Patch Algorithm

daniele.ceccarelli

February 2021

1 Patch BPINN

In a real clinical application, a grid of electrodes is used to record activation times on heart chambers surfaces. A multi electrode catheter is applied to all the surface and we can map the activation times in the chamber. An example of multi electrode cathether and its use on a heart chamber is shown in Figures 1 and 2.

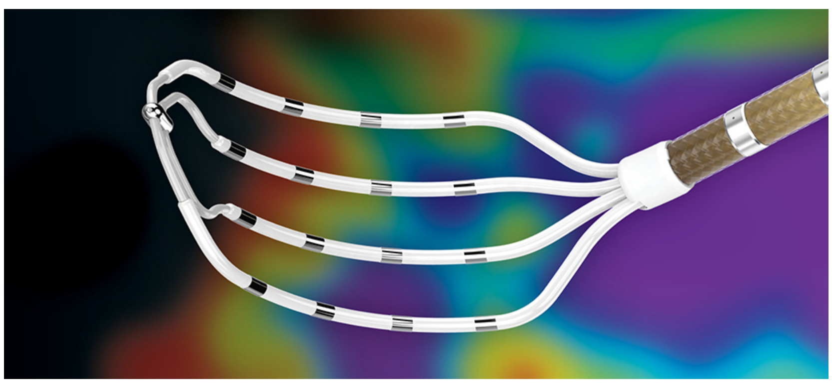

Figure 1: Multi electrode catheter

As shown by these example, a multi electrode catheter has a grid of sensors (16 in the example) and is able to measure the activation times in that locations in real time: step by step the electrode is moved along all the surface to map all the surface. For this reason, the mapping procedure at every step of the procedure is accurate in that particular region (that coincides with the multi electrode catheter area). The domain Ω is splitted in some smaller subdomain and we are mapping all of them one after the other.
We have exploted this idea of subdomain splitting also in our BPINN procedure, ending up in a new algorithm, called "Patch BPINN", where take advantage of both domain splitting and Transfer Learning in a Neural Network [2].

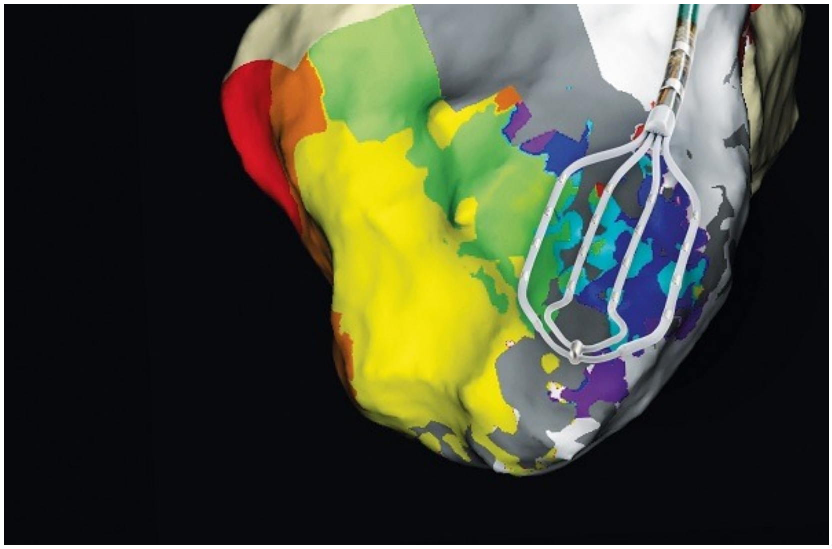

Figure 2: Multi electrode catheter on a Ventricle

1.1 Transfer Learning

Transfer Learning (TL) is a Machine Learning method that use previous knowledge to solve a problem, trying to get better results than starting the training from scratch. We use the knowledge gained with a previous problem that has to be different but also related in some way to the actual problem. To be effective, we need a proper way to share the knowledge: in Neural Networks this is usually done by sharing weights of parameters. There are two different to use a Pre-Trained Network [4] in field of Transfer Learning:

- use the previous weights as initial guess for our Neural Network and train for a few epochs;
- fix the firsts layers with the previous weights and train only the last few layers.

In this work we use the first approach. Some new approaches specifically for Bayesian Networks are presented in [1] but not considered here.
There are three types of performance improvement we can expect applying a Transfer Learning approach, as shown in Figure 3:

- higher start;
- higher slope;
- higher asymptote.

We could expect to find at least one of them, but sometimes also all the three improvements are found. For the purpose of our algorithm, presented

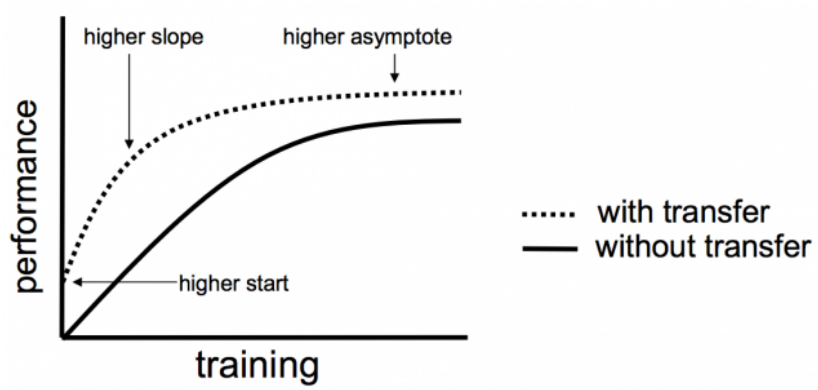

Figure 3: Transfer Learning: three types of performance improvement

in the next section, we look for an higher start and a higher slope in the first epochs: this will make possible to repeat the same faster training on some small portion of the surface and still get accurate results with just few epochs.

1.2 Domain decomposition and TL

The domain Ω is split into K subdomains: $\Omega_i, \quad i = 1, \ldots, K$. For each subdomains we have a dataset $\mathbf{R}_i$ of collocation points and $\mathbf{D}_i$ of exact noisy measurements. We can imagine that every $\mathbf{D}_i$ is a measurements of the multi electrode catheter in a single location.
The idea behind this domain decomposition is simple: after a full training on the first subdomain Ω_1, where we train our BPINN with its datasets $\mathbf{D}_1$ and $\mathbf{R}_1$, we can apply the Transfer Learning approach to all the other subdomains. Figure 4 shows the process on a 2D domain:

- first, we train our BPINN with N epochs on Ω_1;
- then, we reuse the final parameters vector $\boldsymbol{\theta}_{\Omega_1}$ as initial guess (instead of a random guess like Glorot or He initializer [3]) in every $\Omega_i, \quad i = 2, \ldots, K$, using just $\frac{N}{10}$ epochs.

Getting a good and accurate result with just $\frac{N}{10}$ epochs will be very challenging for this task, but the performance boost we get from Transfer Learning will help the algorithm. If this process will be accurate enough, we can save a lot of computational time and resources, since the most of time is spent on Ω_1, while in the others we spend just 10% of it. This is a big improvement toward direction of clinical application, where this procedure can be used during a cardiac mapping process.

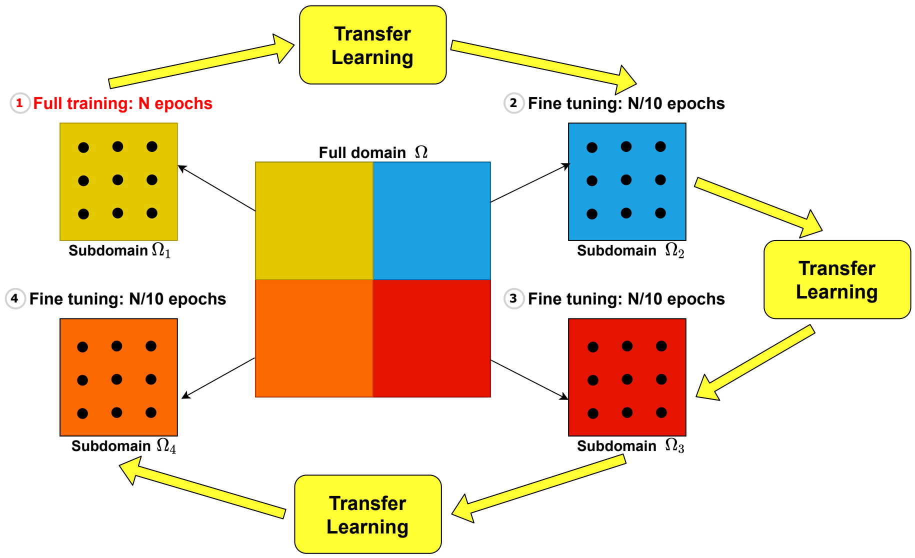

Figure 4: Algorithm on a 2D domain example

1.3 2D test

Let's start from a 2D Isotropic test case, a in domain $\Omega = (0,1)^2$, subdivided in 4 smaller domains $\Omega_1 = (0, 0.5) \times (0.5, 1)$, $\Omega_2 = (0.5, 1) \times (0.5, 1)$, $\Omega_3 = (0.5, 1) \times (0, 0.5)$ and $\Omega_4 = (0, 0.5) \times (0, 0.5)$. In this simple example all the subdomains have their own conduction velocity, as shown in Figure 5.
For this test case, we employ a SVGD Bayesian PINN implementation with 15 particles for simplicity. We set the number of epochs N equal to 100, so in the other subdomain expect the first we use just 10 epochs. The number of collocation points and 10000 and 100 respectively, divided in 25000 and 25 for each subdomain.
The experiment is conducted as follows: we first train all the 4 subdomains for N=100 epochs separately. This is done for comparison purpose. Then we apply our Patch BPINN algorithm: after collecting $\boldsymbol{\theta}_1$, we use it as initial guess for $\boldsymbol{\theta}_i, \quad i = \{2, 3, 4\}$ and train them separately for just 10 epochs. An additional training with Transfer Learning and N=100 epochs in all the subdomains is done in order to compare also the TL performance boost with a longer training. Results are compare looking at the total loss, defined as a sum of quadratic error on T on exact data ($\mathbf{D}_i$) plus quadratic error on residuals ($\mathbf{R}_i$):

$$Tot.\ loss_i = \frac{1}{n_{ex}^i} \sum_{j=1}^{n_{ex}^i} |T_{i,j}^{\boldsymbol{\theta}} - T_{i,j}|^2 + \frac{1}{n_{coll}^i} \sum_{l=1}^{n_{coll}^i} |R_{i,l}^{\boldsymbol{\theta}} - 0|^2, \quad i = 2, 3, 4. \tag{1}$$

Results are shown in Figures 6, 7 and 8.

For what concerne Ω_2 and Ω_4, 10 epochs with TL seems acceptable: in Ω_2

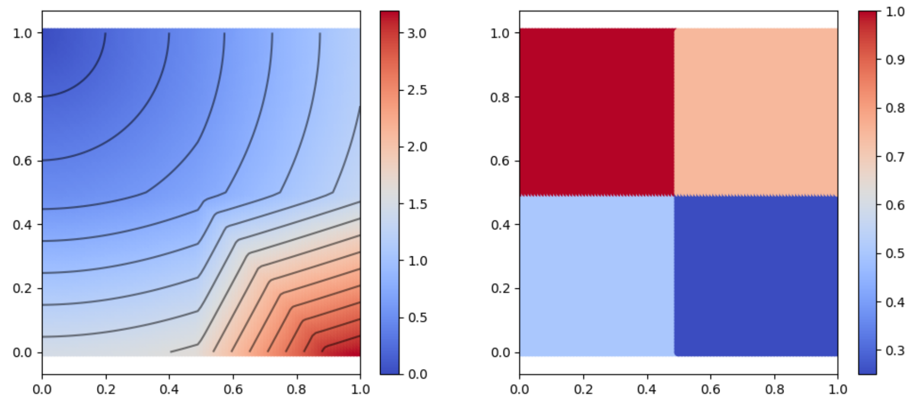

Figure 5: Activation times and conduction velocity in Ω, 2D Isotropic Patch BPINN example

Transfer Learning is so effective that with just 10 epochs we could get a lower loss than with a full training, while in Ω_4 with 10 epochs and TL we get the result of 40 ca. epochs in a "wthout TL" setting. On the contrary, on Ω_3 the transfer learning efficacy is so low that with just 10 epochs we can't see a big difference with respect to the standard case. This problem could be caused by the particular form of activation times T in Ω_3 (lower right quadrant), as shown in Figure 5. Respect to the other subdomains, in Ω_3 we have the collision of two different wave fronts, that produce that particular shape of T. This problem can be easily overcome redefining the activation maps in every subdomains, for instance using always a source localization in the upper left vertices of the subdomain.

Finally we notice that considering all the 100 epochs, Transfer learning provide always a performance boost. In particular, we found all the three types of performance improvement defined in the previous section in Ω_2 and Ω_4, while only two (a better intecept and a better slope) in Ω_3.

1.4 3D test

3D example

References

[1] Roger Luis, L. Enrique Sucar, and Eduardo F. Morales. Transfer learning for bayesian networks. pages 93–102, 2008.

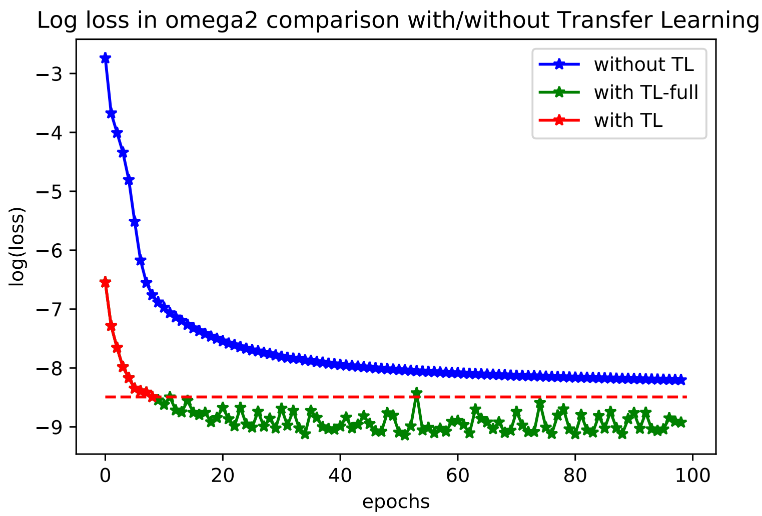

Figure 6: Caption

[2] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[3] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. 28(3):1139–1147, 17–19 Jun 2013.

[4] Dong Yu and Michael Seltzer. Improved bottleneck features using pretrained deep neural networks. pages 237–240, 01 2011.

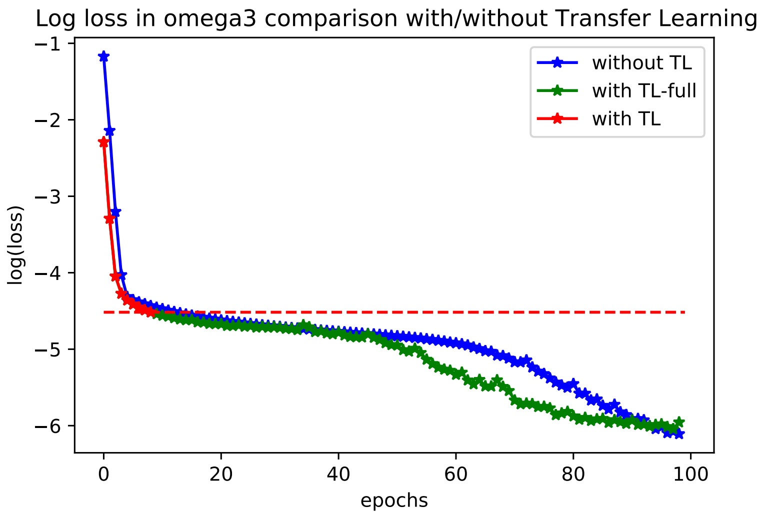

Figure 7: Caption

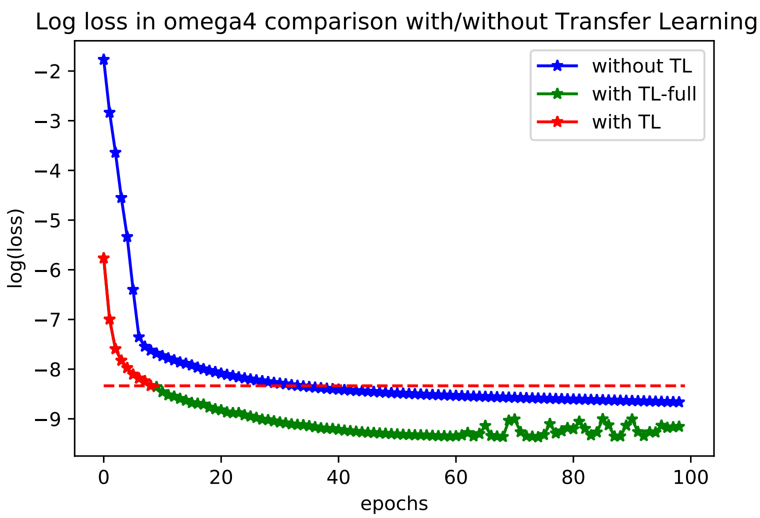

Figure 8: Caption