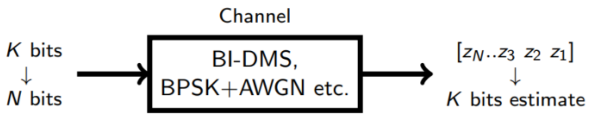
On Deep Learning-Based Channel Decoding (Report)

Dheeraj Racha
EE17BTECH11033

I. Introduction

Deep Neural Networks are used for one-shot decoding of polar codes. This report is an evidence that the nueral networks can learn decoding algorithms (polar codes in this case). Throughout the report, 16-bit block length codes with rate, r = 0.5 are used for simulations.



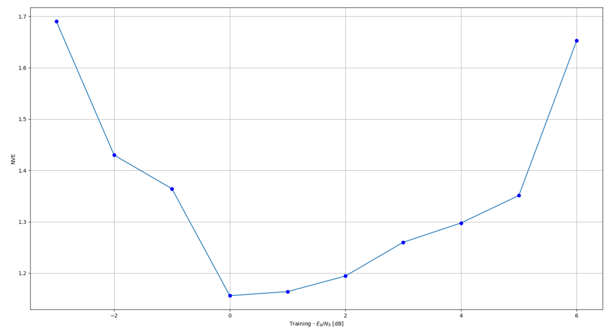
The training set χ consists of all possible codewords $x_i \epsilon \mathcal{F}_2^N$ with $\mathcal{F}_2 \epsilon \{0, 1\}$ (the labels being the corresponding information bits) and is given by $\chi = \{x_0, ..., x_{2^{k}1}\}$
For the sake of simplicity, binary phase shift keying (BPSK) modulation and an additive white Gaussian noise (AWGN) channel is used.
Each hidden layer employs a ReLU activation function and sigmoid activation function at the output layer which gives the probabilites that the decoded bits are 0 or 1.
We train our Neural Netwrok Decoder (NND) in epochs. In each epoch, the gradient of the loss function (Binary Cross entropy) is calculated over the entire training set χ using Adam optimizer.

II. What is the best training SNR?

What is the SNR (dB) that should be considered while training NND?
Lets investigate the performance of the NND trained at different SNR values to find the correct SNR value that should be considered for taining the NND.
To compare the performance of NND trained at a particular SNR, introduce a new metric called Normalized Validation Error (NVE). Given a model trained at an SNR value, compute the mean of ratio of BER performance of this model to MAP performance for a range of SNR values. Mathematically,

$$NVE(\rho_t) = \frac{1}{S} \sum_{s=1}^{S} \frac{BER_{NND}(\rho_t, \rho_{v,s})}{BER_{MAP}(\rho_{v,s})}$$

Clearly, to achieve MAP performance, NVE of our trained model should be close to 1.
Here, we compute NVE over S = 20 differnt SNR points from 0 dB to 5 dB with a validation set size of 20,000 examples for each SNR.
As the codeword length N = 16 and rate r = 0.5, generate all possible valid codewords and encode them. These will be the input to out Neural Netowrk Decoder. The BPSK Modulation and Addition of AWGN noise is taken care by the NND, reducing the training set size to 2^k.
To begin with, consider a Neural Network with 128-64-32 hidden layers, Adam optimizer and Binary Cross Entropy loss function, and train for 2^{16} epochs.



From the above plot it can be observed that, the NND trained at SNR around 0 dB to 1 dB achieves close to MAP performance. It is also observed that,

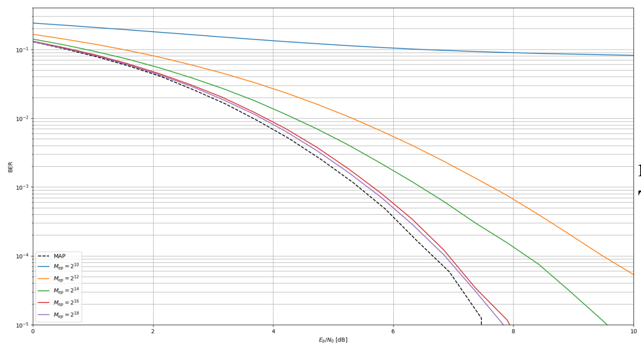
- when trained at low SNR, the NND is unable to learn the underlying structure of the code.
- When trained at high SNR, the NND is unable to handle noise.

Therefore, from now on for further investigations, it is assumed that the NND is trained at an SNR of 1 dB.

III. How many training epochs are necessary?

In the previous section, we've trained the NND for 2^{16} epochs. Can the Neural Network give better performace if we simply increase the number of training epochs?
Let us investigate this by training the NND for different number of epochs (2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}).
As stated in the previous section, training the NND at SNR = 1 dB, gives better performace. So, lets train the NND with 128-64-32 hidden layers at 1 dB for different number of epochs. For BER

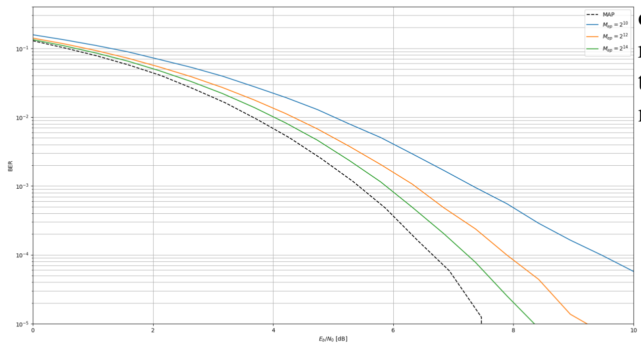
simulations 1,000,000 codewords per SNR point are used.



It is observed that, larger the number of training epochs, the closer the gap between MAP and NND performance. Also, close to MAP performance is achieved for $M_{ep} = 2^{18}$ epochs.

IV. NND Trainied at mixed SNRs

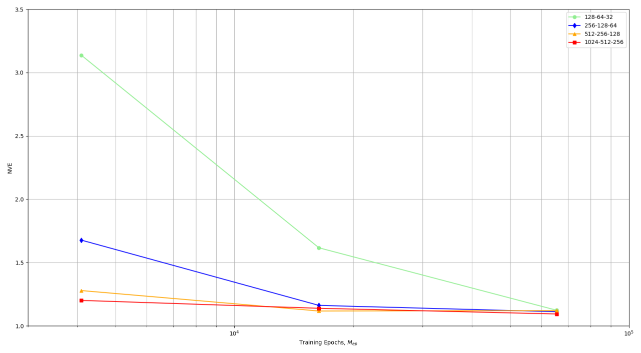
What will be the performance of the NND trained at multiple SNR values, instead of training the NND at one particular SNR value?
From the previous two sections, when the NND is trained at 1 dB SNR for around 2^{16} epochs gives close to MAP performance. In this section, the number of training epochs required get optimal performance is investigated when trained over a range of SNR values.
So, the NND (128-64-32) is trained starting at a high training SNR (6 dB) and then gradually reducing the SNR (till -3 dB).For BER simulations 1,000,000 codewords per SNR point are used.



- Comparing the above plot with BER plot from previous section, it can be observed that training the NND with mixed SNR values gives better performance compared to training with only one SNR value (if both trained for same number of epochs)
- Close to MAP performance is obtained by training for 2^{14} epochs, when the NND is trained with mixed SNR values.
- Whereas, when the NND is trained with only one SNR (1 dB), close to MAP performance is achieved by training for 2^{16} epochs.

V. How many layers and nodes should the NN employ?

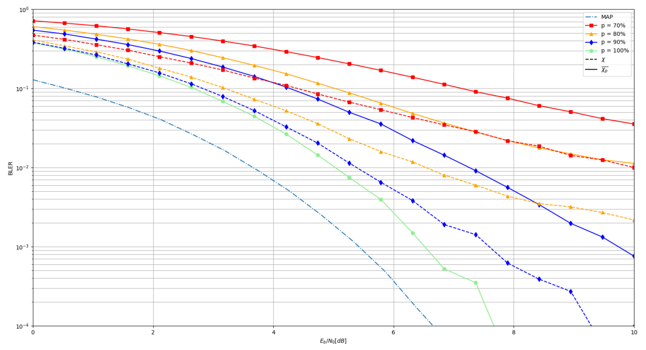
Does the NND give better performance if we just make it more complex?
To address this, we train NNDs with different sizes.



From the above plot, the NND is able to achieve close to MAP performance for all NND architectures. Also, larger the network, the less training epochs required to reach MAP performance. Contrary to what is common in classic machine learning tasks, increasing the network size does not lead to overfitting since the network never sees the same input twice.

VI. Capability of Generalization

Is the Neural Network that we are training actually learning the structure the code?
To investigate this, the NND is trained only on a fraction of the entire possible valid codewords and tested on the remaining codewords set (just to compare, NND is also tested on the entire valid codewords). Instead of BER, a new metric Block Error Rate (BLER) is used for evaluation.



It can be observed from the above plot that the NND is able to predict for unseen codewords also for some extent.
Hence, we can say that the NND is able to generalize in two ways:

- The Neural Network can generalize from input channel values with a certain training SNR to input channel values with arbitrary SNR.
- The Neural Network is able to generalize from a subset χ_p of codewords to an unseen subset $\bar{\chi_p}$.

VII. Conclusions

- One important advantage of using Deep Learning in decoding is that the time to deocde drastically decrease as the complexity reduces from exponential to polynomial. (observed while simulating the results.)
- Neural nets are able to generalize for structures codes.
- With advancements in the computational resources, neural networks has a great role to play in decoding algorithms, as the State-of-the-art decoding algorithms currently suffers from
  * high decoding complexity
  * Lack of parallelization
  * critical decoding latency