Transfer Learning by Adaptive Merging of Multiple Models

Robin Geyer GEYERR@STUDENT.ETHZ.CH
Luca Corinzia LUCA.CORINZIA@INF.ETHZ.CH
Viktor Wegmayr VWEGMAYR@INF.ETHZ.CH
ETH Zurich, Institute for Machine Learning, Zurich, Switzerland

Abstract

Transfer learning has been an important ingredient of state-of-the-art deep learning models. In particular, it has significant impact when little data is available for the target task, such as in many medical imaging applications. Typically, transfer learning means pre-training the target model on a related task which has sufficient data available. However, often pre-trained models from several related tasks are available, and it would be desirable to transfer their combined knowledge by automatic weighting and merging. For this reason, we propose T-IMM (Transfer Incremental Mode Matching), a method to leverage several pre-trained models, which extends the concept of Incremental Mode Matching from lifelong learning to the transfer learning setting. Our method introduces layer wise mixing ratios, which are learned automatically and fuse multiple pre-trained models before fine-tuning on the new task. We demonstrate the efficacy of our method by the example of brain tumor segmentation in MRI (BRATS 2018 Challange). We show that fusing weights according to our framework, merging two models trained on general brain parcellation can greatly enhance the final model performance for small training sets when compared to standard transfer methods or state-of the art initialization. We further demonstrate that the benefit remains even when training on the entire Brats 2018 data set (255 patients).

Keywords: Transfer Learning, Lifelong Learning, Segmentation, Brain, MRI

1. Introduction

Machine learning, especially deep learning, has produced impressive results in supervised learning tasks, given that large and densely annotated training data is available (LeCun et al., 2015; Wainberg et al., 2018). However, the generalization performance of deep learning models deteriorates quickly when training data becomes scarce. This condition is one of the reasons that have prevented the extensive use of deep learning models in applications which require expensive annotation, as is often the case in health care.

Transfer learning (TL) (Pan et al., 2010) is a common approach in machine learning to mitigate the the lack of target data. It is based on the intuition that humans can learn new tasks quickly even without many examples, because they can rely on previous, similar experiences. Similarly, TL pre-trains a model on a task, which is similar to the target task, but has sufficient training data available. More specifically, the weights of the model are adjusted to minimize the loss of the first learning task, before they are used as initialization for the target task, as shown in fig. 1(*b*). Besides improving generalization, TL also offers a way to share information without sharing sensitive data, because only the model parameters are revealed to the community of interest. Again, this advantage is particularly evident in medical applications, which often exhibit a co-existence of many privately

maintained models. It can be beneficial for the development of new applications to have access to such prior models, but to date it is not clear how to merge knowledge from multiple models at once.

In order to address this question we propose T-IMM (Transfer-Incremental Mode Matching), an algorithm for transfer learning with multiple prior models. The concept of IMM appears in context of life-long learning (Lee et al., 2017). It differs from transfer learning as its original purpose is not better initialization, but the sequential merging of models, which still retains good performance on all the prior tasks (fig. 1(*c*)). Our work provides a useful re-interpretation of IMM for transfer learning. Moreover, our extension T-IMM enables automatic, and *adaptive* merging of multiple models, depicted in fig. 1(*d*). By the example of brain tumor segmentation in MR images, we demonstrate that T-IMM provides a better initialization than common IMM, which represents the corner case of uniform model merging. [1]

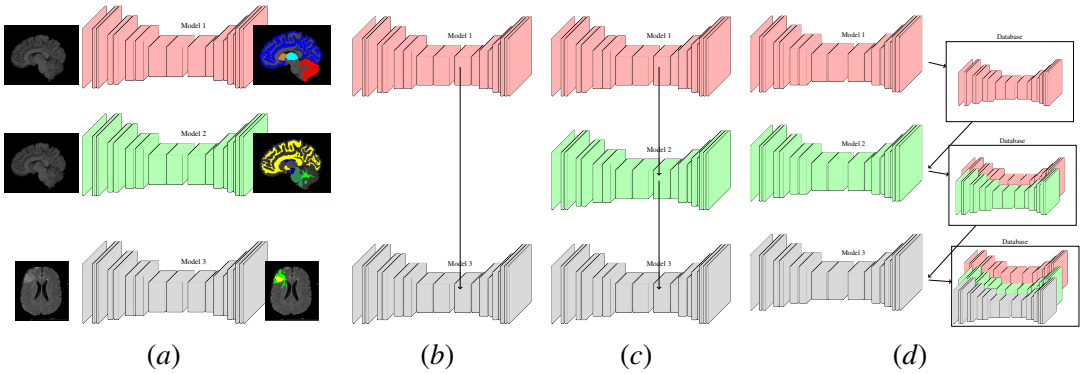


Figure 1: T-IMM framework compared to standard transfer learning approaches. (a) red+green: prior tasks, grey: target task (b) Common transfer learning (c) Sequential IMM (d) T-IMM.

2. Related Work

Transfer Learning TL encompasses methods that discover shared parameters between prior tasks and a target task (Pan et al., 2010). More specifically, TL improves learning of a target task in three ways (Tommasi et al., 2010): (i) better initial performance, (ii) steeper performance growth, (iii) higher performance at the end of training. Moreover, TL is an important part of many state-of-the-art methods in image classification and segmentation. In these cases, TL is mainly performed by reusing the filter parameters of convolutional neural networks (CNN) such as in the work of (Oquab et al., 2014). They use a CNN pre-trained on ImageNet to compute mid-level image representations for object classification in PASCAL VOC images (Everingham et al., 2012), leading to significantly improved results. To this date, the top scoring submissions to the PASCAL VOC challenge continue to use TL, e.g (Chen et al., 2018) pre-trained on the Coco-data set or (Iglovikov and Shvets, 2018) pre-trained on ImageNet. Despite these success stories, little research has been done on leveraging knowledge from multiple models for a new task. Some work is based on ensemble methods (Gao

1. All the code is available at https://github.com/cyrusgeyer/TIMM.git

et al., 2008), which is problematic when the number of available sources is large. A different direction of merging multiple models relies on the particular choice of the SVM model (Tommasi et al., 2010).

Lifelong learning Lifelong Learning (LL) describes the scenario when new tasks arrive sequentially, and should be incorporated into the current model one at a time. In contrast to the TL setting, in LL we require to maintain high performance over prior tasks, too. The reason is, when tasks are learned sequentially, performance typically decreases significantly on earlier tasks. This effect is called catastrophic forgetting (Goodfellow et al., 2013), but it is irrelevant for TL, because we usually only care about performance on the target task. Recent developments in LL such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016) and Learning Without Forgetting (LwF) (Li and Hoiem, 2016)) attempt to overcome catastrophic forgetting by regularization of the target loss function. Incremental Moment Matching (IMM) (Lee et al., 2017) does not change the target loss function, but instead provides a parameter merging scheme for a pair of prior models. Specifically, IMM approximates the posterior distribution of parameters for every prior task as a Gaussian with diagonal co-variance, and then computes the parameter distribution for the new task as the best approximation of the prior mixture of Gaussians.

3. The T-IMM Method

3.1. Adaptively fusing parameters

Let us consider T different but related tasks. Moreover, we assume T models that share the same architecture, and they are trained incrementally on the T tasks. Incremental training means that the parameters of model i are used as initialization for model $i+1$. This procedure is in fact necessary, otherwise it would be impossible to maintain a correspondence between parameters. The parameter set Φ of each model is partitioned into two sets of parameters, i.e. $\Phi = \mathscr{P} \cup \mathscr{S}$, $\mathscr{P} \cap \mathscr{S} = \emptyset$. The first set $\mathscr{P}$ contains all parameters used for fusion, e.g. convolution filters. The second set $\mathscr{S}$ contains all task-specific parameters, e.g. batch normalization parameters or the weights of the top-level layers. Following the work of (Lee et al., 2017), we approximate the parameter co-variance matrix of each model with the diagonal of the inverse empirical fisher information matrix. Ignoring off-diagonal entries in the fisher information matrix is critical for our approach, because it allows simple splitting into parameter subsets. Moreover, we can even introduce multiple IMM-mixing ratios for different subsets of $\mathscr{P}$, e.g. one ratio for each layer. This enables layer-specific, adaptive merging of models. More formally, let the parameter subset $\mathscr{P}_t$ of each task t be composed of N parameter vectors: $\mathscr{P}_t = \{\boldsymbol{\theta}_i^t\}_{i=1}^N$.

We also introduce the set $\mathscr{F}_t$ which holds the Fisher information matrices for each parameter vector in $\mathscr{P}_t$, i.e. $\mathscr{F}_t = \{\boldsymbol{F}_i^t\}_{i=1}^N$.

Lastly, set $\mathscr{A}$ holds mixing coefficients according to which the parameters in $\mathscr{P}$ will be fused:

$$\mathscr{A} = \{\boldsymbol{\alpha}^t\}_{t\in\{1\dots T\}} \quad \text{where } \boldsymbol{\alpha}^t = (\alpha_1^t \dots \alpha_N^t)^{\mathrm{T}} \quad \text{and} \quad \sum_t^T \boldsymbol{\alpha}^t = \mathbb{1}^N \tag{1}$$

The fused parameters of the new model $T+1$ are given by (Lee et al., 2017)

$$\begin{aligned} &\mathscr{P}_{T+1} = \mathscr{P}_{T+1}(\mathscr{A} | \{\mathscr{P}_t, \mathscr{F}_t\}_{i=1}^T) = \{\boldsymbol{\theta}_i^{T+1}\}_{i\in\{1\dots N\}} \\ &\text{where } \boldsymbol{\theta}_i^{T+1} = \left(\sum_t \alpha_i^t \boldsymbol{F}_i^t\right)^{-1} \sum_t \alpha_i^t \boldsymbol{F}_i^t \boldsymbol{\theta}_i^t \end{aligned} \tag{2}$$

3.2. Equally weighted IMM Transfer

Without any further information, the choice of the mixing coefficients is arbitrary. The common IMM method assumes equally weigthed merging, hence it sets $\alpha_i^t = 1/T$, for all tasks t and layers i. After the task-specific parameters $\mathscr{S}_{T+1}$ are randomly initialized, the entire model Φ_{T+1} is fine-tuned on task $T+1$.

3.3. T-IMM

To achieve the best possible performance on the new task, we desire to find a better non-uniform mixing. However, it is clearly impractical to search the space of all possible $\mathscr{A}$ manually, in particular if T is large. For this reason, the Transfer-IMM (T-IMM) method splits transfer learning into two stages: a short adaption stage and an extensive fusing stage.

Adaption Stage In the adaption stage, we aim to learn the mixing coefficients that are best suited for transferring knowledge. We start by randomly initializing the task-specific layers in $\mathscr{S}_{T+1}$. The merged parameters $\mathscr{P}_{T+1}$ are initialized according to eq. (2), as a function of $\mathscr{A}$. The adaption stage optimization can be formalized as:

$$\underset{\mathscr{A},\mathscr{S}}{\text{minimize}} \quad Loss_{T+1}(\mathscr{A},\mathscr{S}|\mathscr{P}_{T+1}(\cdot)) \quad \text{subject to} \quad \sum_t^T \boldsymbol{\alpha}^t = \mathbb{1}^N \textit{ and } \boldsymbol{\alpha}^t \succeq 0, \forall t \tag{3}$$

The constraints on the mixing coefficients can be enforced reparametrizing the mixing ratios. We introduce a set of new unconstrained variables $\{\boldsymbol{\delta}^t\}_{t\in\{1,\dots,T\}}$ and using the sigmoid activation function σ we can write the mixing ratios as:

$$\alpha_i^t = \frac{\sigma(\delta_i^t)}{\sum_{j=1}^T \sigma(\delta_j^t)} \quad \text{with} \qquad \delta_j^t \in \mathbb{R}$$

The adaption stage terminates once the loss converges, and returns a set of mixing coefficients adapted to the new task.

Fine tuning stage After determination of the mixing ratios $\mathscr{A}$ according to eq. (3), all parameters $\Phi_{T+1} = (\mathscr{P}_{T+1}, \mathscr{S}_{T+1})$ are fine-tuned until convergence on a validation set. We also reuse $\mathscr{S}_{T+1}$ from the adaption stage as initialization for the fine-tuning stage. More formally:

$$\underset{\mathscr{P},\mathscr{S}}{\text{minimize}} \quad Loss_{T+1}(\mathscr{P},\mathscr{S})$$

The T-IMM method is depicted in fig. 2.

4. Experiments and Results

4.1. FCNN architecture

Medical image segmentation Fully convolutional neural networks (fCNN) are state of the art in medical image segmentation (2D and 3D). For instance, in segmentation of brain MRI, both BRATS and MRBrainS challenges are lead by fCNN-approaches. This is also the case for interactive segmentation, where 2D- and 3D-fCNNs define the state-of-the-art (Wang et al., 2017a,b). Therefore,

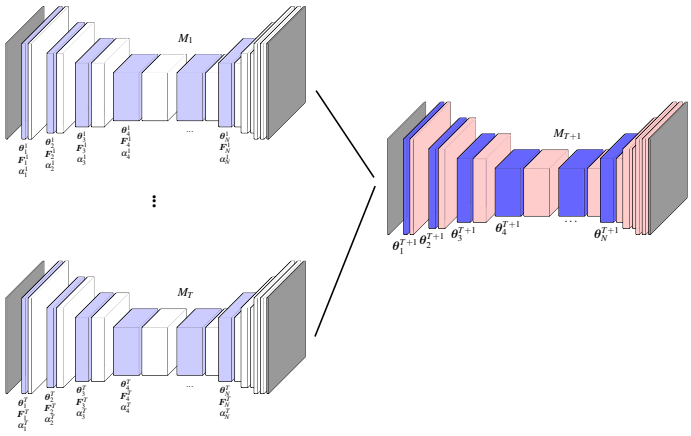


Figure 2: T-IMM framework: T models trained on T different tasks are merged as initialization for task $T+1$. All the parameters in the light-blue layers are in the $\mathscr{P}$ set. The parameters of the new model $\{\boldsymbol{\theta}_i^{T+1}\}$ in the dark blue layers are merged according to eq. (2). The parameters of the red layers (parameters in $\mathscr{S}$) account for task specific layers (i.e. batch-norm, instance normalization layers, or the last classification layers) and hence are randomly initialized.

we also choose this type of neural network for our demonstration of T-IMM in medical image segmentation. Specifically, we use the fCNN-architecture proposed by (Isensee et al., 2017). This network is inspired by the U-net (Ronneberger et al., 2015), and is comprised of 30 3D-convolutional layers, and 27 instance normalization layers. The task specific parameters $\mathscr{S}$ contain the three top-level segmentation layers, and all instance normalization layers. The remaining convolutional filter parameters comprise the set $\mathscr{P}$ of transferred parameters.

4.2. Data and Tasks

In our experiments we perform three different tasks of brain tissue and brain tumor segmentation. Two tasks concern the parcellation of different brain regions, while the third task is about brain tumor segmentation. While the first two tasks are learned incrementally, the third task is learned with different initializations: random, transfer from either of the two prior tasks, initialization with IMM of model 1+2, and initialization with T-IMM of model 1+2.

Task 1 & Task 2 The data used for the first and second task are brain MR images from the Human Connectome Project. The data set holds a total of 58 brain images with T1+T2 weighting. For each brain, 14 different classes were annotated by (Karani et al., 2018), using the FreeSurfer software: (1) cerebellum gray matter (2) Cerebral gray matter, Cortex and Accumbens, (3) Thalamus, (4) Amygdala and Choroid Plexus, (5) Caudate, (6) Pallidum, (7) Celebrospinal Fluid, (8) Cerebellum white matter, (9) Cerebral white matter, (10) Hippocampus, (11) Ventricel, (12) Putamen, (13) Ventral DC ,(14) Brainstem. The data is split into three non-overlapping groups of size 23,23 and 12 respectively. Task 1 is defined as segmenting labels 1 through 7 on the first split of 23 brains. Task 2 is defined as segmenting labels 8 through 14 on the second split of 23 brains. The third split of 12 brains is the test set both tasks can be evaluated on.

Table 1: Description of the different tasks and datasets.

		Total	Training	Validation	Labels
Task 1		23	18	5	1:7
Task 2		23	18	5	8:14
	4 %	10	8	2	ET, TC, WT
Task 3	8 %	20	16	4	ET, TC, WT
	100 %	255	215	40	ET, TC, WT
			Total		
Test set Task 1 & Task 2			12		1:14
Test set Task 3			40		ET, TC, WT
Online-Val set Task 3			66		ET, TC, WT

Task 3 The third task is brain tumor segmentation as defined by the BRATS-2018 challenge. The data set holds 255 patients. For each patient we have four different modalities (T1,T1w,T2,Flair) and an expert's annotation of the enhancing tumor (ET), the tumor core (TC) and the whole tumor (WT). Furthermore, we are provided an online validation set (Online-Val) of 66 non-annotated patients. We evaluate our framework for different portions of the total data: using 4 %, 8% and 100% of the Brats data-set. 40 brains are used as a test set to evaluate the individual experiments on. For the 100 %-experiments, theses 40 brains are used as the validation set and the online, non-annotated set of 66 patients is used as a test set. This is in order to make our experimental results comparable to each other but also to state of the art benchmarks. An outline of the different tasks and datasets is available in table 1.

Data preprocessing We conduct very simple data preprocessing. For data used in Task 3 (Brats), ANTS N4-Bias field correction is conducted. Furthermore, all data is histogram-normalized to filter out irrelevant differences.

4.3. Experiments

Testing the Framework We start by training an fCNN model on Task 1 (M1). After convergence we use the parameters of M1 to initialize M2, which is then trained on task 2 until convergence. Having trained M1 and M2, the main experiments are conducted. For each data portion (4%, 8% and 100%) the following five initialization methods for task 3 are tested: Xavier random initialization (referred to as 'No Transfer'), Parameter Transfer from Model 1, Parameter Transfer from Model 2, Parameter Transfer using IMM and Parameter Transfer using T-IMM.

Understanding T-IMM In order to better understand T-IMM, we further conduct the following three experiments (only for the 8% portion due to computational reasons): parameter transfer from a model that was trained on all HCP-data and on all labels 1:14, parameter transfer from a model that was trained on all HCP-data but only on labels 1:7 and parameter transfer from a model that was trained on all HCP-data but only on labels 8:14. All these are then compared to 8% T-IMM.

Transfer Learning and catastrophic forgetting In a last experiment we evaluate how much of tasks 1 & 2 is remembered by the different initialization models. For this, the task specific sets of parameters $\mathscr{S}_1$,$\mathscr{S}_2$ of M1 and M2 are used in combination with the parameter sets $\mathscr{P}_1$,$\mathscr{P}_2$,$\mathscr{P}_{IMM}$,$\mathscr{P}_{T-IMM}$.

4.4. Evaluation

The metric we chose to asses an fCNN's performance is the Dice Coefficient (DC) averaged over all relevant labels reached on the test set examples. table 2 holds our main results for 4%, 8% and 100% of Brats data. We are dealing with paired-samples, i.e. for each patient in the test set we evaluate the DC difference between T-IMM and all other methods. The differences are visualized in fig. 4. The numbers reported on the 100%-scenario are the feedback of submissions of the 66-unannotated examples to the BRATS validation leader-board. Table 3 shows the length of the different training stages of T-IMM and the final validation score reached at the end of the adaption stage. The distribution of the mixing ratios after the adaption stage is displayed in the appendix Table 4 evaluates how well the models used for transfer remember tasks 1 and 2. This is also visualized in fig. 3.

Table 2: Mean Dice Coefficient of ET,TC and WT for different transfer scenarios and different portions of Brats data

Transfer:	No	Model 1	Model 2	IMM	T-IMM	All HCP all labels	All HCP labels 1:7	All HCP labels 8:14
4%	0.30	0.39	0.52	0.55	**0.58**	-	-	-
8%	0.55	0.60	0.61	0.63	**0.65**	0.60	0.63	0.63
100%	0.79	0.81	0.81	0.81	**0.82**	-	-	-

Table 3: Epochs needed and validation dice score reached for adaption stage and fine-tuning stage

	Adaption Stage		Fine Tuning Stage	
	Epochs	Val-score	Epochs	Val-score
4 %	16	0.44	88	0.72
8 %	20	0.40	178	0.68
100 %	10	0.67	129	0.77

Table 4: Mean dice coefficient when the models used for initialization are evaluated on the test set of Task 1 & Task 2

	Labels 1:7	Labels 8:14
M1	0.89	0.00
M2	0.01	0.89
IMM	0.38	0.42
T-IMM	0.50	0.39

5. Discussion

We can assert multiple things from the results in table 2 and fig. 4. We see the fact confirmed, that the benefit of transfer learning grows with smaller training sets. This was shown in several studies before. We also see that M1 and M2 are not equally well suited for parameter transfer. Especially for the 4% and 8%-scenario, model 2 clearly brings more advantage than model 1. However, initializing with model 1 still outperforms no transfer. T-IMM manages to solve the dilemma of having to choose a priori which model to transfer knowledge from. The experiments for 8% of Brats data show that initializing training with T-IMM even outperforms initialization with a model that was trained on all tasks and all data that T-IMM is able to fuse from. table 4 shows, that both IMM

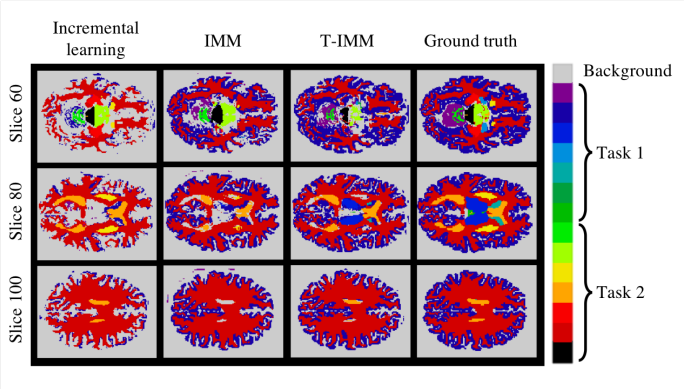


Figure 3: Catastrophic forgetting of task 1, which is learned before task 2. Each task consists of the parcellation of seven different brain regions (colors), and background (grey). **First column**: Prediction after fine-tuning on task 2 using model from task 1 as initialization. The model forgot to predict the blue classes from task 1. **Second column**: Prediction of model obtained with IMM between models trained on task 1 and 2. Clearly reduced forgetting of task 1. **Third column**: Prediction of model obtained after the *adaption stage* of T-IMM. Even though the model was fused to perform a third task, it remembers tasks 1+2 very well. **Fourth column**: Ground truth class labels.

and T-IMM do overcome catastrophic forgetting to a certain extend and manage to remember task 1 and task 2 (even though with lower performance). For T-IMM this is especially interesting, as the model underwent the adaption stage, were it trains to suit task 3. This reassures the assumption that indeed, feature representations from both models/tasks are reused by T-IMM. We were able to show that fusing different CNN for parameter transfer using T-IMM can give a decisive advantage over settling for a singe transfer source, especially when training data is sparse. We further show that the advantage shrinks but remains significant even for large data sets.

References

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. URL http://arxiv.org/abs/1802.02611.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291. ACM, 2008.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. *CoRR*, abs/1801.05746, 2018. URL http://arxiv.org/abs/1801.05746.

Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H. Maier-Hein. Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. *2017 International MICCAI BraTS Challenge*, 2017. URL https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI_BraTS/MICCAI_BraTS_2017_proceedings_shortPapers.pdf.

Neerav Karani, Krishna Chaitanya, Christian Baumgartner, and Ender Konukoglu. A Lifelong Learning Approach to Brain MR Segmentation Across Scanners and Protocols. *arXiv e-prints*, art. arXiv:1805.10170, May 2018.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

Sang-Woo Lee, Jin-Hwa Kim, JungWoo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *CoRR*, abs/1703.08475, 2017. URL http://arxiv.org/abs/1703.08475.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *CoRR*, abs/1606.09282, 2016. URL http://arxiv.org/abs/1606.09282.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, 2010.

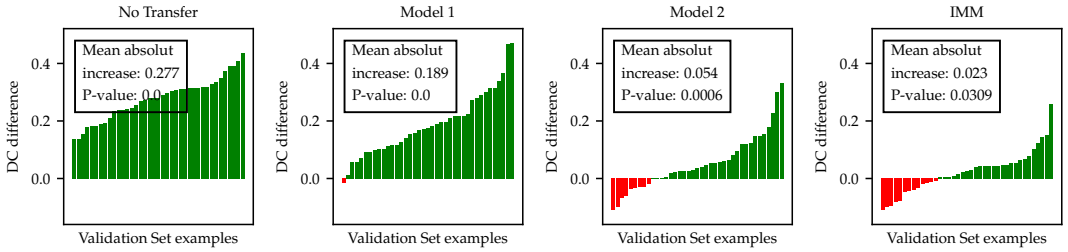
Michael Wainberg, Daniele Merico, Andrew Delong, and Brendan J. Frey. Deep learning in biomedicine. *Nature Biotechnology*, 36:829–838, 2018.

Guotai Wang, Wenqi Li, Maria A. Zuluaga, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Interactive medical image segmentation using deep learning with image-specific fine-tuning. *CoRR*, abs/1710.04043, 2017a. URL http://arxiv.org/abs/1710.04043.

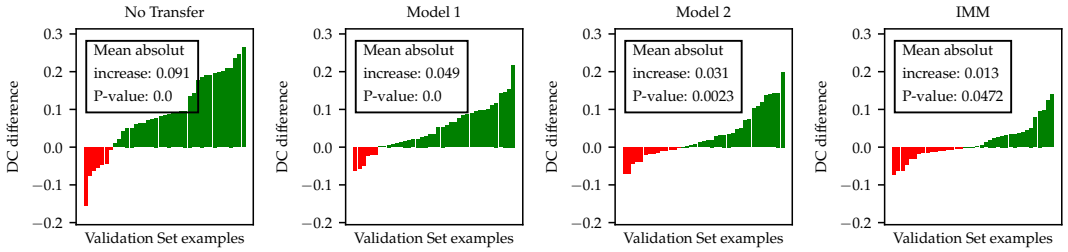
Guotai Wang, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Deepigeos: A deep interactive geodesic framework for medical image segmentation. *CoRR*, abs/1707.00652, 2017b. URL http://arxiv.org/abs/1707.00652.

Appendix A. Subject-level comparison of T-IMM vs. other initializations

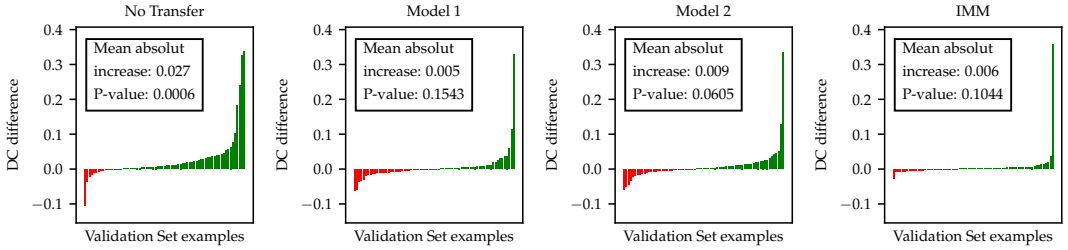
In this section we show the sample-wise performance of T-IMM method compared to other methods on the Brats set. The samples are sorted in ascending order according to the performance of T-IMM.



(*a*)



(*b*)

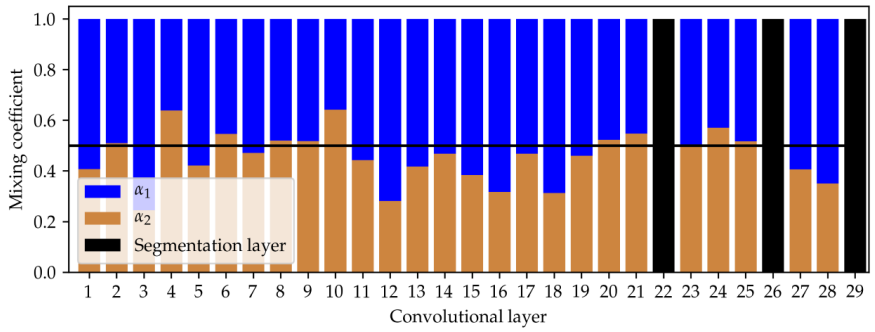


(*c*)

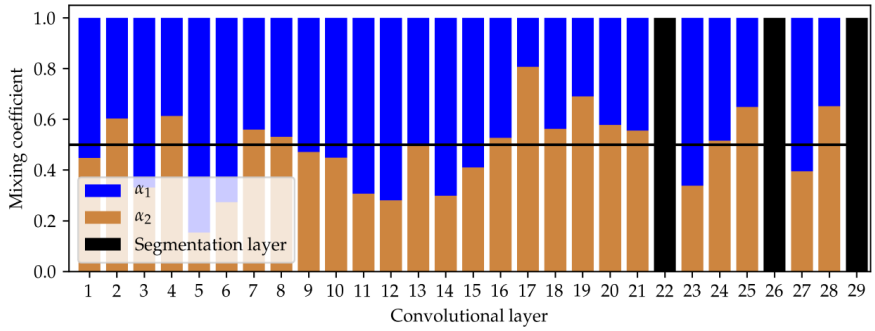
Figure 4: Absolute Dice coefficient increase of T-IMM compared to other transfer methods for training an fCNN on (a): 4 % of Brats data, (b) 8% of Brats data and on (c) 100% of Brats data (the 100% is evaluated on the online validation set)

Appendix B. Mixing ratios after adaption stage of T-IMM

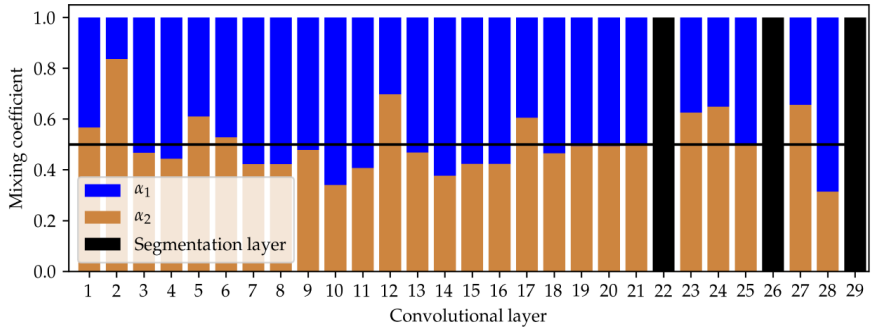
In this section we show the distribution of mixing ratios after the adaption stage of T-IMM was completed and before the fine tuning stage started.



(*a*)



(*b*)



(*c*)

Figure 5: Mixing ratios for each convolutional layer in the set $\mathscr{P}$ of transferable weights, after the adaption stage of T-IMM was completed. Experiments of (a): 4 % of Brats data, (b) 8% of Brats data and on (c) 100% of Brats data (the 100% is evaluated on the online validation set)