

# Problema MNIST com pré-processamento auto-organizado

IZABELLA THAÍS OLIVEIRA GOMES

12/0152401

igomesizabella@gmail.com

LETÍCIA CÂMARA VAN DER PLOEG

12/0152771

leticiaavdploeg@gmail.com

Junho/2016

## Resumo

*Neste documento é descrito o processo de implementação de um mapa auto-organizável, ou rede de Kohonen, para o pré-processamento de um problema de reconhecimento de padrões. A rede é treinada de maneira não supervisionada afim de reduzir a dimensão. Os padrões são obtidos da base de dados MNIST, que fornece imagens manuscritas dos dígitos de 0 a 9.*

## I. INTRODUÇÃO

O reconhecimento de padrões é um dos processos mais importantes para o desenvolvimento da Inteligência Artificial, uma vez que, dado uma classe de caracteres pré-definidos, é possível a identificação de novos caracteres pelo sistema inteligente. Este trabalho configura uma implementação inicial que aproxima a ideia de uma entrada desconhecida do computador que possa ser identificada ao contexto tecnológico atual.

A rede de Kohonen, ou mapa auto-organizável (Self-Organizing Map, SOM), desenvolvida por Teuvo Kohonen em 1982 permite a representação de dados multidimensionais em espaços de dimensão menor, geralmente uma ou duas dimensões. Além da quantização vetorial, a rede é capaz de guardar informações topológicas dos dados, gerando uma classificação em clusters.

Um SOM é formado por nós e é dividido em duas camadas: *camada de entrada* e *camada de Kohonen*. Cada nó da primeira camada recebe todas as entradas e distribui os valores para

a próxima camada. Além disso, possui a ele associado uma posição na configuração tabular do mapa e um conjunto de pesos  $w$ .

Diferentemente das redes neurais estudadas anteriormente, como a rede Perceptron, a aprendizagem da rede de Kohonen é sem supervisão e competitiva. Este método avalia os neurônios de saída da rede determinando como resultado o neurônio que possui maior ativação.

### i. O algoritmo

#### *Passo 1*

Os vetores de peso de dimensão  $N$   $w_1, w_2, \dots, w_m$  das  $m$  unidades computadas são selecionados aleatoriamente. Um raio inicial  $r$ , uma taxa de aprendizado inicial  $\eta$  e uma função de vizinhança  $\phi$  são selecionados.

#### *Passo 2*

Deve-se selecionar um vetor de entrada  $\xi$  utilizando a distribuição de probabilidade desejada sobre o espaço de entrada.

#### Passo 3

A unidade  $k$  ganhadora, que possui a menor distância euclidiana 1 em relação à entrada (distância entre  $w_i$  e  $\xi$  é mínima,  $i = 1, \dots, m$ ), é selecionada.

$$dist = \sqrt{\sum_{i=0}^n (V_i - W_i)^2} \quad (1)$$

#### Passo 4

Os vetores de peso são atualizados utilizando a função de vizinhança e a regra de atualização:

$$w_i = w_i + \eta \phi(i, k) (\xi - w_i) \quad (2)$$

#### Passo 5

Deve-se parar se o número máximo de épocas  $n$  for atingido, caso contrário deve-se modificar  $\eta$  e  $\phi$  como definido e seguir para o passo 2.

Na implementação realizada neste trabalho, o algoritmo foi seguido, alterando-se a forma de decisão do ganhador no aprendizado competitivo como mostra a equação da distância quadrática 3, a taxa de aprendizado 4 e a função de vizinhança 5.

$$d_{j,i}^2 = ||\mathbf{r}_j - \mathbf{r}_i|| \quad (3)$$

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), n = 0, 1, 2, \dots \quad (4)$$

$$h_{j,i}(\mathbf{x}) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right), n = 0, 1, 2, \dots \quad (5)$$

## II. METODOLOGIA

A implementação da rede de Kohonen foi realizada utilizando o ambiente de simulação *Matlab* e está disponível no arquivo *KohonenAlgorithmSOM.m*. Para a análise do desempenho da rede será utilizada também a implementação desenvolvida no trabalho 1 [3], em que o reconhecimento de padrões MNIST será feito considerando uma dimensão menor. As imagens do MNIST possuem 28x28 pixels, ou seja,

uma dimensão original igual a 784. O SOM deve reduzir a imagem para  $N \times N$ , onde  $N$  está entre 1 e 27. Foi adotada a seguinte metodologia:

- A rede Kohonen é treinada com o conjunto de 60000 imagens, variando o número de épocas, a configuração do mapa ( $N$ ), a constante de atualização da taxa de aprendizado e a constante de atualização da função de vizinhança;
- Verifica-se o resultado da rede isoladamente como classificadora;
- A rede Perceptron Multicamada é treinada com o conjunto de 60000 imagens já processadas pela rede de Kohonen para as mesmas condições do primeiro item;
- A rede Perceptron Multicamada é testada com o conjunto de 10000 já processadas pela rede Kohonen.

Nas seções a seguir, a metodologia é aplicada e os resultados são apresentados e analisados.

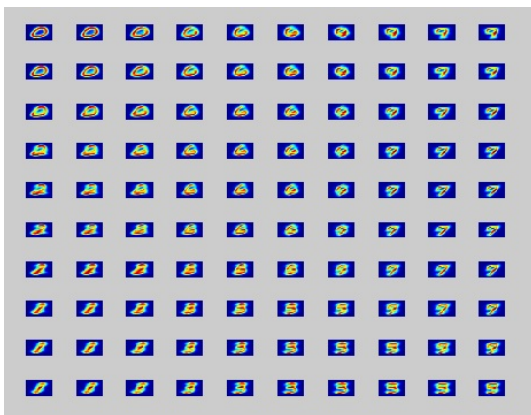
## III. RESULTADOS

### i. Rede Kohonen

Para um número de épocas muito alto, a rede apresentou um desempenho baixo associado a um tempo de processamento muito longo. Portanto, optou-se por utilizar um número de épocas menor (20 épocas) e as configurações foram ajustadas considerando os outros parâmetros citados na seção II.

Para todas as configurações testadas a seguir, a rede mostrou um comportamento classificatório de forma que todos os nós com pesos semelhantes foram dispostos próximos uns aos outros. A figura 1 exemplifica este resultado.

Para cada configuração o mapa foi treinado com base nas 60000 imagens e testado com base nas 10000 imagens. O erro foi calculado considerando os testes, como mostrado a seguir.



**Figura 1:** Resultado classificatório com a rede de Kohonen para base de dados MNIST dada a configuração 1

**Configuração 1:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **1000**, constante de tempo para atualização da função de vizinhança igual a **1000/log( $\sigma$ )**  
 Erro associado de 5263.

**Configuração 2:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **500**, constante de tempo para atualização da função de vizinhança igual a **1000/log( $\sigma$ )**  
 Erro associado de 5951.

**Configuração 3:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **100**, constante de tempo para atualização da função de vizinhança igual a **1000/log( $\sigma$ )**  
 Erro associado de 4989.

**Configuração 4:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **1000**, constante de tempo para atualização da função de vizinhança igual a **1500/log( $\sigma$ )**  
 Erro associado de 4542.

**Configuração 5:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **500**, constante de tempo para atu-

alização da função de vizinhança igual a **1500/log( $\sigma$ )**  
 Erro associado de 5910.

**Configuração 6:** Rede **10x10**, constante de tempo para cálculo da taxa de aprendizado igual a **3000**, constante de tempo para atualização da função de vizinhança igual a **50/log( $\sigma$ )** e razão inicial da vizinhança igual a **1/5**  
 Erro associado de 4575.

## ii. Rede Backpropagation pré-processada

Utilizando a saída do mapa de Kohonen como entrada para a rede Backpropagation, espera-se obter um percentual de erro menor, ou seja, a entrada pré-processada deve implicar em um desempenho melhor da rede. A melhor configuração da rede Backpropagation obtida no trabalho 1 é descrita por:

- Função de ativação: logística;
- Número de épocas: 20;
- Taxa de aprendizado: 0,05;
- Número de camadas: 2;
- Neurônios por camada: 100, 10 (saídas).

A configuração para o mapa corresponde a **configuração 4**, que apresenta melhor desempenho. Normalizando a saída do mapa de Kohonen (distâncias euclidianas) e utilizando como entradas para o Backpropagation, realiza-se o treinamento com 60000 imagens e o teste com 10000 imagens. O treinamento da rede apresentou o erro quadrático médio que é mostrado na figura 2. Após os testes, a rede apresentou erro de 49,84%. Este valor está acima do esperado, visto que sem o pré-processamento a rede apresentou erro de 12,13%.

Alterando apenas o número de épocas para 100, o treinamento da rede apresentou o erro quadrático médio que é mostrado na figura 3. Após os testes, a rede apresentou erro de 29,96%. Este valor, apesar de melhor, continua acima do erro apresentado pela rede no trabalho 1.

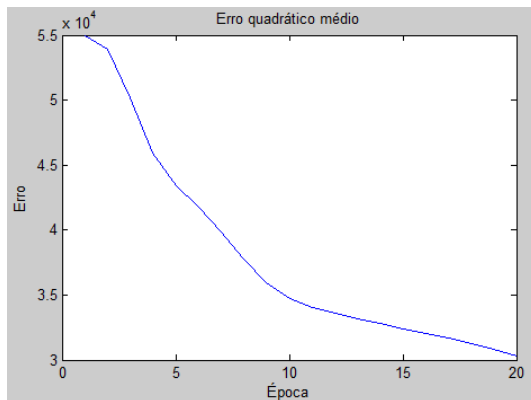


Figura 2: Erro quadrático médio para MLP descrita

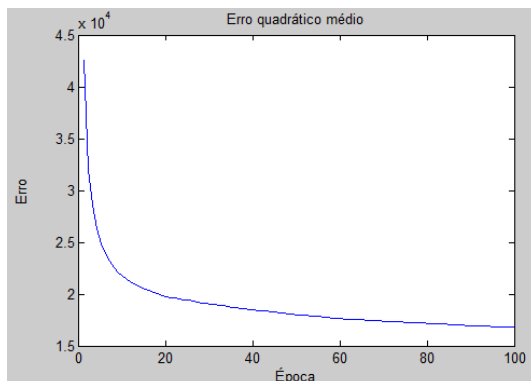


Figura 3: Erro quadrático médio para MLP com 100 épocas

#### IV. ANÁLISE

##### i. Rede Kohonen

Observando os resultados das configurações apresentadas na seção III.i, percebe-se que o resultado da rede para a aplicação proposta não é satisfatório. O melhor resultado (menor erro) é dado pela configuração que possui uma constante de tempo da taxa de aprendizado de 1500 e uma constante de tempo para atualização da função de vizinhança igual a  $1000/\log(\sigma)$  num mapa  $10 \times 10$ . Para a redução de dimensionalidade de 2,8, o erro é de 45,42%.

##### ii. Rede Backpropagation pré-processada

Os resultados mostrados na seção III.ii mostram que as duas configurações adotadas pos-

suem um erro superior ao obtido no trabalho anterior, ou seja, o pré-processamento com o mapa de Kohonen foi falho. Apesar disso, é possível identificar que com um número maior de épocas, levando um custo de processamento de 6 horas, o erro foi reduzido para, aproximadamente, 30%. A aplicação, portanto, não é a mais adequada para o pré-processamento utilizado.

#### V. CONCLUSÃO

Após a implementação, conclui-se que o mapa de Kohonen desenvolvido para atuar como método de redução de dimensão dos dados obtidos da MNIST resulta em um resultado fraco. O número de erros do mapa unicamente como classificatório é elevado, apesar das diversas configurações testadas e da otimização do algoritmo. Aplicando este mapa à rede Backpropagation, observa-se uma taxa de erro superior ao que foi encontrado com a rede sem pré-processamento, o que confirma que o mapa não é a melhor alternativa para o pré-processamento do banco de imagens utilizado.

#### REFERÊNCIAS

- [1] Haykin, Simon. Neural Networks: A Comprehensive Foundation, 1999. Pearson Education, Inc.
- [2] A.I junkie. Kohonen's Self Organizing Feature Maps Disponível em: <<http://www.ai-junkie.com/ann/som/som1.html>>
- [3] Gomes, Izabella e Van Der Ploeg, Letícia (2016). Implementação de perceptron multicamada para reconhecimento de padrões. Disciplina Introdução ao Controle Inteligente e Numérico.