Learning to Segment Multiple Sclerosis Lesions using Convolutional Encoder Networks

***

[1] ***
[2] ***
[3] ***

Abstract. In this paper, we propose a novel approach for segmenting hyperintense T2 lesions in magnetic resonance images of multiple sclerosis patients. In contrast to the recently proposed patch-based classification approaches, our method performs segmentation of the entire image at once. Segmentation is finding a function that maps images to lesion masks. Advantages: 1) don't need to select patches, 2) scales better to large images than patch-based approaches, 3) automatically learned features, no predefined features, and 4) combines feature-learning and classification phase which allows the supervised fine-tuning of features. We have evaluated our method on publicly available data set from the MICCAI 2008 MS lesion segmentation challenge, to allow a direct comparison of our method with state-of-the-art lesion segmentation methods. Our method performs en par with the state-of-the-art method on that data set. In addition, our model scales to large data set and can leverage the large amounts of labelled data achieving a DSC of 57 % in that case.

Keywords: Segmentation, T2 lesion, multiple sclerosis, machine learning, unbalanced classification, deep learning, convolutional neural networks

1 Introduction

Motivation

- MS is a neuro-degenerative disease of the central nervous system characterized by the formation of lesions
- Accurate, reproducible and automatic segmentation of lesions is important to assess disease progression, treatment affect, lesion LL an important endpoint of clinical trials

Related Work

- Lesion segmentation is treated as a voxel classification problem, where algorithms mostly differ in the choice of features and the type of classification algorithm.

- The classification problem itself can then be solved either in a supervised way using, e.g., artificial neural networks [1] or random forests [2], or unsupervised using clustering methods with one outlier class [3] or by treating lesions as an outlier of a generative model [4].
- Trend from simple intensity based features to more complex features extracted or learned from image patches.
- Early approaches use the intensity values of different modalities at a particular voxel as the input features [1].
- However, simple intensity features can be sensitive to intensity variations between images. Carefully chosen context-rich features are more robust [2] to intensity variations.

- Youngjin et. al proposed to learn domain specific features from image patches from an unlabelled data set using unsupervised feature learning [5].
- In the context of the segmentation of cell membranes, Ciresan et. al proposed to perform classification of image patches directly using a convolutional neural network without a dedicated feature extraction step [6]. Features are learned indirectly within the lower layers of the neural network during training, while the higher layers can be regarded as performing the classification.

- Advantage, allows the fine-tuning of features that are useful for the classification task.
- The time required to train complex patch-based feature extraction methods can make the approach infeasible when the size and the number of patches is large. [6] reported a training time of more than a week to train their patch-based segmentation model using 4 GPUs using 2D images with a resolution of 512×512 pixels. To scale patch-based classification to 3D images with a resolution of $256 \times 256 \times 50$, Youngjin et. al used only a small fraction (0.1 %) of the possible patches for training, which might lead to suboptimal learning due to introducing a bias of the learned model towards the selected patches.

Proposed Method

- In this paper, we propose a novel method for segmenting MS lesions that outperforms the state-of-the-art on the MICCAI 2008 lesion segmentation challenge data set, the most widely used publicly available clinical data set for comparing lesion segmentation methods. (Both is to the best of our knowledge). Might rephrase to comparable to the state of the art on this data set depending on how my latest cross-validation experiment goes.
- Our network is a combination of a convolutional and a deconvolutional neural network. The first layer is a convolutional layer [7] that extracts features from multi-modal MRIs at each voxel location. The second layer is a deconvolutional layer [8] that uses the features extracted by the first layer to classify each voxel of the image in a single operation. Given a training set consisting of MRIs and lesion masks, the parameters of the model can be learned using stochastic gradient descent.
- Our network is similar in architecture to convolutional auto-encoders [9] but instead of predicting the input itself, the output layer represents the

predicted lesion masks given a stack of MRIs of the same subject but different modality. Due to the similarity to auto-encoders, we will call our network architecture a convolutional encoder network (CEN).
– A major advantage compared to patch-based approaches is that our approach does not require the selection of patches because it uses all voxels of an image.
– Our model is fast because we can segment an entire image in a single feed-forward pass through the network. Allows to evaluate the segmentation performance at training time. Allows direct maximization of the similarity between predicted and ground truth segmentation during the training stage.
– Combined feature learning and classification like the membrane paper. Learns features that are tuned for the classification task, instead of using a set of general features. But scales much better to high-resolution 3D volumes.
– Traditionally, NNs are trained using sum of squared differences. This metric is not suitable for the segmentation of MS lesions, because MS lesion segmentation is a highly unbalanced classification problem and a neural network trained with SSD would greatly favour one class.
– The second contribution is our new proposed objective function that allows NN to be applied to the voxel-wise classification in the case of very unbalanced classification problems. We propose to use a weighted sum of sensitivity and specificity error to better balance. We will show how convolutional neural networks can be trained using the modified objective function.

2 Methods

In this paper, the task of segmenting MS lesions is defined as finding a function s that maps multi-modal images I to corresponding lesion masks S. Given a set of training images I_n, $n \in \mathbb{N}$, and corresponding segmentations S_n, we treat finding an appropriate function for segmenting MS lesions as an optimization problem of the following form

$$\hat{s} = \arg\max_{s \in \mathcal{S}} \sum_n \mathrm{sim}(S_n, s(I_n)). \tag{1}$$

where sim denotes a function that calculates the similarity between ground truth segmentations and predicted segmentations, and $\mathcal{S}$ is the set of possible segmentation functions.

The set of possible segmentation functions is modeled by the convolutional encoder network illustrated in Figure 1. Our network consists of two layers, a convolutional layer that extracts automatically learned features from multi-modal images, and a deconvolutional layer that uses the extracted features to segment MS lesions. The convolutional layer is a deterministic function of the following form

$$y_j^{(1)} = \max\left(0, \sum_{i=1}^{C} \tilde{w}_{ij}^{(1)} * x_i^{(1)} + b_j^{(1)}\right) \tag{2}$$

where $x_i^{(1)}$ denotes an image representing the ith modality, $y_j^{(1)}$ denotes the feature map corresponding to the jth feature, w_{ij} and $b_j \in \mathbb{R}$ are trainable

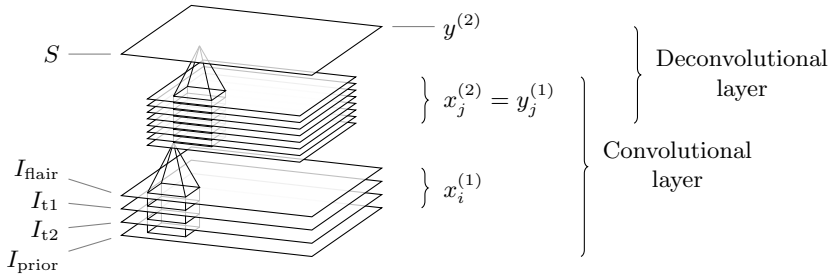

Fig. 1. Convolutional encoder network used to segment MS lesion of images from the MICCAI 2008 lesion segmentation challenge. The first two layers form a convolutional neural network, and the last two layers form a deconvolutional neural network.

parameters of the model, $*$ denotes valid convolution, and $\tilde{w}_{ij}$ denotes a flipped version of w_{ij}. The convolutional and deconvolutional layer are connected by setting the inputs of the deconvolutional layer to the outputs of the convolutional layer, i.e. $x_j^{(2)} = y_j^{(1)}$. The output of the deconvolutional layer can be calculated with

$$y^{(2)} = \text{sigm}\left(\sum_{j=1}^{F} w_j^{(2)} \circledast x_j^{(2)} + b^{(2)}\right) \tag{3}$$

F is the number of features. Also add C as the number of input channels or modalities to the convolutional layer.

where $w_j^{(2)}$ and $b^{(2)}$ are trainable parameters, and $\circledast$ denotes full convolution.

Class of Segmentation Functions

– The parameters can be trained by back-propagation using the delta rule

$$E = \frac{1}{2}\sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \tag{4}$$

$$\delta^{(2)} = \left(y^{(2)} - S\right) y^{(2)}\left(1 - y^{(2)}\right) \tag{5}$$

$$\frac{\partial E}{\partial w_j^{(2)}} = \delta^{(2)} * \tilde{x}_j^{(2)} \tag{6}$$

$$\frac{\partial E}{\partial b^{(2)}} = \frac{1}{N^3}\sum_{\boldsymbol{p}} \delta^{(2)}(\boldsymbol{p}) \tag{7}$$

And for the convolutional layer

$$\delta_j^{(1)} = \left(w_j^{(2)} \circledast \delta^{(2)}\right)\mathbb{I}\left(y_j^{(1)} > 0\right) \tag{8}$$

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = x_i^{(1)} * \tilde{\delta}_j^{(1)} \tag{9}$$

$$\frac{\partial E}{\partial b_j^{(1)}} = \frac{1}{M^3}\sum_{\boldsymbol{q}} \delta_j^{(1)}(\boldsymbol{q}) \tag{10}$$

Similarity Measure

- SSD is problematic for unbalanced classification tasks as the learning will greatly favor one class.
- To overcome this problem, we calculate the mean squared difference for lesion and non-lesion voxels separately and then calculate the weighted sum of the two terms form the final error measure

$$E = r_{\text{sen}} \frac{\sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 S(\boldsymbol{p})}{\sum_{\boldsymbol{p}} S(\boldsymbol{p})} + (1 - r_{\text{sen}}) \frac{\sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \left(1 - S(\boldsymbol{p})\right)}{\sum_{\boldsymbol{p}} \left(1 - S(\boldsymbol{p})\right)} \quad (11)$$

  In the binary case, the first term is equal to 1 minus the sensitivity and the second term is equivalent to 1 minus the specificity. We will therefore call these to terms sensitivity and specificity error.
- The sensitivity ratio r_{sen} can be used to assign different weights to the two terms. Due to the large number of non-lesion voxels, weighting the specificity higher than the sensitivity gives better segmentation results in practice. We've found that a sensitivity ratio between 0.1 and 0.01 gives works best in practice. While the actual choice of the sensitivity ratio has a big impact on the optimal threshold, the segmentation result is not sensitive to the sensitivity ratio.
- Equations (6) to (10) are a consequence of the chain rule of derivatives and independent of the chosen similarity measure, so we only need to derive the new update rule for $\delta^{(2)}$. With $\alpha = 2r_{\text{sen}}(\sum_{\boldsymbol{p}} S(\boldsymbol{p}))^{-1}$ and $\beta = 2(1 - r_{\text{sen}})(\sum_{\boldsymbol{p}}(1 - S(\boldsymbol{p})))^{-1}$ we can rewrite E as

$$E = \frac{1}{2} \sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \alpha S(\boldsymbol{p}) + \frac{1}{2} \sum_{\boldsymbol{p}} \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \beta \big(1 - S(\boldsymbol{p})\big) \quad (12)$$

$$= \frac{1}{2} \sum_{\boldsymbol{p}} \big(\alpha S(\boldsymbol{p}) + \beta(1 - S(\boldsymbol{p}))\big) \left(S(\boldsymbol{p}) - y^{(2)}(\boldsymbol{p})\right)^2 \quad (13)$$

  The first term does not depend on y and is therefore constant with respect to the model parameters. The second term is identical to the sum of squared difference objective function. The derivatives and therefore delta 2 is also identical to the SSD, whereby the constant term is carried over to the delta update as follows

$$\delta^{(2)} = \big(\alpha S + \beta(1 - S)\big)\big(y^{(2)} - S\big) y^{(2)} \big(1 - y^{(2)}\big) \quad (14)$$

Prevent Overfitting

- Two main sources: 1) bias terms highly tuned to lesion locations observed in the training data, and 2) filters tuned to the intensity range observed in the training data

– Use shared bias terms and add lesion prior calculated from a large data set plus smoothing
– Use data augmentation, i.e. during training, randomly change the brightness contrast and gamma correction of the training images to artificially increase the intensity variability in the training set

Training Pipeline

– Downsample training images and training segmentations from 0.5 mm × 0.5 mm × 0.5 mm to 1 mm × 1 mm × 1 mm voxel resolution.
– Perform brain extraction
– Crop to smallest ROI
– Pad all images to standard size
⇒ Calculate combined cropping parameters
– Perform training of the NN on the downsampled training set
– Calculate probability maps for the entire downsampled training set
– Upsample the probability maps to the native resolution
– Crop lesion masks in native resolution to fit the same ROI as the upsampled probability masks
– Choose the threshold that maximizes the DSC in the training set

Testing Pipeline

– Downsample test image
– Perform brain extraction
– Apply combined cropping parameters
– Infer probability using the downsampled images
– Upsample the probability map to the native resolution
– Apply threshold
– Crop lesion masks to combined cropping parameters in native resolution space
– Compare segmentation

3 Experiments and Results

Experiments to come

– Visualize segmentation performance. Show a good and a bad example.
– Comparison with state-of-the-art methods. I will compare our method to Geremia, dictionary learning and Souplet in terms of TPR, PPV, and DSC. Need to check if I can get Tianming Zhan and BRICQ Stéphanie papers. Can't compare to Xavier Tomas-Fernandez, because he didn't compute TPR, PPV, or DSC.
– Sensitivity analysis. Will be performed with best parameters as baseline. Parameters to be varied are: a) sensitivity ratio (0.05, 0.02, 0.01), b) number of epochs for training, c) filter size (13, 11, 9, 7, 5), d) number of filters (64, 32, 16), and e) number of training cases (2-fold, 3-fold, 4-fold, and 5-fold CV). In all cases, I will compare the influence on the training and testing DSC and the best threshold.

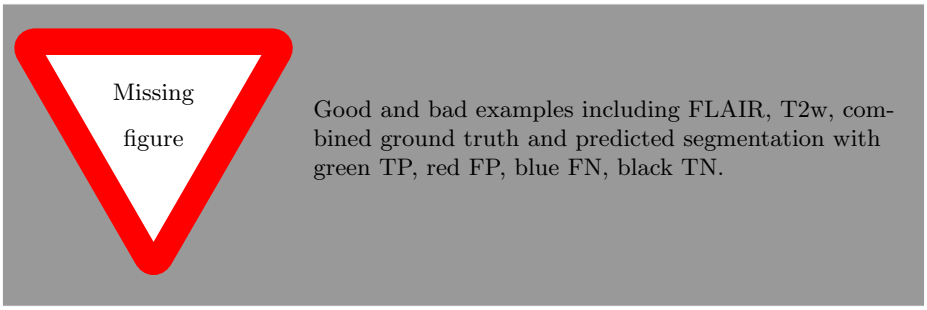

Fig. 2. Example segmentation using our method. Comment while the method performed poorly. Maybe it just was a very difficult case.

- Does the train performance correlate with the lesion load? Visualize good, average and bad example. If it correlates with lesion load, visualize examples with high, average, and low lesion load.
- I need to get my challenge ranking. Inclusion of these results is optional.
- *Optional* Stride-1 tests: Requires memory efficient bias term handling for shared bias terms. Might yield improved results and nicer filters and I don't need to worry about explaining strided convolutions.
- *Optional* Visualization of filters.
- *Optional* Evaluation on BioMS using stride-1 convolutions and new preprocessing pipeline with and without lesion prior, with individual and shared bias terms. Metrics: TPR, PPV, DSC, correlations with lesion load and clinical scores.

Table 1. Comparison of state of the art methods with our method.

Method	TPR	PPV	DSC
Souplet et. al (2008)	20.65	30.00	—
Weiss et. al (2013)	33.00	36.85	29.05
Geremia et. al (2010)	39.85	40.35	—
Our method	39.71	41.38	35.52

4 Conclusions

- Future work: use more layers to achieve a hierarchical segmentation method, but this paper, focus on the simplest possible network to evaluate the potential of such an approach.

Table 2. Segmentation results measured using the Dice Similarity Coefficient. Two layer auto-encoder, stride size of $2 \times 2 \times 1$, 32 filters, sensitivity ratio 0.05. Threshold finding methods: a) using a global threshold that optimizes the average DSC of the entire data set, b) using the optimal thresholds for each sample, and c) using predicted thresholds for each sample.

Method	DSC per Lesion Load Category					
	0.0–4.0	4.0–7.8	7.8–14.7	14.7–28.5	> 28.5	Average
Global threshold	0.298	0.546	0.601	0.650	0.680	0.543
Optimal thresholds	0.351	0.568	0.618	0.667	0.727	0.573
Predicted thresholds	0.338	0.550	0.587	0.653	0.717	0.556

– We have demonstrated the potential of our approach for MS lesion segmentation, although the method is not inherently limited to this kind of segmentation. We are planning to apply this framework to other segmentation problems and we anticipate that other groups will adopt this approach to a variety of segmentation problems.

Acknowledgements ****
****

References

1. Zijdenbos, A.P., Dawant, B.M., Margolin, R.A., Palmer, A.C.: Morphometric analysis of white matter lesions in mr images: method and validation. Medical Imaging, IEEE Transactions on **13**(4) (1994) 716–724
2. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial decision forests for ms lesion segmentation in multi-channel mr images. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010. Springer (2010) 111–118
3. Souplet, J.C., Lebrun, C., Ayache, N., Malandain, G.: An automatic segmentation of t2-flair multiple sclerosis lesions. In: The MIDAS Journal-MS Lesion Segmentation (MICCAI 2008 Workshop). (2008)
4. Weiss, N., Rueckert, D., Rao, A.: Multiple sclerosis lesion segmentation using dictionary learning and sparse coding. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013. Springer (2013) 735–742
5. Yoo, Y., Brosch, T., Traboulsee, A., Li, D.K., Tam, R.: Deep learning of image features from unlabeled data for multiple sclerosis lesion segmentation. In: Machine Learning in Medical Imaging. Springer (2014) 117–124
6. Ciresan, D., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. Advances in Neural Information Processing Systems (2012) 1–9
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
8. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2018–2025

9. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Artificial Neural Networks and Machine Learning–ICANN 2011. Springer (2011) 52–59