{n}})}{\mathbf{tr}(\hat{H}^H \hat{H})} \\
\tau_t^2 &= \frac{1}{N_t}\mathbf{tr}(C_t C_t^H) v_t^2 + \\
&\quad \frac{\theta_t^2}{N_t}\mathbf{tr}(W_t R_{\hat{n}\hat{n}} W_t^H)
\end{aligned}
\tag{10}
$$

where $C_t = I - \theta_t W_t \hat{H}$. The nonlinear estimator η_t for estimating $\hat{x}_{t+1}$ instead is revised and it is constructed by the divergence free estimator

$$
\eta_t(r_t, \tau_t^2, \phi_t, \xi_t) = \phi_t(\mathbf{E}\{x|r_t, \tau_t\} - \xi_t r_t) \tag{11}
$$

where $\mathbf{E}\{x|r_t, \tau_t\}$ is computed as for OAMP.

4.5. LVAMP

LVAMP is the neural network that results from the unfolding of the iterations of VAMP. The LVAMP network consists in two modules as VAMP that can be divided in two parts each. Therefore, four steps compose the LVAMP network, a LMMSE estimation, decoupling stage, shrinkage estimation, an another identical decoupling stage. The LMMSE stage uses as parameters $\tilde{\theta} = \{U_t, s_t, V_t, \sigma_{wt}^2\}$ for each iterations t. When the channel is not iid Gaussian and there are correlations, it is important to consider the covariance matrix. In this case the parameters of the network becomes $\tilde{\theta} = \{G_t, K_t\}$ and the LMMSE stage is defined as

$$
\tilde{\eta}(\tilde{r}_t; \tilde{\sigma}_t, \tilde{\theta}_t) = G_t \tilde{r}_t + K_t y \tag{12}
$$

where $G_t \in \mathcal{R}^{N_t \times N_t}$ and $K_t \in \mathcal{R}^{N_t \times N_r}$. The shrinkage stage instead has as parameter θ_t that is used in the denoising function $\eta(\cdot)$. Therefore the parameters to be learnt are expressed as $\{\tilde{\theta}_t, \theta_t\}_{t=0}^T$. It is suggested to inizialize U, s, V as the SVD values of H and σ_w^2 at the average value of $N_t^{-1}\|y\|^2$.

5. Method

The thesis is based on an experimental and quantitative methodology. The algorithms that are taken in consideration for the thesis and therefore for the experiments are MMSE, VAMP, OAMP, LVAMP, OAMPNet2, and also MMNet. MMNet is a deep learning neural network that can be considered the state of the art when trained online. In the thesis it is trained offline for coherence with the other DL-based algorithms.

The experiments consist in different simulations based on 5G scenarios. The simulations differ for size of the MIMO system, channel model, QAM size, SNR values and type of training.

For each experiment, the same range of SNR values are considered. The range that has been selected is from 18 dB to 23 dB, that are the values that are common in the comparison of MIMO detectors.

Finally, the algorithms, in particular the ones that are DL-based, are trained and tested in three different ways:

- The training phase is conducted with iid Gaussian channel matrices, and tested with matrices generated from the same channel model.
- The training phase uses Kronecker channel matrices with different correlation parameters, and the testing is conducted with Kronecker channel matrices.
- The training is based on iid Gaussian channel model, but the testing is done with Kronecker matrices in order to verify the adaptability of the algorithms.

The metric that is used to compare the algorithms in the experiments is the SER performance metric.

$$SER = \frac{no.of symbols in error}{total no.of transmitted symbols} \tag{13}$$

5.1. Dataset

The training dataset is split in parts of equal size called batches before the start of the learning procedure. When the training phase starts the algorithm iterates over epochs. The dataset is composed by samples with three sources of randomness: the signal x, the channel matrix H and the noise n. The signal x is sampled form the constellation randomly and uniformly. The channel matrix is sampled following the structure of the channel model selected for the experiment. The noise is derived from the sampling of the standard deviation sigma that is derived from the SNR of the experiment. For each sample in the batch, the SNR value is chosen randomly in the range 18-23 dB. Therefore, each batch that is used during training is a tuple of four elements $\{(y^{(d)}, H^{(d)}, \sigma^{(d)}, x^{(d)})\}_{d=1}^{D}$ where $x^{(d)}$ are the values that have to be predicted. The generated training dataset is split in two parts in order to create the validation set. The size of the validation set is 25% of the generated dataset. In order to avoid overfitting, early stopping, dropout and cross validation are used. The algorithms have been trained for 2000 epochs, with Adam optimizer and learning rate of 0.001. Each batch has size of 1000 samples.

5.2. Experiments

The number of combinations that can be generated and that have been tested during the thesis work are hundreds. In the thesis only the most significant ones are reported. Seven experiments have been described in the thesis and they can be summarized through Table 1.

6. Results and Analysis

In this section, the results of the experiments are shown and analysed. Due to the fact that the 32 transmitters and 64 receivers configuration is the most common and realistic for comparing algorithms in a MIMO scenario, only the results of Experiment 5, 6 and 7 are shown in this executive summary.

6.1. Experiment 5

In Figure 1 the results of the experiment are presented. All the algorithms have very good performances, with SER values that are always (a part from 18 dB of SNR) under 0.1. The OAMPNet2 algorithm outperforms the others, being the only one having all SER under 0.1, and reaching 10^{-4} for 23dB of SNR. VAMP and LVAMP algorithms instead does not perform well in this configuration, with values that are very similar to the one of MMSE. The OAMP algorithm is the second best algorithm with great results despite it is not a DL-based algorithm. For what concerns MMNet instead, it has a particular curve. In fact, at the beginning the curve and the SER values are very close to the ones of OAMP, but when the SNR values increase, it does not follow the OAMP curve, having an improvement of accuracy lower.

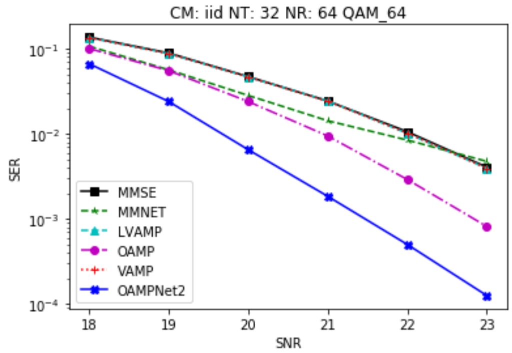

Figure 1: Experiment 5 results

Table 1: Experiments summary

Experiment	N_r	N_t	Shape	Modulation	C.M. training	C.M. testing
1	4	4	Squared	QAM-16	i.i.d Gaussian	i.i.d Gaussian
2	32	32	Squared	QAM-64	i.i.d Gaussian	i.i.d Gaussian
3	32	32	Squared	QAM-64	Kronecker	Kronecker
4	32	32	Squared	QAM-64	i.i.d Gaussian	Kronecker
5	64	32	Rectangular	QAM-64	i.i.d Gaussian	i.i.d Gaussian
6	64	32	Rectangular	QAM-64	Kronecker	Kronecker
7	64	32	Rectangular	QAM-64	i.i.d Gaussian	Kronecker

6.2. Experiment 6

Experiment 6 is divided in two cases, that are based on training and testing phases conducted with Kronecker channel matrices.

6.2.1 Experiment 6a

The first case consists in an experiment where $\rho_R = \rho_T = 0.3$. In Figure 2 the results of the experiment are reported. The MMNet algorithm performs very badly, with SER values that are very high. This algorithm is fragile when trained offline and this experiments shows the difficulties for MMNet to adapt to a more complex scenario. For what concerns the other algorithms, they behave similarly to experiment 5. In fact, VAMP and LVAMP have performances very close to the ones of MMSE. OAMP provides again good results, while OAMPNet2 outperforms the other algorithms. As it possible to see, OAMPNet2 is the only algorithm that reaches a SER value close to 10^{-4} with a SNR value of 23 dB. The SER values are a bit worse than the ones in experiment 5, but this is due to a more complex and realistic MIMO system in which the experiment is conducted.

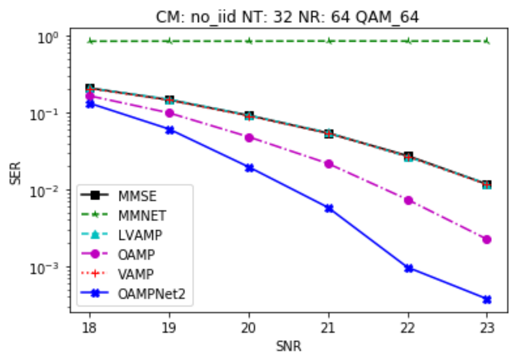

Figure 2: Experiment 6a results

6.2.2 Experiment 6b

The second case changes the correlation parameters respect to the experiment 6a. In fact in this case $\rho_R = \rho_T = 0.5$. The result of the experiment are presented in Figure 3. This time MMNet has SER values very close to 1, meaning that quite all the estimations are wrong. With an highly correlated Kronecker channel model, MMNet is not able to estimate the transmitted signal if trained offline. An interesting result of the experiment is represented by OAMPNet2. In fact for the first time, it has a behaviour very close to the one of OAMP for SNR equal to 18, 19 and 20 dB. For the last three values of SNR, instead, OAMPNet2 performs better than the others. Again, VAMP and LVAMP have accuracy very close to the one of MMSE, overlapping graphically. Also OAMP is closer to MMSE for 18dB of SNR, but then its performances improve.

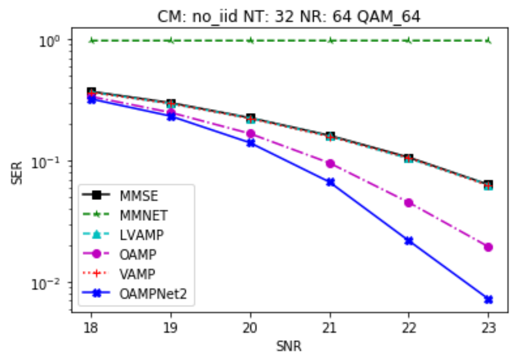

Figure 3: Experiment 6b results

6.3. Experiment 7

The final experiment is based on a training phase conducted with iid Gaussian channel matrices and then the algorithms are tested on a Kronecker channel model. This represents the innovative part of the thesis. Thanks to this experiment it is possible to analyse the adaptability of the DL algorithms, trained with a simple channel model, and tested on a more complex one. Also in this experiment, two cases are analysed.

6.3.1 Experiment 7a

The first case uses $\rho_R = \rho_T = 0.3$ as correlation parameters for the Kronecker channel model. The results of the experiment are shown in Figure 4. The results are very similar to the ones obtained in experiment 6a, with some SER values that are also better. Again, MMNet is not able to adapt to a more complex channel model, having high SER values for each SNR considered. VAMP and LVAMP performs as MMSE. OAMPNet2, despite the training phase conducted with iid Gaussian channel matrices, outperforms the other algorithms. Again it reaches a value of SER in order of 10^{-4} for SNR equal to 22 dB and 23 dB.

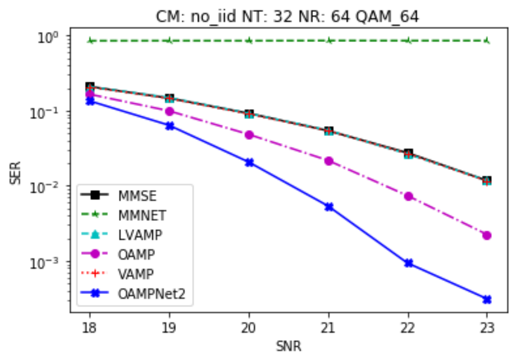

Figure 4: Experiment 7a results

6.4. Experiment 7b

The second case has $\rho_R = \rho_T = 0.5$, therefore an highly correlated channel. The results of the experiment are very similar to the ones of experiment 6b and they can be seen in and Figure 5. MMNet again misses the estimations of all the points, achieving a SER very close to 1 for all the SNR values. Also in this case, VAMP, LVAMP and MMSE perform the same. OAMPNet2 is also in this case the best algorithm, but for low SNR values it performs like OAMP. For higher SNR values instead it improves its performances. Also OAMP has performances that are a bit worse for the lowest SNR values, and better when SNR increases.

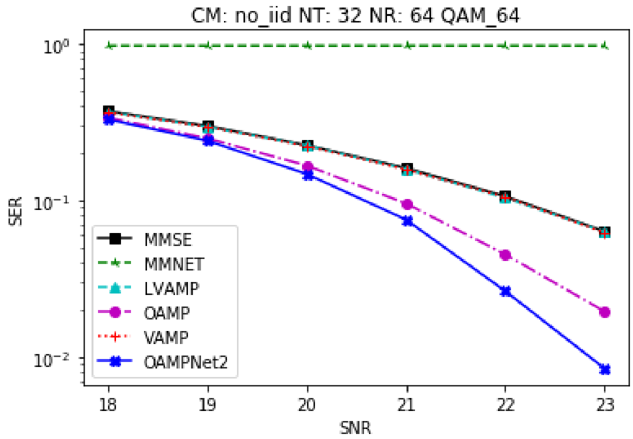

Figure 5: Experiment 7b results

7. Complexity Analysis

A summary of this complexity analysis is shown in Table 2 where each column represents an algorithm, the first row the computational complexity and the second row the number of iterations/layers T for convergence.

8. Conclusions

Both LVAMP and OAMPNet2 algorithms can be considered sub-optimal solutions for the MIMO detection problem. In fact, LVAMP can be built with a complexity lower than MMSE and it obtains SER values that are equal or better than MMSE in all the scenarios considered for the experiments. Also OAMPNet2 is a sub-optimal solution to the problem because it outperforms MMSE in all the scenarios considered, having a complexity that is just a few greater than MMSE, thanks also to a very fast convergence.

Another important insight discovered thanks to the experiments is that the two deep networks are able to adapt very well to different conditions. In fact they perform well on both iid Gaussian and Kronecker channel models. They are also able to perform well on realistic channel models like the Kronecker one, with both medium and high correlation parameters. Moreover, they can perform well on this type of channel, also with a simple training phase conducted with iid Gaussian channel matrices.

Table 2: Complexity and number of iterations/layers for convergence of each algorithm

	MMSE	MMNet	VAMP	LVAMP	OAMP	OAMPNet2
Complexity	$\mathcal{O}(N_r^3)$	$\mathcal{O}(TN_r^2)$	$\mathcal{O}(2TN_rN_t)$	$\mathcal{O}(2TN_rN_t)$	$\mathcal{O}(TN_r^3)$	$\mathcal{O}(TN_r^3)$
T	1	10-14	5-6	5-6	4-5	4-5

9. Future work

- Find new methods in order to solve the MIMO detection problem.
- compare and verify that all the different version of VAMP and LVAMP performs the same or if there is a version that provides better results.
- Trying to rethink the OAMPNet2 algorithm in order to lowering its complexity.
- Test all the algorithms on the 3GPP channel model.
- Using other QAM types, a wider range of SNR values, different sizes of MIMO system, different $\frac{N_t}{N_r}$ ratios.
- Changing the loss function during the training phase of the DL-based algorithms

References

[1] Mark Borgerding, Philip Schniter, and Sundeep Rangan. Amp-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, Aug 2017.

[2] Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. A model-driven deep learning network for mimo detection. *IEEE Transactions on Signal Processing*, 68:1702–1715, Feb 2020.

[3] Mehrdad Khani, Mohammad Alizadeh, Jakob Hoydis, and Phil Fleming. Adaptive neural signal detection for massive mimo. *IEEE Transactions on Wireless Communications*, 19(8):5635–5648, 2020.

[4] Junjie Ma and Li Ping. Orthogonal amp, 2017.

[5] Sundeep Rangan, Philip Schniter, and Alyson K. Fletcher. Vector approximate message passing. *CoRR*, abs/1610.03082, 2016.

[6] Andrea Scotti. Graph neural networks and learned approximate message passing algorithms for massive mimo detection, 2020.