

Hyperparameter Optimization of a Deep Learning-Based Channel Decoder

Marcelo Vinícius Cysneiros Aragão

Nicole Souza

Instituto Nacional de Telecomunicações – Inatel

TP555 – Inteligência Artificial e Machine Learning

Santa Rita do Sapucaí, 29 de julho de 2021

Sumário

1. Introdução
2. Revisão bibliográfica
3. Métodos
4. Resultados
5. Discussão
6. Conclusões e perspectivas para novos trabalhos
7. Novos experimentos, resultados e conclusões

Introdução

Introdução

- Decodificadores de canal baseados em redes neurais
 - Tempo de treinamento reduzido
 - Alta capacidade de generalização
 - Facilidade de implementação em *hardware*
- Decodificador baseado em redes profundas (*deep learning*)
- Otimização de hiperparâmetros
 - Avaliação de diferentes otimizadores
 - SGD, RMSprop, Adam, Adadelta, Adagrad, Adam, AdaMax e Nadam

Revisão bibliográfica

Revisão bibliográfica

- Bruck e Blaum (1989)
 - “decodificação de um código de bloco linear corretor de erros é equivalente a encontrar um máximo global da função de energia de uma determinada rede neural”
- Zeng, Hush e Ahmed (1989)
 - MLD usando uma RRN que pode ser treinada em tempo polinomial e generalizável/aplicável à decodificação de qualquer tipo de código.
- Caid e Means (1990)
 - Decodificador neural corretor de código aplicável a cenários sem ruído gaussiano branco aditivo (AWGN) e/ou canal simétrico binário (BSC)

Revisão bibliográfica

- Tallini e Cull (1995) e Wu, Tseng e Huang (2002)
 - Mais implementações bem-sucedidas de decodificadores neurais.
- Di Stefano et al. (1991)
 - Avaliaram o uso de redes do tipo Counter e Back Propagation
- Abdelbaki, Gelenbe e El-Khamy (1999)
 - RNs aleatórias → fácil implementação em *hardware* (*chips* dedicados)
- Nachmani, Be'ery e Burshtein (2016) e **Gruber et al. (2017)**
 - Desenvolvimento de decodificadores usando redes neurais profundas

Métodos

Métodos

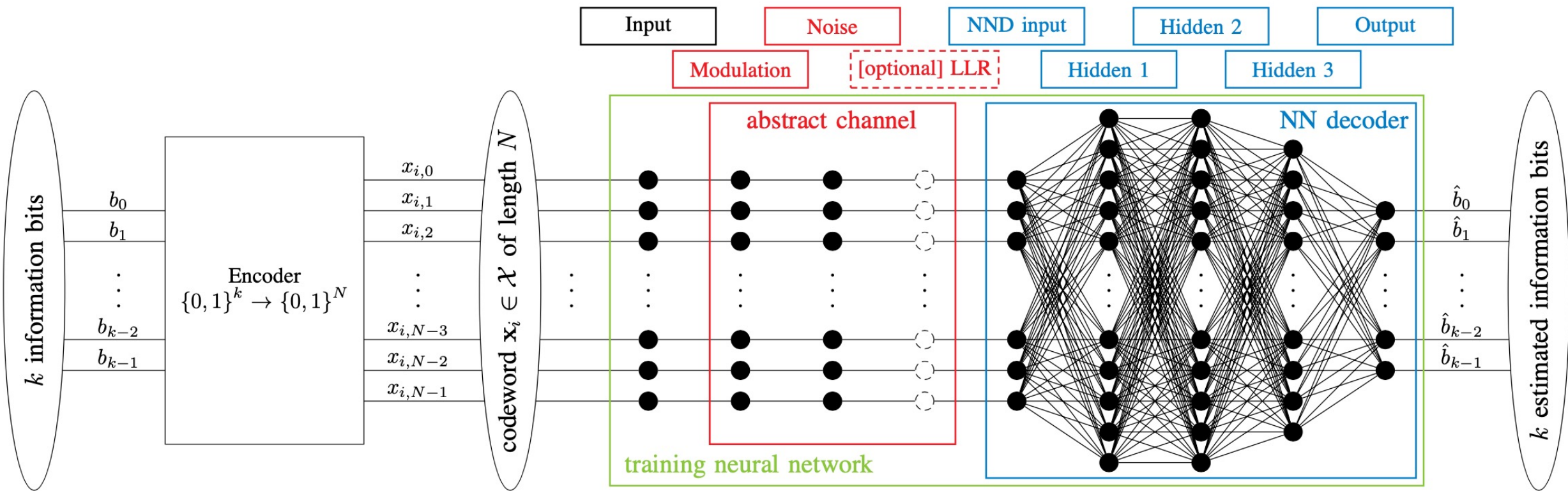


Fig. 1: Deep learning setup for channel coding.

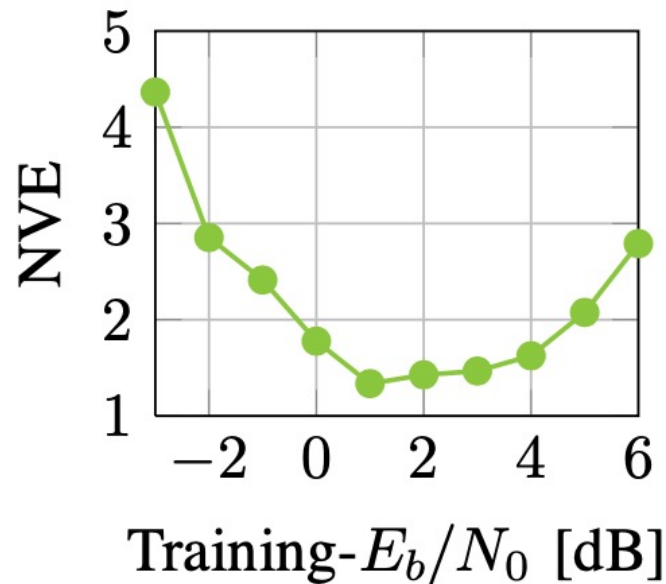
Métodos

Levando em consideração tanto a codificação de código aleatório quanto de estruturado (polar), os seguintes experimentos são conduzidos no trabalho:

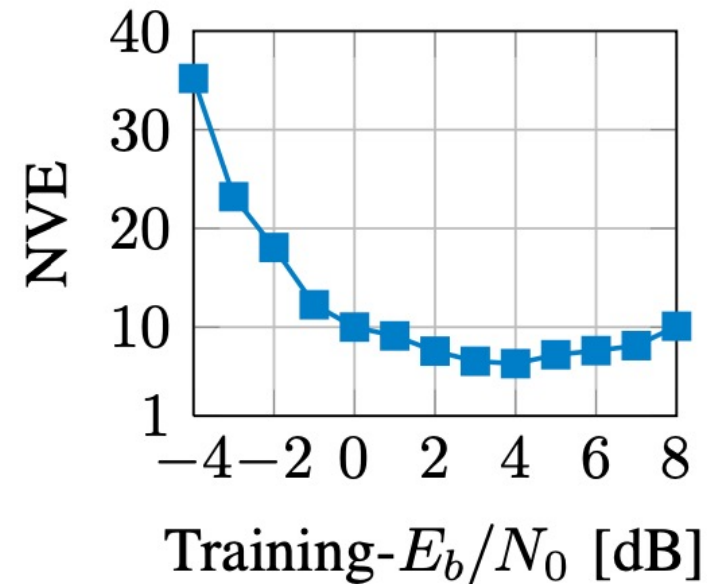
1. Influência da SNR ($E_b/N_0 = -4, -3, \dots, +6$ dB) no NVE, durante a etapa de treinamento.
2. Influência da qtd. de épocas de treinamento ($M_{ep} = 2^{10}, 2^{12}, \dots, 2^{18}$) na BER, considerando:
 - a) Uso (ou não) dos valores de verossimilhança logarítmica na saída do transmissor;
 - b) Uso de diferentes funções de erro na otimização (MSE – *Mean Squared Error* ou BCE – *Binary Cross-Entropy*);
 - c) Quantidades crescentes de neurônios nas três camadas escondidas do decodificador (128-64-32, 256-128-64, 512-256-128 e 1024-512-256);
3. Escalabilidade do decodificador neural considerando informações de tamanhos crescentes ($k = 8, 9, \dots, 14$ bits) e taxas de código variáveis ($p = 70\%, 80\%, 90\%, 100\%$).
4. Capacidade de decodificar palavras-código inéditas avaliando a BLER (*Block Error Rate*) para diferentes SNRs e particionamentos de dados entre conjunto de treinamento e de testes (100%/0%, 90%/10%, 80%/20% e 70%/30%).

Resultados

Resultados



(a) Polar Code



(b) Random Code

Fig. 2: NVE versus training- E_b/N_0 for 16 bit-length codes for a 128-64-32 NN trained with $M_{\text{ep}} = 2^{16}$ training epochs.

- **Resultado 1:** o código polar demanda uma SNR de treinamento consideravelmente menor (1 dB) do que o código aleatório (4 dB) para atingir um NVE aceitável.

Resultados

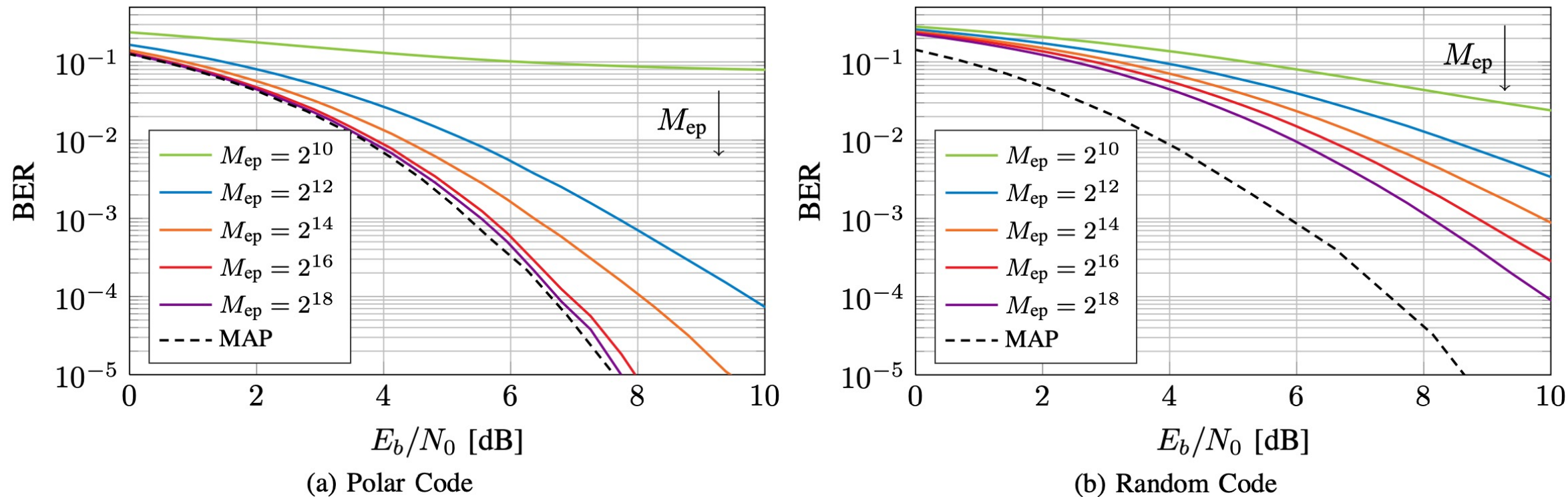


Fig. 3: Influence of the number of epochs M_{ep} on the BER of a 128-64-32 NN for 16 bit-length codes with code rate $r = 0.5$.

- **Resultado 2:** códigos estruturados são de fato mais fáceis de aprender do que códigos aleatórios, ou seja, menos períodos de treinamento são necessários.

Resultados

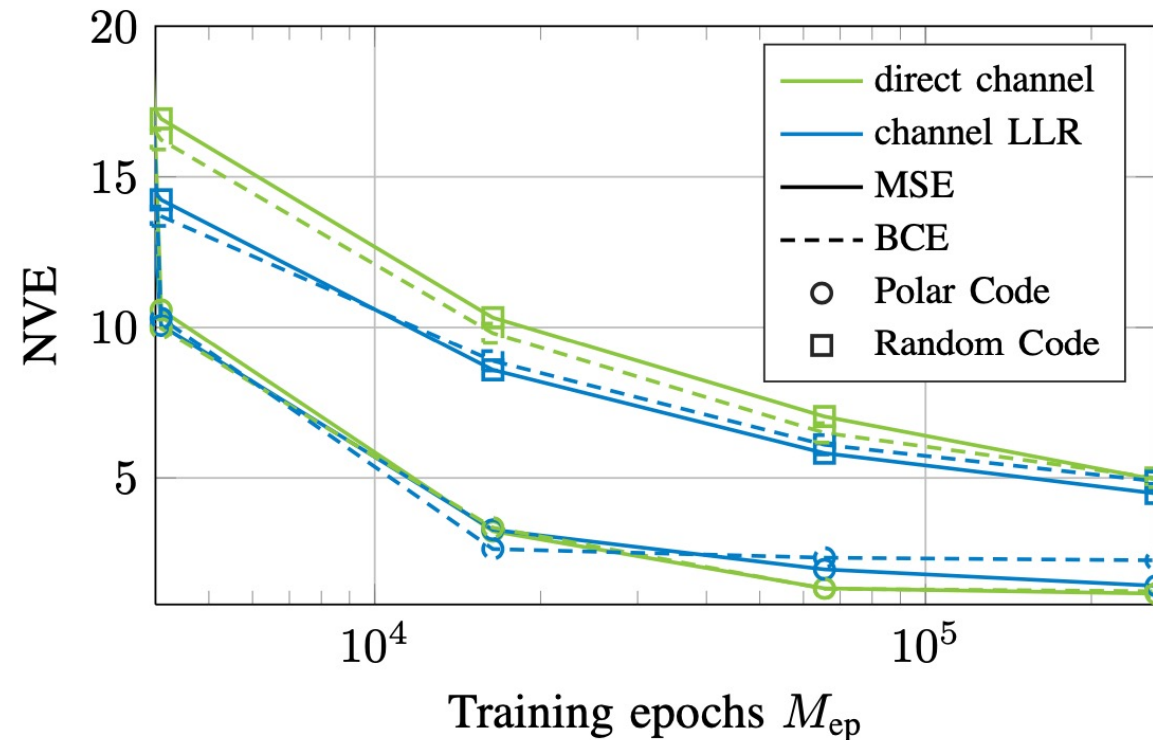


Fig. 4: Learning curve for 16 bit-length codes with code rate $r = 0.5$ for a 128-64-32 NN.

- **Resultado 3:** quando a RN é treinada com um grande número de épocas, o emprego de LLR ou valores de canal, a normalização dos valores de entrada e a escolha da função de perda não influenciam no resultado.

Resultados

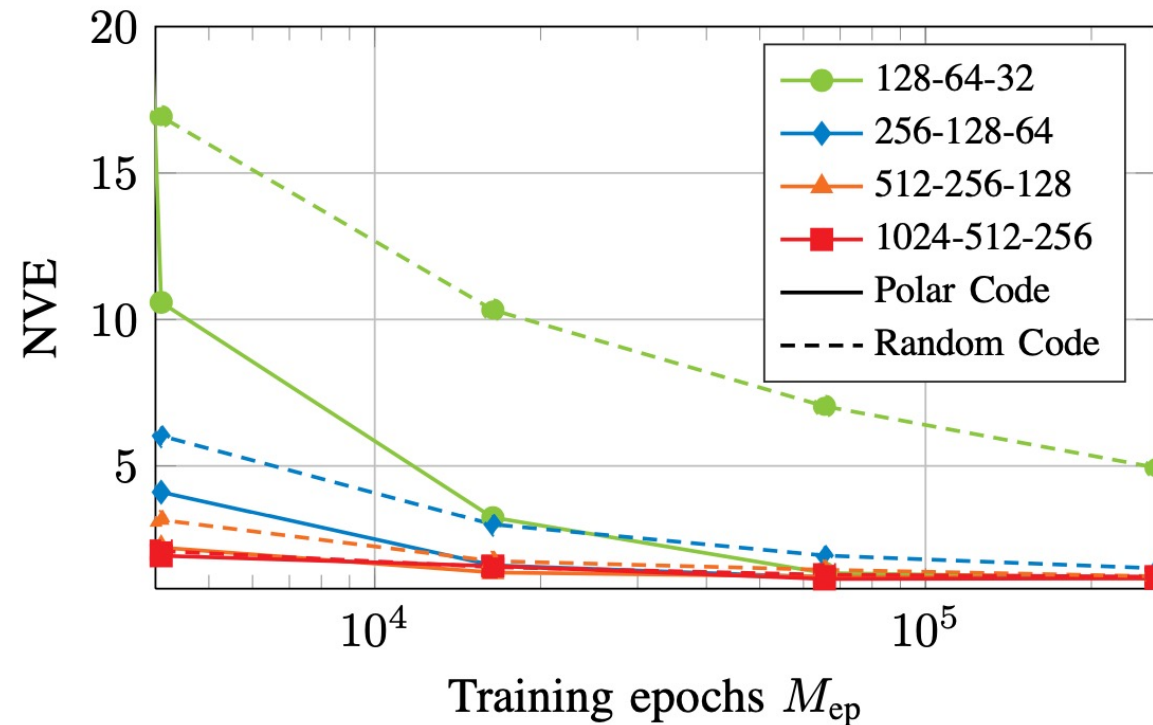


Fig. 5: Learning curve for different NN sizes for 16 bit-length codes with code rate $r = 0.5$.

- **Resultado 4:** quanto maior a quantidade de neurônios nas camadas ocultas da rede, menos épocas são necessárias para atingir um NVE aceitável.

Resultados

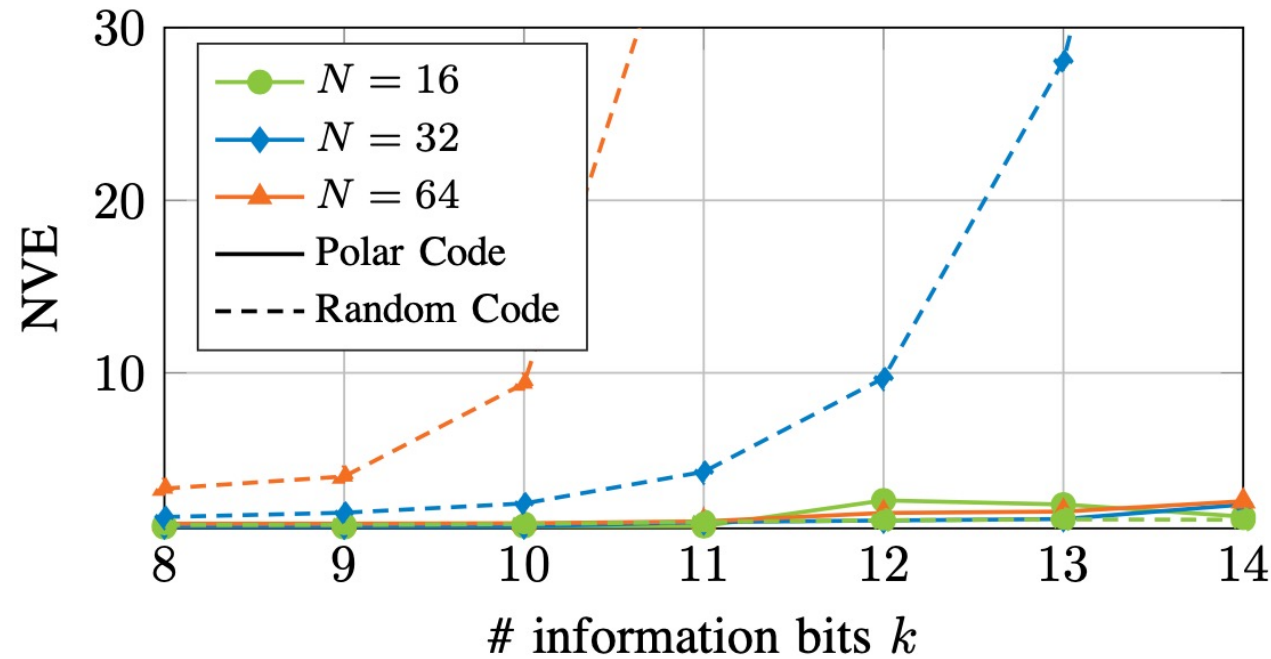
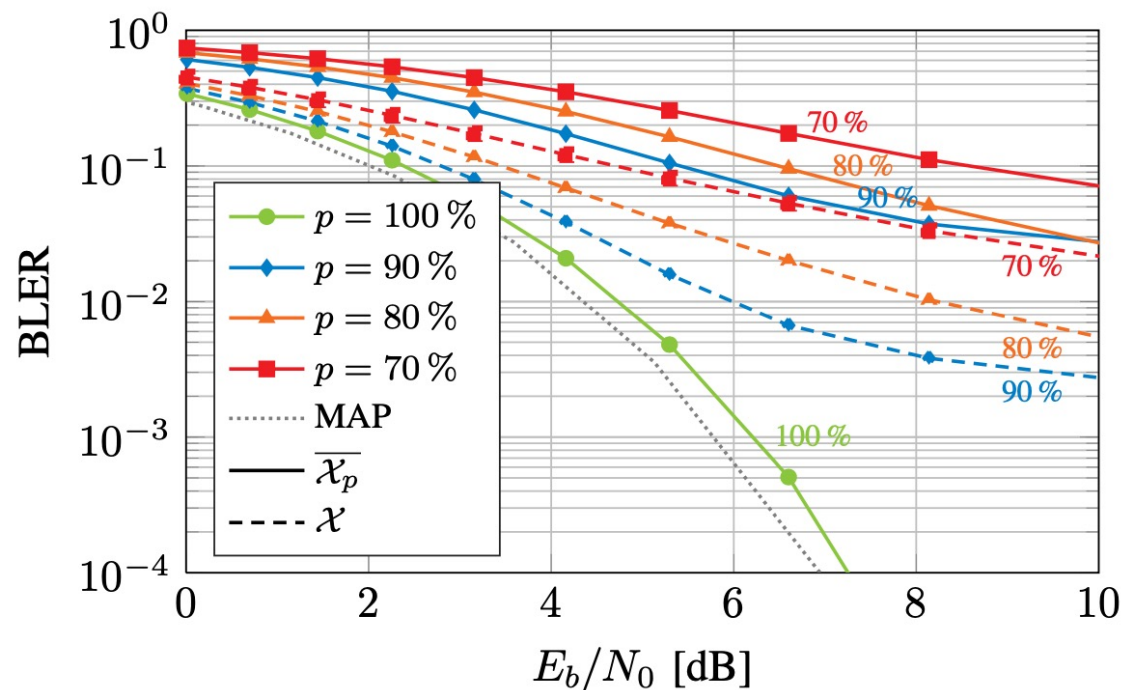


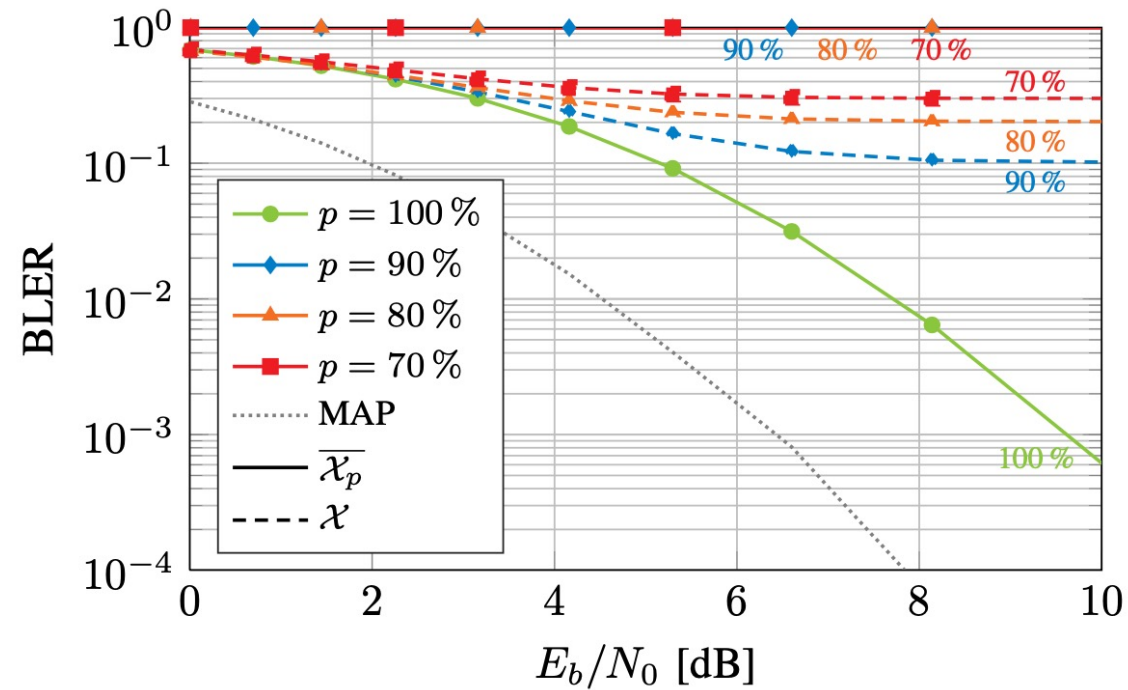
Fig. 6: Scalability shown by NVE for a 1024-512-256 NN for 16/32/64 bit-length codes with different code rates and $M_{\text{ep}} = 2^{16}$ training epochs.

- **Resultado 5:** o NVE aumenta exponencialmente conforme o número de *bits* de informação aumenta.

Resultados



(a) 16 bit-length Polar Code ($r = 0.5$)



(b) 16 bit-length Random Code ($r = 0.5$)

Fig. 7: BLER for a 128-64-32 NN trained on \mathcal{X}_p with $M_{\text{ep}} = 2^{18}$ learning epochs. Solid and dashed lines show the performance on \mathcal{X}_p on \mathcal{X} , respectively.

- **Resultado 6:** quanto menor for conjunto de treinamento, menor será a capacidade de generalização da rede (para quantidades de épocas fixas), especialmente em casos de códigos aleatórios.

Resultados

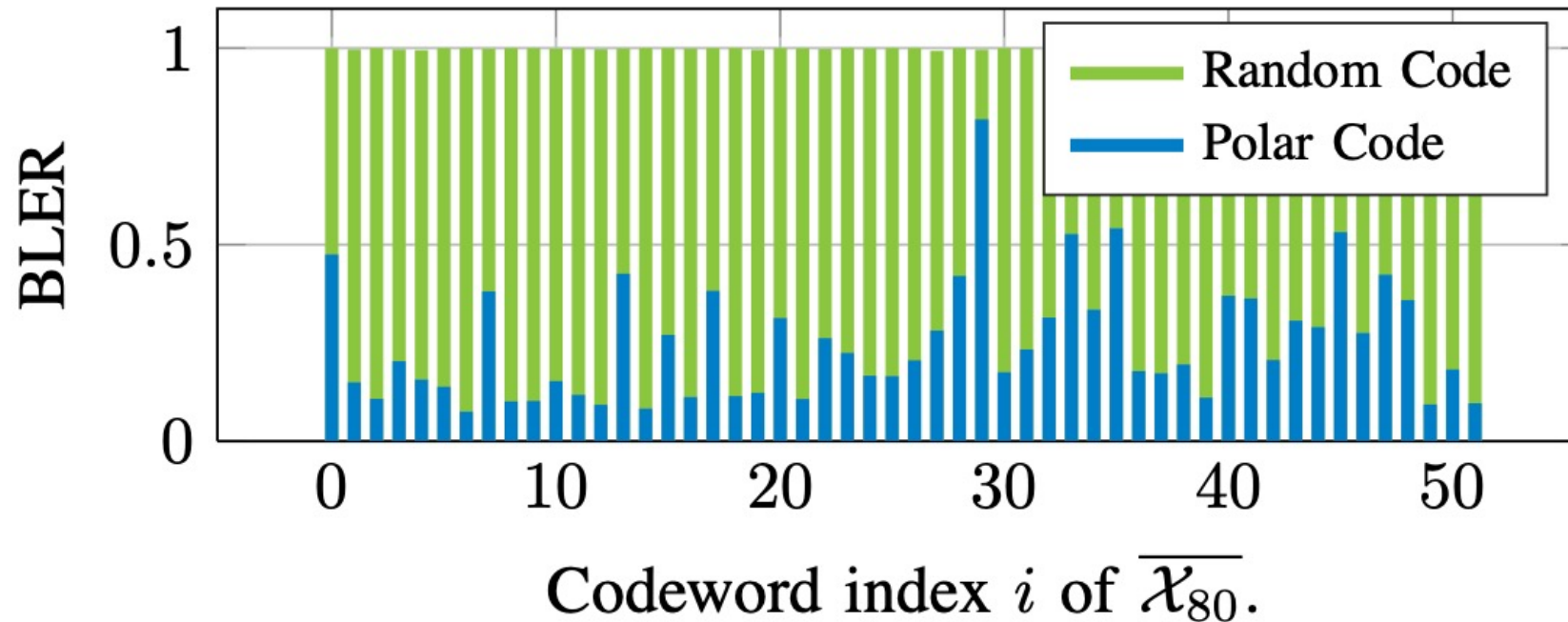


Fig. 8: Single-word BLER for $\mathbf{x}_i \in \overline{\mathcal{X}}_{80}$ at $E_b/N_0 = 4.16$ dB and $M_{\text{ep}} = 2^{18}$ learning epochs.

- **Resultado 7:** RNs são capazes de generalizar ou “interpolar” para o livro de código completo depois de ter visto apenas um subconjunto de exemplos, sempre que o código tiver estrutura.

Discussão

Discussão

- Reafirmação da aplicabilidade de decodificadores neurais.
- Resultados compatíveis com:
 - Lyu et al. (2018)
 - Decodificadores neurais baseados em redes *feedforward* (MLP), convolucionais (CNN) e recorrentes (RNN), que se destacaram.
 - Xu et al. (2018)
 - Modelo híbrido baseado em duas RNs: uma convolucional para equalização do sinal recebido e uma profunda, responsável pela decodificação do sinal.

Conclusões e perspectivas para novos trabalhos

Conclusões e perspectivas para novos trabalhos

- Diversos aspectos podem ser avaliados em trabalhos futuros:
 - Outros modelos de aprendizado de máquina
 - Outros tipos de modulação
 - Diferentes tipos de canais
 - Estimar a variância para ser usada como parâmetro na camada LLR
 - **Realizar a otimização de hiperparâmetros**
 - Incorporar técnicas de regularização
 - Normalizar as entradas da RN para ter média zero e variância unitária
 - Avaliar outras topologias de rede neural

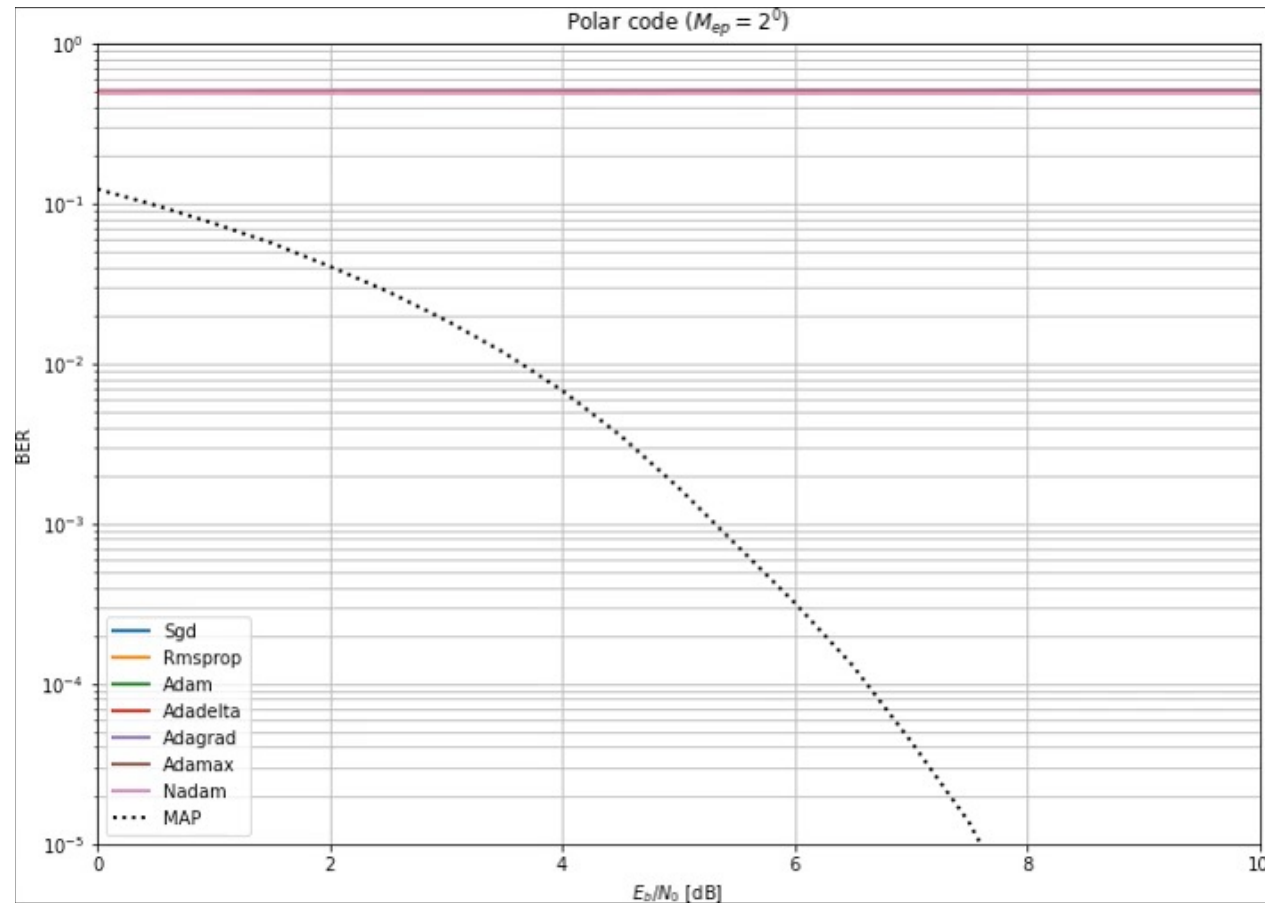
Novos experimentos, resultados e conclusões

Novos experimentos, resultados e conclusões

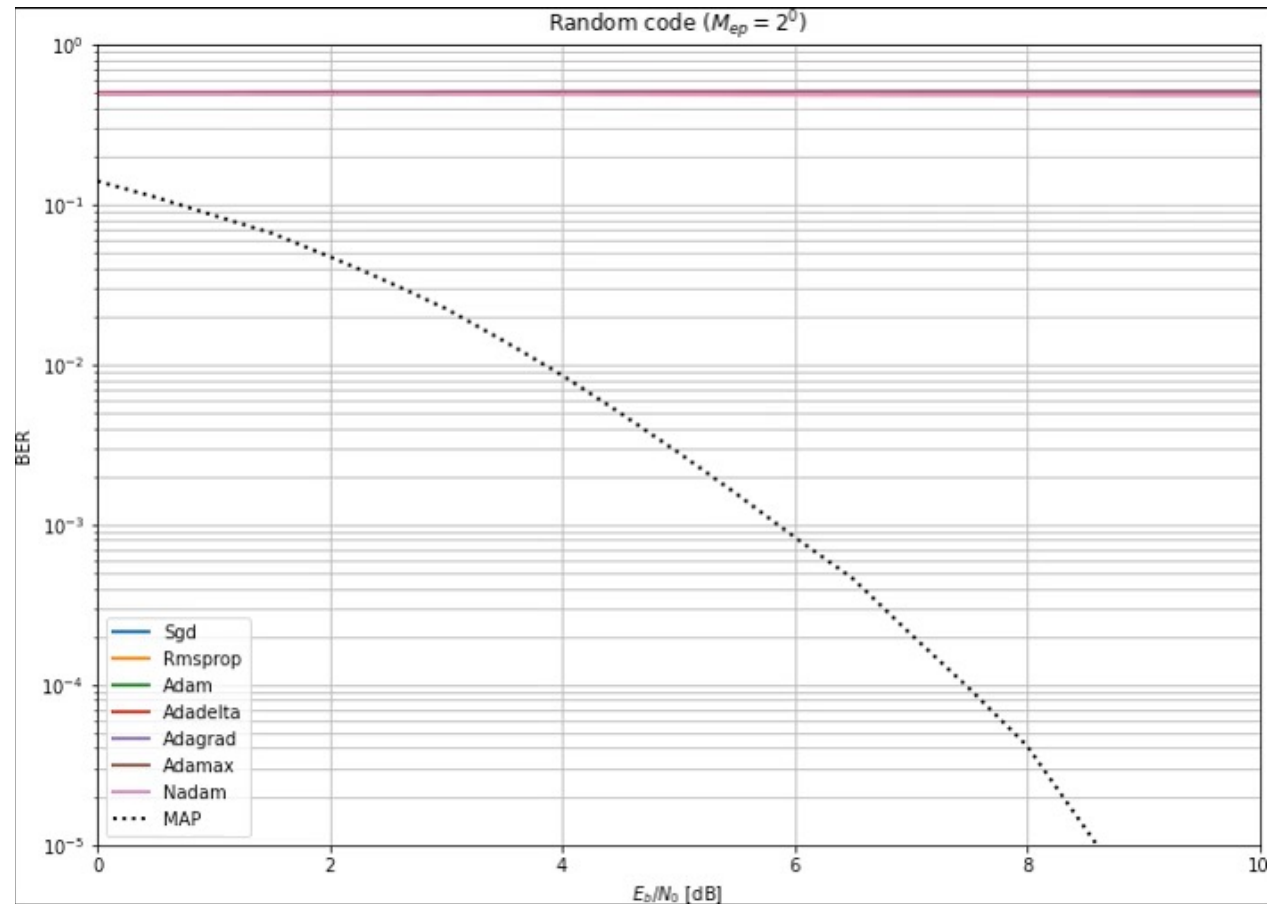
- Foram considerados os seguintes otimizadores:

SGD	atualização imediata dos pesos
Adagrad	taxa de aprendizado adaptativa
Adadelta	taxa de aprendizado adaptativa restrita
RMSprop	Adagrad com fator de decaimento
Adam	Adadelta/RMSprop com momentum
AdaMax	Adam com convergência mais estável
Nadam	Adam com momentum de Nesterov

Novos experimentos, resultados e conclusões



Novos experimentos, resultados e conclusões



Novos experimentos, resultados e conclusões

- É possível obter resultados melhores com decodificadores neurais mediante a otimização de hiperparâmetros.
- Neste caso, foram considerados outros otimizadores, mas há diversas possibilidades a serem exploradas.
- Além dos pontos mencionados anteriormente, sugere-se o emprego de *early stopping* para monitorar o decaimento do erro e evitar que a rede neural se sobreajuste.