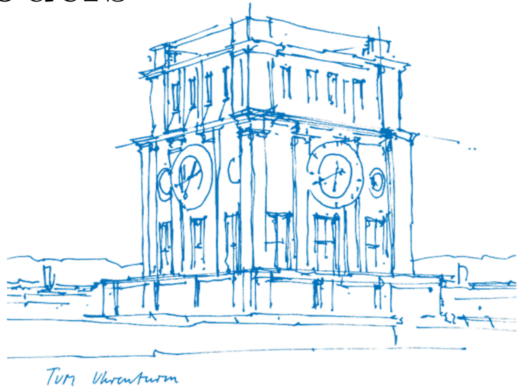
Geometric and Probabilistic Constellation Shaping with Autoencoders

Research Internship



David de Andrés Hernández
Technical University of Munich

Institute for Communications Engineering

October 13, 2022

Agenda

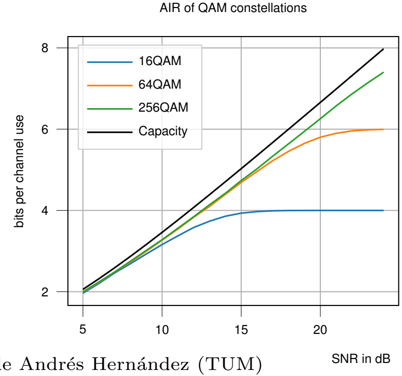
1. Introduction
   - SNR Gap
   - Probabilistic Constellation Shaping
   - Geometric Shaping
2. Autoencoders
   - Challenges
3. Contribution
   - First Implementation
   - Second Implementation
4. Conclusions

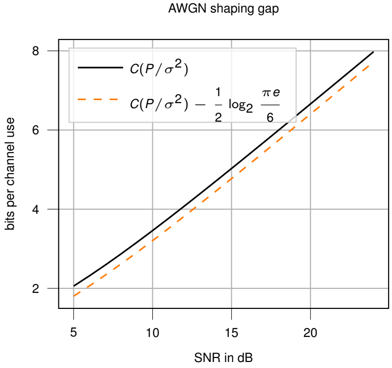
Introduction

- We want to make use of each channel's capacity as efficiently as possible

$$C = \max_{p(X):\mathbb{E}[X^2]\leq P} \mathbb{I}(X;Y)$$

- The optimal $p(x)$ has only been found for specific channels, such as the AWGN, since knowledge of the channel distribution $p(y|x)$ is required.





Closing the SNR Gap

- ASK and QAM modulation schemes are penalized for two reasons:
  1. They use uniform probability densities
  2. The constellation points are equidistant
- Solution 1: Shape the probability of occurrence of the constellation points — Probabilistic Constellation Shaping
- Solution 2: Shape the space location of the constellation points — Geometric Constellation Shaping

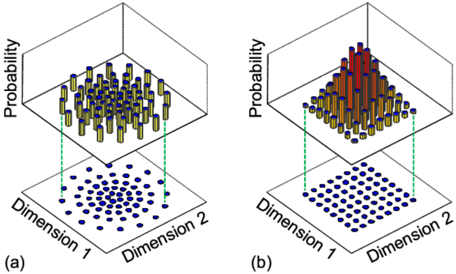


Figure: (a) Probabilistic Constellation Shaping, (b) Geometric Constellation Shaping; [CW19]

Why does it makes sense to use ML for this problem?

- Finding the constellation parameters when $p(y|x)$ is very complex or unknown can be mathematically untractable
- NN have the property of being universal function approximators [HSW89]
- O'Shea and Hoydis [OH17] pioneered the idea of interpreting the complete communication system as an autoencoder

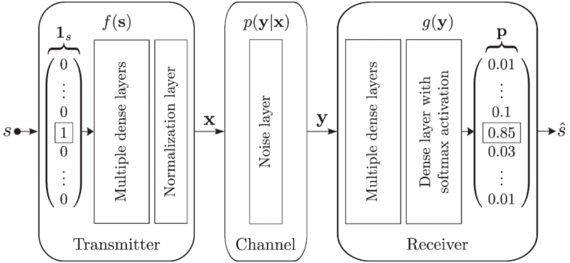


Figure: Autoencoder architecture proposed by [OH17].

Autoencoders

- Idea: transmit a particular representation of the input data so that at the output, it can be reconstructed with minimal error
- This representations must be robust with respect to the channel impairments (i.e. noise, fading, distortion, etc.) — bottleneck in the autoencoder jargon
- The autoencoder is implemented using Feed-forward Neural Networks (FFNN), and the parameters are learned using Stochastic Gradient Descent (SGD)

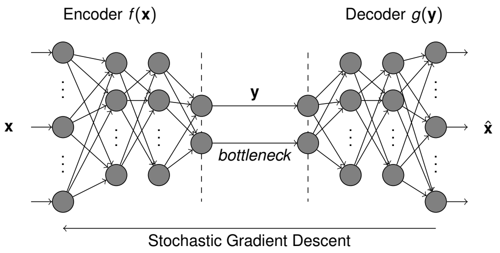
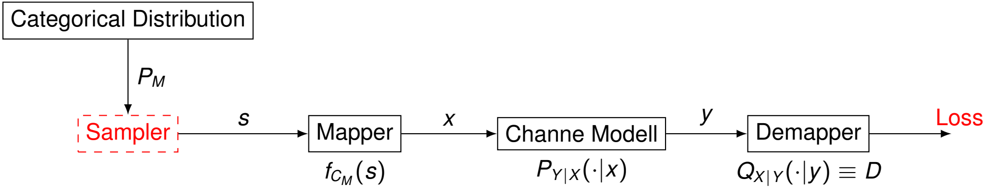


Figure: Representation of the NN of an autoencoder.

Challenges



- To find fitting sets of parameters $\{P_M, C_M, D\}$ we define a loss function, $L(P_M, C_M, D)$, that compares the current output of the autoencoder with the desired output from the training set.
- The most used algorithm is SGD, which trains any parameter, θ, as

$$\theta_{new} = \theta_{old} + \epsilon \nabla L(\theta_{old})$$

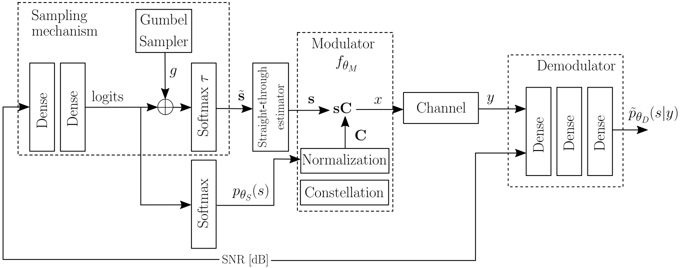
- To compute the gradient efficiently a computational graph stores the transformations to the factors which influenced the loss function.

First implementation [SAAH19]

Trainable parameters:

- P_M, source's probability distribution learnt by the encoder.
- C_M, spatial distribution of the constellation points learnt by the mapper.
- D, posterior probability distribution learnt by the demapper.

Autoencoder architecture:



Loss Function

The goal of probabilistic constellation shaping is to maximize the MI. To this end, defining an appropriate loss function is critical. Starting from the demodulator, the categorical cross entropy loss

$$L(D, P_M, C_M) \triangleq \mathbb{X}(P_{X|Y}||Q_{X|Y}; D) = \mathbb{E}\left[-\log_2(Q(X|Y; D))\right]$$

is appropriate for training D and C_M, but not P_M. To see why, we rewrite the MI as

$$\mathbb{I}(X, Y) = \mathbb{H}(X) - \mathbb{H}(X|Y)$$

$$\mathbb{I}(X, Y) = \mathbb{H}(X) - \mathbb{X}(P_{X|Y}||Q_{X|Y}) + \mathbb{D}(P_{X|Y}||Q_{X|Y}).$$

And the Loss becomes

$$L(D, P_M, C_M) = \mathbb{H}(X) - \mathbb{I}(X, Y) + \mathbb{D}(P_{X|Y}||Q_{X|Y}).$$

So, if L is minimized during training, the source entropy is unwantedly minimized.

Loss Function (cont'd)

To avoid this effect, Stark *et al.* modify the loss function as

$$\hat{L}(D, P_M, C_M) \triangleq L(D, P_M, C_M) - \mathbb{H}(X).$$

With this correction the optimization problem

$$\min_{D, P_M, C_M} \hat{L}(D, P_M, C_M) = \max_{D, P_M, C_M} \{\mathbb{I}(X, Y) - \mathbb{D}(P_{X|Y} || Q_{X|Y})\}$$

maximizes the MI.

Gumbel-Softmax trick [JGP16]

- Solves the problem of backpropagating through stochastic nodes by reparametrizing the samples to avoid breaking the dependency between the samples and the trainable parameters.
- Relaxes the argmax function using a softmax instead, which is smooth in $\tau > 0$. The parameter τ, controls the degree of approximation to the expected value of the categorical distribution.

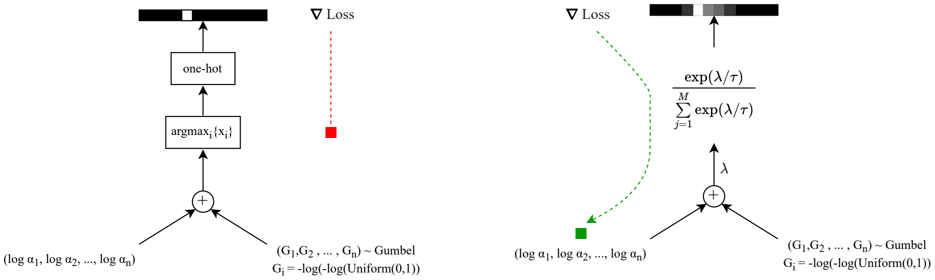
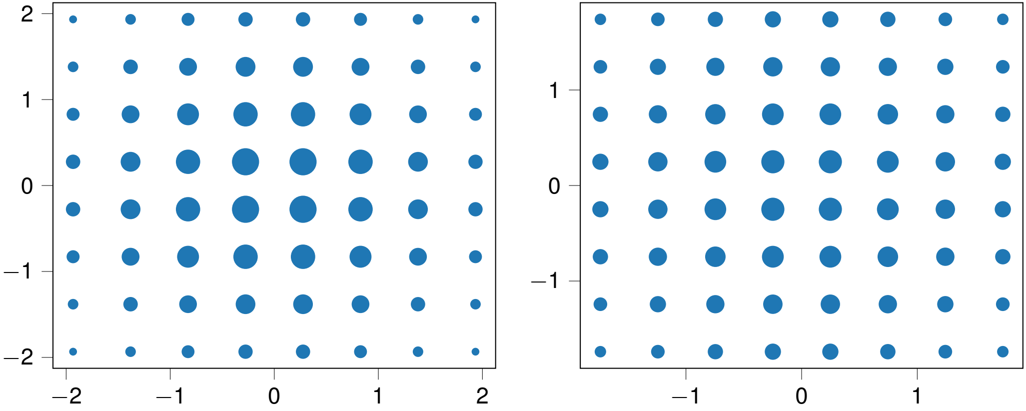


Figure: Non-differentiable path vs. differentiable path using the Gumbel-Softmax trick.

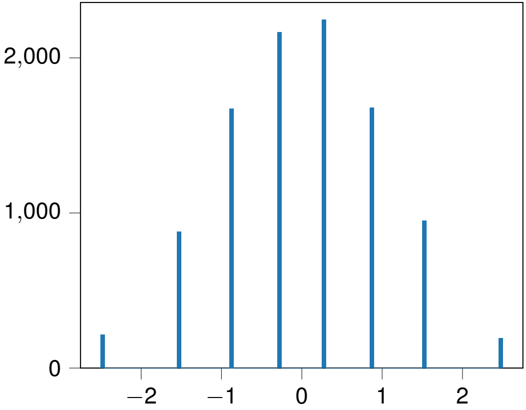
Probabilistic Constellation Shaping



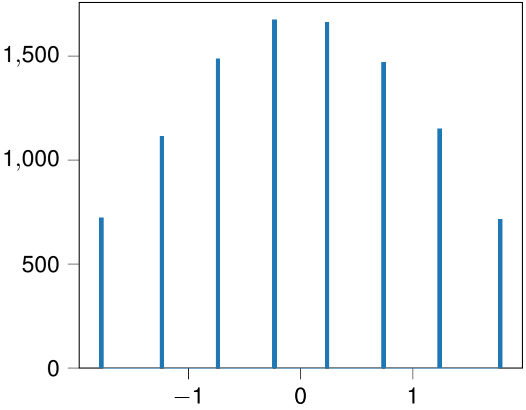
(a) 5dB (b) 18db

Figure: Learnt probabilistic constellation shaping for M = 64. The size of the markers is proportional to the transmission probability of the symbol. When trained under 5dB, the probabilistic shaping approaches a Gaussian. While under 18dB it approaches a uniform distribution.

Joint Probabilistic and Geometric Shaping



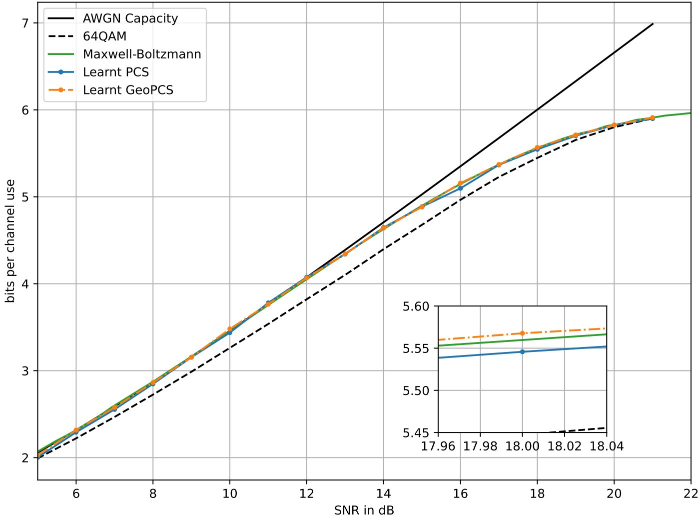
(a) SNR = 5dB



(b) SNR = 18dB

Figure: Learnt joint geometric and probabilistic ASK constellations for M=8.

Overall Performance



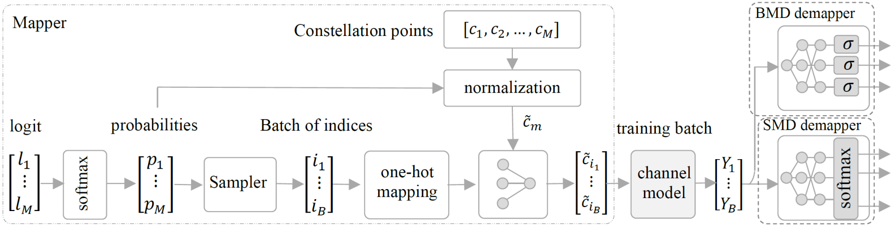
Second implementation [AC21]

What motivates another implementation?

The Gumbel-Softmax trick is complex and numerically unstable.

Trainable parameters:

- P_M, source's probability distribution learnt by the sampler.
- C_M, spatial distribution of the constellation points learnt by the mapper.
- D, posterior probability distribution learnt by the demapper.



Loss Function

The goal is again to to maximize the mutual information

$$\max_{D,P_M,C_M} \mathbb{I}\left(X, Y; D, P_M, C_M\right) = \mathbb{H}(X) - \mathbb{X}(P_{X|Y} \| Q_{X|Y}; D, P_M, C_M). \tag{1}$$

Typically, through SGD we adjust the trainable parameters as:

$$\theta_{new} = \theta_{old} + \epsilon \frac{\partial}{\partial \theta_{old}} \mathbb{I}\left(X, Y; \theta_{old}\right) \tag{2}$$

for all trainable parameters $\theta \in P_M, C_M, D$. And the MI can be numerically approximated by

$$\mathbb{I}\left(X, Y\right) \approx \mathbb{I}\left(X, Y\right)_{\text{num}} = \frac{1}{B} \sum_{i=1}^{B} -\log_2(P(x_i)) + \log_2(Q_{X|Y}(x_i|y_i)) \tag{3}$$

$$= \frac{1}{B} \sum_{i=1}^{B} L(x_i, y_i). \tag{4}$$

Next, the following approximation usually allows to adjust the trainable parameters:

$$\frac{\partial}{\partial\theta}\mathbb{I}(X, Y; \theta) \approx \frac{\partial}{\partial\theta}\mathbb{I}(X, Y)_{\text{num}} = \frac{1}{B}\sum_{i=1}^{B} L(x_i, y_i). \tag{5}$$

However, Aref claims that although this is true for the constellation locations ($\theta \in C_M$) and the demapper parameters ($\theta \in D$), it does not hold for the constellation probabilities $\{p_1, p_2, \ldots, p_M\} = P_M$

$$\frac{\partial}{\partial p_j}\mathbb{I}(X, Y; P_M) \not\approx \frac{1}{B}\sum_{i=1}^{B} \frac{\partial}{\partial p_j} L(x_i, y_i) \tag{6}$$

as $\{p_1, p_2, \ldots, p_M\}$ changes the statistics of the training set.

For this reason, (6) must be computed differently.

- Instead of relying on the autodifferentiation mechanism, Aref and Chagnon compute the gradient analytically.
- The derivative of the mutual information w.r.t. p_j, (6), results to be

$$\frac{\partial}{\partial p_j}\mathbb{I}(X,Y;P_M) \approx -\log_2(p_j) - \log_2(e) + \frac{1}{Bp_j}\sum_{b \text{ if } x=j} \log_2 Q_{X|Y}(j|b) + \frac{1}{B}\sum_{(a,b)} \log_2 Q_{X|Y}(a|b) \tag{7}$$

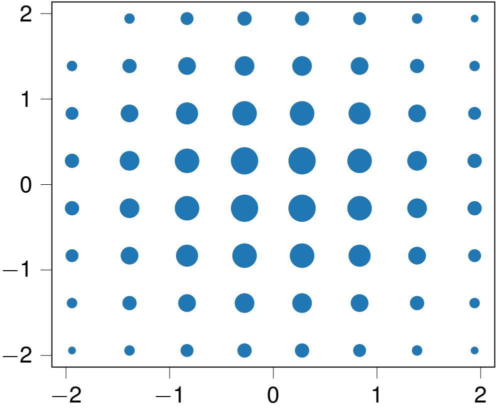
- The following terms can be computed via backpropagation

$$-\log_2(p_j) + \frac{1}{Bp_j}\sum_{b \text{ if } x=j} \log_2 Q_{X|Y}(j|b) = \frac{1}{B}\sum_{i=1}^{B} \frac{\partial}{\partial p_j} L(x_i, y_i) \tag{8}$$

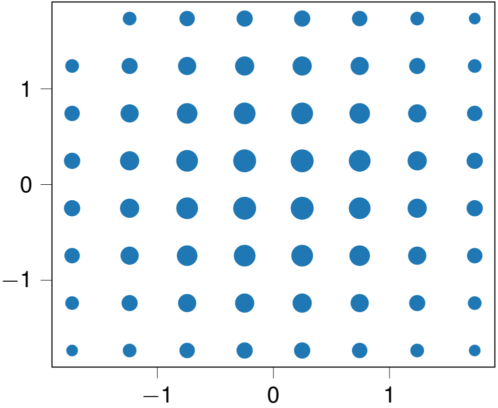
while the remaining ones must be explicitly computed and added to the gradient after backpropagating.

- We call this step *gradient correction* and it is due to the change of statistics in the sampled batch.

Probabilistic Shaping



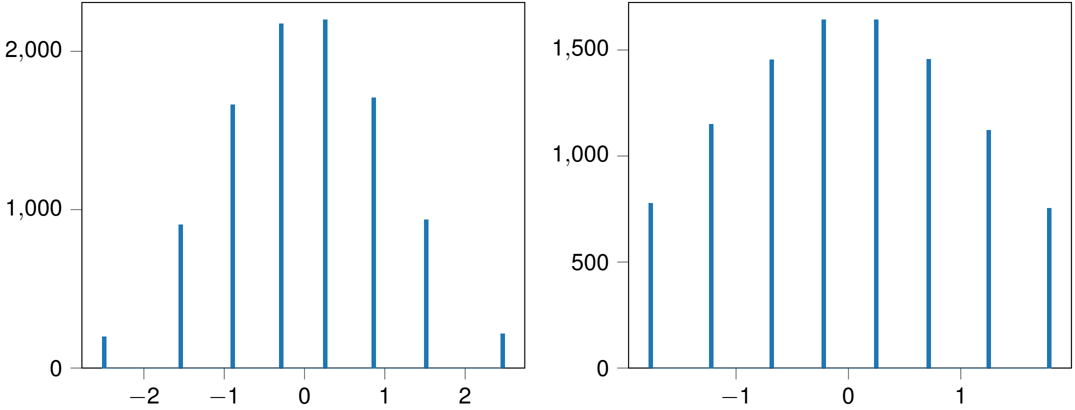
(a) SNR = 5dB



(b) SNR = 18dB

Figure: Learnt probabilistic constellation shaping for M = 64. The size of the markers is proportional to the transmission probability of the symbol. When trained under 5dB, the proababilistic shaping approaches a gaussian. While under 18dB it approaches a uniform distribution.

Joint Geometric and Probabilistic Shaping

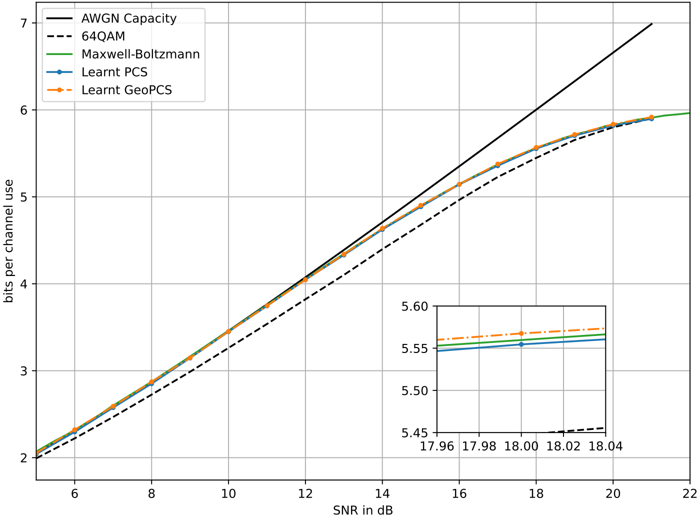


(a) SNR = 5dB

(b) SNR = 18dB

Figure: Learnt joint geometric and probabilistic ASK constellations for M=8.

Overall Performance



Conclusions

- Both autoencoder proposals show close-to-optimal performance over the AWGN channel.
- Common keys for success:
  1. the choice of the loss function
  2. and correct computation of the gradient w.r.t P_M
- The potential of the autoencoder approach is for training over complex channels such as optical fiber.
- Over these channels we expect that they will exhibit different performance.
- Both [AC21] and [SAAH19] introduce $\mathbb{H}(X)$ into the loss function. This in turn has the effect of adding a complementary path to the computational graph for computing the gradient w.r.t P_M without backpropagating through the channel model.
- Training over other channel models is an open research direction with some challenges, such as the limited support of complex data types in the ml frameworks, but promising outcomes

Bibliography

[AC21] V. Aref and M. Chagnon, *End-to-end learning of joint geometric and probabilistic constellation shaping*, 2021. DOI: 10.48550/ARXIV.2112.05050. [Online]. Available: https://arxiv.org/abs/2112.05050.

[CW19] J. Cho and P. J. Winzer, "Probabilistic constellation shaping for optical fiber communications," *Journal of Lightwave Technology*, vol. 37, pp. 1590–1607, 2019.

[HSW89] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[JGP16] E. Jang, S. Gu, and B. Poole, *Categorical reparameterization with gumbel-softmax*, 2016. DOI: 10.48550/ARXIV.1611.01144. [Online]. Available: https://arxiv.org/abs/1611.01144.

[OH17] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017. DOI: 10.1109/TCCN.2017.2758370.

[SAAH19] M. Stark, F. Ait Aoudia, and J. Hoydis, "Joint learning of geometric and probabilistic constellation shaping," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6. DOI: 10.1109/GCWkshps45667.2019.9024567.