Technical University Munich
Department of Informatics

I31 - AI in Medicine and Healthcare
I32 - Computational Imaging in AI and Medicine

Practical exercise 1 — 24. Nov. 2022

Transfer Learning

Submission deadline: 7. Dec. 2022, 11 p.m.

Please submit your solutions (via Moodle).
The corresponding tutorial session is

24. Nov 2022, 4-6 p.m. in lecture hall 5901.EG.051

For questions regarding this exercise sheet, please contact: `paul.hager@tum.de` or `felix.meissen@tum.de`
For general questions, please contact: `course.aim-lab@med.tum.de`

Code and further instructions on this practical can be found at
`https://github.com/compai-lab/aim-practical-3-transfer-learning`

Generating high quality labels for medical datasets is an expensive and time consuming task, especially on tasks such as segmentation where expert radiologists are required. Often we are faced with datasets that have scarce labels or none at all and must rely on selfsupervised pretraining methods to increase our performance. We will continue to use the brain MRI dataset from the previous practicals. The Jupyter Notebook provided contains some preliminary code you can use and some function prototypes that you are expected to fill in. The deliverables for the submission an archive containing the code provided completed as well as a short report explaining your strategies and choices for each task in this practical.

1. (40%) Task 1 : **MONAI and Low Label Regime**

   We will concern ourselves with segmentation again and we will be using the medical imaging library MONAI `https://monai.io/`. Read through some of the documentation, and take a look at the useful functions they have like Dice-CE Loss and UNet. In this part of the exercise we will examine how our models respond to the low label regime. After establishing a full-data baseline, we will investigate how our results change if we were to only have three segmented brains as labels.

   (a) (20%) 1a: Full Data vs Low Data

   Set appropriate arguments for the UNet class and train using the full dataset. Reduce the train dataset down to only three samples using the provided code and rerun.

   What happens to our training dynamics? Do we converge faster or slower? Is there a better way to select a model than a set number of steps?

   (b) (20%) 1b: Early-Stopping

   Implement early stopping and rerun the experiments. Try different patience values and comment on how the selected patience influences validation/test performance. How could we set a high patience but still select the optimal model? (Note: You do not need to implement this)

2. (60%) Task 2 : **Autoencoder and Transfer Learning**

   With such few samples our average Dice score has fallen by quite a bit. Lets try some unsupervised learning strategies to make use of our data that don't have labels.

   (a) (20%) 2a: Train an Autoencoder

   Fill the appropriate arguments for UNet. Take care when setting the number of out channels.

   Visualize the reconstructed results of the autoencoder below the original images.

   (b) (20%) 2b: Transfer Learning

   Initialize a new model and transfer the encoder weights from the trained autoencoder to the new model.

   Freeze the encoder weights and retrain. You should see a mean Dice score improvement between 2 and 3 points depending on your architecture.

   (c) (20%) Task 2c: Experiments

   When doing some form of pretraining there usually are quite a lot of decisions that need to be made. Take the time to explore the alternative options and report your findings. What happens if we leave the network fully trainable? What happens if we train our autoencoder to convergence? What about different architectures? Skip connections? Try some things out and report back!

3. (20%) BONUS: MedicalNet

   There are many already trained models that are floating around the internet. For new projects, especially ones without a lot of data it can be quite useful to use their weights as initial starting points, if the architectures match. One such model that is quite useful is MedicalNet `https://github.com/Tencent/MedicalNet`. Download it and use it for our segmentation task. Comment on the results.