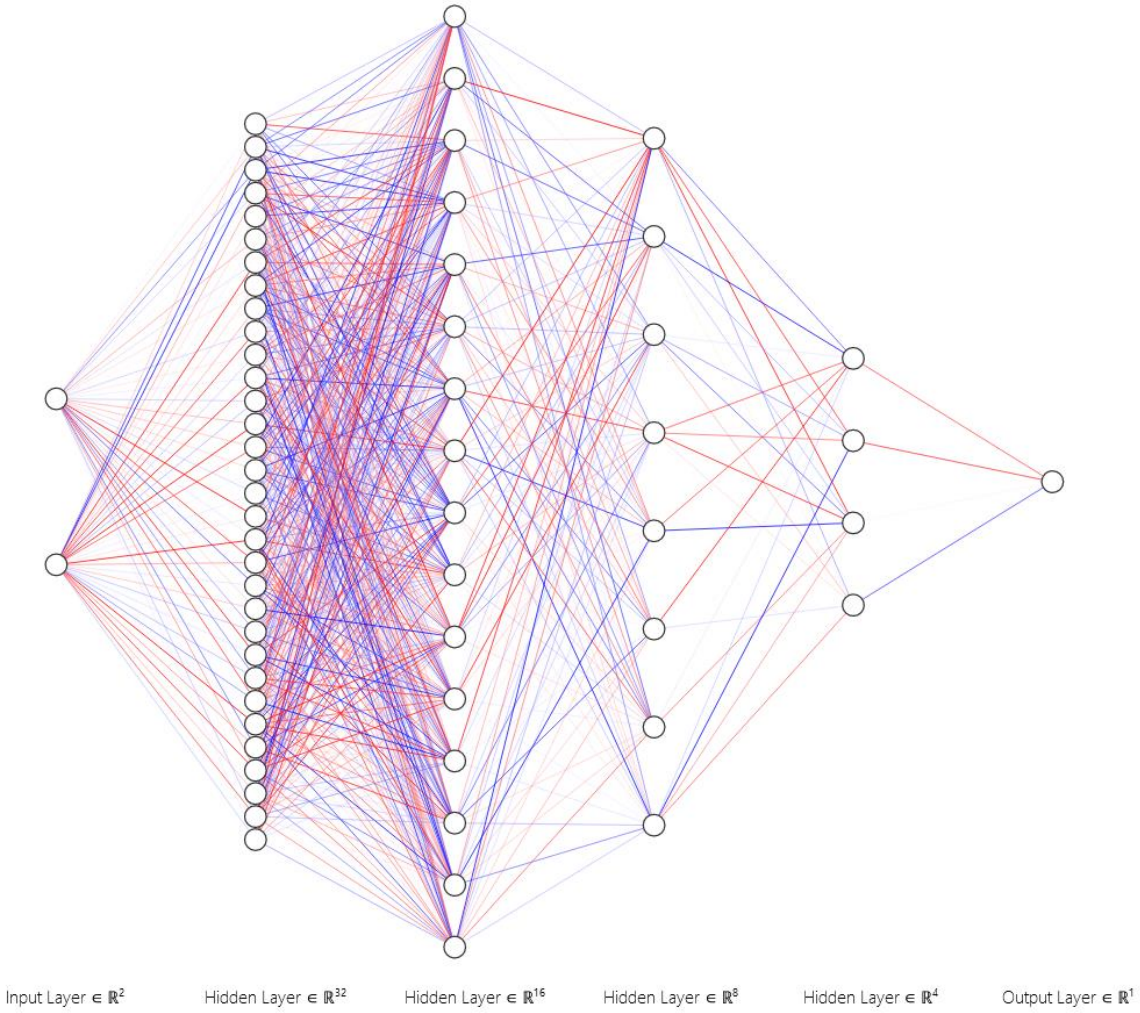
CIRI Report:

Deep Learning Based Channel Decoding



Simplified neural network view, where one node represents an 8-bits long message (vector)

Table of contents

Abstract

Following the master thesis made by *Tobias Gruber*, *Sebastian Cammerer*, *Jakob Hoydis*, and *Stephan ten Brink*, we have tried to reproduce their work on bits decoding neural network.

Although some strong and theorical optimal decoding algorithm already exist – such as the Maximum A Posteriori algorithm (MAP) –, these are known for being very resource consuming, which is a main drawback when considering how long a message, when transferred as a sequence of bits, can be.

By training a neural network using deep learning frameworks and libraries, we aimed at optimizing these processes while keeping the same decoding performances.

Introduction

The main issue when trying to learn a deep learning neural network is how many "labels" are to be classified. When applied to bit telecommunication, the patterns become rapidly huge, as the number of words (or "patterns") is exponentially growing. Thus, when encoding an 8 bits message, there are 2^8 different patterns, which result in 256 "labels". When adding a few more bits, like for instance 2^{14} possible message reaches a staggering number of 16,384 possible initial messages.

Furthermore, without an enormous amount of training examples, little to no perturbation and lots of training epochs, there is no hope in achieving any kind of satisfying results. On top of this, the neural network computation couldn't be run on a normal computer with limited computing capabilities.

Thus, the problem had to be limited in ambitions and specified. The learning dataset had to be featuring limited codeword length, which implies an overall limited references dictionary. The work done below was thoroughly done on 8-bits long words, thus being able to decode a dictionary of only 256 words. As said earlier, any increase in words length would indeed widen the dictionary size, but computing cost (i.e. time) would also raise exponentially.

Although limited when compared to 32-bits or 64-bits messages commonly used in many computing systems, some basic telecommunication applications can still be envisaged on more modest electronical systems.

While less ambitious, the results we managed to reach using structured encoded are plenty satisfying, enjoying the strength of being based on a full dataset, thus being obviously more resistant and accurate than a network trained on a sample of a dictionary (as implied in the original work when citing 64-bits long messages examples).

Technical approach

1) Preamble

This part relates the ambitions, goals and questions targeted by the experiment. The following parts deepen the technical solution deployed, and how they are fitted to answer the raised questions.

If the results are already known, as exposed in the original thesis, this report aims at reproducing the same work and answering the following points:

- What is the best neural network configuration?
- What is the best training white noise parameter (sigma)?
- How many epochs are required in order to reach a satisfying result?
- Is the noise resistance the same depending on the word length?
- In which case a trained neural network should be used, instead of a regular MAP algorithm?

2) Encoding

The transmitted message is originally an 8-bits sequence. It is encoded using the *Polar Code* technique, which can be resumed as followed: the sequence m is subjected to a matrixial product with a polar encoding matrix G, resulting in a to-be-transmitted bit sequence x.

$$x = m*G$$

The matrix G having a shape of 8x16, the encoded message is now a 16-bits sequence. The purpose of the matrix G is thus to both give weight to the original message, and to maximise its hamming distance (by maximising the difference in encoded bits of two similar 8-bits messages).

The now 16-bits encoded message then follows a BPSK modulation, so the simulated message is now composed of 1 and -1 symbols called X.

$$X = 2*x - 1$$

3) Neural Network design

The designed neural network was actually trained to operate these operations as well, so that it takes in input X and returns the original decoded message m. It is composed as followed:

- A BPSK modulation layer that transforms the polar encoded 16-bits message x into the to-be-transmitted message X,
- A noise layer, that simulates the perturbations applied on the signal during the transmission (AWGN channel). It thus allows to tweak the network training, by refining the variance of the gaussian noise.
- A decoder “layer”, composed of 3 hidden layers of respectively 128, 64 and 32 neurons, and an output layer classifying every bit in 2 categories, zeros and ones.

About the decoder, the 3 hidden neurons layers are chosen identically to the master thesis ones. It actually follows a logical decision that doesn’t need to be tested: the input being a 256 words dictionary, the first layer is 128 neurons wide, then 64, then 32… The only one missing is the 16 one, as its impact might not be consequent following 3 larger layers.

The activation functions of the 3 hidden neurons layers are 3 ReLU functions, as it has been long time proved than this one was the most optimal in most cases. The output layer on the other hand features a sigmoid activation function, as it returns a symbol probability between 0 and 1, which allows to jump the symbol to bits decoding step (instead of a classifying function which would return 1 or -1 when evaluated).

The loss, optimizing and errors functions (for backpropagation) are all common, being respectively the mean square error, Adam optimizer, and bit error sum function (resp. bit error ratio) for the decoder (resp. training model).

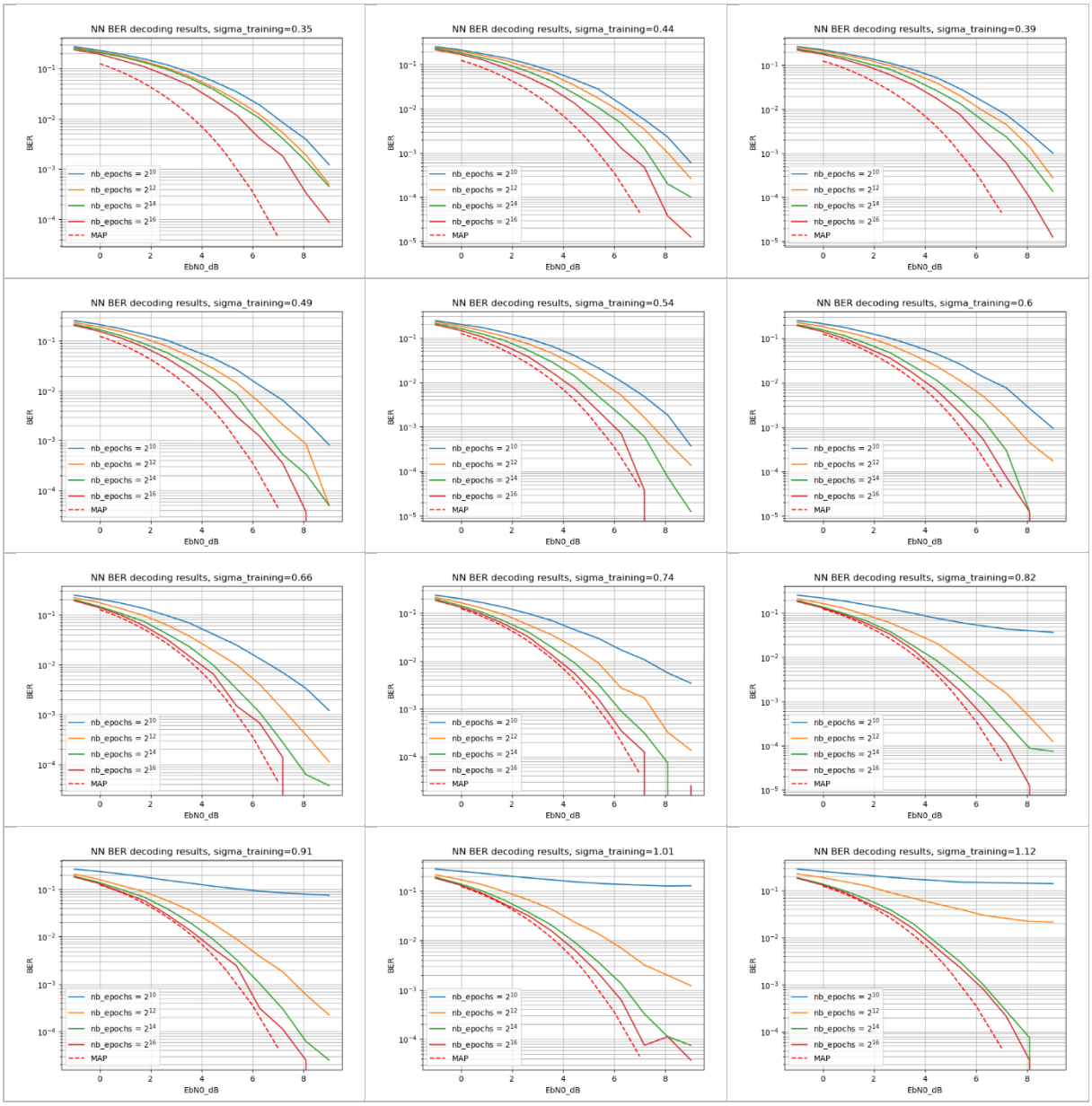
Results

1) Preamble

The following part summarizes the results obtained after training the above neural network for different noise values, and multiple epochs generation. All the results should be compared to the MAP curve, which represents the theorical optimal Bit Error Ratio (BER) obtainable for a given Eb/N0 (dB) value (directly connected to the variance of the white gaussian noise).

2) 80,000 bits

These plots are realized for Eb/N0 (dB) values ranging from -1dB to 9dB. The number of training epochs is included from 2^{10} to 2^{16} epochs. The evaluation method features 10 tries, with 1000 8-bits words tested every iteration. Because of how little this first sample is, the curves are sometimes slightly contradictory.



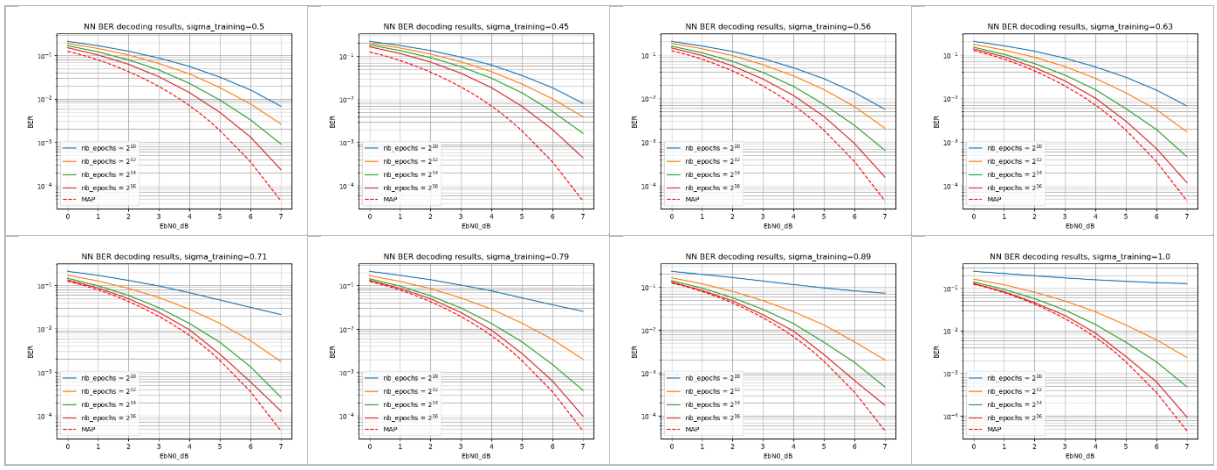
If some curve aberrations can be seen, due to the small amount of evaluation tries, the tendency shown is easily readable, and confirms the results obtained in the original thesis.

Variance values too small result in a network training lacking in noise resistance. Although, it's worth noticing than for less training epochs, the results are far better than those with a stronger noise, supposing a link between converging speed of the neural network training and training noise.

Overall, best results are reached for larger number of epochs, and sigma ≈ 0.66 (Eb/N0 = 3dB).

3) 800,000 bits

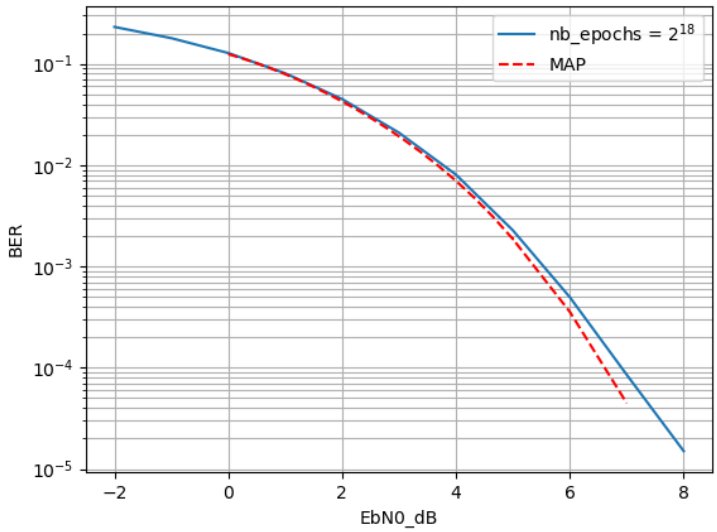
The following 8 curves were realized for a ratio Eb/N0 (dB) included between 0dB and 7dB. The epochs are the same, though thanks to a larger number of evaluation iterations (itt = 100), the curves are way more representative of how the neural network is expected to work on large number of bits transmitted.



The conclusion is now slightly changed. If a smaller training variance still gives better results for small amounts of epochs, the best results are obtained for sigma ≈ 0.89, which corresponds to a ratio Eb/N0 = 1dB, as highlighted in the thesis.

4) 800,000 bits, 2^{18} epochs

The best results are definitely obtained for larger amounts of training epoch. Due to how resource consuming the neural network computing is, only one example should be enough to conclude on how efficient all this study has been. According to the deductions above, sigma ≈ 0.89 i.e. Eb/N0 = 1dB are decided to be the most optimal parameters for a high number of epochs computation.



Conclusion

Overall, the results are satisfying, as we've managed to reproduce the work done in the thesis. With identical results for the same initial conditions, the conclusion is similar to the one of the original document.

About the best training noise, it has been thus confirmed than a ratio Eb/N0 = 1dB seems to be optimal, and the neural network, although "refurbished" when compared to the one done 6 years ago, is globally the same.

Sadly, due to technical limitations, we couldn't test a larger spectrum of word lengths and dictionaries sizes and ended up working on the same conditions as the authors of the thesis, which is a result by itself.

Finally, even though the trained neural network is working, it lacks fields of direct application in its current form, where the MAP algorithm complexity is still satisfying. Other training methods and larger datasets should be used in order to achieve a real breakthrough in the decoding techniques field.