Representation Learning for Cross-Modality Classification

Gijs van Tulder[1] and Marleen de Bruijne[1,2]

[1] Biomedical Imaging Group Rotterdam
Erasmus MC University Medical Center, the Netherlands
[2] Image Group, Department of Computer Science
University of Copenhagen, Denmark

Abstract. Differences in scanning parameters or modalities can complicate image analysis based on supervised classification. This paper presents two representation learning approaches, based on autoencoders, that address this problem by learning representations that are similar across domains. Both approaches use, next to the data representation objective, a similarity objective to minimise the difference between representations of corresponding patches from each domain. We evaluated the methods in transfer learning experiments on multi-modal brain MRI data and on synthetic data. After transforming training and test data from different modalities to the common representations learned by our methods, we trained classifiers for each of pair of modalities. We found that adding the similarity term to the standard objective can produce representations that are more similar and can give a higher accuracy in these cross-modality classification experiments.

Keywords: Representation learning, Transfer learning, Autoencoders, Deep learning, Multi-modal image analysis

1 Introduction

Most classification techniques assume that they will be applied to data that comes from the same domain as the training data. In practice it may be necessary to use training data from a different domain. In medical image analysis, for example, it may happen that annotated training data is available but comes from a different scanner, or was made with different scanning protocols or different imaging modalities. Transfer learning methods handle these differences by transferring the knowledge learned in one domain and applying it to data from another domain. Some approaches do this by transforming the feature spaces, while others use instance weighting to give larger weights to training samples that look more similar to the target data (see [1] for a recent overview).

This paper proposes two representation learning approaches for transfer learning and applies those to a medical imaging problem. Representation learning [2] methods learn efficient, data-driven representations of the training data and have shown good results in same-domain applications. We use these techniques

H. Müller et al. (Eds.): MCV/BAMBI 2016, LNCS 10081, pp. 126–136, 2017.
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-61188-4_12

to learn cross-domain representations that are not just efficient descriptions of the data, but are also similar across domains.

There is an obvious trade-off between learning a representation that provides efficient descriptions of data from one domain and learning a representation that is similar between domains. We discuss a hybrid learning objective that combines a standard representation learning objective, which tries to learn an efficient representation, with a similarity objective that minimises cross-domain differences. We use a weighted combination to find an optimal trade-off.

We suggest two models: a set of domain-specific autoencoders and an axial neural network. Both approaches learn a common representation, with a separate transformation for each domain. With autoencoders, this is achieved by training a separate autoencoder for each domain, whereas the axial neural network uses a single network that combines inputs from all domains. For both models, we include a similarity term in the learning objective to minimise the representation difference between corresponding samples from each domain.

Previous work on the combination of representation learning and transfer learning in medical image analysis can be separated in several groups. A popular approach is to transfer feature descriptors. These approaches reuse features that were learned from images from another domain, such as natural images or a different medical image dataset, and apply those to the target data with data-specific fine-tuning (e.g., [3,4]), but do not generally train cross-domain classifiers. Another group of approaches does train on data from different domains, but does so with a single feature transformation for all domains. Siamese networks [5], neural networks that are trained on data from different domains in parallel, fall in this category. These models are somewhat similar to the networks discussed in this paper, since both types of models are trained on paired samples. However, our methods learn a different transformation for each domain, which may work better if the domains are dissimilar.

We performed our experiments on data from the BRATS tumor segmentation challenge [6]. This multi-modal dataset contains brain scans made with four MRI sequences and manual annotations. In addition, we present experiments on a synthetic dataset derived from the BRATS images. Using the representations learned by our methods as the features, we measured the classification performance of random forest classifiers trained on data from one sequence and applied to another.

The rest of this paper is organised as follows. Section 2 describes our methods. The data and experiments are discussed in sections 3 and 4 and the results in section 5. We end with a discussion and conclusion.

2 Methods

In order to learn the similarities between the different modalities, we assume that our dataset has corresponding samples from each modality. In practical terms: we apply our methods to registered scans of the same subjects scanned with each modality. This allows us to define learning objectives that minimise the representation difference between corresponding patches from each modality.

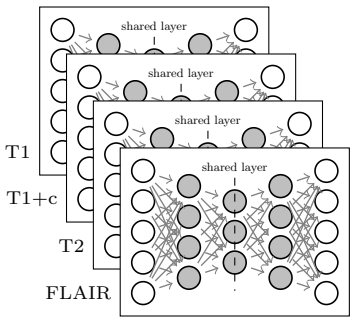

Fig. 1. We train a separate autoencoder for each modality, with a similarity term that connects the central hidden layers across modalities.

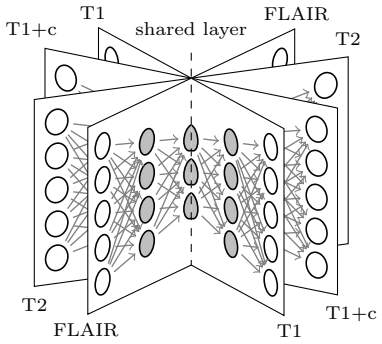

Fig. 2. The axial neural network connects all modalities in a central axis, combining all inputs to get a common representation.

2.1 Autoencoders

Autoencoders [7] are multi-layer neural networks that consist of an input layer, a number of hidden layers and an output layer, with weighted directed connections between nodes in subsequent layers (Fig. 1). The input of the network is an image patch, with each node in the input layer representing one voxel. The first few hidden layers, the encoding part, compute an increasingly small representation of the input. The remaining layers form the decoding part and have an increasing number of nodes, up to the output layer that has the same number of nodes as the input. The network is trained to reconstruct the input, and because the number of nodes in the smallest hidden layer is limited, the model is forced to learn a concise representation of the data. This representation in the central hidden layer can be used as the feature vector for a classifier.

We used autoencoders with rectified linear units (ReLUs) [8] as the hidden nodes and nodes with a linear activation function for the final output layer. The connection weights of the encoding layers were shared by the decoding layers, but the biases of the encoding and decoding parts were independent.

For our experiments on multi-modal data, we trained a separate autoencoder for each of the M modalities. We have corresponding patches in each modality such that sample $\mathbf{x}_{m,i}$ contains the voxel values for patch i in modality m. We denote the values of the central hidden layer of the network for modality m given sample i by $f_m(\mathbf{x}_{m,i})$. Denote the values at the output layer by $g_m(f_m(\mathbf{x}_{m,i}))$. The network for modality m is trained to minimise the mean reconstruction error over all of the N training samples:

$$\mathcal{L}_{\text{err},\,m} = \sum_{i=1}^{N} \left| g_m\left(f_m\left(\mathbf{x}_{m,i}\right)\right) - \mathbf{x}_{m,i} \right|. \tag{1}$$

2.2 Learning similar representations

Training a separate autoencoder for each modality makes it possible to learn a different transformation for each modality, but does not learn a common representation across modalities. We extend the standard learning objective (1) with a similarity term that minimises the difference between the representation of a patch in one modality and its mean representation across modalities. We define the similarity objective for modality m as

$$\mathcal{L}_{\text{sim},\,m} = \sum_{i=1}^{N} \left| f_m\left(\mathbf{x}_{m,i}\right) - \frac{1}{M} \sum_{m'=1}^{M} f_{m'}\left(\mathbf{x}_{m',i}\right) \right|. \tag{2}$$

We combine this similarity objective with the standard autoencoder objective (1) to form a hybrid learning objective

$$\mathcal{L}_{\text{combined},\,m} = \alpha \mathcal{L}_{\text{sim},\,m} + (1-\alpha)\,\mathcal{L}_{\text{err},\,m} \tag{3}$$

where the similarity weight α determines the trade-off between the representation error and the similarity objective. We vary this parameter in our experiments.

2.3 Axial neural networks

The axial neural network (Fig. 2) is a single model that combines all modalities. It has separate encoding layers for each modality, which are joined at a central hidden layer where the incoming representations are averaged into a single, shared representation. This shared representation is used as the input for the decoding part, which is again separate for each modality. For modality m, given the modality-specific encodings $f_{m'}$, the output is defined as

$$g_m\left(\frac{1}{M} \sum_{m'=1}^{M} f_{m'}\left(\mathbf{x}_{m',i}\right)\right). \tag{4}$$

Averaging over the representations encourages the model to learn a common representation that is similar across modalities. The network is trained using a learning objective that minimises the reconstruction error for each modality:

$$\mathcal{L}_{\text{err}} = \sum_{i=1}^{N} \sum_{m=1}^{M} \left| g_m\left(\frac{1}{M} \sum_{m'=1}^{M} f_{m'}\left(\mathbf{x}_{m',i}\right)\right) - \mathbf{x}_{m,i} \right|. \tag{5}$$

Similar to the approach with multiple autoencoders (3), the standard learning objective (5) can be combined with an additional similarity objective to explicitly minimise the differences between the representations coming from each modality:

$$\mathcal{L}_{\text{sim}} = \sum_{i=1}^{N} \sum_{m=1}^{M} \left| f_m\left(\mathbf{x}_{m,i}\right) - \frac{1}{M} \sum_{m'=1}^{M} f_{m'}\left(\mathbf{x}_{m',i}\right) \right| \text{ and} \tag{6}$$

$$\mathcal{L}_{\text{combined}} = \alpha \mathcal{L}_{\text{sim}} + (1-\alpha)\,\mathcal{L}_{\text{err}}. \tag{7}$$

3 Data

We use data of 30 subjects from the BRATS tumor segmentation challenge [6] with four MRI sequences: T1, T1 post-contrast (T1+c), T2 and FLAIR (Fig. 3). The scans of each subject are rigidly registered to the T1+c scan and resampled to 1 mm isotropic resolution. The dataset contains brain masks and labels for four tumor components, some of which can only be identified on one specific sequence or by comparing multiple sequences [9]. Because our experiments require classes that can be identified on any single sequence, we grouped the four components in one foreground class and used the other parts of the brain mask as the background. For each subject we selected a balanced subset of 10 000 patches ($11 \times 11 \times 5$ voxels) for each class, taken at random positions inside the brain mask and at the same position for each sequence. We normalised each patch to zero mean and unit variance. We used the patches from 20 subjects for training, 5 subjects for validation of the random forest parameters and 5 for testing.

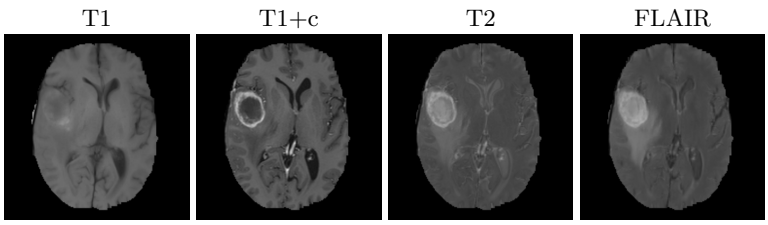

Fig. 3. One slice of the BRATS dataset shown in the four MRI sequences.

We also present experiments with an artifical dataset derived from the BRATS T1+c scans. Using four different MRI sequences makes it harder to see whether a low across-sequence performance is due to the different intensity distribution or simply because some structures are just not visible in one of the sequences. We therefore constructed an artificial dataset by transforming the T1+c scans with the exponential function $f(I) = I^{\gamma}$, where I is the voxel-wise intensity. Because each of the alternative views is derived from the same original scan, each view provides exactly the same information, but with a different distribution of intensity values. Before applying the transformation, we scaled the intensity values to fit between 0 and 1. We used the original intensities from the T1+c scan ($\gamma = 1$) and generated three alternative views computed with $\gamma = \{1.5, 2, 3\}$. After the transformation, each patch was normalised to zero mean and unit variance. We used the same set of training and test scans as in the other experiments.

4 Experiments

In our experiments we trained the autoencoders and axial neural networks on the patches in the BRATS dataset and the dataset with synthetic transformations.

For both scenarios, we trained the models to learn a joint representation for the four modalities. We evaluated multiple values for the weight of the similarity term in the learning objective. Using the learned representation as the feature vector, we then trained random forest classifiers and evaluated these by computing the classification accuracy on the test set. We did this for each pair of training and testing modalities.

We tried several network configurations for the autoencoders and axial neural networks. We used networks with three or five hidden layers, with 100 or 200 nodes in the central layer and 200 to 500 in the others. The number of nodes in the input and output layers was equal to the patch size, i.e., $11 \times 11 \times 5 = 605$ nodes. We trained networks for each combination of these parameters and used the performance on a held-out validation set to select the optimal combination.

As a baseline, we show the results of an approach that does not use different transformations for data from different modalities: we combined the patches from all modalities in one heterogeneous dataset and applied principal component analysis (PCA). We selected the 100 or 200 most important components, depending on the size of the network we compared with, to give the PCA baseline the same number of features as our models.

The networks were implemented in Python using Theano [10]. We trained the networks using stochastic gradient descent with a minibatch size of 50, for 300 epochs with various learning rates (0.0001, 0.001, 0.01 or 0.02). Based on our observations of the reconstruction and similarity objectives, we selected the networks with learning rates 0.001 and 0.01 for our classification experiments. We used the random forest implementation from Scikit-learn [11] for classification, with the number of trees (50, 75 or 100) optimised on the validation set.

5 Results

5.1 Synthetic transformations

First, we present the results on the T1+c data with synthetic transformations. Figures 4 and 5 show the results averaged over the learning rates 0.02 and 0.01, which gave the best results on the training and validation sets. Choosing a similarity weight that is too large leads to suboptimal results, indicating that the networks learn uninformative representations if the similarity term is too strong and the reconstruction term too weak. For the smaller similarity weights, the patterns are different for the same-modality and different-modality scenarios. For same-modality training and test sets, learning representations that are similar across modalities is not important. For different-modality training and test data, the similarity term allows the models to learn similar representations for both datasets. Choosing the right weight for the similarity term brings the performance of different-modality training close to that of the same-modality baseline. This shows that the models learn representations that are similar across modalities.

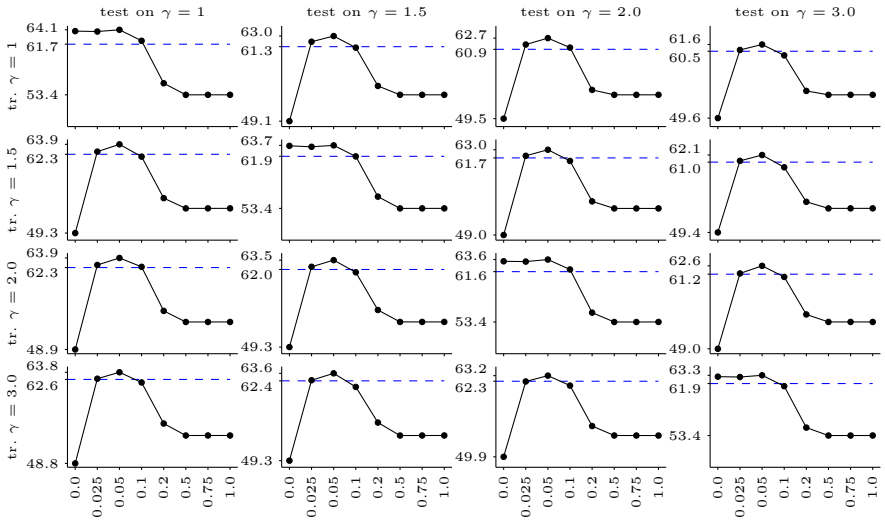

Fig. 4. Results with autoencoders and synthetic transformations of the T1+c scans. Classification accuracies (vertical axes) with features from autoencoders, for different modality pairs (rows and columns) and different weights of the similarity term (horizontal axes, 0 = no similarity). Dashed lines indicate the PCA result.

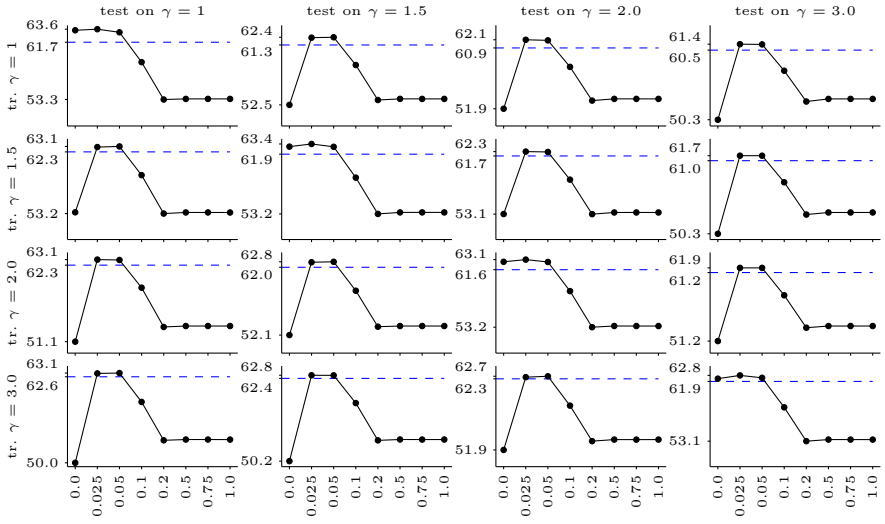

Fig. 5. Results with axial neural networks and synthetic transformations of the T1+c scans. Classification accuracies (vertical axes) with features from axial neural networks, for different modality pairs (rows and columns) and weights of the similarity term (horizontal axes, 0 = no similarity). Dashed lines indicate the PCA result.

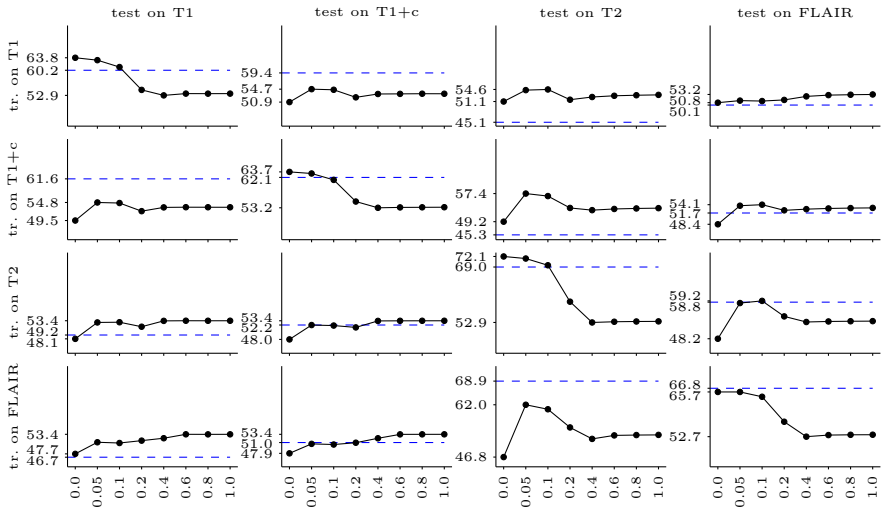

Fig. 6. Results with autoencoders and multi-modal BRATS data. Classification accuracies (vertical axes) with features from autoencoders, for different modality pairs (rows and columns) and different weights of the similarity term (horizontal axes, 0 = no similarity). Dashed lines indicate the PCA result.

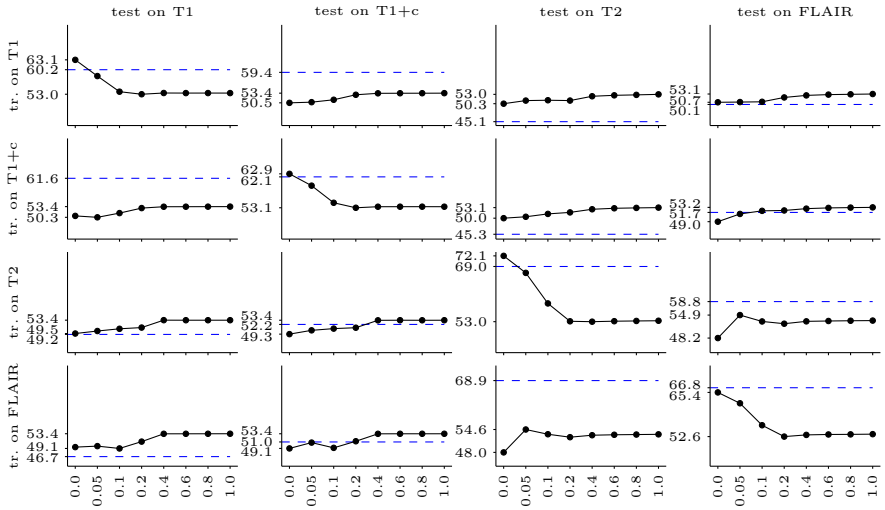

Fig. 7. Results with axial neural networks and multi-modal BRATS data. Classification accuracies (vertical axes) with features from axial neural networks, for different modality pairs (rows and columns) and weights of the similarity term (horizontal axes, 0 = no similarity). Dashed lines indicate the PCA result.

5.2 Four MRI modalities

The results of the experiments on true multi-modal MRI data are shown in Figs. 6 and 7. We show the results averaged over the network sizes and learning rates (0.01 and 0.001). The results of the larger networks tended to be slightly better than those of the smaller networks, but the overall trends were similar to those shown here.

The best classification accuracy was found when the training and testing modalities were the same (the plots on the top-left to bottom-right diagonal). In these scenarios, using a larger weight for the similarity term and a lower weight for the reconstruction error resulted in a lower classification accuracy. With autoencoders, the performance remained relatively stable when the similarity term was added with a small weight (0.1 or less). Overall, the best learned representation often performed equal to or slightly better than the PCA baseline.

The performance in the cross-modality experiments was not as good as in the single-modality experiments. However, adding the similarity term improved the classification accuracy: a mixed learning objective with a small weight for the similarity term performed better than just the representation objective.

In scenarios with different training and testing modalities, the representations learned by our models usually gave a better classification accuracy than the representations made with PCA. PCA worked better for the pairings T1/T1+c and T2/FLAIR, perhaps because those sequences were more similar.

For the autoencoders, for certain modality pairs, the classification accuracy peaked at a similarity weight of 0.05 or 0.1, which corresponds to the plateau of the single-modality experiments. For other modality pairs, a larger similarity component gave a better classification accuracy. A similar pattern appears in the results for the axial neural networks.

6 Discussion and Conclusion

This paper introduced two representation learning approaches for learning similar representations from dissimilar data, using an additional learning objective that minimises representation differences for corresponding patches from different modalities. Our experiments on multi-modal MRI data showed that, when brain and test modalities are different, the representations learned with the similarity objective could produce better classification results than with just the normal learning objective. This effect was strongest in our experiments with simulated modalities derived from a single image, but was also visible in experiments on real multi-modal data.

Although adding the similarity objective can improve results, the weight of the objective should be chosen carefully. Giving a large weight to the similarity component favours learning a representation that is similar over learning a representation that is good at describing the data. This might cause the model to learn a trivial representation, such as all zeros, that may be similar across modalities but is not very useful for classification.

When training and testing on data from the same modality, adding the similarity objective may also lead to a lower performance, especially if the weight of the similarity objective is too large. For the autoencoders in our experiments, adding the similarity objective in a same-domain problem did not significantly decrease the accuracy if the weight of the similarity objective was small enough. This is useful when training a single model for use with multiple modalities.

While training and testing on different modalities may be less relevant in complete, multi-modal datasets such as BRATS, this scenario does have many practical applications. For example, approaches such as those proposed here allow data from different scanning protocols to be used for training a single model. The methods could also be used to suppress differences between scans made with scanners from different vendors. In multicenter studies it is possible to pool data from different modalities in a single model, avoiding the variability that may result from using separate models.

Cross-domain learning can only extract information that is visible in all domains and will have problems learning a common representation for structures visible in only one domain. The methods will therefore be most useful if the domains provide similar information but have different appearances. This is visible in the two sets of experiments in this paper. In the synthetic experiments, the modalities were all derived from the same post-contrast T1 image. This meant that each modality provided the same information – albeit with different intensity distributions – and the models could learn a shared representation that gave a good classification accuracy. In the experiments with real MRI modalities, on the other hand, performance depended on the modality pair. For example, post-contrast T1 was analyzed best by models trained on that modality. T2 and the FLAIR appear to have more in common, but the cross-domain accuracy was still below the same-domain accuracy. This suggests that each modality provides additional information that is not available in the other modalities, which makes it harder to learn a shared representation.

We compared our methods with a fairly simple baseline, principal component analysis (PCA): by extracting the strongest variations in the data, PCA will likely extract common features that are shared between sources. PCA performed quite well for all modality pairs in our experiments on the synthetic dataset, which suggests that PCA is able to learn the artificial transformation that we applied there. This was not the case in our experiments on real multi-modal data, where the differences between the modalities are much more intricate. The power of PCA is limited because it has to use the same transformation for all modalities. This makes it impossible to learn, for instance, if the contrast of one of the modalities is inverted. In contrast, the methods proposed in this paper would be able to model these more complex transformations.

A conceptual advantage of axial neural networks is that they combine data from all domains in a single model, whereas autoencoders require an explicit similarity objective. However, in our experiments we found that autoencoders gave slightly better results, and that the performance of axial neural networks improved if we added an explicit similarity term.

Although the methods in this paper are unsupervised – the classifiers in our experiments were trained after the transformations had been learned – the general approach might also be applied with supervised methods, such as convolutional neural networks. In our axial neural network, for example, the decoding layers could be replaced by a set of output layers that compute class labels.

Using corresponding samples across domains is a powerful and efficient way to learn common representations, but it requires paired training samples. This data is available when, after introducing a new scanner, the same subject is scanned on the old and the new scanner. If there is no paired data, but there is labelled data from both domains, the class labels might provide a weaker form of correspondence between samples from the same class. Without class labels it may be possible to match the feature distribution of each domain.

In this paper we have shown two representation learning models that exploit sample correspondences to learn a common representation for samples from different domains. Using the common representation, a classifier can be trained on data from one domain and applied to data from another. Our experiments showed that classifiers trained on this common representation can, depending on the combination of modalities, achieve a higher accuracy than classifiers trained without the common representation.

References

1. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual Domain Adaptation: A survey of recent advances. IEEE Signal Processing Magazine (2015)
2. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. Technical report, Université de Montréal (2012)
3. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging (2016)
4. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., Liang, J.: Convolutional Neural Networks for Medical Image Analysis: Fine Tuning or Full Training? IEEE Transactions on Medical Imaging (2016)
5. Simo-Serra, E., Trulls, E., Ferraz, L., et al.: Discriminative Learning of Deep Convolutional Feature Point Descriptors. ICCV (2015)
6. Menze, B.H., Jakab, A., Bauer, S., et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging (2015)
7. Bengio, Y.: Learning Deep Architectures for AI. (2009)
8. Hinton, G.E.: A Practical Guide to Training Restricted Boltzmann Machines. Technical report, University of Toronto (2010)
9. Jakab, A.: Segmenting Brain Tumors with the Slicer 3D Software. Technical report, University of Debrecen (2012)
10. The Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. Technical report (2016)
11. Pedregosa, F. et al.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research (2011)