Feature engineering methods for Brain Computer Interfaces using NNs

1 Description

Brain computer interfaces (BCIs) [1, 2] are used to translate electrical signals into commands without the need for motor intervention. They are particularly useful for implementing assistive technologies providing communication and control to people with severe muscular or neural handicaps [3]. More recently BCIs have also found application in other different domains such as gaming [4], virtual reality environments [5], and space applications [6].

BCIs require a decoding component in which brain signals are translated into commands. Usually, classification algorithms are applied to predict the human intention from the analysis of the signals. Several classification algorithms have been used to analyze brain data in the context of BCI applications [7]. They include linear discriminant classifiers (LDA) [8], support vector machines (SVMs) [9], neural networks (NNs) [10], and other classification methods [7].

A key ingredient of successful applications of classification methods to BCIs is feature engineering, i.e., ways to create, by transformations and combinations of the original features, new, more informative, features that can help the classifiers to produce accurate predictions.

2 Objectives

The goal of the project is apply feature engineering techniques to classify BCI data. A set of of datasets of an open BCI challenge will be available[1]. The student can select any of the 8 datasets for testing the feature engineering method.

The project should apply Neural Networks for: I) Find suitable feature representations for this problem that are very usable for other ML classifiers, OR, II) Implement NN-based classifiers for this problem, OR III) The combination of I and II (e.g., using an RBM to find the features and a Multi-layer Perceptron to classify the problem using the extracted features). In case II), the students are free to decide which feature representation is more appropriate for the data. In case I), they can use any classifier with the NN-based features.

The student should: 1) Preprocess the dataset as required. 2) Implement the feature engineering method. 3) Apply the selected classifier. 4) Validate the performance of the classifier. 4) Answer to the following questions in the report:

- What class of problems can be solved with the NN? (e.g., supervised vs unsupervised problems)
- What is the network architecture? (e.g., type and number of layers, parameters, connectivity, etc.).
- What is the rationale behind the conception of the NN?
- How is inference implemented? (e.g., How is the information extracted from the network?). Type of prediction or type of inference process.

[1]This dataset, which is describedin [11], can be downloaded from `http://www.bbci.de/competition/iii/#data_set_i`

- What are the learning methods used to learn the network ? Algorithms used for learning the network.

As in other projects, a report should describe the characteristics of the design, implementation, and results. A Jupyter notebook should include calls to the implemented function that illustrate the way it works.

3 Suggestions

- Read Lotte's paper on classification methods for BCIs `https://hal.inria.fr/inria-00134950/document`.
- See description of the BCI Challenge and datasets [11].
- Implementations can use any Python library.

References

[1] M.A. Lebedev and M.A.L. Nicolelis. Brain-machine interfaces: Past, present and future. *TRENDS in Neurosciences*, 29(9):536–546, 2006.

[2] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, and T.M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

[3] U. Hoffmann, J.M. Vesin, T. Ebrahimi, and K. Diserens. An efficient p300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, 167(1):115–125, 2008.

[4] A. Nijholt. BCI for games: A state of the art survey. pages 225–228. Springer, 2009.

[5] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. Lécuyer. OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, 19(1):35–53, 2010.

[6] L. Rossini, D. Izzo, and L. Summerer. Brain-machine interfaces for space applications. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 520–523, 2009.

[7] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4:R1–R13, 2007.

[8] D. Garrett, D.A. Peterson, C.W. Anderson, and M.H. Thaut. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):141–144, 2003.

[9] A. Rakotomamonjy and V. Guigue. BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller. *Biomedical Engineering, IEEE Transactions on*, 55(3):1147–1154, 2008.

[10] E. Haselsteiner and G. Pfurtscheller. Using time-dependent neural networks for EEG classification. *IEEE Transactions on Rehabilitation Engineering*, 8(4):457–463, 2000.

[11] B. Blankertz, K.R. Muller, D.J. Krusienski, G. Schalk, J.R. Wolpaw, A. Schlogl, G. Pfurtscheller, J.R. Millan, M. Schroder, and N. Birbaumer. The BCI competition III: Validating alternative approaches to actual BCI problems. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):153–159, 2006.