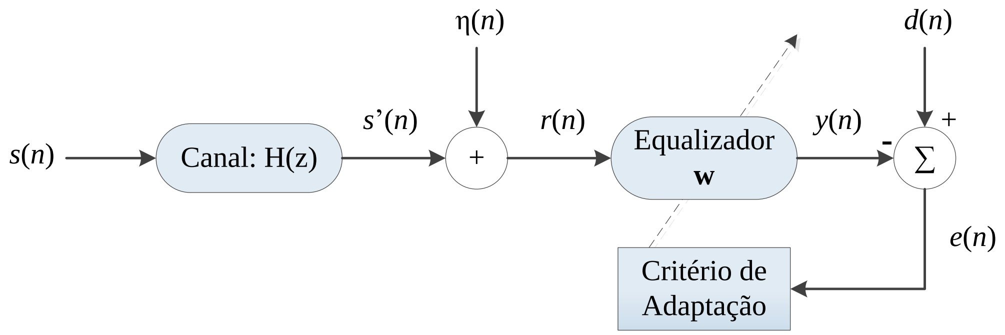
IA353 – Exercícios de Fixação de Conceitos
EFC 2 – 1s2017

Questão 6) (1,5 pontos) Data de entrega: 22/05/2017

Considere o problema de equalização de canais de comunicação, cujos principais elementos são mostrados na figura abaixo:



O objetivo é projetar um filtro, chamado de equalizador, que consiga cancelar as distorções introduzidas pelo canal durante a transmissão do sinal de informação $s(n)$.

O canal é modelado por um sistema linear e invariante com o tempo, com resposta ao impulso finita, dada pelo vetor de coeficientes $\mathbf{h} = [h_0 \ \ldots \ h_{D-1}]$ e função de transferência $H(z) = h_0 + h_1 z^{-1} \ldots + h_{D-1} z^{-(D-1)}$. Portanto, o sinal recebido $r(n)$, que corresponde à entrada do filtro equalizador, é dado por:

$$r(n) = h(n) * s(n) + \eta(n),$$

onde $\eta(n)$ representa um ruído aditivo gaussiano branco (AWGN), de média nula e variância σ^2 e $*$ denota a operação de convolução. A relação sinal-ruído (SNR) indica a potência relativa da parcela de sinal ($s'(n) = h(n) * s(n)$) frente ao ruído ($\eta(n)$) presente nos dados.

Parte I: Entendimento do problema

Neste exercício, o sinal de informação $s(n)$ é um processo aleatório i.i.d. (independente e identicamente distribuído) cujos símbolos pertencem ao alfabeto {+1,-1} (modulação BPSK). O vetor de entrada do equalizador – cuja estrutura será não-linear, implementada por uma rede neural – é formado pela amostra atual e passada do sinal recebido, i.e., $\mathbf{r}(n) = [r(n) \ r(n-1)]^T$. Por sua vez, a saída desejada corresponde a $d(n) = s(n - d)$, onde d é o chamado atraso de equalização.

Por conta da natureza discreta de $s(n)$, o problema de equalização também pode ser visto como uma tarefa de classificação (binária, no caso BPSK), na qual cada vetor de sinal recebido $\mathbf{r}(n)$ deve ser associado à classe correspondente ao valor do símbolo a ser recuperado ($s(n-d)$), dado um valor de atraso de equalização desejado. Particularmente no caso em que $\mathbf{r}(n) \in \Re^2$, é possível visualizar a distribuição dos dados e, deste modo, analisar os desafios envolvidos na separação das classes.

Dois cenários serão considerados neste exercício:

(1) $H(z) = 0{,}9 + z^{-1}$ ou $\mathbf{h} = [0{,}9\ \ 1]$, para atraso $d = 0$.
(2) $H(z) = 1 + 0{,}5z^{-1} + 0{,}2z^{-2}$ ou $\mathbf{h} = [1\ \ 0{,}5\ \ 0{,}2]$, para atraso $d = 2$.

(a) Utilize a rotina [visualiza_cenario_equalizacao.m], fornecendo o vetor **h** e o atraso *d* como argumentos, para mostrar o posicionamento dos estados do canal, que correspondem aos possíveis vetores de sinal recebido $\mathbf{r}(n)$ quando não existe ruído na transmissão. Nas mesmas figuras, exiba também um conjunto de dados (i.e., vetores $\mathbf{r}(n)$) que contenham ruído, considerando uma SNR de 15 dB. Para gerar estes dados, use a rotina [gera_dados_equalizacao.m], fornecendo os argumentos de entrada necessários. Com base nas figuras geradas (uma para cada canal), discuta, então, as características dos cenários (e.g., com respeito à facilidade de separar os estados).

Parte II: Análise da superfície de erro

Sabe-se que o ajuste de pesos sinápticos de uma rede neural MLP está sujeito a mínimos locais e a pontos de sela da superfície de erro. Esses mínimos locais do processo de otimização, e principalmente os pontos de sela, causam mais dificuldades no caso de arquiteturas profundas (*deep learning*). Mesmo no caso de arquiteturas com uma única camada intermediária, é possível visualizar mínimos locais com certa facilidade.

(b) Apresente, então, a superfície de erro de uma rede MLP que mostra a ocorrência de ao menos um mínimo local, ao variar somente dois pesos sinápticos, mantendo os demais fixos. Com isso, a superfície de erro pode ser visualizada no espaço $\Re^3$.

OBS: Junto com o relatório, entregue também um arquivo [RA do aluno].mat, cujo conteúdo deve ter:

- A matriz $\mathbf{W}_1$ de pesos da camada intermediária, de dimensão $(m+1) \times N$, onde $m = 2$ é o número de entradas da rede e N é o número de neurônios na camada intermediária.
- A matriz $\mathbf{W}_2$ de pesos da camada de saída, de dimensão $1 \times (N+1)$.
- A matriz $\mathbf{X}$ com os dados de entrada (treinamento), de dimensão $m \times T$, onde T denota o número de padrões do conjunto de treinamento.
- A matriz $\mathbf{Y}$ com as saídas desejadas, de dimensão $1 \times T$.
- O vetor $\mathbf{p}_1$ indicando o primeiro peso que sofreu variação, com 5 elementos, tal que $\mathbf{p}_1 = [i\ j\ k \text{ min max}]$, onde (i,j) representa a coordenada do peso, $k = 1$ se o peso for da matriz $\mathbf{W}_1$ e $k = 2$ se for de $\mathbf{W}_2$, e [min,max] são os limites do intervalo de variação considerado.
- O vetor $\mathbf{p}_2$ indicando o primeiro peso que sofreu variação, com 5 elementos, tal que $\mathbf{p}_2 = [i\ j\ k \text{ min max}]$, onde (i,j) representa a coordenada do peso, $k = 1$ se o peso for da matriz $\mathbf{W}_1$ e $k = 2$ se for de $\mathbf{W}_2$, e [min,max] são os limites do intervalo de variação considerado.

Parte III: Aplicação de redes perceptron de múltiplas camadas (MLP)

(c) Projete uma rede MLP utilizando um algoritmo de otimização de sua escolha para desempenhar o papel de equalizador. Se o método de otimização assim exigir, utilize um esquema de busca unidimensional para escolher o passo de adaptação. Lembre-se de empregar uma metodologia de validação cruzada ou então de *holdout*. Sejam criteriosos na escolha dos parâmetros (e.g., número de neurônios na camada intermediária) e justifiquem todas as opções relevantes feitas.

(d) Analise o comportamento do algoritmo de treinamento por meio da curva de evolução do erro quadrático (médio) em função do número de épocas. Mostre também a progressão do erro em relação ao conjunto de validação. Use escala logarítmica se auxiliar na visualização. Comente os resultados.

(e) Aplique a rede projetada sobre um conjunto de teste com 1000 amostras. Mostre a saída gerada para cada entrada e calcule o erro quadrático médio de teste, assim como o percentual de erro.

(f) Mostre o mapeamento entrada-saída ($\Re^2 \rightarrow \Re^1$) efetivamente produzido pela rede otimizada, assim como a fronteira de decisão gerada no espaço dos dados de entrada. Para isto, utilize a rotina [map_fronteira_mlp_equalizacao.m]. Comente o que foi observado.

Parte IV: Aplicação de redes com função de base radial (RBF)

(g) Projete uma rede neural RBF, empregando o algoritmo *k-means* para a definição da posição dos centros. Não é necessário empregar regularização na definição dos pesos da camada de saída. Justifique as escolhas relevantes feitas para os diversos parâmetros.

(h) Modifique o programa [map_fronteira_mlp_equalizacao.m] para, agora, gerar o mapeamento e a fronteira de decisão associados à rede RBF projetada. Nesta última figura, inclua também as posições encontradas pelo algoritmo *k-means* para os centros das funções de base radial.

(i) Aplique a rede projetada sobre um conjunto de teste com 1000 amostras. Mostre a saída gerada para cada entrada e calcule o erro quadrático médio de teste, assim como o percentual de erro. Discuta os resultados obtidos