Classification of Alzhermer Disease and Normal Congnitive Status with Recurrent Neural Networks in Resting State fMRI

Tao Sun

Ohio University

2016

Classification of Alzhermer Disease and Normal Congnitive Status with Recurrent Neural Networks in Resting State fMRI

Introduction

A human brain is a complex system composed of structural regions that are functionally specialized. Due to the conclusion that these locally segregated regions are actively interconnected even when a subject is at resting-state (Biswal, Yetkin, Haughton, Hyde, 1995), the resting-state functional Magnetic Resonance Imaging (fMRI), which is a neuroimaging procedure that measures the changes of signals associated with blood flow, has become a prevailed tool for investigation of brain functional networks. Since functional connectivity in the brain is an significant measure that could indicates disease-induced changes in the network, it could provide assist to the diagnosis of brain diseases such as Alzheimer Disease (AD) or its early stage Mild Congnitive Impairment (MCI).

With the typical assumption that the functional networks in a brain is stationary, many diagnosis methods of MCI and AD with resting-state fMRI (rs-fMRI) model the network with correlation analysis such as Pearson's correlation, independent component analysis (Li et al., 2012). However, recent studies (**hutch**) suggest that significant temporal changes exist in functional connectivity. Thus, valuable information could be lost when connectivity estimation is based on analysis restricted to a single value obtained from the entire scanning time.

In this paper, we present a novel method to classify subjects with AD and Normal healthy Control (NC) by combining Deep Auto-Encoder (DAE) and Recurrent Neural Networks (CNN). Initially rs-fMRI images data is preprocessed and mean time series of Regions of Interest (ROIs) are extracted. Then high-dimensional time-series data is reduced to a lower dimensionality by the DAE and then splitted into multiple identical-sized sub-series. A RNN classifier is trained on the sub-series which can tag each sub-series as AD or NC. Finally, the diagnosis for a subject is made by ensemble of the outputs of the sub-series classifier. Tests shows that accuracy of the method approaches 70% on test data.

Problem Definition and Algorithm

Task Definition

Data set and Preprocessing. The data used for training and test of the proposed classifier are retrieved from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. After filtering, images of 33 AD subjects and 50 NC subjects, are downloaded. With most of these subjects are scanned more than once, we have 89 AD examples and 139 NC examples in the data set.The data set are divided into training data, test data, and validation data with a ratio of 7:2:1.

The preprocess can be divided into 4 phases ((Suk, Lee, & Shen, 2015), (**chen**)) :

1. Preprocessing of anatomical images
2. Preprocessing of functional images
3. Anatomical standardization of functional images
4. Removal of noise signa

The results of the preproces are a set of mean time series

$$F^{(n)} \in \{F | F = [\mathbf{f}_1, \cdots, \mathbf{f}_t, \cdots, \mathbf{f}_T], \mathbf{f}_t \in \mathbb{R}^R\}, n = 1, \cdots, N \ ,$$

where $N = 228$ is number of the scans, $R = 120$ is the number of ROIs, and $T = 135$ is the length of a time series.

Dimentionality Reduction. Suk, Lee, & Shen (2015) proposed a DAE can used as an intermediate building block for deeper models in neuroimaging analysis. DAE is an unsupervised neural networks, the goal of which is setting a latent representation of feature vectors of a scan

$$G = [\mathbf{g}_1, \cdots, \mathbf{g}_t, \cdots, \mathbf{g}_T], \mathbf{g}_t \in \mathbb{R}^R$$

from its original form

$$F = [\mathbf{f}_1, \cdots, \mathbf{f}_t, \cdots, \mathbf{f}_T], \mathbf{f}_t \in \mathbb{R}^R$$

by training a nonlinear approximation function $h(\mathbf{f}_t) \approx \mathbf{f}_t$. After training, only the first part of the DAE is used to transform each $\mathbf{f}_t \in \mathbb{R}^R$ into $\mathbf{x}_t \in \mathbb{R}^r$, where $r < R$. As a result, the encoded representation of a scan becomes

$$X = [\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T]$$

to fed into the classification model to come.

High-level RNN classifier and ensemble for classification. In the classification model, a encoded time-series is splitted into identical-sized sub-series

$$X = [\mathbf{x}_1, \cdots, \mathbf{x}_s, \cdots, \mathbf{x}_S], \text{ where } T = n * S, n \text{ is an integer.}$$

Suk et al. (2015) Wee, Yang,

Study 1

This study is designed as a pilot study and serves to identify the appropriate difficulty for the Petri net material. The amount of places and transitions and their interconnection should make the task feasible, meaning the recall rate should be far above a randomly drawn new Petri net, but still each recalled drawing should contain some error. Successful analyses have been conducted with accuracy measure from 58% correct recall (**egan1979chunking**) up to 95% (**moss2006role**) which is hence taken as the desired lower and upper boundary respectively.

Method

Participants. Three first-year students, three third-year students who have taken at least one Petri net class and three scientific employees who work with Petri net related material on a daily basis shall participate in this study. The students' participation in experiments is mandatory and part of their curriculum, the scientific assistants get cookies as a reward.

Materials. Petri nets are randomly generated (cf. Appendix B) and visually formatted (cf. Appendix C). A range of 5 and 30 graph elements (n) is used with a number of interconnections (f) between $f = n - 1$ and $f = round(3/2 * n)$. Grid lines are drawn onto the background so that each place or transition is on an intersection of one of the vertical and horizontal lines.

Procedure. In the beginning the participant is introduced in using **renew** (**renew**) as a tool for drawing Petri nets. Further participant is informed that any element which is not on one of the intersections between the vertical and horizontal lines is removed during the scoring process. Each participant does 13 runs from which the first run is discarded as a demo run. In that way each participant sees 13 different Petri nets. The participant sits in front of the screen and is presented one of the generated Petri nets for 5 seconds. After that immediately the recall phase starts. Using **renew** the participant is asked to reconstruct the previously seen net. On the background the same grid lines are presented and the participant is asked to position the places and transitions on the grid lines as the participant has seen it before. After the participant believes that everything which could be remembered has been drawn, the file is saved, printed and closed. The participant is then asked to indicate the used strategy on the print-out and circle which symbols were grouped together.

Scoring

The scoring is inspired on a previous study on mechanical engineers (**moss2006role**) which used an extended scoring system of **chase1973mind** The scoring of each recall is done in the following manner:

1. Each place and transition which is not drawn on the grid intersections is deleted (including the arcs which connected the deleted element with other elements).

2. Each place and transition which has been omitted on the grid is added to the element omission score.

3. Each place and transition which appears on a wrong position on the grid is added to the element insertion error score.

4. Each arc which connects two previously unconnected elements is added to the arc wrong connection score.

5. Each arc which is omitted is added to the arc omission score.

6. Each arc which points into the wrong direction is added to the arc wrong direction score.

The total weighted error score for a drawing is the weighted sum of all the errors mentioned above. Each error score is weighted with 1 except the arc wrong direction score which is weighted with 0.5.

Data Analysis

For each participant and each Petri net first the error score is calculated. Then correct recall score is calculated as a sum of all correctly positioned elements. Then the ratio of the correct recall score to the sum of both the total weighted error score and the correct recall score is calculated. If this ratio is in the range of 58 to 95%, it is selected as appropriate material. Otherwise the Petri net is discarded. Further the Petri nets are ranked according to their ratio and the rank list is partitioned into three proportions of similar sizes. The partitions are labeled with their respective difficulty level, viz. easy, medium and difficult.

Study 2

In this study the prepared material of the last study is used in a similar setup. As a new component a distractor task is added to deteriorate the visual-spatial sketchpad (cf. **baddeley1986working**). Further a second measure, the inter-response time analysis, is taken, to gain more insight into the nature of chunks.

Method

Participants. 10 first-year students, 10 third-year students who have taken at least one Petri net class and 10 scientific employees who work with Petri net related material on a daily basis shall participate in this study. As number of professors and teaching assistants of the University of Hamburg might not be sufficient, assistance from other universities which are also involved in Petri net research might be asked to help. The students' participation in experiments is mandatory and part of their curriculum, the professionals get some cookies as a reward. Having participated in study 1 excludes the participants from study 2.

Materials.

Petri nets. The Petri nets which have been approved in study 1.

Distractor task. An unsolvable 15-puzzle (cf. **ratner1986finding**) which runs as a program on the same computer

Procedure. In the beginning the participant is introduced in using **renew** (**renew**) as a tool for drawing Petri nets. Further the participant is informed that any element which is not on one of the intersections between the vertical and horizontal lines is removed during the scoring process. Each participant does 13 runs from which the first run is discarded as a demo run. The pool of Petri nets for one participant contains 6 drawings connected with the distractor – no distractor condition. In the no-distractor condition the net will be shown immediately whereas in the distractor condition between the presentation and the recall a 30s delay happens. The six drawings shown in both conditions equal 12 runs.

For each run the Petri net and its connected condition is drawn randomly from the pool. The participant sits in front of the screen and is presented the drawn Petri net for 5 seconds. Depending on the drawn condition either the distractor task is shown or the recall starts immediately. Using **renew** the participant is asked to reconstruct the previously seen net. Mouse clicks and the screen are recorded throughout the process. On the background the same grid lines are presented and the participant is asked to position the places and transitions on the grid lines as the participant has seen it before. After the participant believes that everything which could be remembered has been drawn, the file is saved, printed and closed. The participant is then asked to indicate the used strategy on the print-out and circle which symbols were grouped together.

Data Analysis

Performance Analysis. First the error scores are calculated as described in *scoring* of study 1. A mixed-design analysis of variance (ANOVA) with the Petri net difficulty level (simple, medium, difficult), the distractor task and the total weighted error score as a within-subjects factor and expertise level as the between-subjects factor is conducted. As a post-hoc test Tukey's HSD is chosen. Significant differences are expected between the three groups and between the three Petri net difficulty levels. First-year students are expected to show significant differences between the distractor task and the no distractor task condition while scientific employees are expected not to have a significant difference.

Inter-Response Time Analysis. The mouse clicks and screen recordings are analyzed using a single-linkage hierarchical clustering algorithm (cf. **moss2006role**). The found chunks are supposed to correspond to the units indicated by the respective participant. Since each drawing was both presented in the distractor and no distractor condition, for the same drawing for each participant two hierarchical clusters exist. They are compared using the correlation between the cophenetics of each hierarchical cluster (cf. **fowlkes1983method**). The two cophenetics are expected to strongly correlate for scientific employees since their are expected to be unaffected by the distractor condition whereas a rather low correlation between the two cophenetics is expected for first-year students. In the no distractor condition first-year students are expected to have rather complex (probably both nested and overlapping) chunks in working memory which help to recall the drawing with comparably little error. In the distractor condition first-year students need to rely on incidental encoding to their long-term working memory. Maybe the presented drawing can utilize existing knowledge about superficially similar visual languages like flow chart diagrams or other, probably highly personal, mnemonic strategies. Without making elaborated assumptions about the used strategies, they are supposed to result in a different cluster than in the immediate recall condition.

Implementation of this Project

The experiment tried to target at many yet not well understood spots and it is novel in several aspects. First, the expertise studies mostly focused on professions for which memorizing material added significant value for the professionals' daily life(cf.

oulasvirta2006surviving). Whether this is the case for Petri nets has yet been unexplored. This study helps to identify for which kinds of tasks long-term working memory can be helpful. The special nature of Petri nets is that their concept is quite abstract since they only describe how different states are interrelated, which is far from the previous studies which mostly focused on work in which the configuration resembled a specific meaning, e.g. winning or loosing a game or delivering the right dish to the right table. Second, this study looks not only at whether the chunks of lays show a different quality but it also examines in how chunks are deteriorated. This gives some valuable insight in how information declines – whether mainly the interrelations between the low-level chunks suffer (cf. **moss2006role**) or whether more, new patterns of recall errors can be observed.

References

Li, Y., Wang, Y., Wu, G., Shi, F., Zhou, L., Lin, W., & Shen, D. (2012). Reasoning with exceptions: an event-related brain potentials study. *Neurobiology of aging*, *33*(2), 427.e15–427.e30.

Suk, H.-I., Lee, S.-W., & Shen, D. (2015, October). A hybrid of deep network and hidden markov model for mci identification with resting-state fmri. In *International conference on medical image computing and computer-assisted intervention* (pp. 573–580). Springer International Publishing.

Appendix A

Visual vocabulary of an Elementary System Net

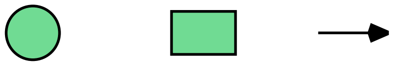

Figure A1. A circle depicts a place, a rectangle depicts a transition and an arrow depicts and arc. The arrow connects places with transitions and can be bent for that purpose. The green color is the standard color of the

Appendix B

Random Petri Net Generation Algorithm

```
Input
  n: number of graph elements
  f: number of interconnections
Output
  P: set of places
  T: set of transitions
  F: set of f el P x T U T x P

Algorithm
  tmp := gaussian(location=n)
  while tmp < 0:
    tmp := gaussian(location=n)
  sizeP := n - tmp
  sizeT := n - sizeP
  P := {p_1, p_2,..., p_sizeP}
  T := {t_1, t_2,..., t_sizeT}

  P_2 = {}
  T_2 = {}
  p_last := random element from P
  t_last := random element from T
  while f > 0:
    p_Pool := P \ P_2 if (P \ P_2) != {} else P
    t_Pool := T \ T_2 if (T \ T_2) != {} else T
    p := random(p_alt, random element from p_Pool)
    t := random(t_alt, random element from t_Pool)
    P_2 := P_2 U p
```

```
T_2 := T_2 U t
if random(TRUE, FALSE):
  F := F U (p, t)
  p_last := random element from P_2
  t_last := t
else:
  F := F U (t, p)
  p_last := p
  t_last := random element from T_2
f--
if places or transitions are not yet connected:
  discard Petri net
```

Appendix C

Layout Algorithm

```
Input
  P: set of places
  T: set of transitions
  F: set of f el P x T U T x P
Output
  Layout


Algorithm
  Draw grid
  Search for whether there is p el P with no incoming arc
  If yes: P_2 := all p el P with no incoming arc
  Else: P_2 := {random p el P}
  i := 0
  while F_2 != F:
    draw all p el P_2 on the ith grid line vertically arranged if not yet present
    draw all t for which (p, t) el F on the (i+1)th grid line if not yet present
    draw the arcs between p and t for all drawn t if not yet present
    P_2 := {p | (t, p) el F for all drawn t}
    F_2 := F_2 U (t, p) U (t, p) for t, p part of the drawing
    i++
```