Robust Medical Imaging and Domain Adaptation

Tianqin Li (tianqinl) [1] **Geyang Zhang (geyangz)** [1]

1. Introduction

Machine learning methods in medical diagnosis have grown increasingly helpful in quantification of imaging biomarkers, computer aided interpretation and computer aided diagnosis, etc. Unfortunately, training data is difficult to obtain especially when the diagnosis goal requires repeated visit. Additionally, imaging data are usually within high dimensions, further increases the difficulty to fit a generalized model. For both reasons, overfitting has been a common issue that faced by medical imaging prediction models. In this paper, we focus on the task to determine the conversion from mild cognitive impairment(MCI) to Alzheimer's disease (AD), using magnetic resonance imaging (MRI) data.

To increase robustness, a popular strategy is to apply adversarial training as it forces the model to utilize features mostly related to semantic label rather than features generated by bias. Label shuffling has been shown to cause convolution neural network (CNN) remember the data itself, rather than it's semantic representations (Wang et al.), thus might be a way to use as adversarial data. Furthermore, the adversarial training idea can be used in domain adaptation, where the variants between source domain and target domain were subtracted during training (Ganin et al., 2016). So can be apply to transfer learning task from AD-cognitive normal(CN) to MCI conversion. The former, has more separable features thus is easier to classify.

On extension to these ideas, in this project, we propose two methods to increase the robustness on classification on MCI conversion task: Firstly, we try to adversarially train MRI data with a label shuffled supplementary learner; Secondly, we try to apply the adversarial training idea to MCI related tasks, such as AD-MCI or AD-CN. Our ultimate goal is to find those most meaningful representations in this disease progress.

2. Related work

MCI is considered as the prodromal state of AD. Each year, around 10 15 % people with MCI are identified to have AD (RC, 2004) thus this period has a significant mean for AD diagnosis and pre-treatment. Several studies were conducted on the task to distinguish MCIs that progressed to AD (pMCI) or stay in MCI state (sMCI)using MRI data. These studies generally reported an accuracy around 80 %, however, most of them have been reported to have data leakage or used imbalanced dataset thus lead to an overestimated performance (Ganin et al.). The flaw in handling data is mostly resulting from the lack of dataset available for this task. To be used for classification, the sample is usually required to have follow up scans in at least 3 years thus making the general size of dataset being around a few hundreds participants with around a thousands sessions for each category.

To solve this issue, some preprocessing techniques has been used to enlarge the dataset. As MRI data is in 3d-format, some paper choose to slice it into 2-d images or 3d-patches and use them as individual inputs. However, the correlations within each individuals' MRI results can lead to a biased output; also, the cancel of relations between slices or patches were shown to hurt the performance. As a results, the classification between pMCI and sMCI was most accurate taking each individual as one subject, using the whole results or regions with interests (ROI). Both methods have the number of inputs remains the same as sample size (Ganin et al.).

Alternatively, one may choose to generate artificial samples. This includes the classical augmentation including translation, rotation, flipping, etc [3]. It has also been shown that Deep Convolutional GAN(DCGAN) were able to generate qualified data on liver lesion CT dataset, with only a few hundreds true image given (Frid-Adar et al., 2018). Specifically, they used this method in ROI of liver lesion CT and the generated images were proved to improve the classification.

Another path to increase robustness is to use invariant features among differently distributed dataset. Given the small size of MCI data, it is highly possible that the distribution of training dataset are biased. To find such feature, (Ganin et al., 2016) proposed to adversarially learn data label and

[1] Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

domain label, either in supervised or unsupervised manner. By subtracting the gradient learned from distinguishing domains, their classifier were able to classify target domain with improvement over 40% compared to trained with source domain only.

Also, if we loosen the requirement that labels are generated from the same distribution, domain shift can be performed on even more far related dominoes using transfer learning. (Cheng et al., 2012) utilized the fact that AD and cognition normal(CN) data have more separable features, and learned a cross-domain kernel to co-train with pMCI-sMCI data. They found this auxiliary domain helped the classification of target task on not only with MRI data, but multiple measures.

Labeling shuffling may be another simple but effective way to exclude unrelated features. When data are with random label, CNN can still fit the data but tends to remember the data rather than learning label related features. I was further suggested that the features learned with label shuffled dataset were less generalizable, due to the abandon of feature meaning and label meaning(Wang et al.). Thus, this might help us to select features that are mostly related to labels.

3. Methods

In this project, we aim at increasing the robustness of medical imaging classification with adversarial training. Our model will mainly have 3 parts that extends the structure of DANN: a feature extractor, a label classifier, and a supplementary learner. Here we propose two methods:

1. using the supplementary learner to learn the label shuffled training set and subtract the gradient at feature extractor at back propagation. This is based on the assumption that the learner would learn label unrelated features with label shuffled data, thus to force the feature extractor weigh on meaningful features;

2. perform transfer learning on this network, where using the label classifier to learn the classification of source domain, and the supplementary learner learn the domain difference between source domain and target domain, also with domain label shuffled.

3.1. Baseline model

Our baseline model structure is identical to DANN (Ganin et al., 2016), Figure 1. except that we use supervised domain learning. For comparison to class label shuffling method, we simply abandon the supplementary learner, where the remaining is a simple CNN; for transfer learning task, we use DANN as the baseline model. Let $G_f(\cdot;\theta_f)$, $G_y(\cdot;\theta_y)$ and $G_d(\cdot;\theta_d)$ denote the feature extractor, label predictor and supplementary learner respectively, then the prediction for sample x^i class label $\hat{y^i}$ and domain label $\hat{d^i}$ is:

$$\hat{y^i} = G_y(G_f(x^i;\theta_f);\theta_y)$$

$$\hat{d^i} = G_d(G_f(x^i;\theta_f);\theta_d)$$

For both predictor, we use cross entropy loss at the end, and update the parameters as:

$$\theta_f \leftarrow \theta_f - \mu(\frac{\delta L_y^i}{\delta\theta_f} - \lambda\frac{\delta L_d^i}{\delta\theta_f})$$

$$\theta_y \to \theta_y - \mu\frac{\delta L_y^i}{\delta\theta_y}$$

$$\theta_d \to \theta_d - \mu\frac{\delta L_d^i}{\delta\theta_d}$$

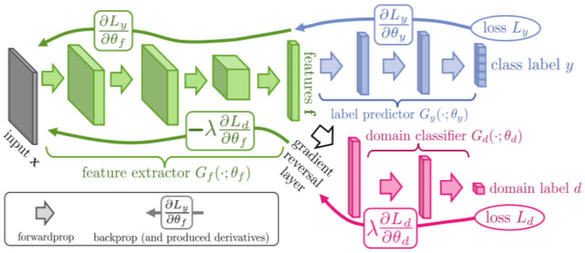
$\frac{\partial L_y}{\partial\theta_f}$
$\frac{\partial L_y}{\partial\theta_y}$
loss L_y
class label y
features f
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial\theta_f}$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
input x
feature extractor $G_f(\cdot;\theta_f)$
loss L_d
$\lambda\frac{\partial L_d}{\partial\theta_d}$
forwardprop
$\frac{\partial L_y}{\partial\theta_f}$
backprop (and produced derivatives)

Figure 1. DANN network structure from (Ganin et al., 2016).

3.2. Class label shuffling

In this part, for our input data X and corresponding label Y, we both perform class prediction at label predictor G_y and supplementary learner G_s, however, data labels are randomly shuffled at G_s as $Y_{shuffle}$. The parameter update is then

$$\theta_f \leftarrow \theta_f - \mu(\frac{\delta L_y^i}{\delta\theta_f} - \lambda\frac{\delta {L_y}^i_{shuffle}}{\delta\theta_f})$$

$$\theta_y \to \theta_y - \mu\frac{\delta L_y^i}{\delta\theta_y}$$

$$\theta_s \to \theta_s - \mu\frac{\delta {L_y}^i_{shuffle}}{\delta\theta_{yshuffle}}$$

Currently, we still directly subtract the gradient of supplementary learner. However, as doing the same classification task, even though the supplementary learner were expected to learn the label unrelated features, it might hurt the label predictor learning at the beginning. Thus, whether λ should be changing over time, or the subtraction should be down once per several epoch, is subjected to change.

3.3. Domain Label shuffling

As the original goal for DANN is to perform domain adaptation, we would like to extends this idea to transfer learning, where the source and target domain does not necessarily generate from the same distribution. Even though learning the domain class as proposed in DANN may still work, we would like to know whether domain label shuffling could also help in this circumstance. Here, the supplementary learner is still a domain label classifier as in DANN, however, the domain labels here are shuffled as $D_{shuffle}$. As such shuffle is expecting the learner to learn unrelated features to domain label, we will add the gradient of supplementary learner to feature extractor here. The updates are:

$$\theta_f \leftarrow \theta_f - \mu(\frac{\delta L_y^i}{\delta\theta_f} + \lambda\frac{\delta L_{d_{shuffle}}^i}{\delta\theta_f})$$

$$\theta_y \to \theta_y - \mu\frac{\delta L_y^i}{\delta\theta_y}$$

$$\theta_d \to \theta_d - \mu\frac{\delta L_{d_{shuffle}}^i}{\delta\theta_{dshuffle}}$$

4. Experiments

4.1. Unsupervised domain adaptation using mnist and svhn dataset

First, we want to test our idea of shuffling to see the performance of our label shuffling idea. Therefore, experiments are conducted in the following ways. Our goal is to using the MNIST(LeCun et al., 1998) dataset to train a classifier and then using as the SVHN(Netzer et al., 2011) dataset as target dataset to perform unsupervised domain adaptation, i.e. given no label of the target domain, we wish to train the classifier to generalize well on the target dataset using source dataset. Supervision comes from the source dataset labels but the vanilla feature learnt by Convolutional Neural Network (CNN) can easily capture the parts that is unessential to the semantic of the data although they can somehow contribute to the final classification task. In order to learn a more robust features, we tried using the shuffled method proposed above on the MNIST and SVHN dataset when training and test the domain adaptation results on the testing dataset (SVHN).

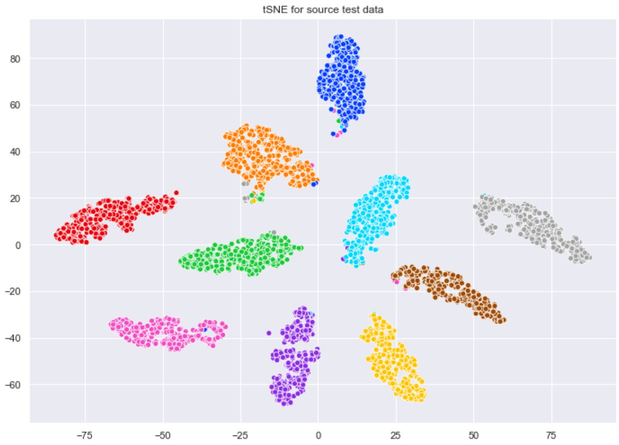

Figure 2. tSNE plot for source domain after transformed by encoder.

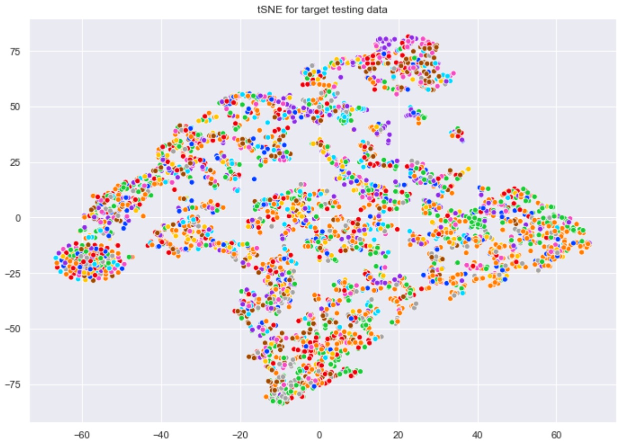

Figure 3. tSNE plot for target domain (baseline).

First we perform the baselines of our method where we train the our CNN encoder on the source data to extract the useful abstract features for classification and also a Multilayer Perceptron is used to perform the final classification task. In the baseline method, the MNIST dataset is our training set while we use SVHN dataset to test our classification ability. The baseline method has the target testting accuracy as 20% on svhn data while the test accuracy on the source domain (MNIST dataset) is nearly 97%. This shows that using the MNIST dataset which is a less noisier version of digits images to train a CNN can generalize pretty badly on the new dataset (SVHN) which consists the digits photo taken on the street. Figure 2 shows the tSNE plot of the source domain testing data. We can clearly see that the encoder CNN learnt the representation that makes the classifier easily to perform classification task. However, Figure 2 shows the tSNE plot of the encoder output when feeding into target domain data. The results is very disappointed comparing to the source domain suggesting that the encoder is not doing a good job at generalization.

Next experiments is to perform the Domain-Adaverisal Nerual Network (DANN) method to see whether they can force the network to learn the true feature that is useful to the underlying semantic rather some combination of features that is highly sensitive to the domain noise. The DANN method used domain adversial training to force the

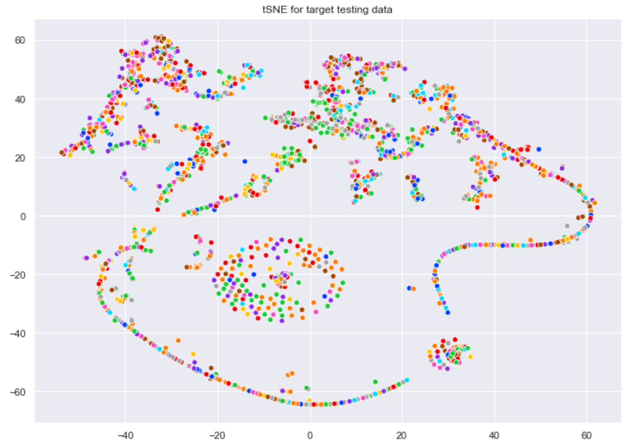

Figure 4. tSNE on DANN results (target)

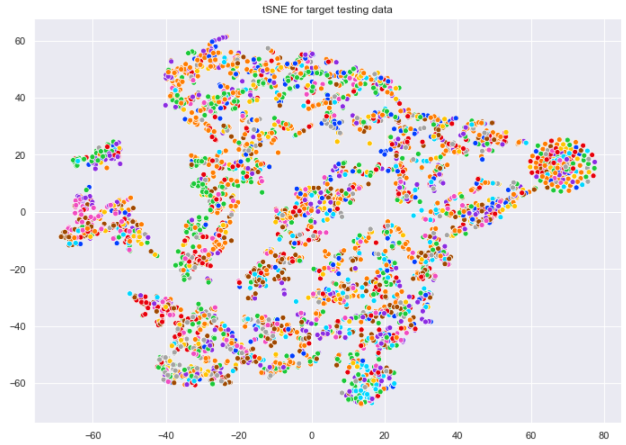

Figure 5. tSNE on domain label shuffling results (target)

network to obtain the features that is invariant to the domains. As is shown in Figure 1, they utilize the gradient reverse layer to perform the learning. Although they managed to transfer from the SVHN domain into MNIST domain with good accurracy, we perform the DANN method but got only to 22% improvement in our primary implementation. It may requires more parameter tuning efforts to improve the performance further but we don't see a hugh improvement comparing to our baseline. To further reason behind the accuracy, we plot the tSNE plot of the DANN as shown in Figure 4. Comparing to Figure 3, we do see that the data is seperated more but the labeling is not consistent within each cluster. It may because the DANN directly forcing the neural network to learn the invariant feature has some negative effects on the classification task which leads to our label shuffling method.

Then we tried our proposed labeling shuffling method as a soft version of DANN method and experiments are performed where the training dataset is the MNIST and the testing dataset is the SVHN. During the training, we utilize the domain label shuffling method where except for the normal encoder and classifier, we add a domain shuffling classifier which classify incoming domain of the each image. However, in order to confuse the encoder to learn the domain invariant representations, we shuffled the domain labels when feed into the domain shuffling classifier. The testing accuracy is 25% which is slightly higher than the DANN results. We plot the tSNE of the testing dataset as shown in Figure 5. As we can compare to the Figure 4, the domain label shuffling method shows less clear clusters as it did in original DANN results. However, it shows that using shuffling method, the layer tSNE plot is a lot more like the baseline method but in the accuracy, it's slightly better than the orginal DANN. This demonstrates that the shuffling method is slightly modifying the learnt features but it's modifying the useful one. Comparing to the DANN, it more close to the original tSNE structure which also makes the classifier easier to learn the right classification rules which may be the reason that the accuracy is better than the DANN results.

Exploring class label shuffling idea

After performing the domain label shuffling method, we also explored the idea of shuffling the class label during the training to supress the noise feature inside the training dataset. This procedure begin with normal training processs by feeding into a CNN encoder and also a classifier supervised by the class label. Then the we randomize the class label of the training set and feed it into the network in a slighter manner, i.e. we train the neural network by the normal cross entropy loss but with a weight equals to 0.01. By training the neural network in a forward manner with some random example inside the network, we wish the network to learn how to classify the correct label through these features that is not too closely tight to the class label. In other words, we are using this method to avoid the overfit of the neural network. Suppose the neural network will rely on the features that is highly related to the noise in the dataset, when shuffling the labels, it will reduce the amount of weight that contributes to generating the noise related feature.

However, using this method, we still didn't get too much robust features aross the domains as we test on the target domain with 22% accuracy. Nevertheless, by doing the class label shuffling, we generate the features that makes the data less related to each other within each label group. This can be shown in Figure 7 and Figure 8. Comparing Figure 2 and Figure 8, we can observe that each cluster is streched when using class label shuffling method.

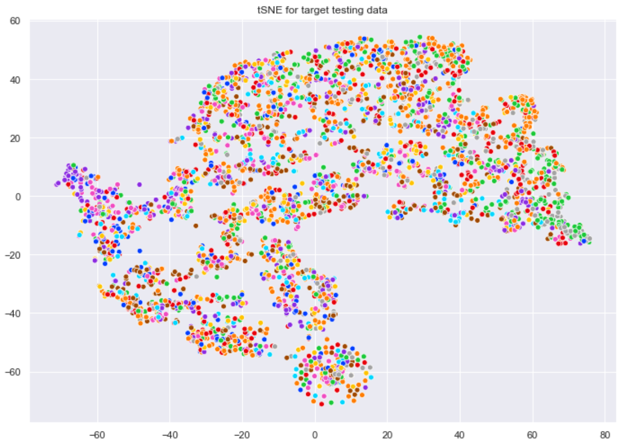

Figure 6. tSNE on class label shuffling results (target)

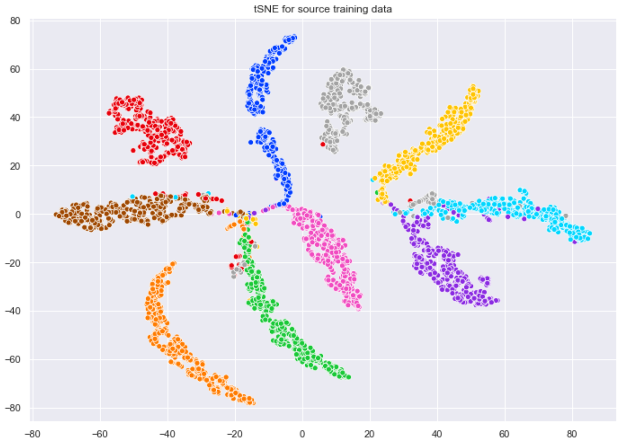

Figure 7. tSNE on class label shuffling results (source train)

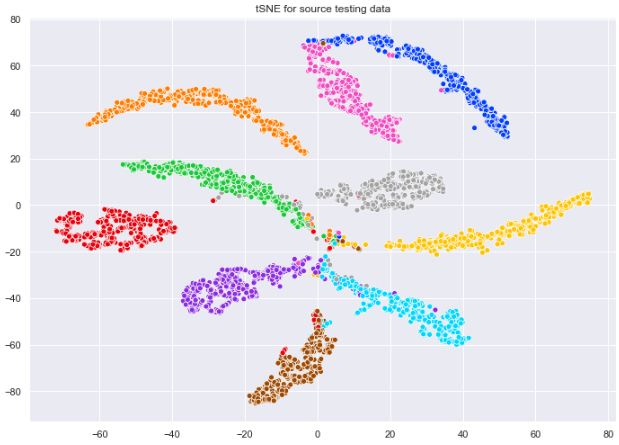

Figure 8. tSNE on class label shuffling results (source test)

5. Future Direction

5.1. Domain adaptation

To achieve better domain generalization task, we can have more exploration based on our current tested methods.

Although the domain label shuffling method improves our performance a little, it may only learn the domain invariant features. However, we argue that the domain invariant feature doesn't necessarily means the features that is good for classification. The noise that coexists in two domains is also not worth to consider and it would not have the power to increase the prediction accuracy of the classifier. Therefore, we need to find a way to constrain the domain classifier such that it would not encourage the noise enter the learnt features.

Beside, we are thinking towards causal inference for constraining the domain invariant classifier as it is believed that human has the ability to generalize across different situation because we have a causal understanding of the context and the subject such that we can easily identify the causal features of an objects to make the classification task. Therefore, maybe trying to find a way to constrain the features such that they resambles some of the causal cues of the classification task would increase the domain adaptation task better.

5.2. Applying on medical image dataset

This project is aimed to improve the classification ability of seperating the sMCI and pMCI from fMRI dataset using the knowledge learnt in classifying AD/non-AD fMRI dataset. Therefore, although doing the data experiments above, we wish to apply our method into the AD/non-AD data and transfer the knowledge from AD/non-AD dataset into the sMCI/pMCI classification method. Therefore, unsupervised domain adaptation would be helpful for transferring knowledge between the two domains.

5.3. Data augmentation

As is noticed in the method, both DANN and the domain label shuffling method requires huge amount of data. However, the data in the ADNI data only has around thousands of fMRI image for each domain where the MNIST data set has 60,000. Therefore, data augmentation should be performed such that the data is sufficient for the network to learn useful features and generalize well on the target domain.

References

Cheng, B., Zhang, D., and Shen2, D. Domain transfer learning for mci conversion prediction. pp. 15(0 1): 82–

–90., 2012.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Convolutional neural networks for classification of alzheimer's disease: Overview and reproducible evaluation. *Domain-Adversarial Training of Neural Networks*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Some studies in machine learning using the game of checkers. *Domain-Adversarial Training of Neural Networks*, 17:1–35, 2016.

LeCun, Y., Bottou, L.and Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. pp. 2278—2324, 1998.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

RC, P. *Mild cognitive impairment: aging to Alzheimer's Disease*. Oxford University Press, 2004.

Wang, H., Wu, X., Huang, Z., and Xing, E. P. High frequency component helps explain the generalization of convolutional neural networks.