



DINCON 2017

CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES

30 de outubro a 01 de novembro de 2017 – São José do Rio Preto/SP

Classificação de dinâmicas de sistemas utilizando redes neurais LVQ adaptativas

Renan Fonteles Albuquerque¹

Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Ceará

Arthur Plínio de Souza Braga²

Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Ceará

Bismark Claire Torrico³

Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Ceará

Laurinda Lúcia Nogueira dos Reis⁴

Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Ceará

Resumo. A classificação de dinâmicas em sistemas é uma área de pesquisa emergente no campo de análise de sinais e séries temporais. O reconhecimento de padrões em séries temporais é um problema de interesse em diversas áreas, contendo aplicações interessantes para o uso de técnicas de inteligência computacional. Entre as soluções de algoritmos para reconhecimento de dinâmicas em sistemas, as Redes Neurais Artificiais, por sua capacidade de generalização, tornam-se interessante para tratar o problema. Este trabalho apresenta uma variação adaptativa do algoritmo Learning Vector Quantization (LVQ) com o objetivo de ser aplicado na classificação de dinâmicas em sinais e sistemas. A metodologia de avaliação do algoritmo proposto consiste de um estudo de caso que trata da classificação de atividades diárias realizadas por pessoas a partir dos sinais obtidos de um acelerômetro. A acurácia média obtida com o algoritmo proposto é de aproximadamente 80.05% e os resultados obtidos foram comparados com outras variações do LVQ.

Palavras-chave. Reconhecimento de padrões, Redes Neurais Artificiais, Learning Vector Quantization, Classificação de dinâmicas.

1 Introdução

O reconhecimento de padrões em dinâmicas tem sido alvo de inúmeras pesquisas relacionadas a classificação. Muitos trabalhos utilizam dados contínuos de sensores, tais

¹renanfonteles@gmail.com

²arthurp@dee.ufc.br

³bismark@dee.ufc.br

⁴laurinda@dee.ufc.br

como acelerômetros, sinais de Eletrocardiograma (ECG), Eletroencefalograma (EEG), além de outras fontes de dados para classificar padrões de sinais que representam um estado específico do sistema. Por exemplo, em [1] os autores apresentam um estudo sobre a utilização de Redes Neurais Artificiais e métodos estatísticos para classificação de sinais de transmissão em comunicação entre satélites. Em [2,3] os autores apresentam um novo modelo de classificação de sinais de EEG utilizando *Deep Neural Networks (DNN)*. No trabalho [4] os autores classificaram sinais ECG a partir de uma análise linear de *features*.

Outra área de pesquisa explorada na classificação de dinâmicas é a de reconhecimento de atividades humanas utilizando dados de acelerômetros. Muitos trabalhos utilizam dados de acelerômetros acoplados em diversas regiões do corpo para detectar padrões como: caminhar, andar, trabalhar no computador, dirigir, dentre outras tarefas [5]. Diferentes grupos de pessoas apresentam padrões diferentes ao realizar uma mesma atividade [6]. Outro fator importante a ser avaliado é a localização dos sensores. Existem estudos que comparam o desempenho e a acurácia de reconhecimento de atividades considerando diferentes regiões de sensores [6,7]. Muitos trabalhos envolvendo classificação de atividades foram desenvolvidos recentemente[5-9]. Este trabalho propõe uma variação adaptativa do algoritmo LVQ com o objetivo de ser aplicado em classificação de dinâmicas de sinais e sistemas. Neste trabalho foi realizada uma comparação entre os algoritmos LVQ original de Kohonen [12], LVQ adaptativo (LVQa)[11] e o LVQ proposto pelos autores(LVQp), utilizando como estudo de caso a aplicação desses métodos em um problema de classificação de atividades humanas utilizando a base de dados [10].

As demais seções deste artigo estão organizadas conforme descrito a seguir: na seção 2 é apresentada a teoria de Learning Vector Quantization, e o algoritmo LVQ proposto. A seção 3 descreve a metodologia do trabalho, incluindo uma breve descrição da base de dados e os experimentos realizados. A seção 4 apresenta os resultados e comparações entre variações do LVQ utilizadas. A seção 5 encerra o artigo com as conclusões.

2 Learning Vector Quantization (LVQ)

O termo *Learning Vector Quantization (LVQ)* denomina uma classe de algoritmos de aprendizado supervisionado que realiza a tarefa de classificação de dados em espaços de alta dimensão. O LVQ é aplicado a problemas diversos como reconhecimento de padrões em imagens, sinais biomédicos, detecção de falhas, entre outras [11]. A classificação do LVQ é baseada em uma medida de distância, geralmente a distância euclidiana, que quantifica a semelhança de dados de entrada com os neurônios da rede ou protótipos. O LVQ é facilmente implementável e intuitivo, facilitando na adaptação do algoritmo a problemas específicos, como na classificação de dinâmicas em séries temporais.

2.1 LVQ adaptativo

Redes neurais LVQ adaptativas são estruturas capazes de ajustar sua arquitetura para melhorar o desempenho da rede durante o processo de treinamento. Em geral, as mudanças consistem na inclusão e remoção de protótipos. As estratégias de inclusão e/ou remoção de protótipos da rede LVQ adaptativa determinam seu desempenho de treinamento. O

algoritmo tradicional de aprendizado da rede LVQ sempre mantém a estrutura da rede enquanto as redes LVQ adaptativas podem sofrer alterações na quantidade de protótipos que possibilitem a melhoria em seu desempenho.

2.1.1 Proposto (LVQp)

O algoritmo a seguir (Algoritmo 1) descreve o pseudocódigo do treinamento da rede LVQp. O algoritmo LVQp, inicialmente, utiliza o método não-supervisionado Mapa de Kohonen [12] para determinar os protótipos iniciais da rede LVQ. Após a inicialização dos pesos, o algoritmo inicializa as iterações de treinamento do LVQ, podendo a rede sofrer mudanças em sua estrutura em cada iteração.

Algoritmo 1: Pseudo código para o LVQproposto

```

Input: MaxEpochs,  $\alpha_0$ , X, Y ; // Número de épocas, erro mínimo, taxa de aprendizado, entradas, saídas desejadas
Output:  $w_{opt}$  ; // Protótipos ótimo
1 while epoch < MaxEpochs do
2   for i = 1 to N do
3     getNearestPrototype( $w_j, c_j$ ); // Encontre o protótipo mais perto da amostra j (Protótipo vencedor)
4     /* Se a classe do neurônio vencedor é igual a da amostra i */
5     if  $c_j = y_i$  then
6        $w_j(t+1) = w_j(t) + \alpha(t) * (x_i - w_j(t))$ ; // Aproxima o protótipo da amostra
7        $A_j = A_j + 1$ ;
8     else
9        $w_j(t+1) = w_j(t) + \alpha(t) * (x_i - w_j(t))$ ; // Afasta o protótipo da amostra
10       $B_j = B_j + 1$ ;
11    end
12     $\alpha = \alpha_0 * (1 - (\frac{i}{N}))$ ; // Atualiza a taxa de aprendizado
13     $E_{it} = calculateError$ ; // Cálculo do erro
14    if  $E_{it} < E_{min}$  then
15      // Se o erro for mínimo, salva os protótipos
16       $w_{opt} = w$ ;
17       $E_{min} = E_{it}$ ;
18    end
19    LVQremove(); // Remove protótipos da rede
20    LVQadd(); // Adiciona protótipos da rede
21  end
22  shuffleData(X,Y); // Embaralha os dados para próxima iteração
23  it = it + 1;
24 end

```

2.1.1.1 Estratégia Proposta de Inclusão de Protótipos

O algoritmo LVQp se diferencia do LVQa principalmente na adição de protótipos. A adição de protótipos na rede LVQp (Algoritmo 2) é feita a partir do método Mapa de Kohonen [12] aplicado em amostras mal classificadas. A quantidade de protótipos adicionados depende das amostras classificadas erroneamente. Quanto maior for a presença de amostras mal classificadas, representada por U_i , para uma determinada classe i , maior a adição de novos protótipos nesta classe. A utilização do Mapa de Kohonen para determinar novos protótipos permite melhorar o desempenho da rede em regiões mal representadas. Porém, este método de *clustering* agrega um esforço computacional adicional ao algoritmo. A rede LVQp também utiliza uma variável de *Budget*, semelhante ao LVQa para limitar o crescimento da rede. A decisão da quantidade de protótipos que serão adicionados para uma classe i , representado por Q_i , é dada pela seguinte equação:

$$Q_i = size(U_i) * \gamma \quad (1)$$

Onde γ representa um fator de crescimento o qual determinará a proporção de protótipos que serão adicionados por classe. Os dados U_i são utilizados como vetores de entrada para execução de um algoritmo Mapa de Kohonen [12] para adição de novos protótipos.

Algoritmo 2: Pseudo código para o LVQadd

```

Input:  $U, N_{classes}, \gamma;$  // Amostras classificadas erroneamente, Quantidade de classes, taxa de crescimento
1 for  $i = 1$  to  $N_{classes}$  do
2    $Q_i = size(U_i) * \gamma;$ 
3    $p_{new} = Cluster(U_i, Q_i);$  // Retorna lista de novos protótipos a serem adicionados
4    $addNewPrototypes(p_{new});$  // Adiciona os novos protótipos
5 end

```

2.1.1.2 Estratégia Proposta para Remoção de Protótipos

A remoção de protótipos da rede (Algoritmo 3) é determinada por uma pontuação (Equação 2). A remoção de um protótipo será feita sempre que esta pontuação for menor que um limiar de remoção (ψ). Pontuações baixas indicam que os protótipos erram com frequência ou não contribuem significativamente com a classificação da atividade.

$$score_j^{LVQp} = A_j - B_j \quad (2)$$

Onde A_j conta quantas vezes um protótipo j classificado corretamente foi movido em direção a amostra e B_j quantas vezes foi movido para longe da amostra.

Algoritmo 3: Pseudo código para o LVQremove

```

Input:  $N_P, \psi;$  // Quantidade de protótipos, limiar de decrescimento da rede
1 for  $j = 1$  to  $N_P$  do
2    $p_j = A_j - B_j;$  // Pontuação para decisão de retirada de protótipos
3   if  $p_i < \psi$  then
4      $removePrototype(i);$  // Remove protótipo  $i$ 
5   end
6 end

```

3 Metodologia

3.1 Descrição da Base de Dados

Os dados utilizados foram coletados a partir do repositório *UCI Machine Learning Database* [10]. A base de dados representa dados coletados de um acelerômetro acoplado ao tronco de 15 participantes, praticando 7 atividades distintas. A tabela 1 apresenta as descrições das classes de atividades utilizadas neste trabalho.

Tabela 1: Classes de atividades

Classe	Descrição
1	Trabalhando no computador
2	Está em pé, caminhar e subir ou descer escada
3	Está em pé
4	Caminhando
5	Subindo ou descendo escada
6	Caminhando e conversando
7	Conversando em pé

3.2 Pré-processamento dos dados

Os dados de entrada da rede LVQ são as amostras das acelerações nos eixos x, y e z do acelerômetro. Para melhor representar os padrões de atividades a serem classificados, foram extraídas janelas de amostras. Cada janela é formada por um conjunto de amostras em uma faixa de tempo discreto. A figura 3 ilustra a representação de uma janela do sinal de aceleração. Nesta figura, cada janela é formada por 30 amostras de aceleração de cada eixo são consideradas. As janelas serão utilizadas como padrões de entradas da rede LVQ. Para isso, os dados das janelas são serializados em um único vetor conforme a figura 3.

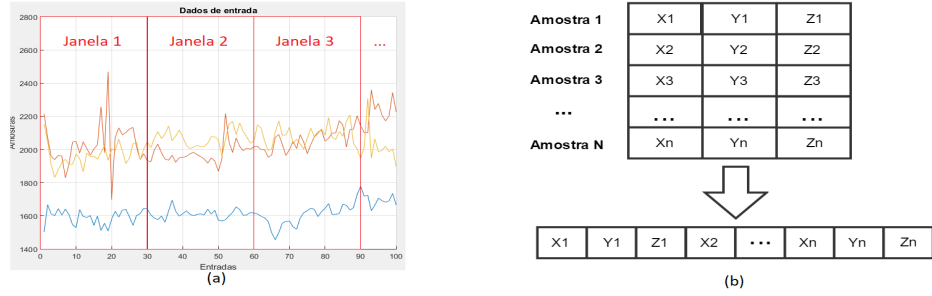


Figura 1: a) Extração de janela b) Serialização da janela.

4 Resultados

O experimento realizado tem como objetivo verificar a acurácia do algoritmo proposto LVQp e comparar com os resultados das variações LVQ e LVQa. O experimento considerou um tamanho da janela de 150 amostras, 150 protótipos e taxa de aprendizado inicial de 0.3. Foram utilizados 80% dos dados para treinamento e 20% dos dados para validação, aplicada em uma base de dados com quantidade de amostras iguais por classe. A tabela 2 apresenta os resultados de acurácia de validação para cada variação do algoritmo LVQ, considerando o treinamento aplicado a base de dados com 1, 5 e 15 pessoas.

Tabela 2: Desempenho de classificação entre variações do LVQ

C	Acurácia (%)								
	1 Pessoa			5 Pessoas			15 Pessoas		
	LVQ	LVQa	LVQp	LVQ	LVQa	LVQp	LVQ	LVQa	LVQp
1	100,00	100,00	100,00	97,37	97,62	98,03	89,47	94,33	95,65
2	88,16	99,34	96,05	78,95	94,08	96,71	78,29	79,61	85,52
3	70,39	78,29	79,63	71,71	78,95	76,97	50,00	80,26	83,55
4	75,66	71,71	74,05	82,89	73,68	68,42	61,18	66,45	69,84
5	75,66	78,29	86,55	84,87	80,26	85,72	79,58	82,87	84,21
6	48,03	48,03	56,58	17,76	33,55	49,34	40,79	39,47	42,76
7	75,00	80,26	87,63	76,97	79,61	81,02	61,57	69,08	82,89

A tabela 3 apresenta a matriz de confusão do treinamento da rede LVQp com os sinais dos 15 voluntários. Observa-se nas Tabelas 2 e 3 que as classes 4 e 6 são as que apresentaram acurácia mais baixas, e na matriz de confusão verifica-se ainda que o classificador confunde estas classes. Tais erros de classificação são justificados pela semelhança entre as atividades representadas por estas classes (ver Tabela 1): Classe 4 - Caminhando e Classe 6 - Caminhando e conversando.

Tabela 3: Matriz de confusão (LVQp) para 15 pessoas

C	1	2	3	4	5	6	7
1	0,9565	0,0238	0	0	0	0	0,0197
2	0	0,8552	0	0	0	0,1448	0
3	0	0,0066	0,8355	0,0066	0,0066	0,0263	0,1184
4	0	0,0197	0,0132	0,6984	0	0,2490	0,0197
5	0	0,0395	0,0461	0,0197	0,8421	0	0,0526
6	0,0066	0,0132	0,1513	0,2829	0,0658	0,4276	0,0526
7	0,0066	0,0132	0,1513	0	0	0	0,8289

5 Conclusão

Neste trabalho foi apresentada uma variação do algoritmo LVQ aplicado na classificação de dinâmicas em sinais de um acelerômetro para reconhecimento de atividades diárias. Considerando a acurácia, o LVQp apresentou-se superior às outras variações. A acurácia média de validação do algoritmo LVQp foi de 77.77% para treinamento com dados de 15 pessoas, 79.46% para 5 pessoas e 82.92% para 1 pessoa. A inclusão de múltiplos protótipos em regiões de baixa acertabilidade foi fundamental para o melhoramento da rede, porém é preciso analisar a proporção ótima para determinar a quantidade de protótipos a serem inclusos. A inclusão de novos protótipos com a presença de padrões semelhantes entre classes pode resultar em uma degradação da rede. Os resultados obtidos apresentaram que o algoritmo proposto (LVQp) é aplicável para classificação de dinâmicas e apresenta em geral melhor desempenho de classificação em comparação às variações LVQ e LVQa.

Agradecimentos

Agradecemos o apoio financeiro do projeto CNPq chamada universal 442573/2014-6 e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa concedida.

Referências

- [1] Smith, Aaron; EVANS, Michael; DOWNEY, Joseph. Modulation Classification of Satellite Communication Signals Using Cumulants and Neural Networks. 2017.

- [2] Taqi, Arwa M. et al. Classification and discrimination of focal and non-focal EEG signals based on deep neural network. In: Current Research in Computer Science and Information Technology (ICCIT), 2017 International Conference on. IEEE, 2017. p. 86-92.
- [3] Kumar, Shiu et al. A Deep Learning Approach for Motor Imagery EEG Signal Classification. In: Computer Science and Engineering (APWC on CSE), 2016 3rd Asia-Pacific World Congress on. IEEE, 2016. p. 34-39.
- [4] Naseer, Noman; NAZEER, Hammad. Classification of normal and abnormal ECG signals based on their PQRS intervals. In: Mechanical, System and Control Engineering (ICMSC), 2017 International Conference on. IEEE, 2017. p. 388-391.
- [5] Casale, P., Pujol, O., e Radeva, P. (2011, June). Human activity recognition from accelerometer data using a wearable device. In Iberian Conference on Pattern Recognition and Image Analysis (pp. 289-296). Springer Berlin Heidelberg.
- [6] Lester, Jonathan; CHOUDHURY, Tanzeem; BORRIELLO, Gaetano. A practical approach to recognizing physical activities. *Pervasive Computing*, p. 1-16, 2006
- [7] Atallah, Louis et al. Sensor positioning for activity recognition using wearable accelerometers. *IEEE transactions on biomedical circuits and systems*, v. 5, n. 4, p. 320-329, 2011.
- [8] Mannini, A., e Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2), 1154-1175.
- [9] Cleland, I., Kikhia, B., Nugent, C., Boytsov, A., Hallberg, J., Synnes, K., e Finlay, D. (2013). Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7), 9183-9200.
- [10] Lichman, M. (2013). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
- [11] Grbovic, Mihajlo; VUCETIC, Slobodan. Learning vector quantization with adaptive prototype addition and removal. In: Neural Networks, 2009. IJCNN 2009. International Joint Conference on. IEEE, 2009. p. 994-1001.
- [12] Kohonen, The Self-organizing Map, *Proceedings of the IEEE*, pp. 1464-1480, 1990.