

Equalização Cega de Canais SIMO utilizando Autocodificadores Variacionais

Sâmya Catta Preta Sena e Claudio José Bordin Jr.

Resumo— Os autocodificadores variacionais (VAEs) são redes neurais amplamente utilizadas em tarefas de aprendizado não supervisionado. Deduz-se neste trabalho uma extensão do algoritmo de equalização cega baseado em VAEs de Caciularu *et al.* para canais SIMO (*Single-Input Multiple-Output*). Propõe-se, ainda, uma formulação alternativa deste método com complexidade computacional reduzida. O desempenho dos algoritmos propostos foi avaliado através de simulações numéricas, que indicaram vantagens em relação a técnicas anteriores.

Palavras-Chave— Equalização Cega; Aprendizado de Máquina; Autocodificadores Variacionais

Abstract— Variational Autoencoders (VAEs) are neural networks widely employed in unsupervised learning tasks. In this work, we deduce an extension of the blind equalization algorithm by Caciularu *et al.* to SIMO (*Single-Input Multiple-Output*) channels. We also propose an alternative formulation for this method with reduced computational complexity. Numerical simulation results indicate that the proposed algorithms outperform previous methods.

Keywords— Blind Equalization; Machine Learning; Variational Autoencoders

I. INTRODUÇÃO

Técnicas de aprendizado de máquina têm sido amplamente empregadas na solução de problemas de processamento de sinais [1]. Dentre elas, destacam-se as redes neurais com múltiplas camadas (*deep learning*) [2], em diversas formulações. Os autocodificadores variacionais (*variational autoencoders* - VAEs) [3] são redes neurais profundas que visam reconstruir um sinal impondo estrutura ao mesmo. Os VAEs podem ser empregados para compressão de sinais ou redução de ruído, dentre outras aplicações [4].

O uso de VAEs para problemas de equalização cega [5] é novo [6]. Outros tipos de redes neurais, porém, já foram empregados para este fim. Em [7], por exemplo, propõe-se o uso de uma rede neural convolucional para a equalização cega de um canal de comunicação utilizando uma função custo inspirada na do algoritmo do módulo constante (CMA) [8].

Em [9], Caciularu *et al.* empregaram VAEs para a equalização cega de canais de comunicação SISO (*Single-Input Single-Output*). Neste trabalho, propõe-se a extensão de um dos algoritmos propostos em [9] para canais SIMO (*Single-Input Multiple-Output*). O novo método é deduzido sob a hipótese de que o ruído aditivo seja independente nos múltiplos

receptores. Propõe-se, ainda, uma variação deste método com complexidade computacional reduzida, que substitui uma esperança condicional necessária para o cálculo da função custo por uma aproximação estocástica. O desempenho dos métodos propostos é, então, avaliado através de simulações numéricas.

O texto a seguir está organizado da seguinte forma: na Seção II, descreve-se o modelo de sinal considerado. Na Seção III, por sua vez, apresenta-se a dedução da extensão do algoritmo de [9] para canais SIMO. Em seguida, na Seção IV, descreve-se uma versão do algoritmo anterior com complexidade computacional reduzida. Finalizando, o resultado de simulações numéricas é relatado na Seção V e as conclusões deste trabalho na Seção VI.

II. MODELO DE SINAL

Neste trabalho, será utilizado um modelo de canal seletivo em frequência FIR com uma entrada e múltiplas saídas (SIMO). Especificamente, o sinal recebido pelo subcanal de índice $r \in \{1, \dots, R\}$ no instante n é formulado como

$$y_{n,r} = \sum_{k=0}^{L-1} x_k h_{n-k,r} + w_{n,r}, \quad (1)$$

em que R é o número de subcanais, $x_k \in \{-1, 1\}$ é a sequência de símbolos BPSK transmitidos, $w_{n,r}$ é o ruído aditivo no r -ésimo subcanal e $h_{n,r}$ é a resposta ao impulso do r -ésimo subcanal, sem perda de generalidade suposta de ordem L . Supõe-se que x_k seja uma sequência de símbolos equiprováveis e i.i.d. (independentes e identicamente distribuídos), e que o $w_{n,r}$ seja um processo Gaussiano i.i.d. de média nula, variância σ_r^2 , independente de x_k e de $w_{n,s}$, $s \neq r$.

Os algoritmos analisados neste trabalho operam em batelada, processando de cada vez N amostras dos sinais recebidos. Assim, utilizando-se a notação

$$\begin{aligned} \mathbf{x} &\triangleq \{x_0, x_1, \dots, x_{N-1}\}, \\ \mathbf{w}_r &\triangleq \{w_{0,r}, w_{1,r}, \dots, w_{N-1,r}\}, \\ \mathbf{y}_r &\triangleq \{y_{0,r}, y_{1,r}, \dots, y_{N-1,r}\}, \\ \mathbf{y} &\triangleq \{\mathbf{y}_1, \dots, \mathbf{y}_R\}, \\ \mathbf{h}_r &\triangleq \{h_{0,r}, h_{1,r}, \dots, h_{L-1,r}\}, \end{aligned}$$

pode-se escrever $\mathbf{y}_r = \mathbf{x} * \mathbf{h}_r + \mathbf{w}_r$, em que $*$ denota a operação de convolução.

O objetivo dos algoritmos de equalização cega analisados a seguir é estimar \mathbf{x} dadas somente as observações \mathbf{y} .

III. AUTOCODIFICADOR VARIACIONAL PARA CANAIS SIMO

Nesta sessão, é descrita uma generalização do algoritmo VAE [9] para canais SIMO.

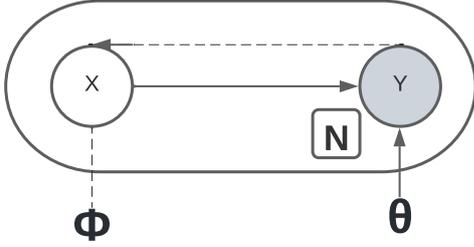


Fig. 1. Modelo Gráfico Direcional para Canal de Comunicação e para a Estrutura da Rede para Inferência. O modelo generativo $p_{\theta}(y|x)$ é denotado pela linha sólida. A aproximação variacional $q_{\Phi}(x|y)$ para $p_{\theta}(x|y)$ é denotada pela linha tracejada. (Adaptado de [9])

O modelo de sinal descrito na Seção II implica em que a função de massa de probabilidade (f.m.p.) de \mathbf{x} seja dada por

$$p(\mathbf{x}) = 2^{-N}. \quad (2)$$

Definindo o conjunto de parâmetros $\theta \triangleq \{\mathbf{h}_1, \dots, \mathbf{h}_R, \sigma_1^2, \dots, \sigma_R^2\}$, a função densidade de probabilidade (f.d.p.) condicional de \mathbf{y} dados \mathbf{x} e θ pode ser escrita como

$$\begin{aligned} p_{\theta}(\mathbf{y}|\mathbf{x}) &= \prod_{r=1}^R p_{\theta}(\mathbf{y}_r|\mathbf{x}), \\ &= \prod_{r=1}^R \mathcal{N}(\mathbf{x} * \mathbf{h}_r, \sigma_r^2 \mathbf{I}_N), \\ &= \prod_{r=1}^R \frac{1}{(2\pi\sigma_r^2)^{\frac{N}{2}}} \cdot e^{-\frac{\|\mathbf{y}_r - \mathbf{x} * \mathbf{h}_r\|^2}{2\sigma_r^2}}. \end{aligned} \quad (3)$$

sendo a primeira igualdade decorrente da independência do ruído nos múltiplos subcanais, em que $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Xi})$ denota uma f.d.p. Gaussiana N -variada com vetor de médias $\boldsymbol{\mu}$ e matriz de covariância $\boldsymbol{\Xi}$, e \mathbf{I}_N denota uma matriz identidade de tamanho N .

Para se determinar a estimativa de *máxima verossimilhança* (ML) de θ , é necessário calcular e maximizar

$$p_{\theta}(\mathbf{y}) = \sum_{\mathbf{x} \in \{\mathbf{X}\}} p_{\theta}(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \quad (4)$$

em que $\{\mathbf{X}\}$ denota o conjunto de todas as 2^N sequências possíveis \mathbf{x} , o que é inviável para $N \gg 1$.

O método de estimação variacional permite obter estimativas ML aproximadas substituindo $p_{\theta}(\mathbf{y})$ por um limitante inferior chamado de limitante inferior variacional ou limitante inferior de evidência. Para isto, explora-se o fato de

que [3]:

$$\begin{aligned} \log p_{\theta}(\mathbf{y}) &\geq \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [-\log q_{\Phi}(\mathbf{x}|\mathbf{y}) + \log p_{\theta}(\mathbf{x}, \mathbf{y})] \\ &= -\mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} \left[\log \left(\frac{q_{\Phi}(\mathbf{x}|\mathbf{y})}{p(\mathbf{x})} \right) \right] - \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [\log p(\mathbf{x})] \\ &\quad + \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [\log p_{\theta}(\mathbf{x}, \mathbf{y})], \\ &= \underbrace{-D_{KL}[q_{\Phi}(\mathbf{x}|\mathbf{y})||p(\mathbf{x})]}_{\triangleq A} + \underbrace{\mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [\log p_{\theta}(\mathbf{y}|\mathbf{x})]}_{\triangleq B}, \\ &\triangleq -\mathcal{L}(\theta, \Phi, \mathbf{y}). \end{aligned} \quad (5)$$

em que $q_{\Phi}(\mathbf{x}|\mathbf{y})$ é uma f.d.p. *arbitrária*, parametrizada por Φ , e $D_{KL}[\cdot||\cdot]$ indica a divergência de Kullback-Leibler [10].

Pelo método variacional, portanto, ao invés de se maximizar diretamente $p_{\theta}(\mathbf{y})$, minimiza-se a função custo $\mathcal{L}(\theta, \Phi, \mathbf{y})$ conjuntamente em θ e Φ . Pode-se mostrar [11] que, minimizando-se a mesma sobre θ e todas as densidades condicionais possíveis $q_{\Phi}(\mathbf{x}|\mathbf{y})$, obtém-se a estimativa ML de θ . A Figura 1 ilustra através de um modelo gráfico o problema de inferência em questão.

Tipicamente, ao se usar um VAE, tanto $p_{\theta}(\mathbf{y}|\mathbf{x})$ como $q_{\Phi}(\mathbf{x}|\mathbf{y})$ são aproximadas através de redes neurais. A densidade $p_{\theta}(\mathbf{y}|\mathbf{x})$ é computada através de um *decodificador* e $q_{\Phi}(\mathbf{x}|\mathbf{y})$ através de um *codificador*. Para o modelo de sinal em questão, não é necessária uma rede neural para implementar o decodificador, pois $p_{\theta}(\mathbf{y}|\mathbf{x})$ possui expressão analítica. O codificador, por outro lado, é implementado em [9] através de uma rede neural que faz o papel de um equalizador.

Em suma, o equalizador VAE opera da seguinte forma (vide Algoritmo 1): as saídas do canal \mathbf{y} são usadas para estimar os parâmetros θ e Φ . Após isso, obtém-se uma estimativa da sequência de símbolos \mathbf{x} usando a saída do codificador $q_{\Phi}(\mathbf{x}|\mathbf{y})$.

Uma vez que os símbolos $x_j \in \{-1, 1\}$, impõe-se que $q_{\Phi}(\mathbf{x}|\mathbf{y})$ seja uma distribuição Bernoulli multivariada com independência estatística entre seus componentes:

$$q_{\Phi}(\mathbf{x}|\mathbf{y}) \triangleq \prod_{j=0}^{N-1} q_{\Phi}(x_j|\mathbf{y}). \quad (6)$$

Denotando por $q_{\Phi,j}(\mathbf{y}) \triangleq q_{\Phi,j}(x_j = 1|\mathbf{y})$ as probabilidades condicionais dos eventos $x_j = 1$ dado \mathbf{y} , tem-se que

$$q_{\Phi}(\mathbf{x}|\mathbf{y}) = \prod_{j=0}^{N-1} (q_{\Phi,j}(\mathbf{y}))^{(1+x_j)/2} (1 - (q_{\Phi,j}(\mathbf{y}))^{(1-x_j)/2}). \quad (7)$$

A implementação do codificador é feita a partir de uma rede neural convolucional. A rede em questão possui uma camada de entrada convolucional, que aplica um filtro FIR separadamente à cada entrada, soma as saídas desses filtros e aplica ao resultado a função de ativação *softsign*, definida por $f(x) = x/(|x| + 1)$. A camada de saída determina $q_{\Phi}(\mathbf{x}|\mathbf{y})$ aplicando a função de ativação *sigmoide* (que assegura que a saída esteja em $[0, 1]$) ao resultado da soma da saída da camada de entrada com uma combinação linear das entradas não processadas. Verificou-se experimentalmente que o uso da função de ativação *softsign* na camada de entrada leva a melhores resultados que o uso de funções como *LeakyReLU*

e \tanh .

A. Cálculo da Função Objetivo

Para se deduzir a expressão de $\mathcal{L}(\boldsymbol{\theta}, \Phi, \mathbf{y})$ (Eq. 5), observa-se inicialmente que

$$\begin{aligned} A &\triangleq \sum_{\mathbf{x}} q_{\Phi}(\mathbf{x}|\mathbf{y}) \cdot (\log p(\mathbf{x}) - \log q_{\Phi}(\mathbf{x}|\mathbf{y})) \\ &= \sum_{\mathbf{x}} q_{\Phi}(\mathbf{x}|\mathbf{y}) \cdot (-N \log 2 - \log q_{\Phi}(\mathbf{x}|\mathbf{y})) \\ &= -N \log 2 + \mathcal{H}[q_{\Phi}(\mathbf{x}|\mathbf{y})], \end{aligned} \quad (8)$$

em que $\mathcal{H}[q_{\Phi}(\mathbf{x}|\mathbf{y})]$ denota a entropia de $q_{\Phi}(\mathbf{x}|\mathbf{y})$ dada por

$$\begin{aligned} \mathcal{H}[q_{\Phi}(\mathbf{x}|\mathbf{y})] &= \mathcal{H} \left[\prod_{j=0}^{N-1} q_{\Phi}(x_j|\mathbf{y}) \right] = \sum_{j=0}^{N-1} \mathcal{H}[q_{\Phi}(x_j|\mathbf{y})] \\ &= - \sum_{j=0}^{N-1} \{ q_{\Phi,j}(\mathbf{y}) \log q_{\Phi,j}(\mathbf{y}) + \\ &\quad (1 - q_{\Phi,j}(\mathbf{y})) \log(1 - q_{\Phi,j}(\mathbf{y})) \}. \end{aligned} \quad (9)$$

Para o termo B, tem-se:

$$\begin{aligned} B &= \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} \left[-\frac{N}{2} \log(2\pi\sigma_r^2) - \sum_{r=1}^R \frac{\|\mathbf{y}_r - \mathbf{x} * \mathbf{h}_r\|^2}{2\sigma_r^2} \right] \\ &= -\frac{N}{2} \log(2\pi) - \frac{N}{2} \log(\sigma_r^2) \\ &\quad - \sum_{r=1}^R \frac{1}{2\sigma_r^2} \underbrace{\mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [\|\mathbf{y}_r - \mathbf{x} * \mathbf{h}_r\|^2]}_{C_r}. \end{aligned} \quad (10)$$

Os termos C_r , interpretáveis como a esperança do erro quadrático de estimação do sinal recebido por cada subcanal, podem ser calculados analiticamente. Manipulando-se a expressão de C_r , pode-se escrever

$$\begin{aligned} C_r &= \sum_{n=0}^{N-1} \left\{ y_{n,r}^2 - 2 \left(y_{n,r} \sum_{k=0}^{L-1} \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k] \cdot h_{n-k,r} \right) + \right. \\ &\quad \left. \sum_{k,l=0}^{N-1} \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k x_l] \cdot h_{n-k,r} h_{n-l,r} \right\}, \end{aligned} \quad (11)$$

em que $h_{n,r} = 0$ para $n \geq L$.

Calculando-se as esperanças em (11), obtém-se que

$$\mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k] = 2q_{\Phi,k}(\mathbf{y}) - 1, \quad (12)$$

$$\mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k^2] = 1, \quad (13)$$

e, para $k \neq l$,

$$\begin{aligned} \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k x_l] &= \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k] \cdot \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_l] \\ &= (2q_{\Phi,k}(\mathbf{y}) - 1)(2q_{\Phi,l}(\mathbf{y}) - 1). \end{aligned} \quad (14)$$

Substituindo-se (12), (13) e (14) em (11) é possível obter uma expressão explícita para C_r . Para isto, observe que [9]:

$$\sum_{k,l=0}^{N-1} \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k x_l] \cdot h_{n-k,r} h_{n-l,r} =$$

$$= \left\{ \sum_{k=0}^{N-1} \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})} [x_k] \cdot h_{n-k,r} \right\}^2 +$$

$$\sum_{k=0}^{N-1} h_{n-k,r}^2 \left\{ 1 - \mathbb{E}_{q_{\Phi}(\mathbf{x}|\mathbf{y})}^2 [x_k] \right\}. \quad (15)$$

Consequentemente,

$$C_r = \sum_{k=0}^{N-1} \|\mathbf{y}_{n,r}\|^2 - 2\alpha_{n,r} + \beta_{n,r}, \quad (16)$$

em que

$$\alpha_{n,r} = \sum_{k=0}^{N-1} y_{n,r} h_{n-k} \cdot (2q_{\Phi,k}(\mathbf{y}) - 1), \quad (17)$$

e

$$\begin{aligned} \beta_{n,r} &= \sum_{k=0}^{N-1} h_{n-k,r} \cdot (2q_{\Phi,k}(\mathbf{y}) - 1) + \\ &\quad \sum_{k=0}^{N-1} (h_{n-k,r})^2 [4q_{\Phi,k}(\mathbf{y}) - 4q_{\Phi,k}^2(\mathbf{y})]. \end{aligned} \quad (18)$$

B. Treinamento

O treinamento do VAE envolve minimizar $\mathcal{L}(\boldsymbol{\theta}, \Phi, \mathbf{y}) = -A - B$ em relação a $\boldsymbol{\theta} = \{\mathbf{h}, \sigma_1^2, \dots, \sigma_R^2\}$ e a Φ . Para se calcular o mínimo em relação às variâncias σ_r^2 , $r = 1, \dots, R$, que só afetam o termo B, calcula-se o gradiente deste termo e iguala-se o resultado a zero. Disto se obtém que o valor ótimo de σ_r^2 é dado por C_r/N . Utilizando este resultado e (8), pode-se escrever que

$$\mathcal{L}(\boldsymbol{\theta}, \Phi, \mathbf{y}) = \frac{N}{2} \sum_{r=1}^R \{\log(C_r)\} - \mathcal{H}[q_{\Phi}(\mathbf{x}|\mathbf{y})] + c, \quad (20)$$

em que $\mathcal{H}[q_{\Phi}(\mathbf{x}|\mathbf{y})]$ é dado em (9), C_r em (16), e c é uma constante que não depende dos parâmetros livres $(\boldsymbol{\theta}, \Phi)$.

Os parâmetros desconhecidos remanescentes (\mathbf{h} e Φ) são estimados através da minimização baseada em gradiente descendente de (20). Cabe notar que não é necessário calcular analiticamente os gradientes; nas simulações relatadas a seguir, os mesmos são calculados numericamente pelo software *Tensorflow* [12].

Finalmente, note que não é necessário se conhecer o valor exato da ordem do canal L para se executar o algoritmo; ao invés disso, basta se ter um limite superior para o qual a resposta ao impulso do canal verdadeiro possa ser bem aproximada.

Estimação dos Símbolos: uma vez determinado Φ que minimiza (20), a sequência de símbolos \mathbf{x} pode então ser estimada a partir de $q_{\Phi,j}(\mathbf{y})$ (Eq. 6), o que pode ser interpretado como uma decodificação suave (*soft decoding*) [13] desta sequência.

O Algoritmo 1 sumariza o método descrito nesta seção.

Algoritmo 1 Algoritmo VAE

 Dados $\mathbf{y}_1, \dots, \mathbf{y}_R$ e t_{max} (número de épocas):

for $t \leftarrow 1$ to t_{max} **do**

- Calcule $q_{\Phi}(\mathbf{x}|\mathbf{y})$ através do codificador (Fig. 2).
- Calcule C_r , $1 \leq r \leq R$ usando a Eq. 16.
- Determine numericamente os parâmetros θ (codificador) e Φ (decodificador) que minimizam a função custo $\mathcal{L}(\theta, \Phi, \mathbf{y})$ (Eq. 20).

end for

 Determine uma estimativa de \mathbf{x} a partir da f.m.p. $q_{\Phi}(\mathbf{x}|\mathbf{y})$.

IV. IMPLEMENTAÇÃO ALTERNATIVA DO ALGORITMO VAE SIMO

 Uma implementação alternativa do algoritmo VAE (vide Algoritmo 2) consiste em substituir a esperança dos termos C_r em (10) por uma aproximação *estocástica*, i.e., fazendo

$$C_r \approx \|\mathbf{y}_r - \tilde{\mathbf{x}} * \mathbf{h}_r\|^2, \quad (21)$$

 em que $\tilde{\mathbf{x}}$ é uma amostra de $q_{\Phi}(\mathbf{x}|\mathbf{y})$, que é determinada pelo codificador.

Esse método tem menor complexidade computacional, pois elimina a necessidade de se calcularem as múltiplas somas de convolução de (11).

Algoritmo 2 Algoritmo VAE-sample

 Dados $\mathbf{y}_1, \dots, \mathbf{y}_R$ e t_{max} (número de épocas):

for $t \leftarrow 1$ to t_{max} **do**

- Calcule $q_{\Phi}(\mathbf{x}|\mathbf{y})$ através do codificador (Fig. 2).
- Gere uma amostra $\tilde{\mathbf{x}}$ de $q_{\Phi}(\mathbf{x}|\mathbf{y})$.
- Usando $\tilde{\mathbf{x}}$ e a expressão para C_r , $1 \leq r \leq R$ da Eq. 21, minimize numericamente a função custo $\mathcal{L}(\theta, \Phi, \mathbf{y})$ (Eq. 20).

end for

 Retorne $\tilde{\mathbf{x}}$ (estimativa dos símbolos).

V. RESULTADOS DE SIMULAÇÃO

 Para avaliar o desempenho dos algoritmos descritos nas Seções III e IV, realizaram-se simulações numéricas que empregaram 600 realizações independentes. Utilizou-se um canal SIMO com $R = 2$ saídas; os subcanais empregados têm resposta ao impulso

$$\mathbf{h}_1 = [0.0661, 0.34376866, -0.9318, -0.0778, 0.0566],$$

$$\mathbf{h}_2 = [0.1594, -0.3816, 0.8888, -0.1836, -0.0725].$$

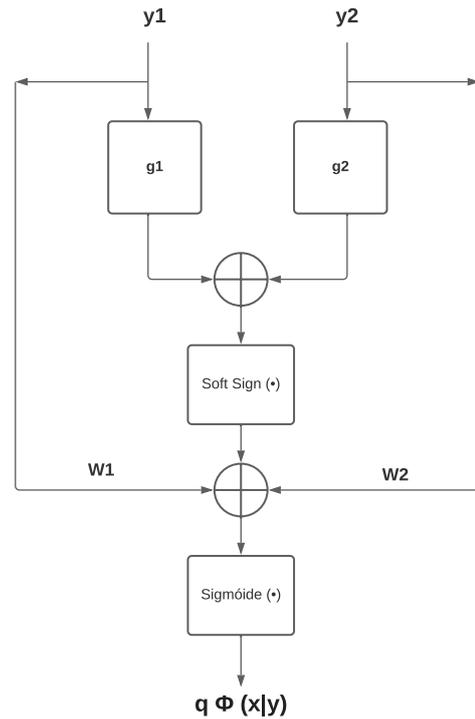
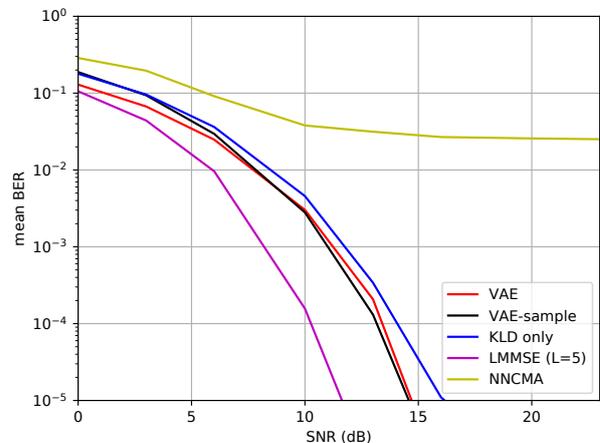
 Os algoritmos empregaram 128 amostras de treinamento. Observe que, por se tratarem de algoritmos *cegos*, o treinamento utiliza apenas amostras dos sinais recebidos. As camadas convolucionais dos algoritmos empregaram filtros com 5 coeficientes. O treinamento dos algoritmos usou o otimizador ADAM [14], com taxa de aprendizado 0,05 durante 300 épocas, utilizando sempre os mesmos dados para treinamento. Para o teste dos algoritmos, foram utilizadas seqüências distintas das de treinamento com 1.000 amostras de sinal recebido.

 Fig. 2. Estrutura do *codificador*, comum para os algoritmos das Seções III e IV. Cada entrada é processada por um filtro separado (g_1 e g_2). w_1 e w_2 são pesos adaptáveis.


Fig. 3. Taxa média de erro de bit (BER) ao longo de 600 realizações independentes para os algoritmos VAE (Algoritmo 1), VAE-sample (Algoritmo 2) e outros descritos na Seção V.

 O *codificador* VAE utilizou a estrutura mostrada na Figura 2. Vale ressaltar que, para canais SIMO, a estrutura com duas camadas convolucionais proposta em [9] exibiu desempenho semelhante à utilizada.

Para comparação de desempenho, simularam-se os resultados, sob as mesmas condições, dos algoritmos:

- i) NNCMA [7], sem linearização, ou seja, com o parâmetro $K = 1$;
- ii) KLD (de *Kullback-Leibler Divergence*), um novo algoritmo que utiliza como função custo apenas o termo A de (5);

iii) LMMSE, definido com o Filtro de Wiener [15] com 5 coeficientes, determinado através dos parâmetros exatos (\mathbf{h}_1 , \mathbf{h}_2 , σ_1^2 , σ_2^2).

Os valores de σ_r^2 foram supostos iguais para todos os subcanais e foram determinados de modo a produzir a relação sinal-ruído (SNR) desejada.

A Figura 3 mostra o desempenho dos algoritmos supracitados em termos da taxa média de erro de bit (BER) em função da SNR. Como se pode observar, os algoritmos VAE têm um desempenho bastante superior ao do NNCMA, cujo desempenho converge para um patamar com BER da ordem de $3 \cdot 10^{-2}$ para SNRs superiores a 10dB, resultado semelhante ao reportado em [7]. No entanto, os algoritmos VAE exibem uma perda de desempenho de cerca de 3dB em relação ao algoritmo LMMSE (para BER de 10^{-4}).

Os algoritmos VAE das Seções III e IV, surpreendentemente, exibiram desempenhos semelhantes, apesar da aproximação empregada pelo algoritmo da Seção IV. Por fim, observe que o algoritmo KLD, que não leva em conta o erro de reconstrução produzido pelo decodificador, tem um desempenho semelhante ao dos algoritmos VAE.

VI. CONCLUSÕES

Propôs-se neste trabalho uma extensão do algoritmo de equalização cega baseado em autocodificadores variacionais de [9] para canais SIMO. Propuseram-se ainda duas modificações deste algoritmo: uma nova versão que emprega aproximações amostrais de certas esperanças e outra com um função custo simplificada.

Os três algoritmos propostos exibem desempenhos semelhantes, e bastante superiores aos do método NNCMA, sendo, porém, inferiores em cerca de 3dB em relação ao filtro de Wiener, calculado com os parâmetros exatos do canal. Como pesquisa futura, pretende-se investigar se o uso de estruturas alternativas para o codificador VAE permite reduzir essa perda de desempenho.

REFERÊNCIAS

- [1] M. A. Little. *Machine Learning for Signal Processing: Data Science, Algorithms, and Computational Statistics*. Oxford University Press, 2019.
- [2] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [3] D. P Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [4] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.
- [5] Z. Ding and Ye Li. *Blind equalization and identification*. CRC press, 2018.
- [6] A. Caciularu and D. Burshtein. Blind channel equalization using variational autoencoders. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2018.
- [7] X. H. Dai. Cma-based nonlinear blind equaliser modelled by a two-layer feedforward neural network. *IEE Proceedings-Communications*, 148(4):243–248, 2001.
- [8] D. Godard. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Transactions on Communications*, 28(11):1867–1875, 1980.
- [9] A. Caciularu and D. Burshtein. Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders. *IEEE Transactions on Cognitive Communications and Networking*, 6(3):1003–1018, 2020.
- [10] S. Kullback. *Information Theory and Statistics*. A Wiley publication in mathematical statistics. Dover Publications, 1997.

- [11] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.
- [12] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.
- [13] J. G. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill, 2008.
- [14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [15] P. S. R. Diniz. *Adaptive Filtering: Algorithms and Practical Implementation*. SpringerLink : Bücher. Springer US, 2012.