15 May 2024

Machine LLRning Based Soft Detection for Extremely High Bandwidth Efficiency 4K-QAM and Beyond

Research Scholarship for M.Eng. Students project under Contract No. M-Eng.-EE-003/2566
Faculty of Engineering, Khon Kaen University

Presented by

Miss Donlaporn Triamwichanon 655040032-8

Supervisor : Assoc. Prof. Dr. Puripong Suthisopapan
Committee : Assoc. Prof. Dr. Watid Phakphisut
Assoc. Prof. Dr. Virasit Imtawil

Outlines

1

Research Objectives

What is QAM?

"QAM"

stands for Quadrature Amplitude Modulation.

- very popular and commercial digital modulation scheme used in modern communication systems to transmit digital signal.

Definition of M-QAM Signal

$$\boxed{s(t) = m_1(t)\cos\omega_c t + m_2(t)\sin\omega_c t,}\quad 0 \le t \le T_s$$

(in-phase) (quadrature)

M : modulation order

T_s : symbol period (its reciprocal is symbol rate)

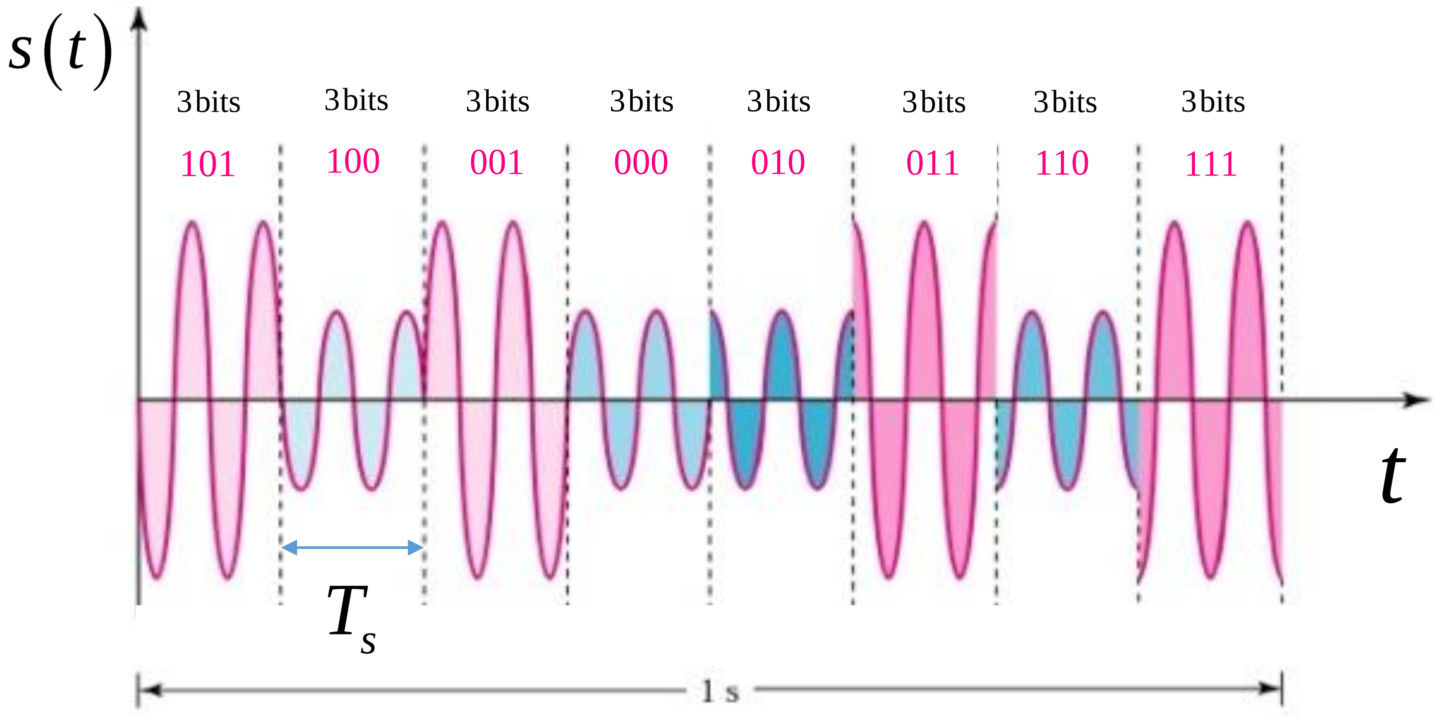
$m = \log_2 M$: the number of bits per symbol

d : minimum Euclidean distance

$$m_1(t), m_2(t) \in \left\{\pm d, \pm 3d, \ldots, \pm\left(\sqrt{M}-1\right)d\right\}$$

Example : 8-QAM Signals

Eight possible waveforms associated with different ***3-bits***.



symbol rate = 8 baud

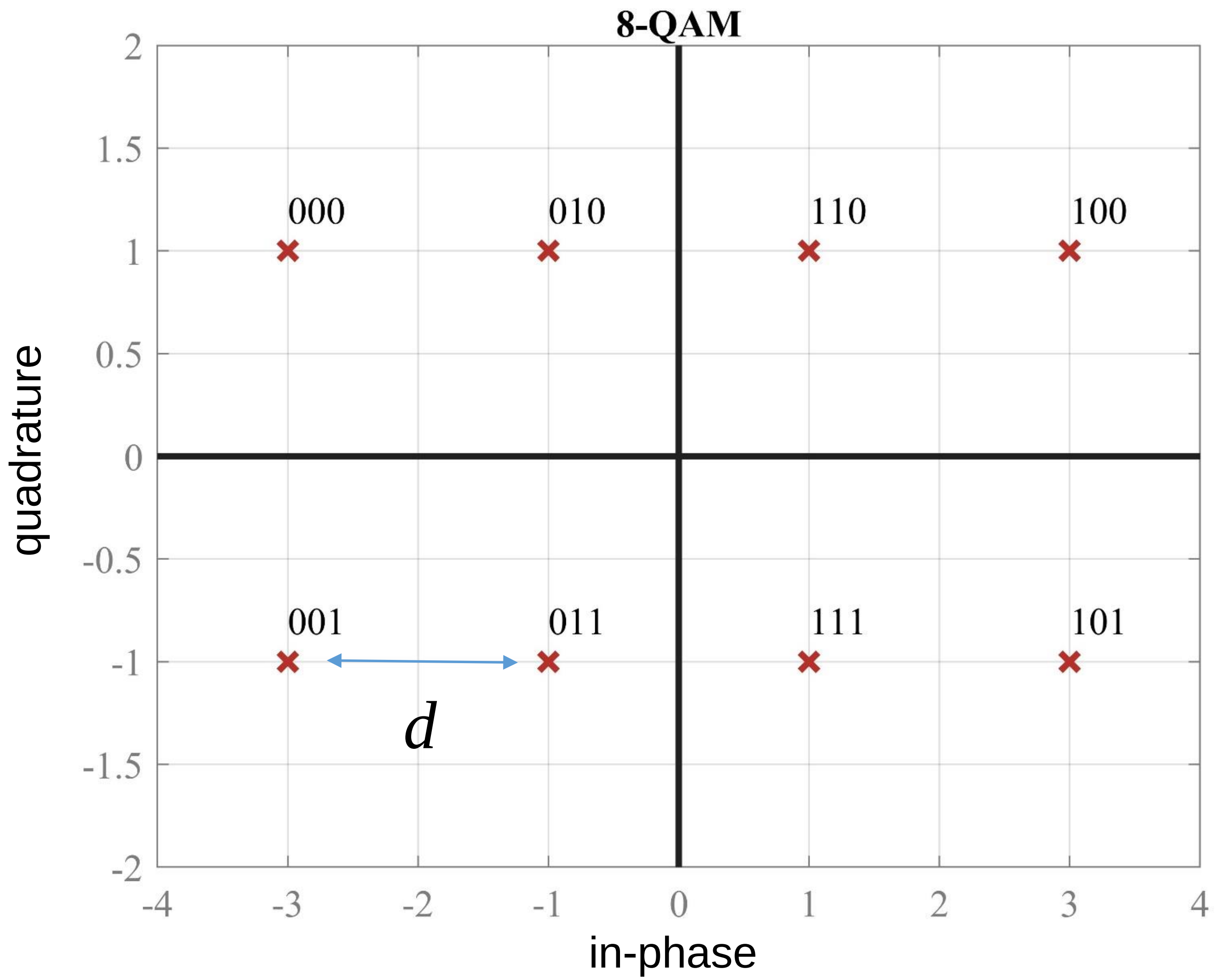
bit rate = $8 \times 3 = 24$ bps

$$M = 8$$

$$m = \log_2 8 = 3$$

One 8-QAM symbol carries ***three*** bits.

Example : 8-QAM Signals (cont.)



$$s = m_1 + jm_2$$

- constellation
- Gray-labelling
- rectangular
- Min. Euclidean distance

Please Imagine !?

8-QAM : 3 bits per symbol

16-QAM : 4 bits per symbol

256-QAM : 8 bits per symbol

4096-QAM : ??? bits per symbol

4096-QAM is known as **4K-QAM**

-more bits
bandwidth efficient
- same bandwidth
- same symbol period

higher order

Pros. & Cons. of Higher-Order QAM

01 high data rate (bps)

02 high spectral efficiency (bps/Hz)

03 supports various data rates

04 very sensitive to interference (dense constellation)

https://www.linkedin.com/advice/3/how-can-higher-order-modulation-improve#.

Existing Communication Standards

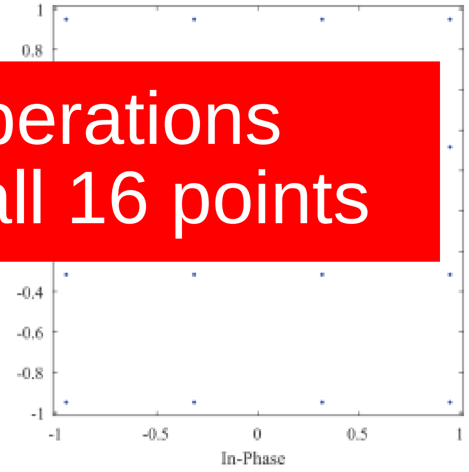
Higher-order QAM in existing communication systems and standards

DVB-C2-future extensions [1]	65536-QAM
Asymmetric digital subscriber loop for copper twisted cables [2]	32768-QAM
power line ethernets [3]	4096-QAM
IEEE802.11be [4]	4096-QAM

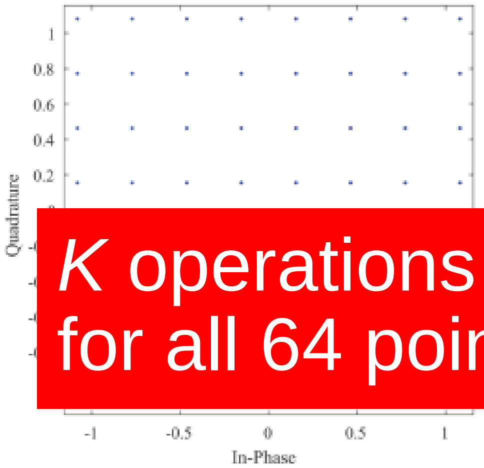
High-order QAM is preferred !

So, we focus on 4K-QAM and beyond !!

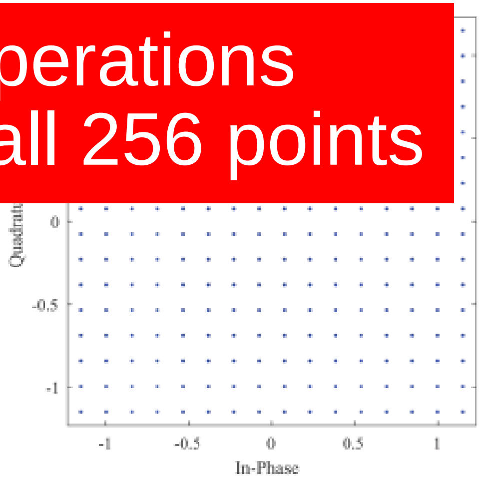
Major Problem : QAM Receiver



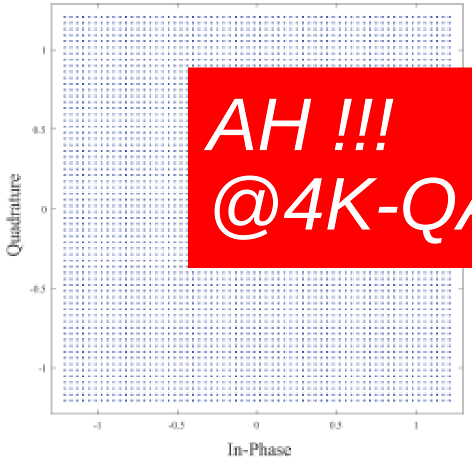
K operations
for all 16 points



K operations
for all 64 points



K operations
for all 256 points



AH !!!
@4K-QAM

Higher order
Higher complexity

not only provide
hard bit estimation

Modern QAM receiver
needs to provide

“soft bit”

more computational time& complexity

Our Approach

We expect that

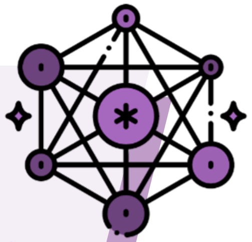
“**Neural Network**” (**NN**)

(from AI/machine learning world)

can solve the problem

Objectives

1. To **propose**
NN-based 4K-QAM soft demodulator



2. To **investigate** the performance
NN-based 4K-QAM soft demodulator
in **coded 4K-QAM systems**

Scopes of Research

1. Focus only on **rectangular QAM** systems under AWGN and Rayleigh fading channels.

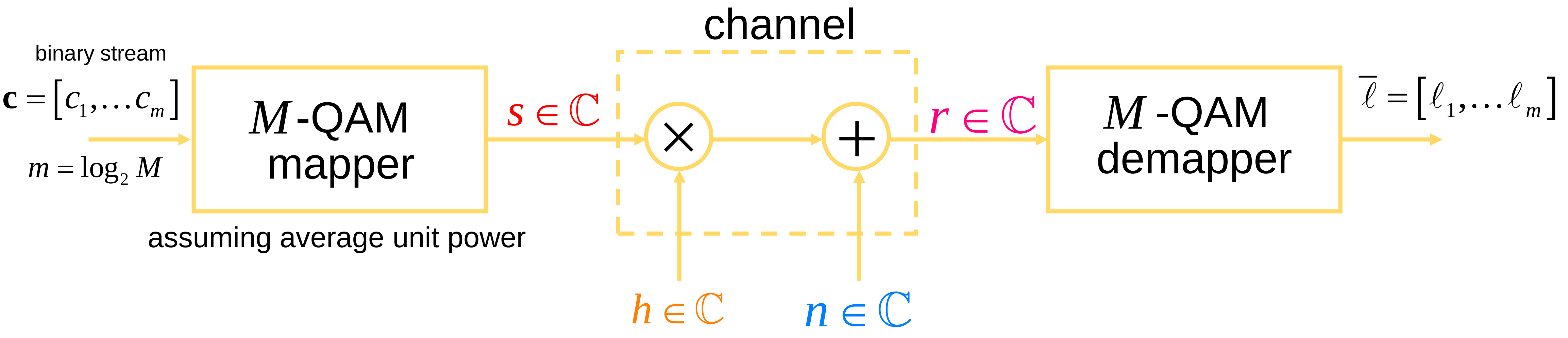
2. Consider **Gray code** mapping for QAM constellation.

3. Using **LDPC codes as FEC**, with structure and parameters based on **Wi-Fi 7 standard**.

2

Background

System Model



$$\boxed{r = hs + n}$$

s : QAM symbol

r : received symbol

h : fading coefficient

n : additive white Gaussian noise

Rayleigh fading channel:

$$h = \frac{1}{\sqrt{2}}(a + bj),$$

$$a, b \sim N(0,1)$$

(two i.i.d. random variables)

AWGN channel: $h = 1$

SNR definition:

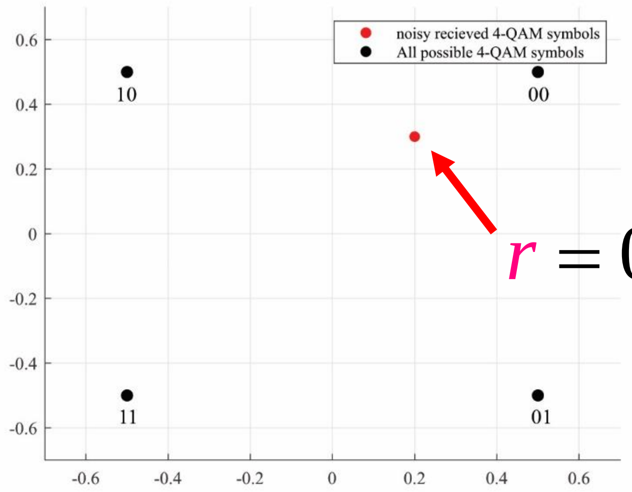
$$\frac{E_s}{N_0} = mR\frac{E_b}{N_0}$$

R : coding rate

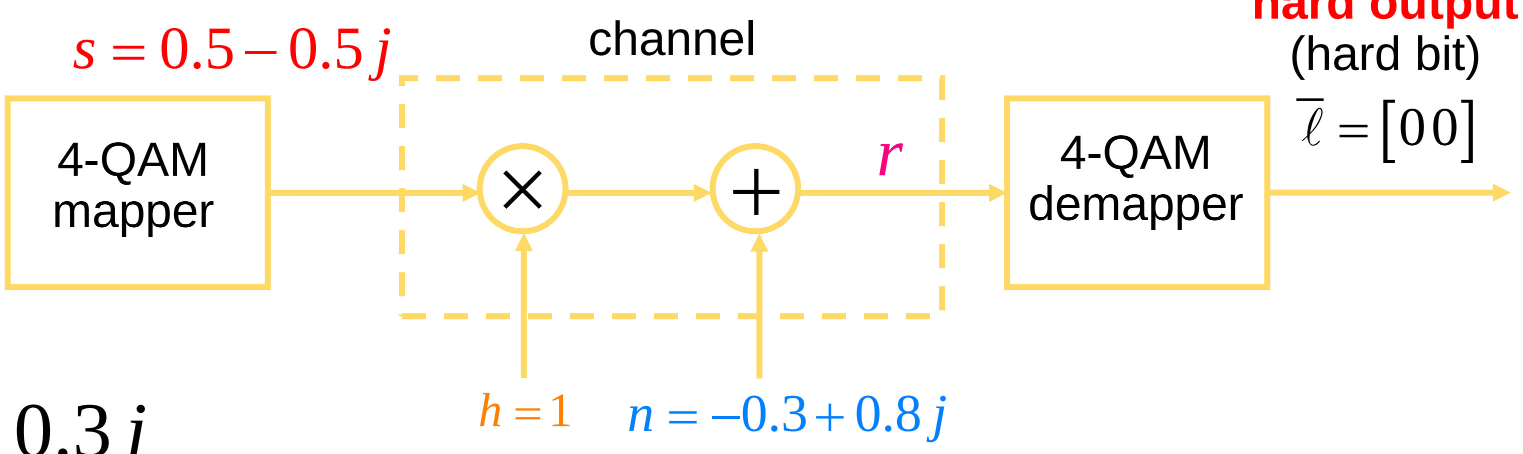
B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," in IEEE Transactions on Communications, vol. 51, no. 3, pp. 389-399, March 2003

System Model (cont.)

Example: 4-QAM



$r = 0.2 + 0.3j$



4-QAM symbols	Gray labeling	received symbol	Euclidean distance
$0.5+0.5j$	00	$0.2+0.3j$	0.36
$0.5-0.5j$	01	$0.2+0.3j$	0.85
$-0.5-0.5j$	11	$0.2+0.3j$	1.06
$-0.5+0.5j$	10	$0.2+0.3j$	0.73

Log-MAP Algorithm

The log-likelihood ratio (LLR) is defined as

$$l_i \triangleq \ln\left(\frac{\Pr\left(c_i = 1 \middle| r\right)}{\Pr\left(c_i = 0 \middle| r\right)}\right), i = 1, \ldots, m.$$

σ^2 : noise variance

S_i^b : sub set of M-QAM symbols, $b \in \{0,1\}$

if $\Pr\left(b_i = 1 \middle| r\right) > \Pr\left(b_i = 0 \middle| r\right)$ then $\log\left(\frac{\Pr\left(b_i = 1 \middle| r\right)}{\Pr\left(b_i = 0 \middle| r\right)}\right) \rightarrow$ positive value

if $\Pr\left(b_i = 1 \middle| r\right) < \Pr\left(b_i = 0 \middle| r\right)$ then $\log\left(\frac{\Pr\left(b_i = 1 \middle| r\right)}{\Pr\left(b_i = 0 \middle| r\right)}\right) \rightarrow$ negative value

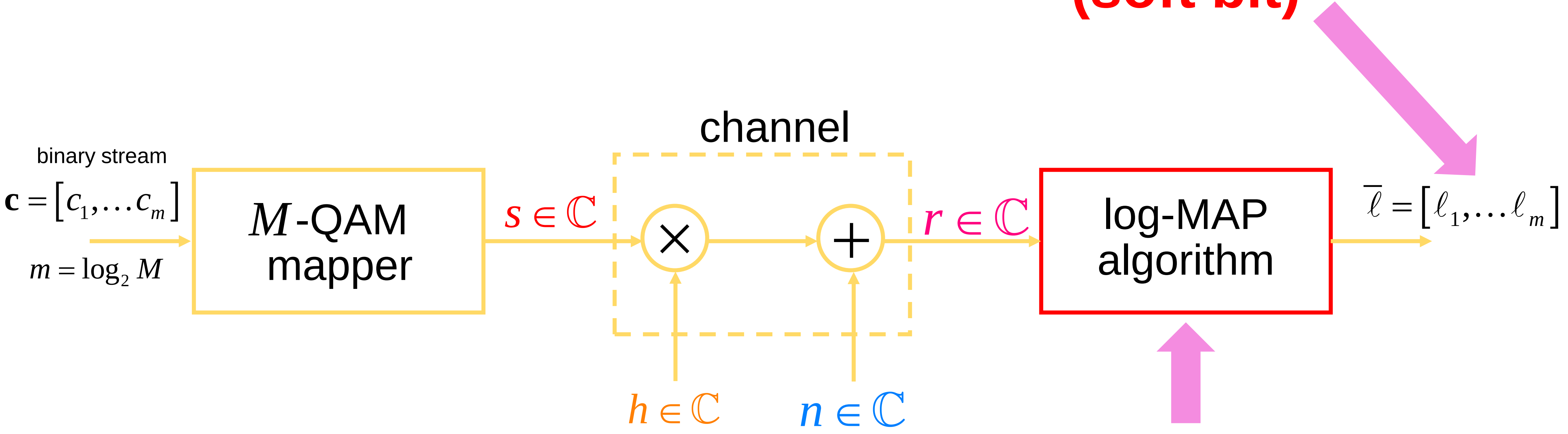
The optimal log maximum aposteriori (MAP) is given by:

$$l_i = \ln\left(\frac{\sum_{s \in S_i^1} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}{\sum_{s \in S_i^0} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}\right), i = 1, \ldots, m$$

- positive LLR
- negative LLR
- magnitude of LLR

Soft Bit

soft output
(soft bit)

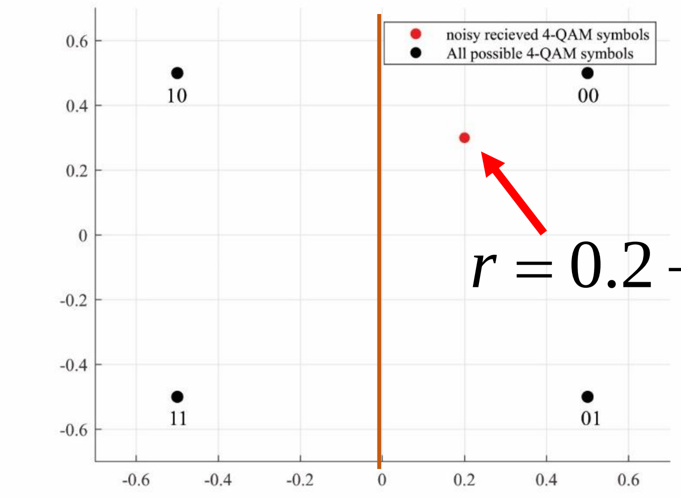


a.k.a. soft demodulator
soft demapper
soft detection
soft detector

Log-MAP Algorithm (cont.)

Example: 4-QAM

Given that,



$r = 0.2 + 0.3j$

$s \in \{\pm 0.5 \pm 0.5j\}$

$h = 1$

$\sigma^2 = 0.05$

$$\ell_1 = \ln\left(\frac{\sum_{s \in S_1^1} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}{\sum_{s \in S_1^0} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}\right)$$

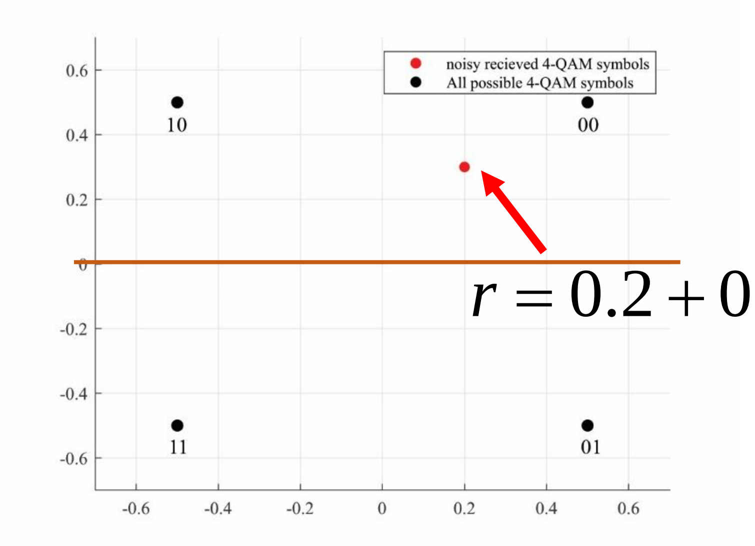
$$= \ln\left(\frac{e^{\left(-\frac{\|(0.2+0.3j)-(-0.5+0.5j)\|^2}{0.05}\right)} + e^{\left(-\frac{\|(0.2+0.3j)-(-0.5-0.5j)\|^2}{0.05}\right)}}{e^{\left(-\frac{\|(0.2+0.3j)-(0.5+0.5j)\|^2}{0.05}\right)} + e^{\left(-\frac{\|(0.2+0.3j)-(0.5-0.5j)\|^2}{0.05}\right)}}\right)$$

$$= -16$$

Log-MAP Algorithm (cont.)

Example: 4-QAM

Given,



$r = 0.2 + 0.3j$

$s \in \{\pm 0.5 \pm 0.5j\}$

$h = 1$

$\sigma^2 = 0.05$

$$\ell_2 = \log\left(\frac{e^{\left(-\frac{|(0.2+0.3j)-(-0.5-0.5j)|}{0.05}\right)} + e^{\left(-\frac{|(0.2+0.3j)-(0.5-0.5j)|}{0.05}\right)}}{e^{\left(-\frac{|(0.2+0.3j)-(0.5+0.5j)|}{0.05}\right)} + e^{\left(-\frac{|(0.2+0.3j)-(-0.5+0.5j)|}{0.05}\right)}} \right)$$

$$= -24$$

log-MAP algorithm (cont.)

$$\left.\begin{aligned}\Pr(b_i=1|r)&=0.1\\ \Pr(b_i=0|r)&=0.9\end{aligned}\right\}\ \log\left(\frac{\Pr(b_i=1|r)}{\Pr(b_i=0|r)}\right)=\log\left(\frac{0.1}{0.9}\right)=-2.1972$$

$$\left.\begin{aligned}\Pr(b_i=1|r)&=0.3\\ \Pr(b_i=0|r)&=0.7\end{aligned}\right\}\ \log\left(\frac{\Pr(b_i=1|r)}{\Pr(b_i=0|r)}\right)=\log\left(\frac{0.3}{0.7}\right)=-0.8473$$

$$\left.\begin{aligned}\Pr(b_i=1|r)&=0.49\\ \Pr(b_i=0|r)&=0.51\end{aligned}\right\}\ \log\left(\frac{\Pr(b_i=1|r)}{\Pr(b_i=0|r)}\right)=\log\left(\frac{0.49}{0.51}\right)=-0.04$$

ambiguity
++++++

Max-Log-MAP Algorithm

Apply this approximation,

$$\log\left(\sum_j \exp\left(-x_j^2\right)\right) \approx \max_j\left(-x_j^2\right)$$

$$\ell_i = \log\left(\frac{\sum_{s\in S_i^1} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}{\sum_{s\in S_i^0} e^{\left(-\frac{\|r-hs\|^2}{\sigma^2}\right)}}\right)$$

log-MAP

well known

suboptimal detection

get rid of exponential & logarithm

$$\ell_i \approx \frac{1}{\sigma^2}\left(\min_{s\in S_i^1}\|r-hs\|^2 - \min_{s\in S_i^1}\|r-hs\|^2\right), i = 1,\ldots,m$$

lower complexity

Artificial Intelligence (AI)

Artificial intelligence (AI)

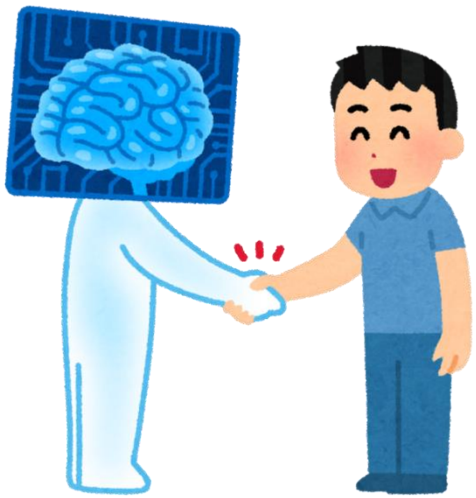
A program that can sense, reason, act, and adapt.

Machine learning

Algorithm that can learn from data and thus perform tasks without explicit instructions.

Deep learning

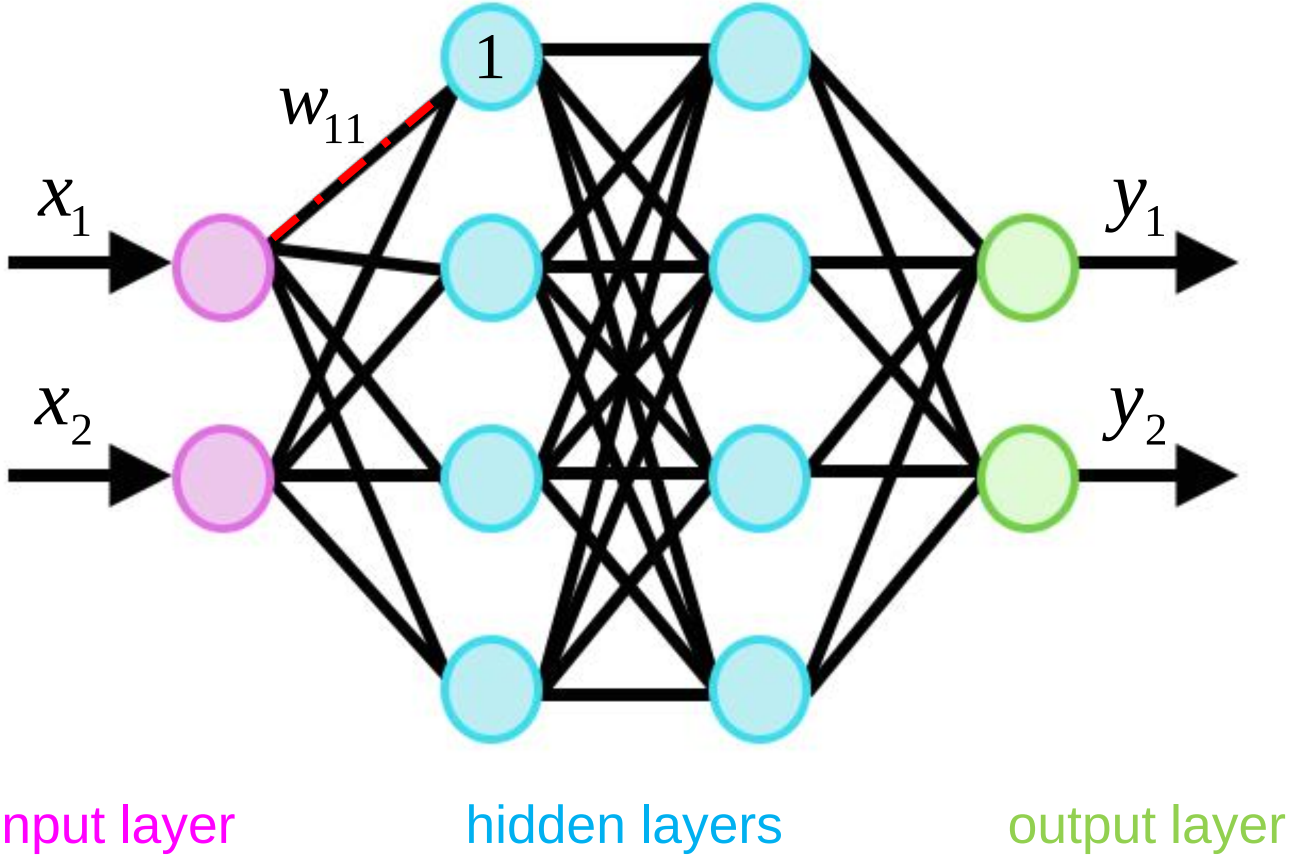
Subset of machine learning in which multilayer neural network.



Neural Network (NN)

A neural network, also known as an artificial neural network (ANN),
is a computational model inspired by the structure and function of the human brain.

ex. NN with 2 inputs + 2 hidden layers + 2 outputs.



- 3 categories of layer
- weights
- fully connected
- #neurons
- #hidden
- shallow / deep
- neuron = $\Sigma \mid f$

Neural Network (NN) (cont.)

NN attempts to **learn**
the relationship between **inputs** and **responses**

$$x^1 \leftrightarrow y^1$$

$$x^2 \leftrightarrow y^2$$

$$\vdots$$

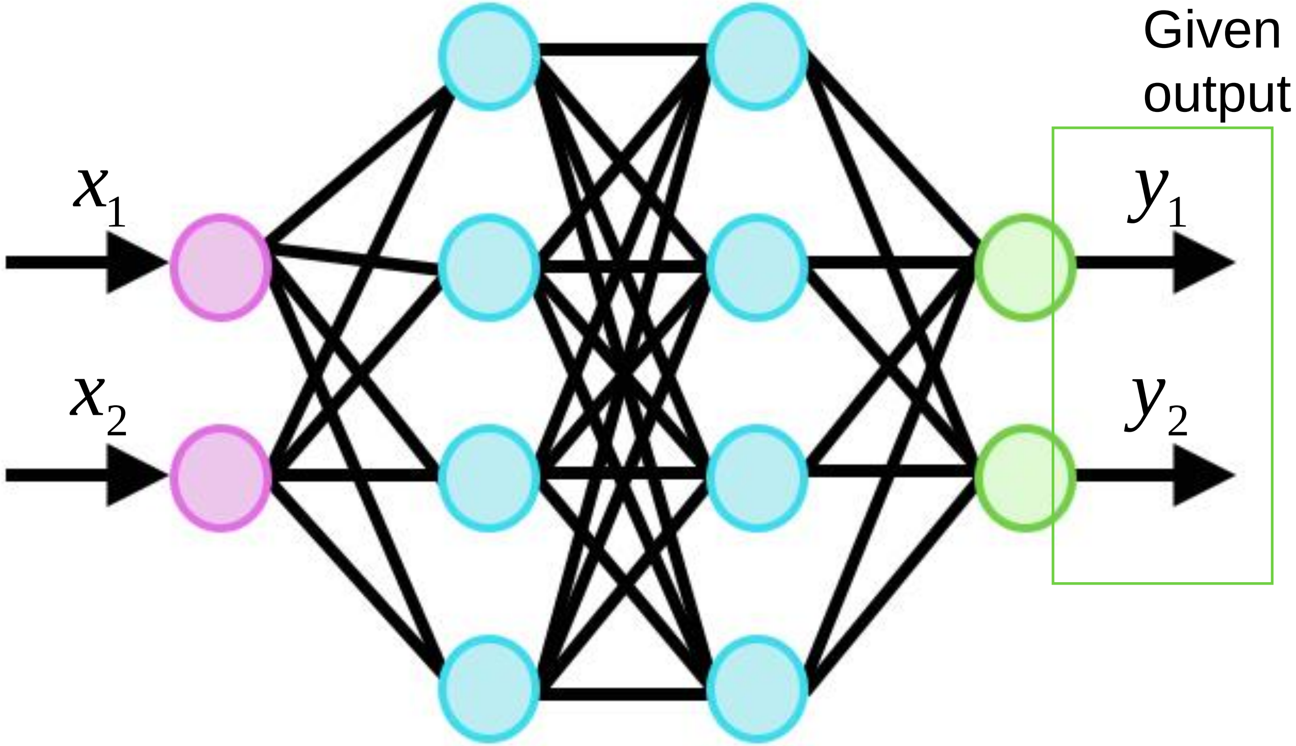
$$x^k \leftrightarrow y^k$$

Relationship can be

scalar to scalar
scalars to scalar
vector to vector
etc.

Neural Network (NN) (cont.)

ex. NN with 2 inputs + 2 hidden layers + 2 outputs.

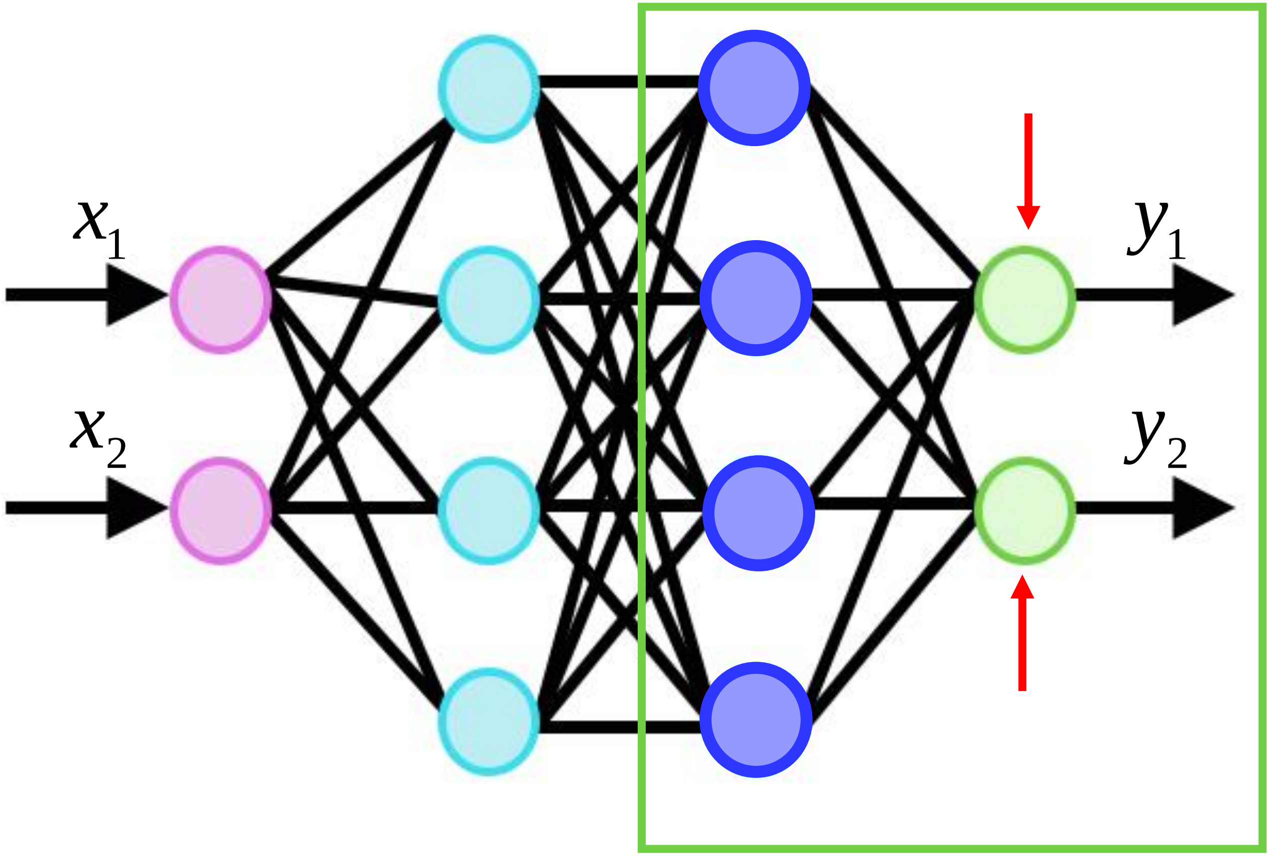


Given the input to NN,
output from NN is often called **prediction.**

The ultimate goal is
to find set of **suitable weights**
that **minimize** the **error**
between predictions
and responses
(real output / ground truth).

Neural Network (NN) (cont.)

ex. NN with 2 inputs + 2 hidden layers + 2 outputs.



Operation in NN can be viewed in matrix / vector form

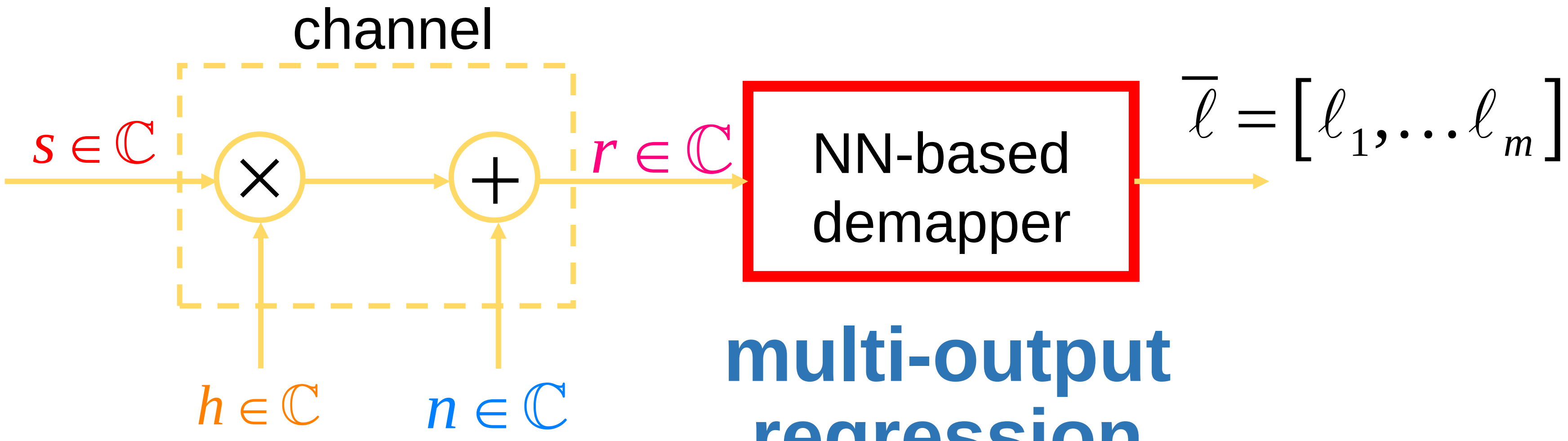
$$\mathbf{y} = f\left(\mathbf{W}\mathbf{z} + \mathbf{b}\right)$$

Overview of NN Generation

1. Data collection: a large number of input/response pairs
2. Dataset generation:
   training set (a%)
   validation set (b%) $\Bigg\}$ a+b+c = 100%
   testing set (c%)
3. Define NN architecture and
   related parameters (e.g., optimizer, loss function)
4. Training process: determine weights for NN
5. Evaluation: assess performance of NN

NN-based Soft Demodulator

given received signal as the input of NN



multi-output regression

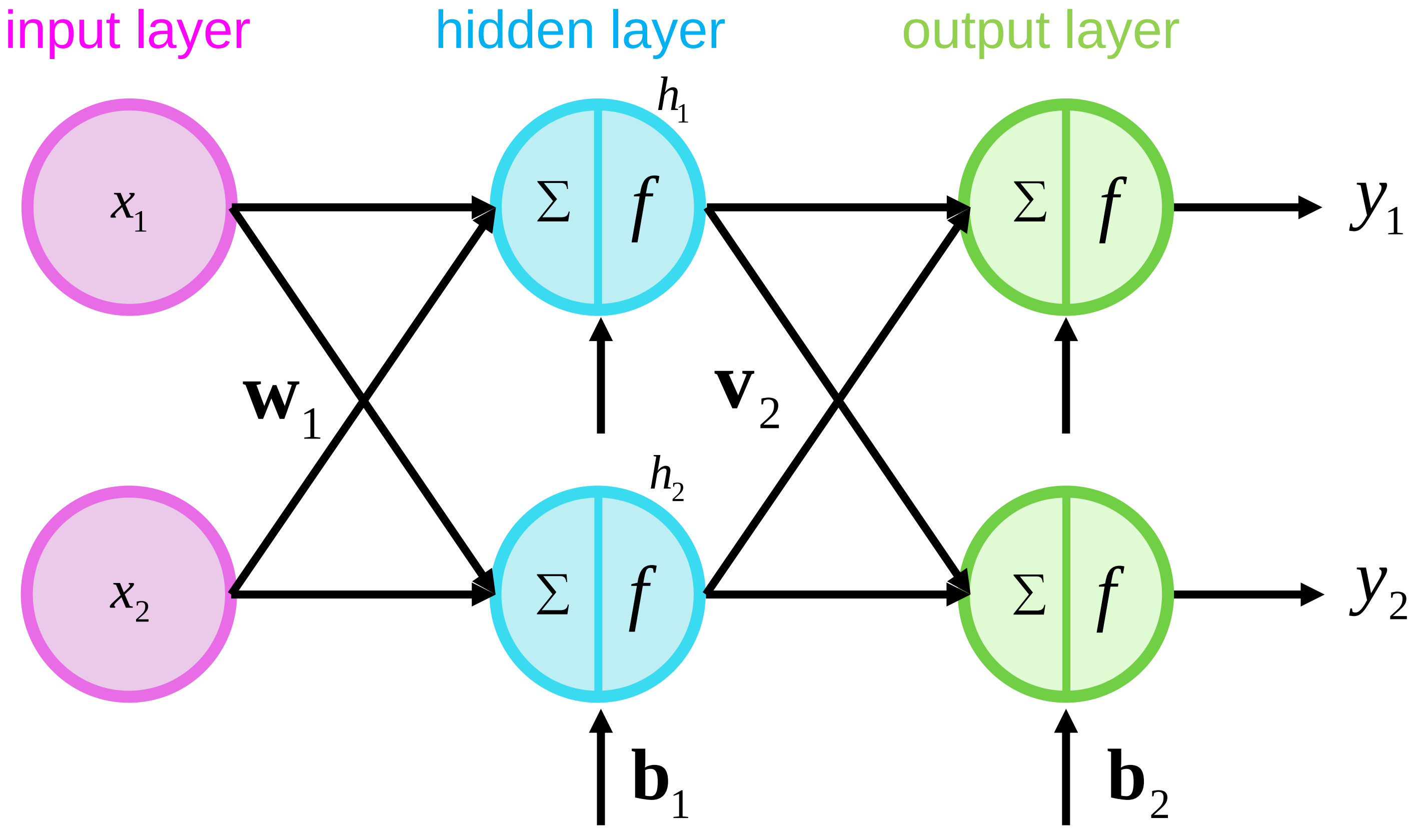
able to estimate LLR

NN-Based Example

NN task: find good
$\mathbf{w}_1, \mathbf{v}_2, \mathbf{b}_1, \mathbf{b}_2$

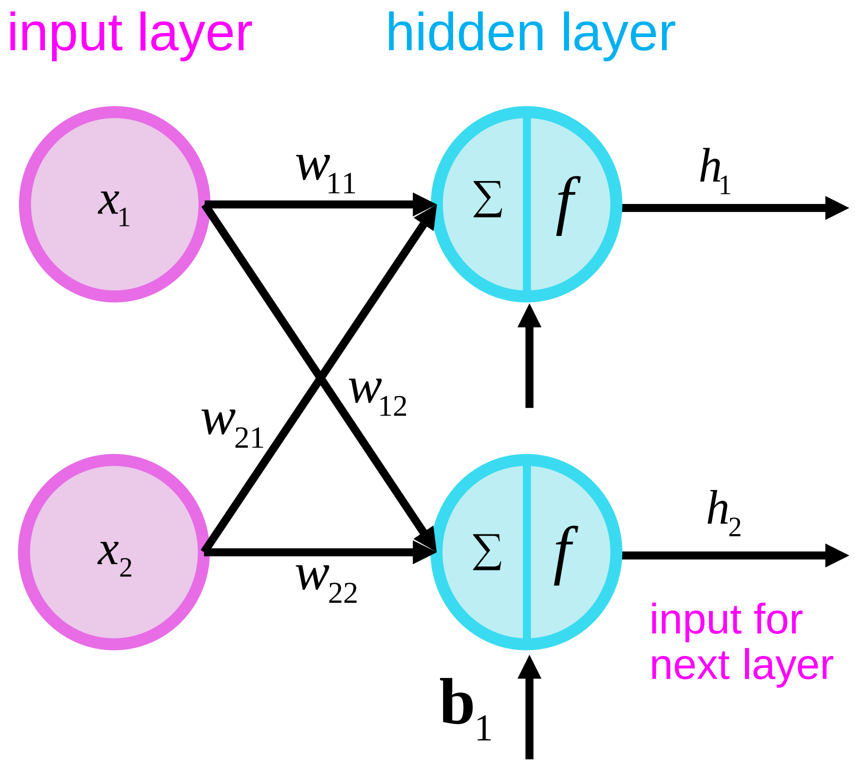
$\mathbf{w}_1, \mathbf{v}_2$: weight

$\mathbf{b}_1, \mathbf{b}_2$: bias



NN-Based Example (cont.)

STEP01: Input layer



assume

$$\mathbf{x} = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} \quad \mathbf{w}_1 = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{w22} \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}\right)$$

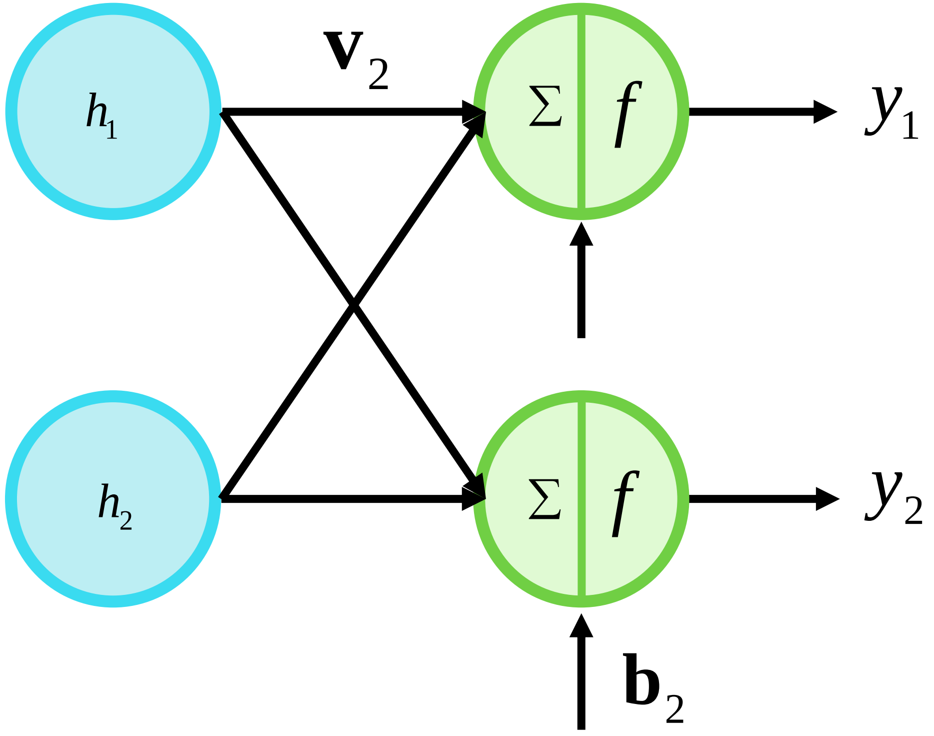
f : Sigmoid function (can be changed)

$$\boxed{\mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} 0.6225 \\ 0.7109 \end{bmatrix}}$$

NN-Based Example (cont.)

STEP02: Input layer

output layer



output from the last layer
will be an input for the next layer

assume

$$\mathbf{h} = \begin{bmatrix} 0.6225 \\ 0.7109 \end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix} 1.0 & 1.5 \\ 2.0 & 2.5 \end{bmatrix} \quad \mathbf{b}_2 = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = f\left(\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}\begin{bmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{bmatrix} + \begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix}\right)$$

prediction
(LLR approximation)

$$\mathbf{y} = \begin{bmatrix} 2.69 \\ 4.53 \end{bmatrix}$$

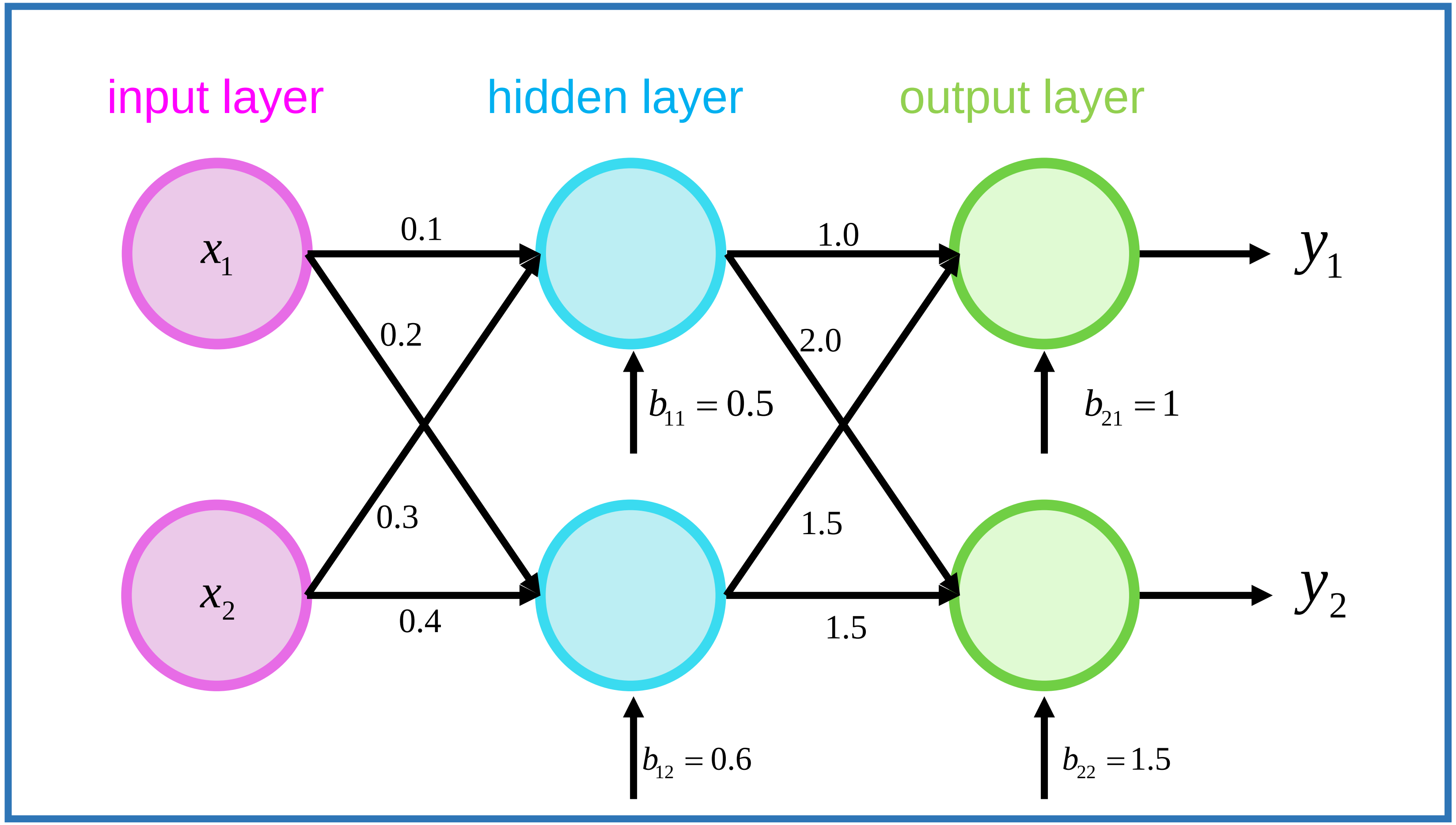
response
(ground truth)

$$\bar{\ell} = \begin{bmatrix} -16 \\ -24 \end{bmatrix}$$

Loss function

Mean Square Error (MSE) ~ 581.64

NN-Based Example (cont.)



called

NN model

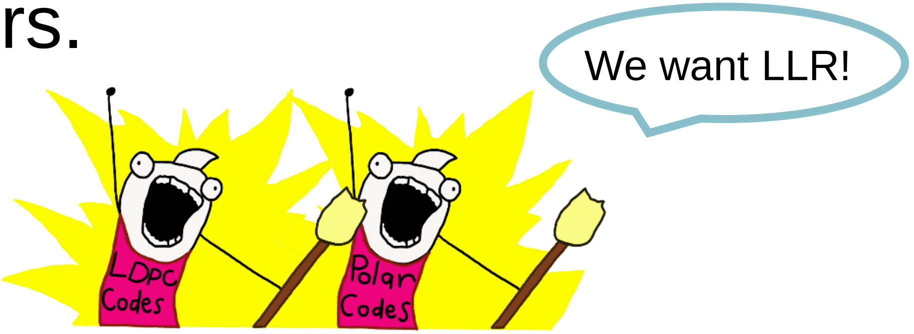
3

Literature Review

Related Works

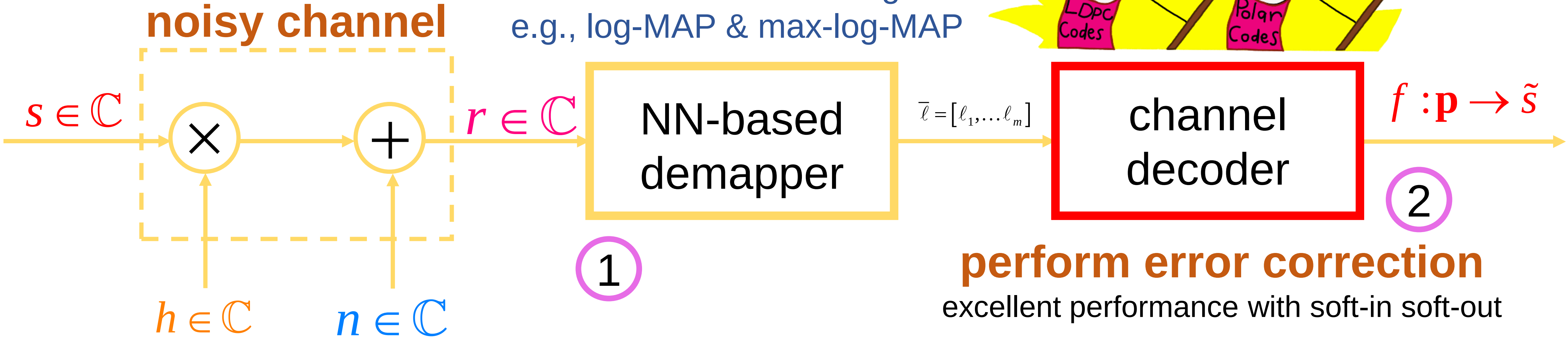
All works in this section constructed "**Neural Network**" to estimate LLR from QAM system with the goal to reduce complexity.

LLR is essential for modern channel decoders.



competitors

non-machine learning



Related Works

2019, O. Shantal and J. Hoydis {shallow}

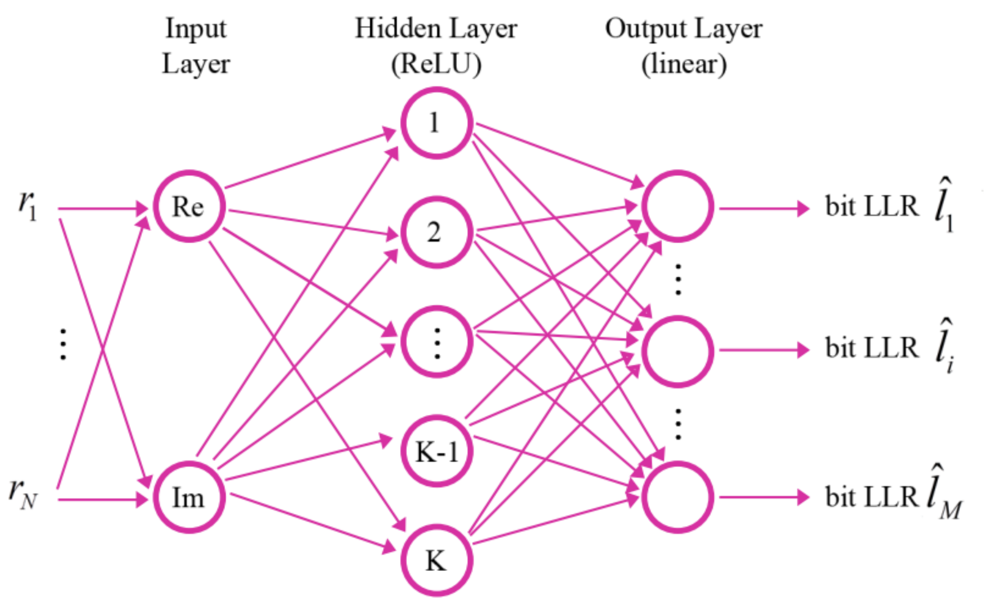
O. Shental and J. Hoydis, ""Machine LLRning": Learning to Softly Demodulate," 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 2019, pp. 1-7

Define this term “Machine LLRning”

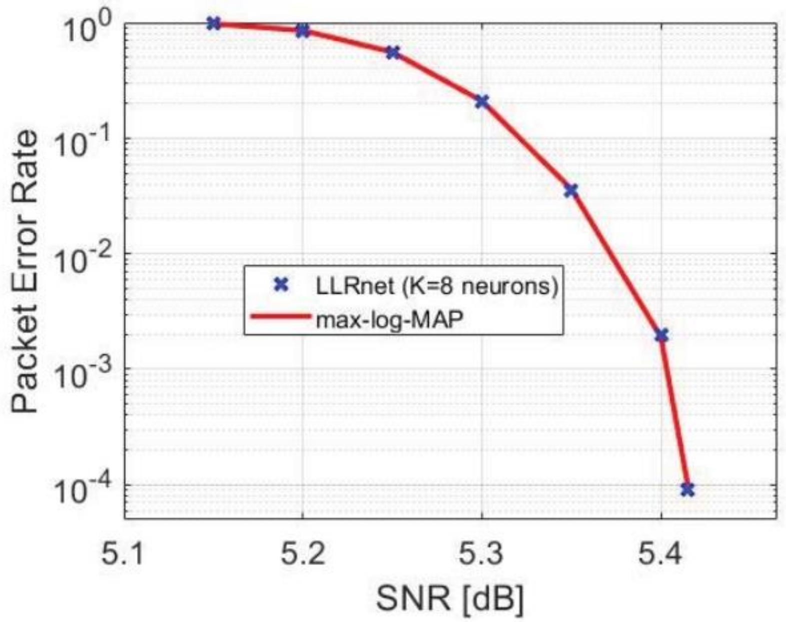
16-QAM (8 neurons) to 1024-QAM (64 neurons)

80% complexity reduction

MSE = 10^{-5}



identical decoding performance



Related Works

2020, R. N. Toledo, C. Akamine, F. Jerji and L. A. Silva {shallow}

R. N. Toledo, C. Akamine, F. Jerji and L. A. Silva, "M-QAM Demodulation based on Machine Learning," 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Paris, France, 2020, pp. 1-6

256-QAM (20 neurons) to 4096-QAM (50 neurons)
6.7% time processing reduction for 4K-QAM
only AWGN is considered

2021, S. K. Vankayala, S. Kumar, K. G. Shenoy et al., {shallow}

S. K. Vankayala, S. Kumar, K. G. Shenoy, D. Thirumulanathan, S. Yoon and I. Kommineni, "Neural Network Architecture for LLR Computation in 5G Systems and Related Business Aspects," 2021 24th International Symposium on Wireless Personal Multimedia Communications (WPMC), Okayama, Japan, 2021, pp. 1-6

1024-QAM (8 neurons) to 4096-QAM (8 neurons)
using special activation function
ignoring some weights during training

Related Works (cont.)

2022, S. Zheng, X. Zhou, S. Chen et al., {deep, 3 hidden layers}

S. Zheng, X. Zhou, S. Chen, P. Qi, C. Lou and X. Yang, "DemodNet: Learning Soft Demodulation from Hard Information Using Convolutional Neural Network," ICC 2022 - IEEE International Conference on Communications, Seoul, Korea, Republic of, 2022, pp. 1-6

16-QAM to 256-QAM (64-128-128)
using convolutional NN with taking hard info. (special preprocessing)

2023, Y. Zhang et al., {deep, 3 hidden layers}

Y. Zhang et al., "Low Complexity Log Likelihood Ratio Estimation Algorithm Based on Neural Network in Decoding," 2023 6th International Conference on Electronics Technology (ICET), Chengdu, China, 2023, pp. 652-657

16-QAM (4-4) to 64-QAM (4-16-8)
construct NN for various SNR.
only AWGN is considered.

Summary

1. Only 5 papers exist.
2. Mainly focus on AWGN channel.
3. NN-based 4K-QAM soft demodulator for **Rayleigh fading** channel has not yet been explored.
4. NN-based QAM soft demodulator for **M>4096** has not yet been mentioned.

4

Research Methodology

NN Architecture

4 inputs:

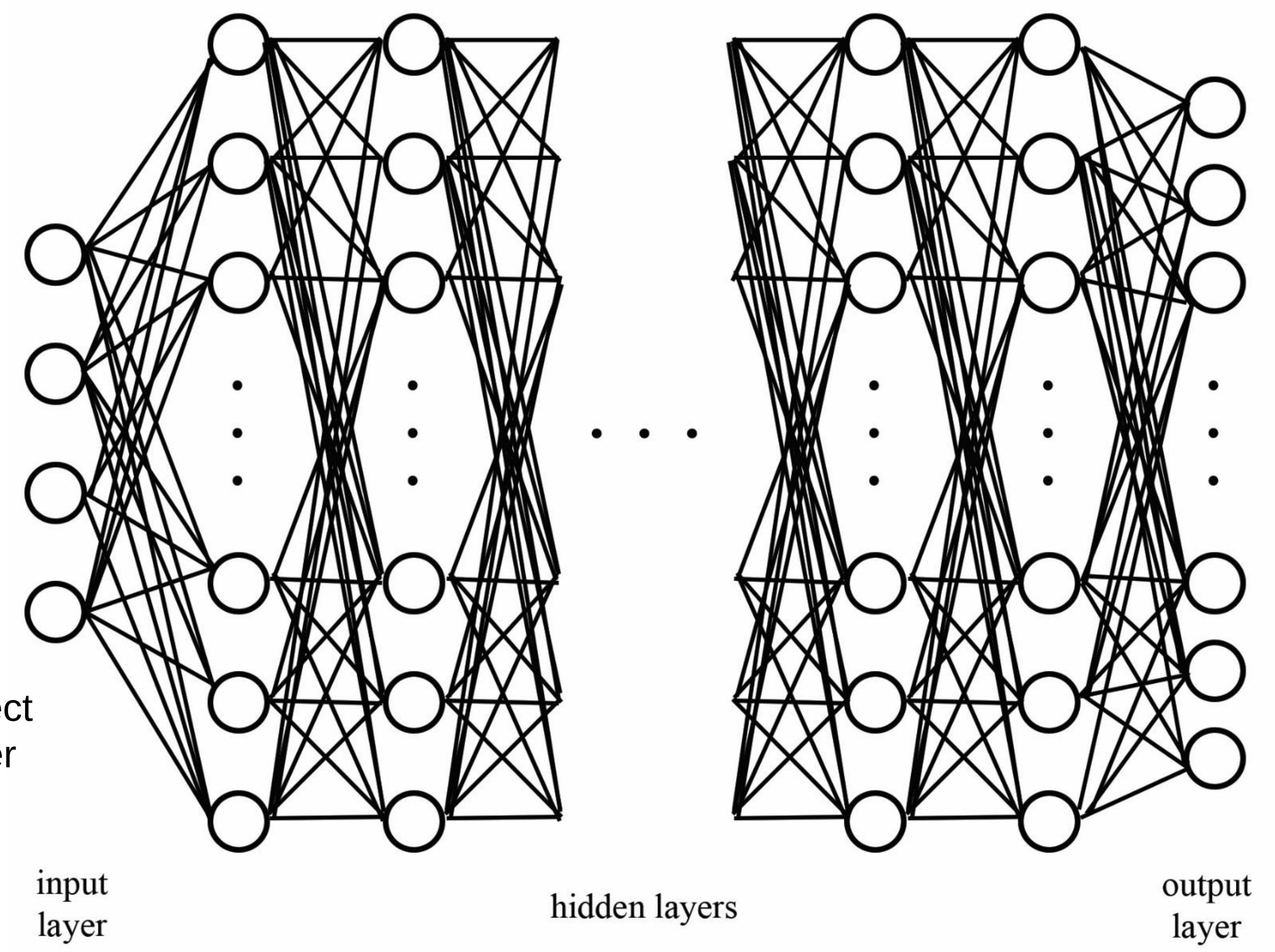
$\mathrm{Re}(r)$

$\mathrm{Im}(r)$

$\mathrm{Re}(h)$

$\mathrm{Im}(h)$

assuming perfect
C.S.I at receiver



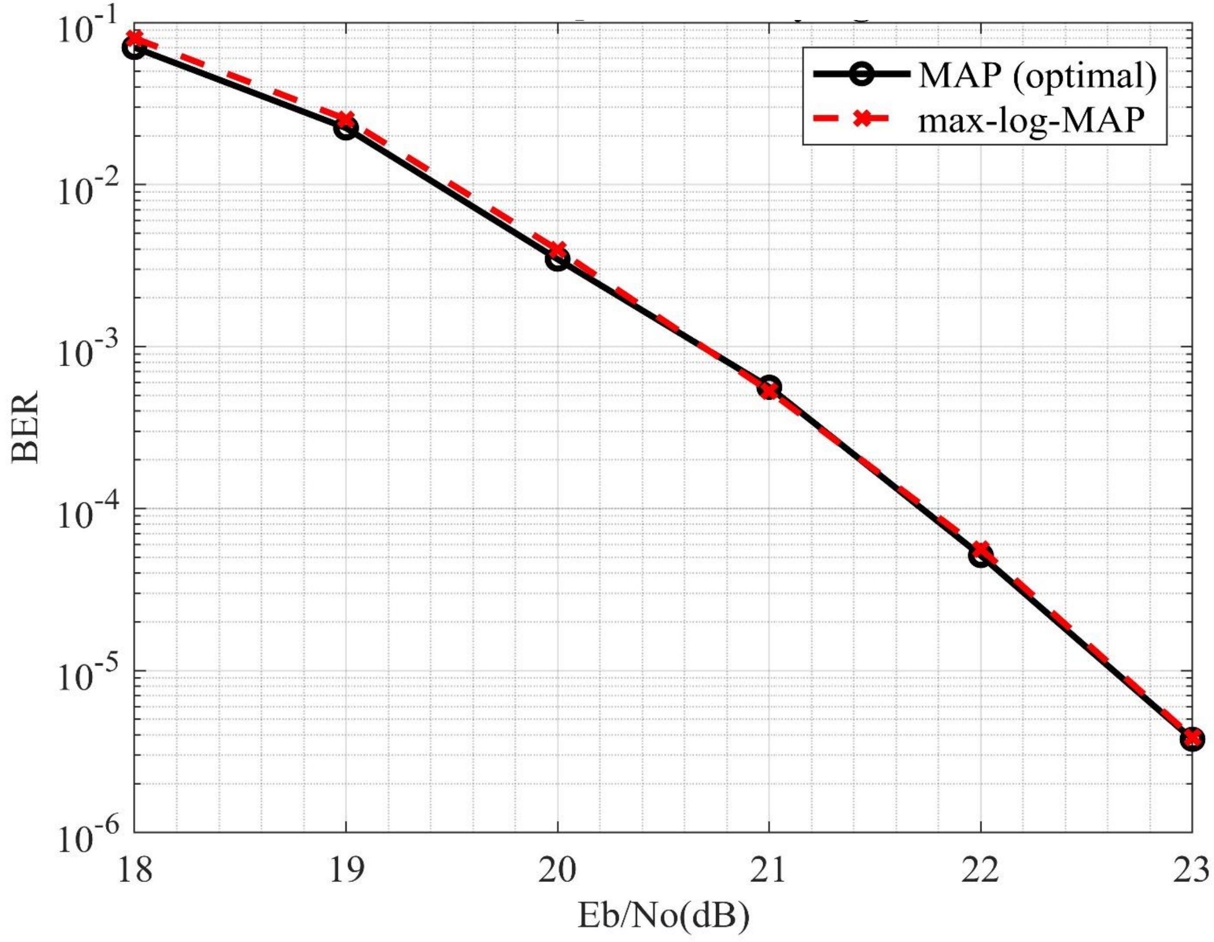
***m* outputs:**

$\overline{\ell} = [\ell_1, \ldots \ell_m]$

LLR

LLR Characteristic in 4K-QAM

(648,324) LDPC with 4K-QAM over Rayleigh channel



We prefer to use LLR calculated from max-log-MAP as an input for NN training.

- numerical stability
- time consuming
- only for 4K-QAM

Data Collection: Input for Training

inputs responses

$$r^1 \leftrightarrow \overline{\ell}^1 = [\ell_1, \ldots \ell_m]$$

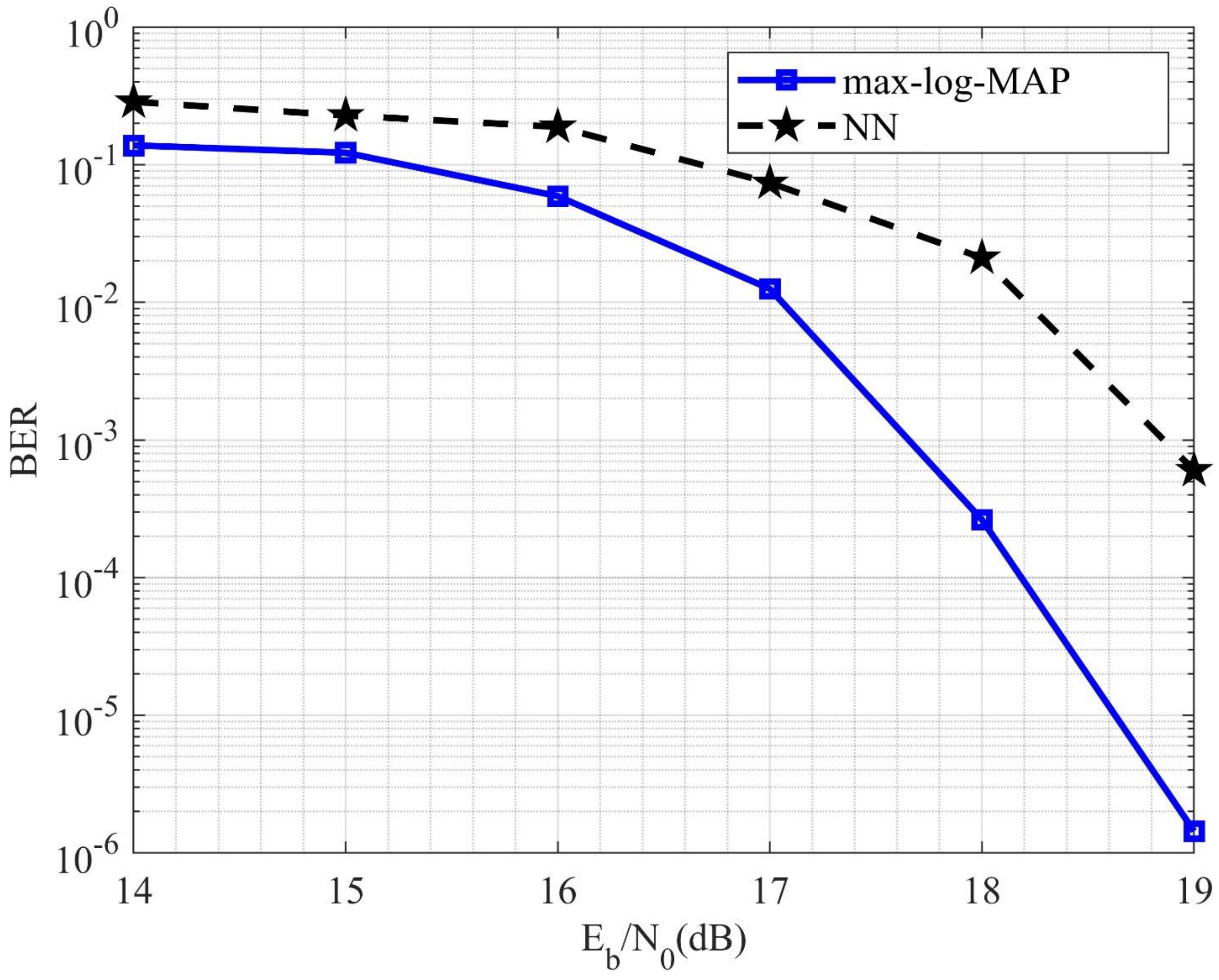
$$r^2 \leftrightarrow \overline{\ell}^2 = [\ell_1, \ldots \ell_m]$$

$$\vdots$$

$$r^k \leftrightarrow \overline{\ell}^k = [\ell_1, \ldots \ell_m]$$

here are feature/response pairs

It Is Not Easy Task

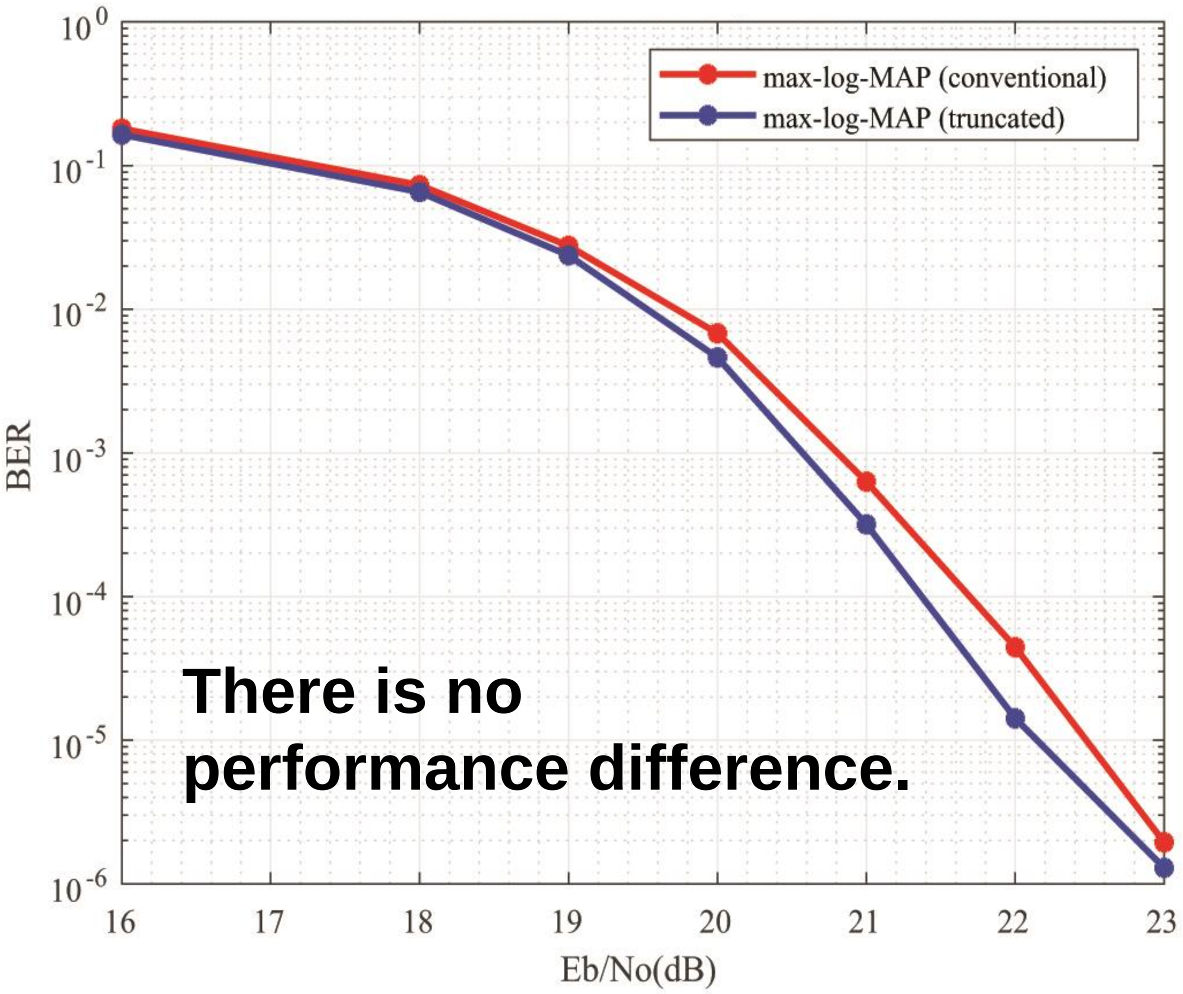


Simply put a large collection of feature/response pairs into NN does not work !

$$k \to \infty$$

Loss does not go to 0.

Take a Look at Training Input



There is no performance difference.

LLR truncation

$$\ell_i = \begin{cases} -20 \;, & \ell_i \le -20 \\ \ell_i \;, & -20 < \ell_i < 20 \\ 20 \;, & \ell_i \ge 20 \end{cases}$$

So, the magnitude of LLR is not sensitive to performance.

Finite range of LLR is hardware-friendly.

Define A New Measurement

Accuracy regarding LLR sign agreement

$$\boxed{\kappa = \frac{1}{km}\sum_{i=1}^{km} f\left(\ell_i^{\text{exact}}, \ell_i^{\text{predict}}\right)}$$

where, $f\left(\ell_i^{\text{exact}}, \ell_i^{\text{predict}}\right) = \begin{cases} 1\ , & \text{sign}\left(\ell_i^{\text{exact}}\right) = \text{sign}\left(\ell_i^{\text{predict}}\right) \\ 0\ , & \text{sign}\left(\ell_i^{\text{exact}}\right) \neq \text{sign}\left(\ell_i^{\text{predict}}\right) \end{cases}$

no sign difference → 100% accuracy

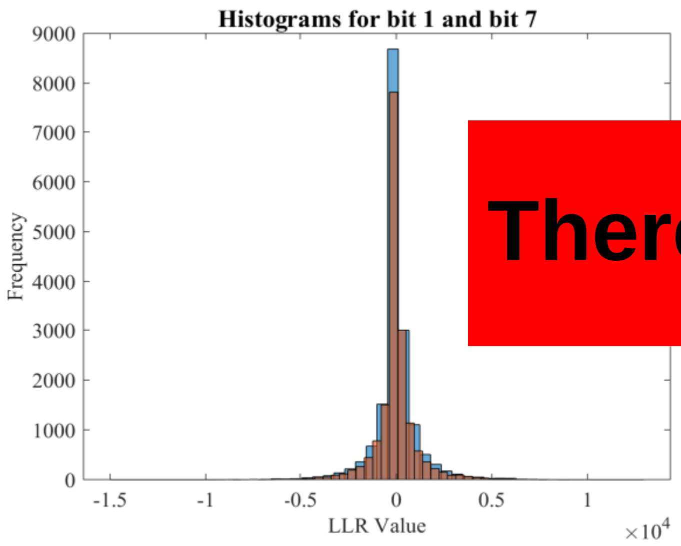
We found this !

Apply previous measurement per bit

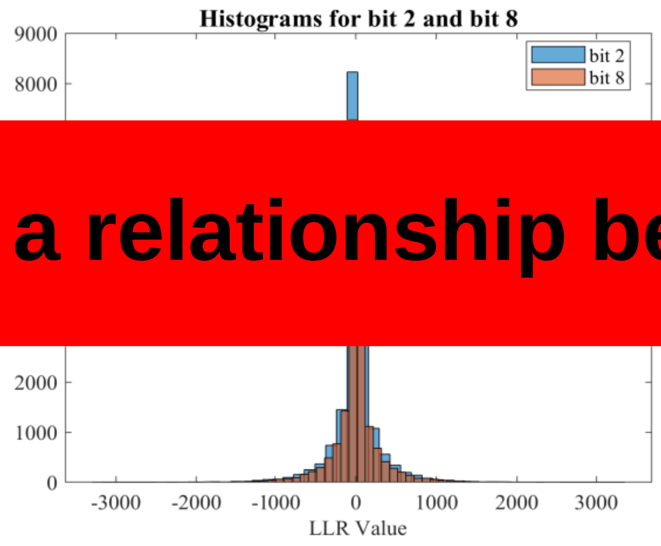
Bit	1	2	3	4	5	6
ACC	1.00	0.99	0.98	0.87	0.57	0.54
Bit	**7**	**8**	**9**	**10**	**11**	**12**
ACC	1.00	0.98	0.97	0.89	0.57	0.56

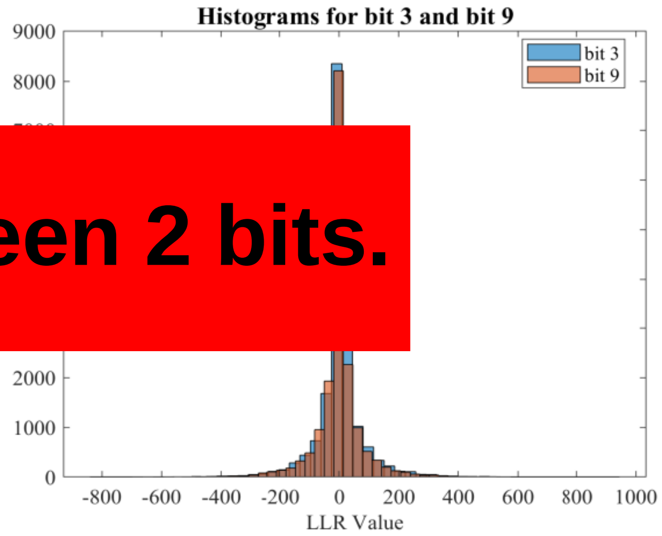
Here is the problem.

AWGN channel
#dataset = 100k
NN (shallow): 256 nodes

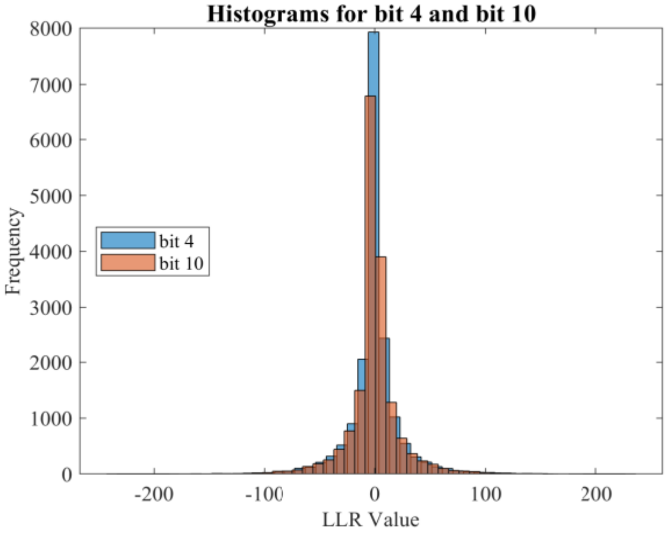


(a) Histogram for Bit 1 and Bit 7.



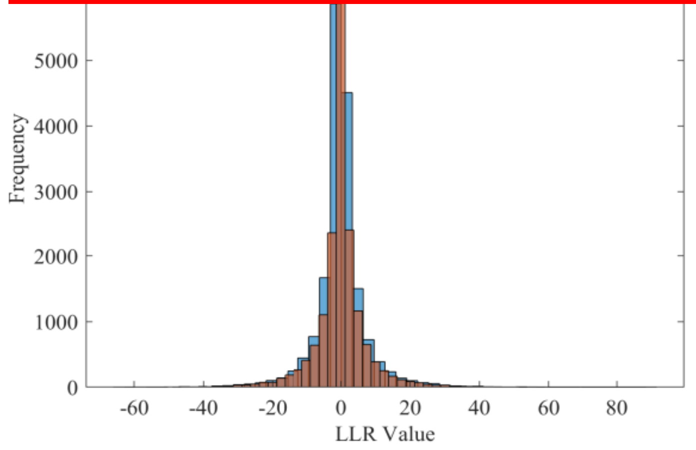


There is a relationship between 2 bits.

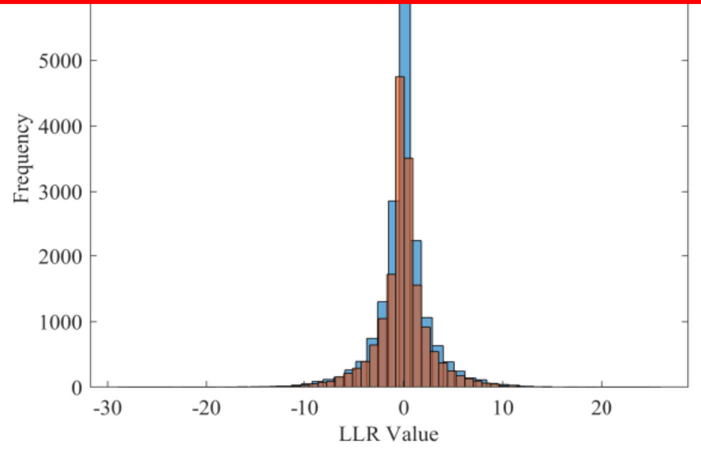


(d) Histogram for Bit 4 and Bit 10.

high level of ambiguity at bit 5 6 11 12

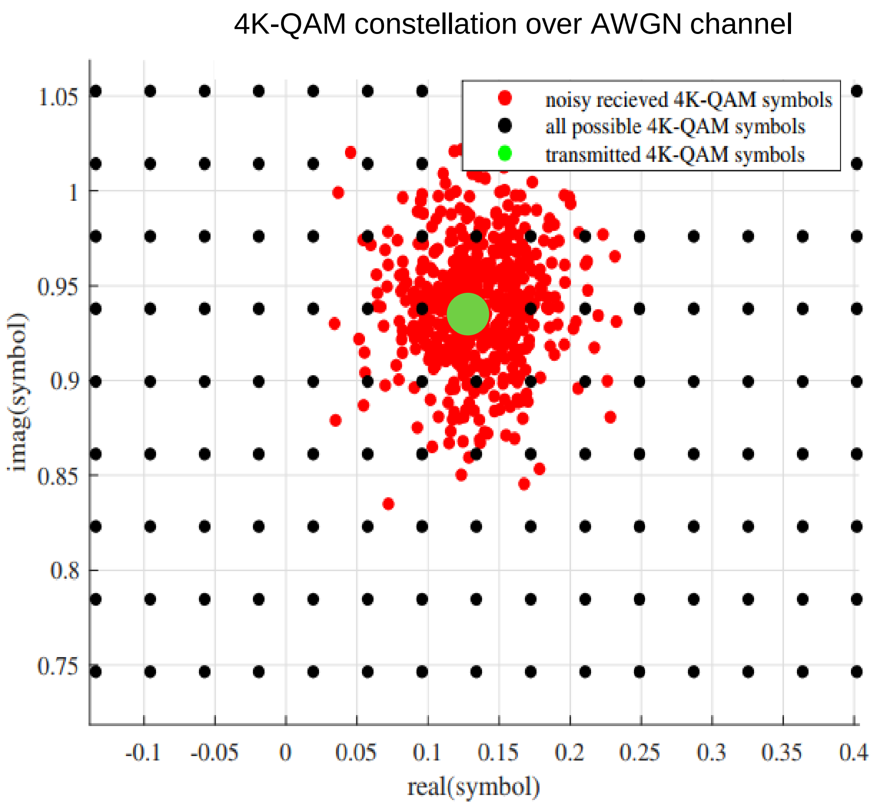


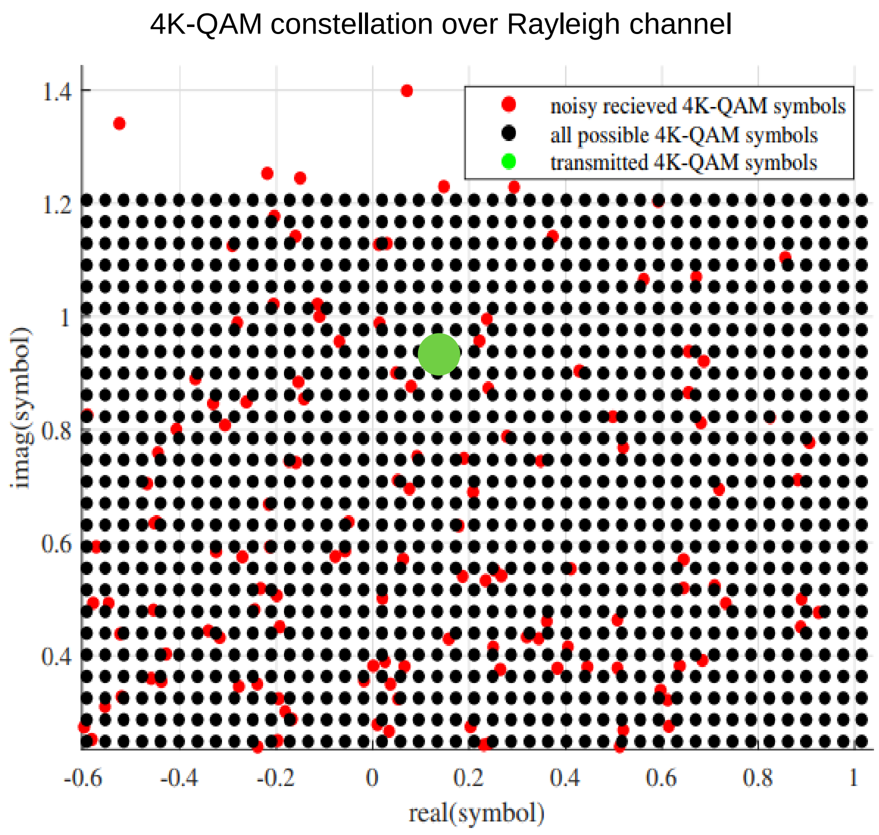
(e) Histogram for Bit 5 and Bit 11.



(f) Histogram for Bit 6 and Bit 12.

Difficulty of Rayleigh Fading Channel





training NN for **Rayleigh** case is more harder than **AWGN**

(need deeper NN architecture)

Proposed Trainable Framework

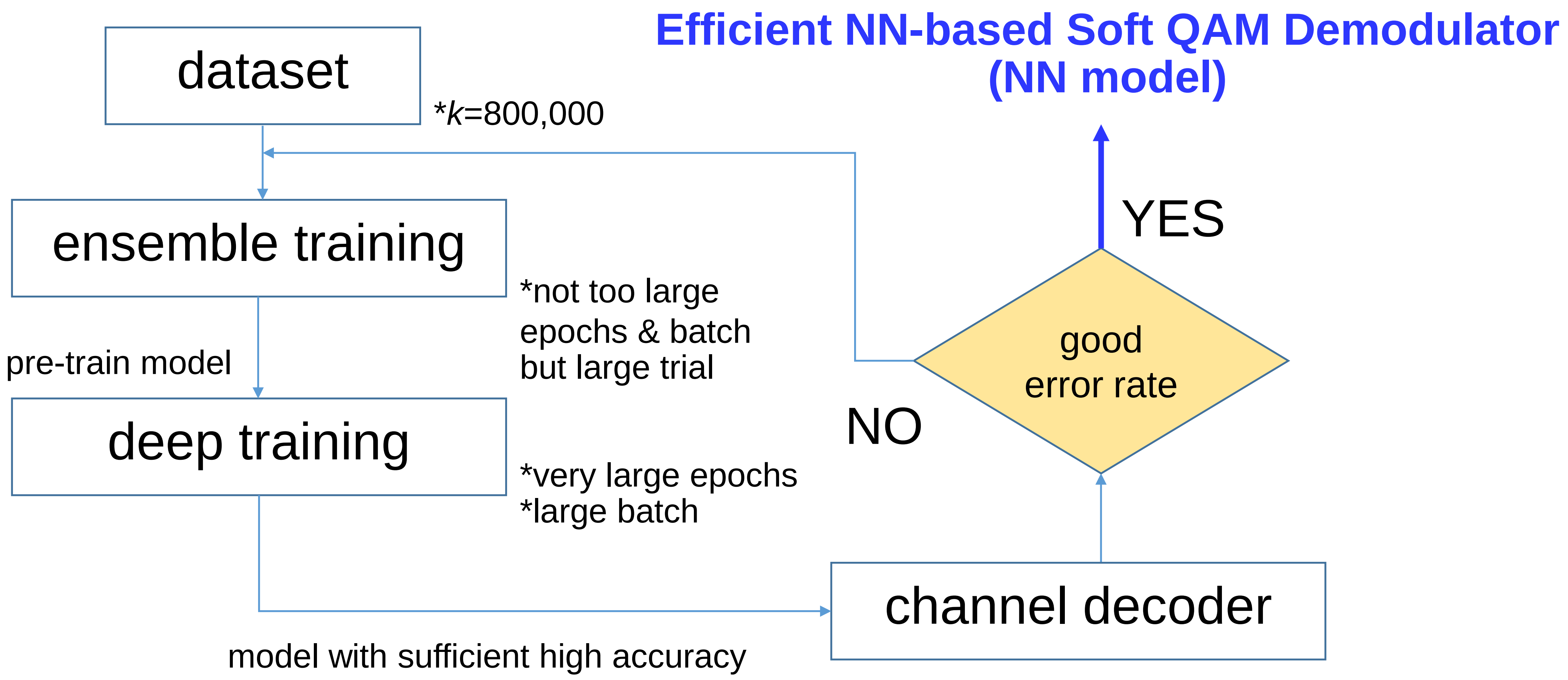
Data-preprocessing :

- 4 input feature needs z-normalization
- input response needs special **min-max scaling**

Specific training parameters :

- fully connected feed forward neuron network
- learning rate : 0.0001
- optimizer: SGDM
- L2 regularization (penalized rate 0.001)
- batch normalization layer
- activation function : LeakyReLu
- loss function: MSE

Proposed Trainable Framework (cont.)



5

Results & Discussion

Accuracy for NN-based 4K-QAM

full option	99 %
only preprocessing	75 %
only network tweaking	69 %
original data	55 %

Train NN with proposed framework
achieve almost perfect accuracy.

Complexity in Terms of FLOPs

For M-QAM

$$\mathrm{FLOPs}_{\mathrm{log-MAP}} = 56M\log_2 M + 51\log_2 M$$

$$\mathrm{FLOPs}_{\mathrm{max-log-MAP}} = 5M\log_2 M + 8\left(\log_2 M\right)^2 - 6\log_2 M$$

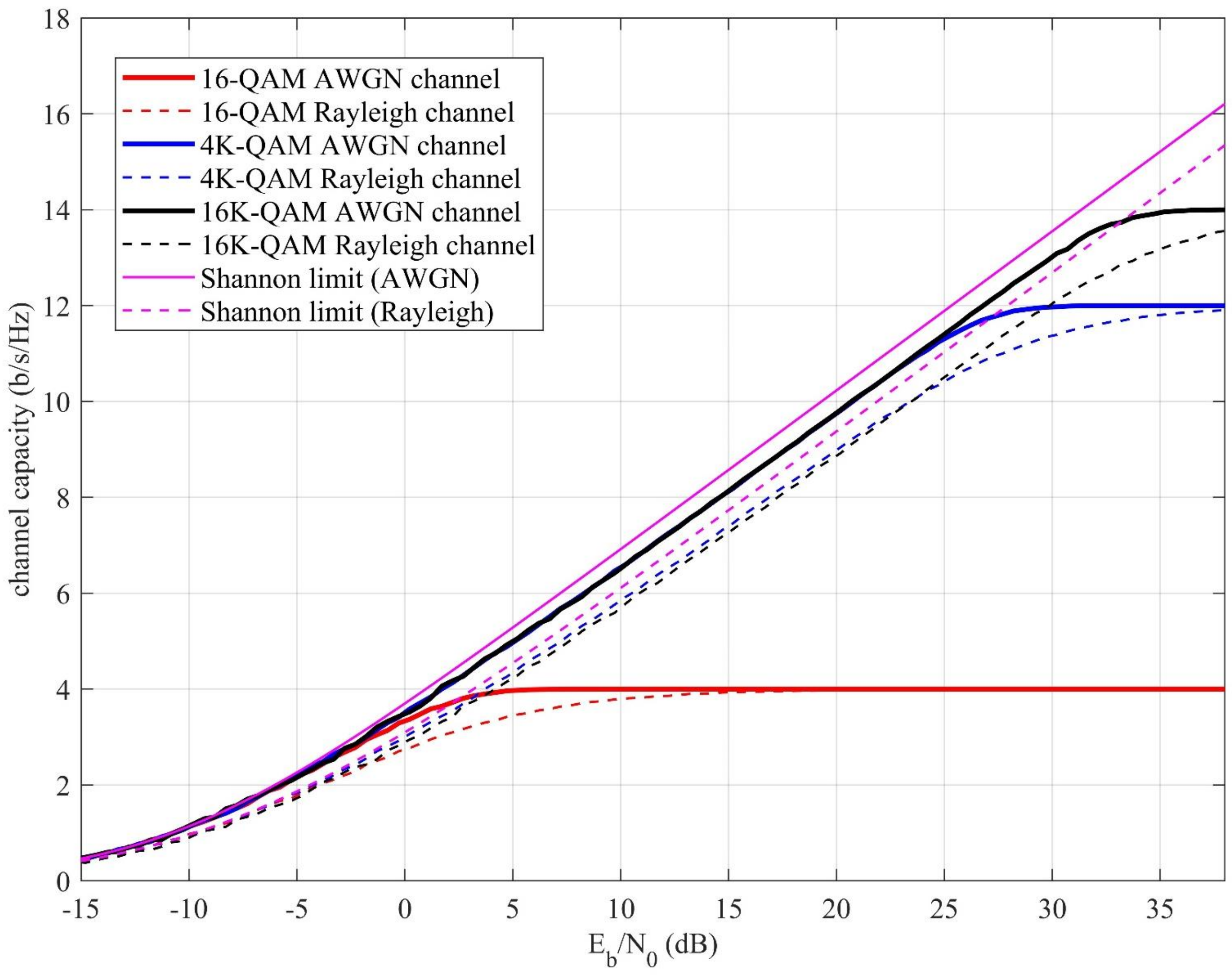
For 4K-QAM

$$\mathrm{FLOPs}_{\mathrm{NN-4inputs}} = 34n_p + \left(n_h - 1\right)\left(2n_p^2 + 4n_p\right) - 1$$

n_p : #nodes per 1 hidden

n_h : #hidden layers

Channel Capacity as Reference



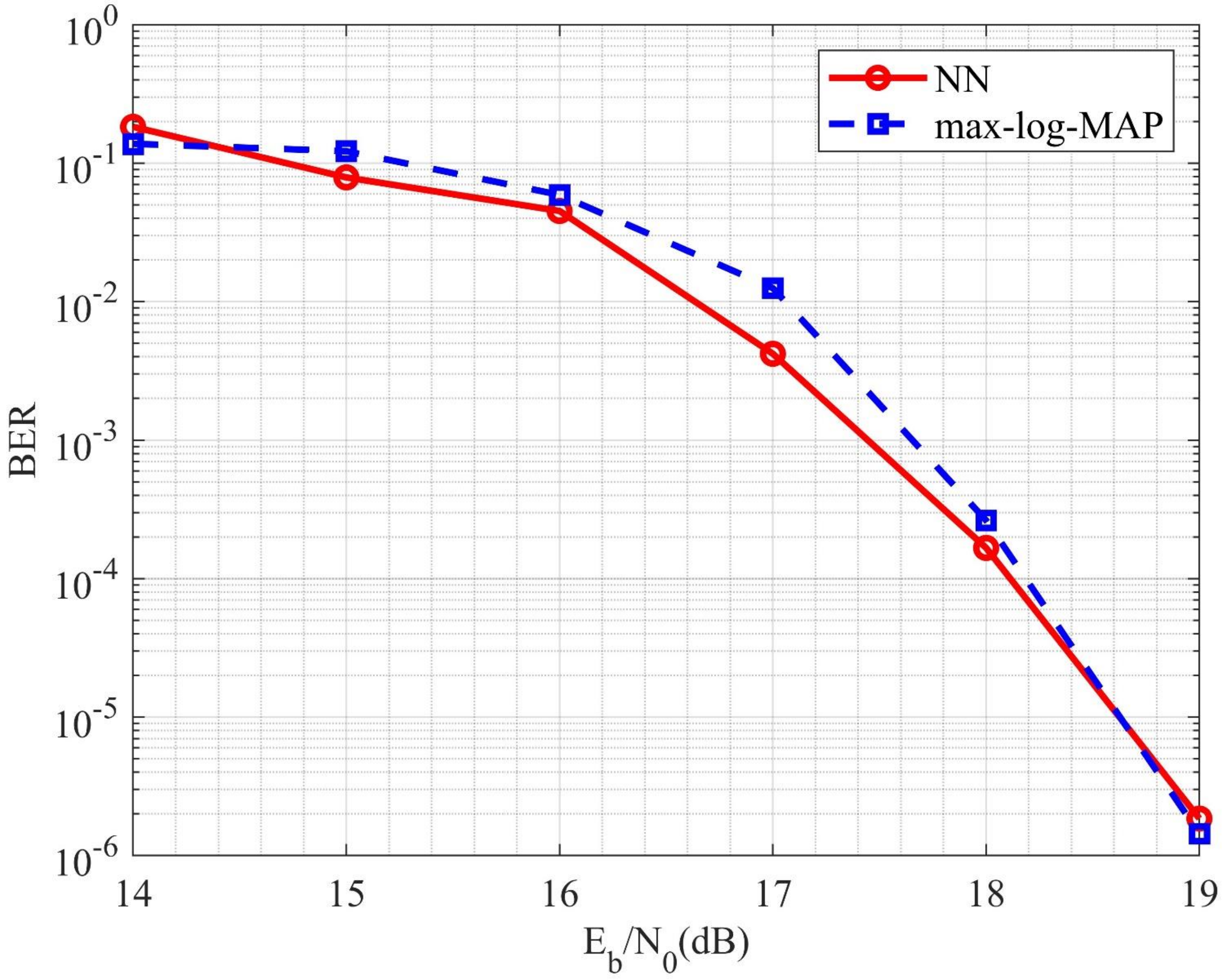
minimum SNR
for reliable transmission

6 bps/Hz ~ 11 dB
(Rate = 1/2 with 4K-QAM)

10 bps/Hz ~ 20 dB
(Rate = 5/6 with 4K-QAM)

7 bps/Hz ~ 13 dB
(Rate = 1/2 with 16K-QAM)

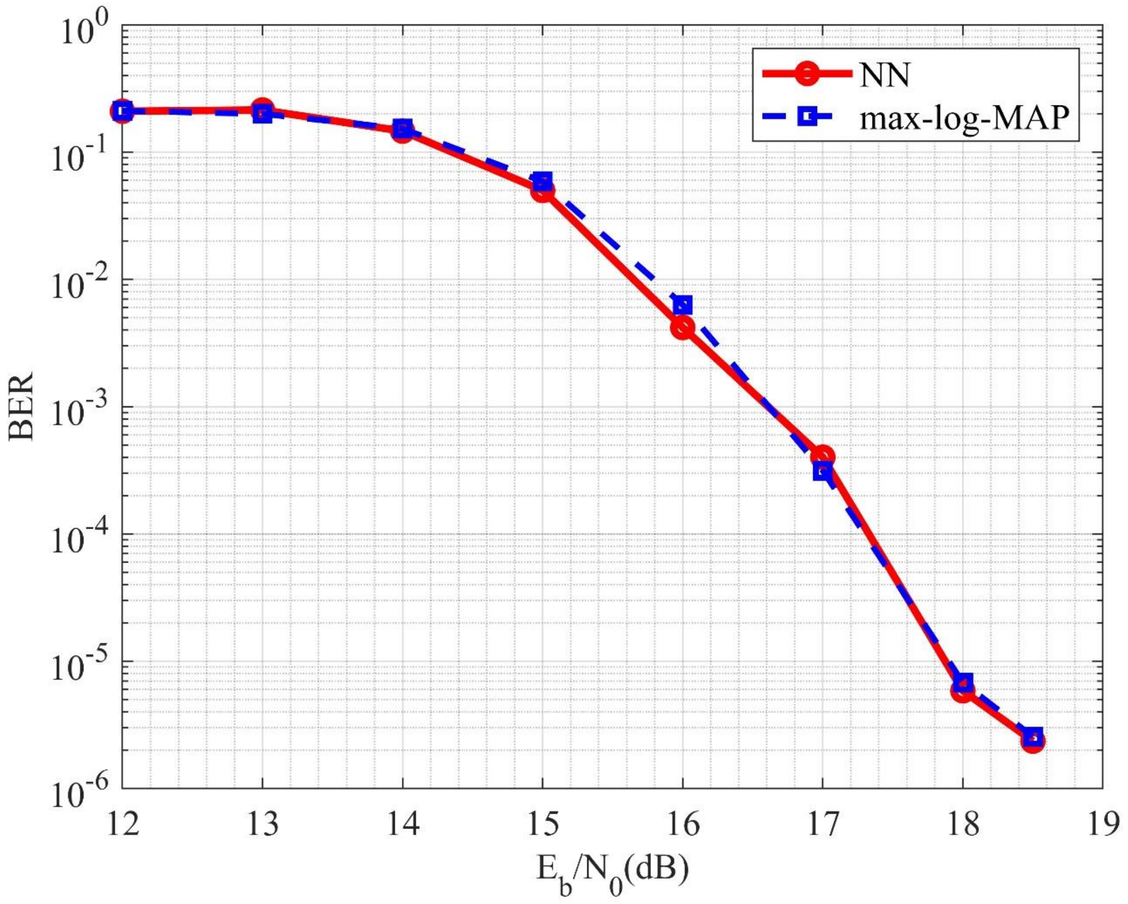
Results



(648,324) LDPC
4096-QAM
AWGN channel
shallow NN (256 nodes)

99.68%
complexity reduction

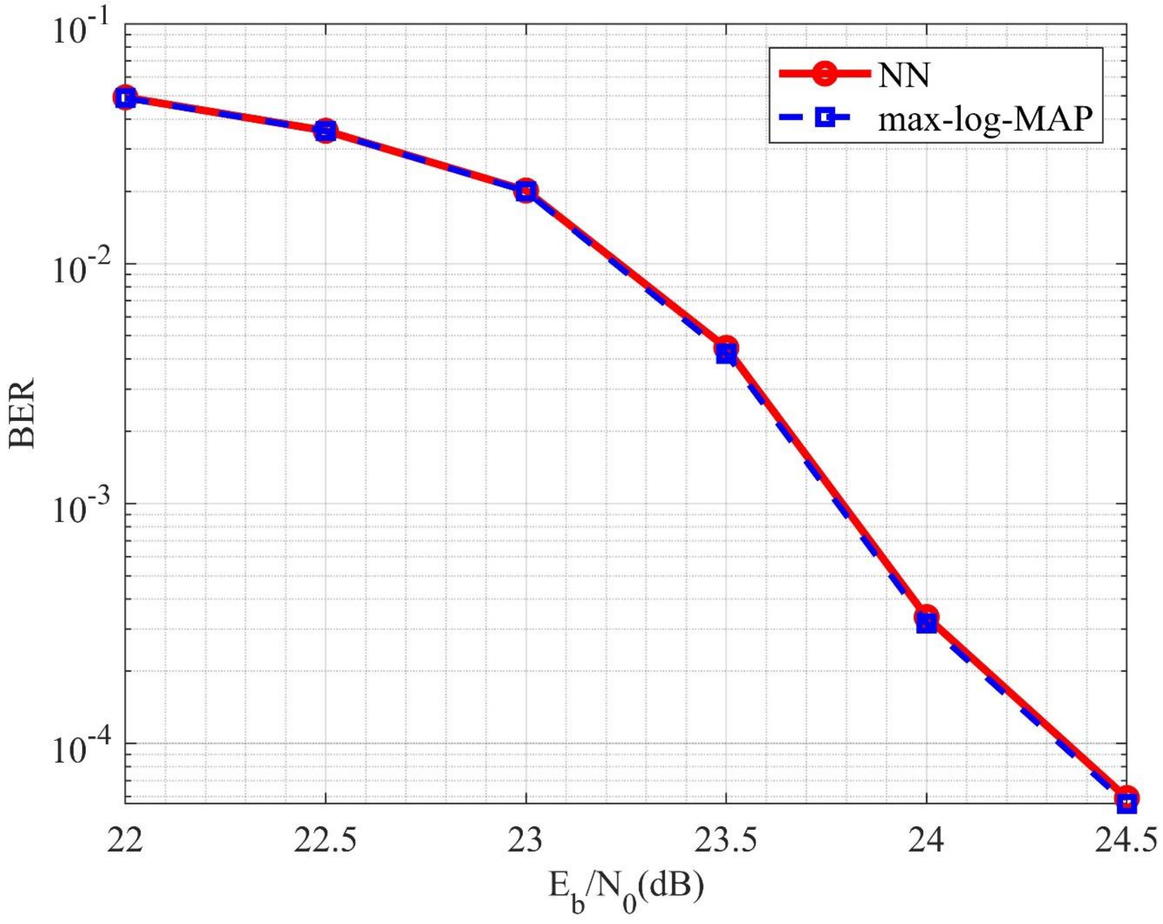
Results (cont.)



(1296,648) LDPC
4096-QAM
AWGN channel
shallow NN (256 nodes)

99.68%
complexity reduction

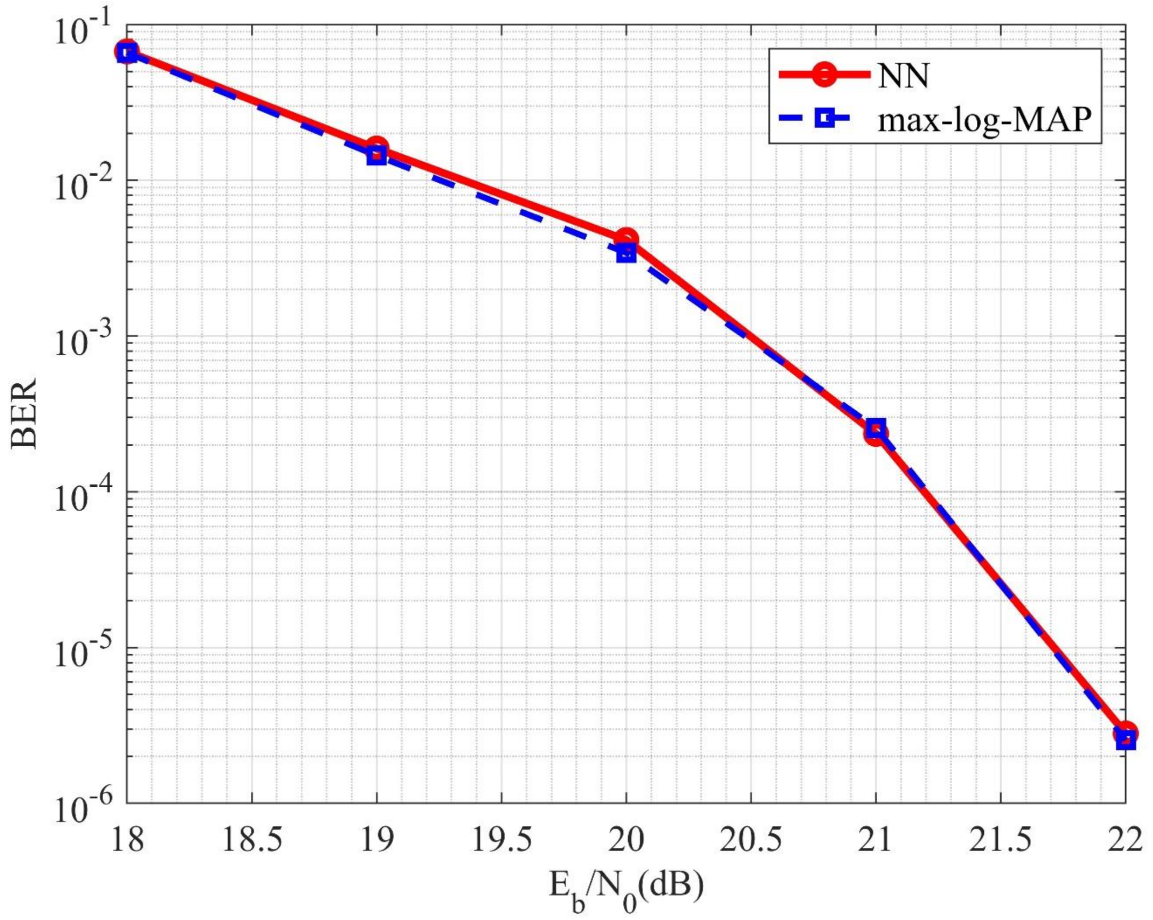
Results (cont.)



(1944,1260) LDPC
4096-QAM
AWGN channel
shallow NN (256 nodes)

99.68%
complexity reduction

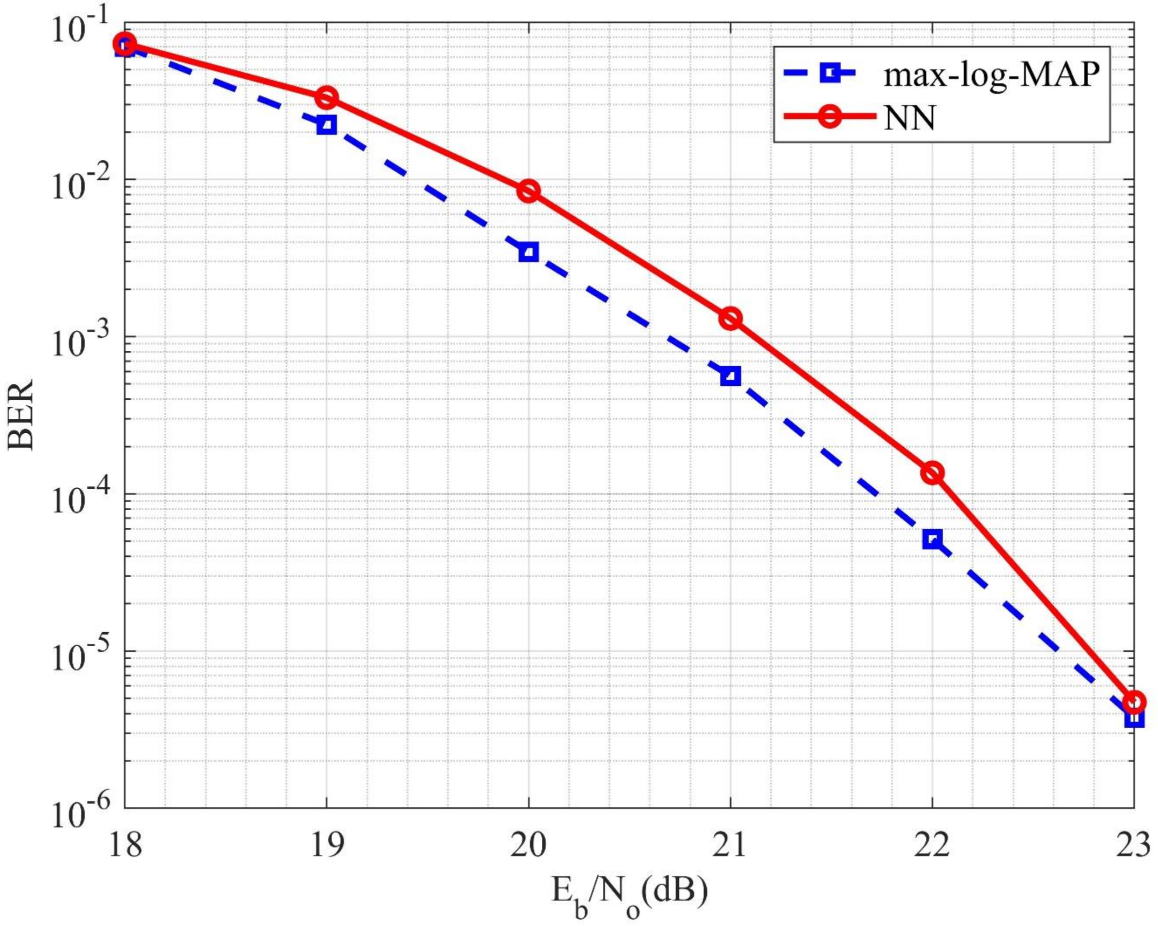
Results (cont.)



(1944,1260) LDPC
16384-QAM
AWGN channel
deep NN
(128-128-128)

97.44%
complexity reduction

Results (cont.)



(648,324) LDPC
16384-QAM
Rayleigh fading channel
deep NN
(128-128-128-128-128-128-128-128-128)

90.32%
complexity reduction

6

Conclusion

Conclusions

To the best of our knowledge,
this is the first work that considers this machine learning problem seriously.

- We propose a ***trainable framework*** for ***NN-based QAM soft demodulator*** for $M \geq 4096$.

- Proposed NN-based 4K-QAM soft demodulator can be efficiently utilized with ***significant complexity reduction***.

Problem : all numerical experiments are time-consuming !

Future work : we expect that NN architecture can be optimized
(for further complexity reduction) !!

Publication

Accepted for presentation in 2024 6th International Conference on Computer Communication and the Internet (ICCCI 2024), Tokyo (with IEEE Conference Proceedings)

Preparing manuscript for IEEE Comm. Letter submission

Invitation to participate in the presentation of thesis defense

INDIE-CT
RESEARCH LAB

Thank you for your kind attention

any questions or suggestions are appreciated

Ms. Donlaporn Triamwichanon
Master student, Electrical Engineering,
Faculty of Engineering, Khon Kaen University

References

[1] D. Jaeger, “DVB-C2—System aspects and performance analysis,” in Proc. DVB-C2 Seminar CableLabs Winter Conf., Denver, CO, USA, Feb. 2010. [Online]. Available: https://www.yumpu.com/en/document/read/37639234/dvb-c2- system-aspects-and-performance-analysis-ict-redesigneu

[2] “Asymmetric digital subscriber line (ADSL) transceivers 2 (ADSL2),” Int. Telecommun. Union, Geneva, Switzerland, ITU-T Recommendation G.992.3, Jul. 2002.

[3] “Unified high-speed wire-line based home networking transceivers— System architecture and physical layer specification,” Int. Telecommun. Union, Geneva, Switzerland, ITU-T Recommendation G.9960, Nov. 2018.

[4] "IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016) , vol., no., pp.1-4379, 26 Feb. 2021, doi: 10.1109/IEEESTD.2021.9363693.

[5] IEEE Standard For Information Technology–Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Networks–Specific Requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, IEEE Standard 802.11, Jul. 2010.