Meta-Learning for Recalibration of EMG-Based Upper Limb Prostheses

Krsto Proroković [1] Michael Wand [1] Jürgen Schmidhuber [1]

Abstract

An EMG-based upper limb prosthesis relies on a statistical pattern recognition system to map the EMG signal of residual forearm muscles into the appropriate hand movements. As the EMG signal changes each time the user puts the prosthesis on, an efficient method for prosthesis recalibration is needed. Here we show that meta-learning is a promising approach for achieving this aim. Furthermore, we show that meta-leaning can be used to recalibrate the prosthesis even when the examples of some movement types are missing in the target session.

1. Introduction

An upper limb prosthesis is an artificial device whose purpose is to restore some functions of the missing hand. The large number of possible movements which need to be performed flexibly and robustly makes prosthesis control a challenging task.

All muscle contractions generate small electrical currents which can be recorded by surface electrodes (Merletti et al., 2001; Merletti & Parker, 2004). An EMG-based prosthesis uses a pattern recognition system which translates the EMG signal recorded by electrodes attached to the forearm into a corresponding movement which is then performed by the controller (Hudgins et al., 1993; Englehart & Hudgins, 2003; Hargrove et al., 2010; Scheme & Englehart, 2011; Amsüss et al., 2014). Given an EMG signal $\mathbf{x}$ the pattern recognizer outputs $\mathbf{y}$, where y_i is the strength of the movement i and multiple simultaneous movements are possible. Hence, EMG-based prosthesis control is a regression task. We only consider systems where movement commands are generated frame-by-frame, without any sequence modelling.

Note that it would be infeasible for the user to wear the prosthesis all the time; rather it needs to be removed and reattached regularly. This causes variations in the EMG signal due to electrode shifts, sweating, etc. In the context of this study, we define a *session* as an uninterrupted recording of EMG data, during which the recording electrodes are not removed. We assume that we want to apply the system to data from a *target* session, and that we additionally have a set of *source* sessions at our disposal. A pattern recognizer trained on the source sessions might not perform equally well on the target session. Therefore, the prosthesis needs to be adapted after the user puts it on. This process is called recalibration. We are interested in using the source sessions in the best possible way so that recalibration does not require a lot of data and is performed quickly.

In this work we use meta-learning for recalibrating EMG-based upper limb prostheses with a neural network as pattern recognizer. We show that meta-learning outperforms approaches based on no pretraining or conventional pretraining, even though just a small number of source sessions is available. We also show that meta-learning can be used to recalibrate the prosthesis even when not all the movement examples are present in the target session.

Previously, (Vidovic et al., 2015) used covariate shift adaptation to recalibrate prostheses with pattern recognizer based on linear and quadratic discriminant analysis.

2. Data

We use the data from 15 subjects: 4 amputees and 11 able-bodied subjects. A ring of 8 electrodes was placed around the forearm of the subjects as in Figure 1 (without the use of the prosthesis). Every EMG sequence lasts 5 seconds and contains the EMG signal of no movement or one of the following isolated movements: fine pinch, hand open, key grip, wrist extension, wrist flexion, and wrist pronation, wrist supination. The sampling rate is 1 kHz.

Every recording session contains 5 examples of each of 7 movements at each of three strength levels and 15 examples of no movement. Hence, there are 120 sequences per session. For able bodied subjects, recordings were performed on five different days with three recording positions with a shift of ±8mm. For amputees, 10 recordings were collected over 5 different days, where the recording position varies naturally. Thus, we have 10 sessions per amputee and 15 sessions

[1] Instituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), USI & SUPSI, Manno-Lugano, Switzerland. Correspondence to: Krsto Proroković <krsto@idsia.ch>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

per able-bodied subject. The total corpus length is slightly above 34 hours.

Four Hudgins-style features (Hudgins et al., 1993) are extracted from the raw EMG data with frame size of 128 ms and a frame shift of 50 ms, namely: mean absolute value, zero crossing rate, slope sign changes and the wave length. So, we have 32 features in total. Features of each recording session are normalized separately to mean 0 and standard deviation 1. All the systems developed are subject-dependent.

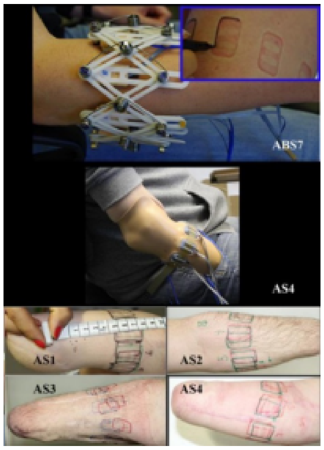

Figure 1. EMG recordings for able-bodied subjects (ABS) and amputees (AS).

3. Methods

We compare three approaches in preparing the pattern recognizer for the target session. All three approaches require the model to be trained on the data from the target session. The main difference is in the way the model is pretrained.

- The first approach we take involves no pretraining. In other words, we initialize the model parameters randomly and train it only on the data from the target session.
- The second approach we take uses the data from the source sessions as a single dataset on which the model is pretrained. We call this conventional pretraining. The resulting parameters are then taken as an initial point for subsequent fine-tuning on the data from the target session.
- The third approach we take is based on learning to learn or meta-learning. Meta-learning deals with reusing previous learning experience in order to quickly adapt to the similar data. One way to perform meta-learning would be to learn a learning algorithm itself, e.g. using evolutionary search (Schmidhuber, 1987; Bengio et al., 1991) or recurrent neural networks (Schmidhuber, 1992; 1993b;a; Younger et al., 2001; Andrychowicz et al., 2016; Ravi & Larochelle, 2017). Here, we use another method which relies on finding model parameters that adapt quickly to the new data by gradient descent. More precisely, we use Model Agnostic Meta-Learning (MAML) (Finn et al., 2017) which optimizes the model to be optimized with a fixed number of gradient steps. In each iteration of the algorithm, the data corresponding to a session is taken and split into two parts, an update part and an evaluation part. Then, model parameters θ are updated with K gradient steps on the update part of the session yielding parameters θ'. After that, the loss is calculated on the evaluation part of the session (using updated parameters θ'), and the original parameters θ are updated by taking a gradient step. This method is described by Algorithm 1. Note that this algorithm includes differentiation through gradient descent and therefore requires second order gradients with respect to model parameters. There are modifications to avoid this (Rajeswaran et al., 2019), but we will not use them here as our model does not contain a huge number of parameters.

Algorithm 1 Model Agnostic Meta-Learning (MAML)

```
Input: A set of sessions $\mathcal{S}$.
Randomly initialize model parameters $\theta$.
repeat
  for each $s \in \mathcal{S}$ do
    Split $s$ into two parts $s^{\text{update}}$ and $s^{\text{eval}}$.
    Compute parameters after $K$ steps of SGD:
    $\theta'_0 = \theta$
    $\theta'_k = \theta'_{k-1} - \nu \nabla_\theta \mathcal{L}(\mathrm{s}^{\text{update}}, \theta'_{k-1})$ for $k = 1, \ldots, K$
    Update $\theta$ using the loss on the eval part.
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\mathrm{s}^{\text{eval}}, \theta'_K)$
  end for
until a stopping condition is met.
```

While we experimented with different neural network architectures, we found the following to work well for all three approaches: feedforward neural network consisting of 2 hidden layers with 200 units, sigmoid activation function, and 20% dropout on the input.

4. Experiments

For every subject, we run a set of experiments. In each experiment, a single session is used as the target session, while the remaining ones are used for pretraining. Out of 120 sequences in the target session, 48 are used for training, 24 for validation, and 48 for testing. We make sure that each type of movement at each strength level is present equally many times in each of three subsets. We observe that the results vary between different runs, so we run 5 experiments for every session being a target one. For hyperparameter tuning we use the data from two amputees and one able bodied subject (we focus on the amputees as the prostheses are intended for them in the first place). The rest of the data is used for the evaluation. As a performance metric we use the coefficient of determination or R^2 score.

4.1. No Pretraining

Here we present an optimized baseline for no pretraining approach. The model is optimized using the ADAM optimizer (Kingma & Ba, 2015) with learning rate 10^{-3} (we also tried 10^{-4}), batch size 64 and early stopping patience 50. The resulting R^2 scores, averaged over all experiments, are shown in Table 1. The results vary drastically between subjects; the average R^2 score is equal to 0.853.

4.2. Conventional Pretraining Followed by Fine-Tuning

Here we present an optimized baseline for conventional pretraining. For pretraining, we optimize the model on the data from source sessions using the ADAM optimizer with learning rate 10^{-3} and batch size 64. A single source session is picked randomly and used for validation with the early stopping patience 50. We then fine-tune the model on the training and validation set of the target session. For fine-tuning we also use the ADAM optimizer with learning rate 10^{-4} (we also tried 10^{-3}).

The results, both without and with fine-tuning are displayed in Table 1. We see that fine-tuning substantially improves the performance of the model on the target session. We also see that conventional pretraining (both with and without fine-tuning) outperforms the approach based on no pretraining on average, meaning that the data from the source sessions is useful when adapting to the target one.

4.3. Meta-Learning

For meta-learning, we use MAML as described in Section 3. In the inner loop we use SGD with learning rate $\nu = 1.0$ (we also tried 0.1, 0.2, 0.5, and 2) and $K = 15$ gradient steps (we also tried 1, 2, 5, and 10). Therefore, we optimize the model to be optimized with 15 steps of SGD with learning rate 1.0. In the outer loop we use ADAM with learning rate $\eta = 0.01$ and meta-batch size of 2 sessions. When splitting the session, 10% of its data is used for the update part and 90% for the evaluation part. As with conventional pretraining, we randomly pick a single session that we use for validation with early stopping patience 50.

We experiment with different ways of fine-tuning. We use two optimizers: SGD with the learning rate 1.0 and whole data from the target session passed in a single batch and ADAM with the learning rate 10^{-4} and batch size 64. First method uses SGD for 15 steps without early stopping. Second method uses SGD with early stopping. Third method uses SGD for 15 steps after what we switch to ADAM while accounting for the validation loss. We remark that for conventionally pretrained networks none of these methods of fine-tuning gave improvement over the one from Section 4.2. The results are displayed in Table 1. We can see that meta-learning substantially outperforms both approaches based on no pretraining and convential pretraining. We also observe that fine-tuning for more than 15 SGD steps improves the performance only marginally leaving the possibility of avoiding early stopping and using all the data for the training. We will not do that here.

4.4. Recalibration when not all the Movement Types are Available

In this section we investigate prosthesis recalibration when the examples of some movement types are not present in the target session. This setup is important as it could simplify the recalibration process by making the user perform only the movements that are easier to recognize.

In our experiments here we remove all the examples of rotation movements from the training set of the target session, i.e. we remove all examples of wrist pronation and supination. We choose rotations because they are easy to distinguish, but the approach taken here might work for the other movement types as well.

Note that in this case it is outright impossible to train a model that generalizes well without any pretraining. Furthermore, we observed that fine-tuning a conventionally pretrained network with as little as single epoch detoriates the performance. Fine-tuning a MAML pretrained network did not give good results compared to conventional pretraining without fine-tuning.

In order to solve this, we make sure that no rotations appear in the update part of the session during MAML pretraining. By doing this, we are optimizing the model to be adapted on the target session without rotations. We notice that the optimization process gets more difficult, so we decrease the number of gradient steps in the inner loop of MAML to $K = 10$ and increase the early stopping patience to 100. We fine-tune the pretrained networks with 10 steps of SGD with learning rate 1.0. The results are displayed in Table 1. We see that this way of meta-learning allows the adaptation to be performed even when not all the movements are available, and besides that, outperforms conventional pretraining and performs comparably with meta-learning with all the movement types.

Table 1. Average R^2 scores of all the methods on the development set.

Method	Amputee 1	Amputee 2	Able-Bodied	Average
No pretraining	0.9412	0.6962	0.9217	0.8530
Conventional Pretraining w/o Fine-Tuning	0.9425	0.7150	0.9036	0.8537
Conventional Pretraining w/ Fine-Tuning	0.9463	0.7236	0.9307	0.8669
MAML with 15 SGD Steps	0.9625	0.7615	0.9277	0.8839
MAML with SGD	0.9626	0.7615	0.9318	0.8853
MAML with SGD + ADAM	0.9624	0.7604	0.9334	0.8854
MAML without Rotations	0.9616	0.7666	0.9260	0.8847

Table 2. Average R^2 scores of all the methods on the evaluation set.

Method	R^2
No pretraining	0.8359
Conventional Pretraining w/o Fine-Tuning	0.7774
Conventional Pretraining w/ Fine-Tuning	0.8457
MAML with 15 SGD Steps	0.8369
MAML with SGD	0.8505
MAML with SGD + ADAM	0.8563
MAML without Rotations	0.8158

4.5. Evaluation

We now test the previously presented methods on the evaluation set, i.e. the data corresponding to the remaining 2 amputees and 10 able-bodied subjects. The results, averaged over all the subjects are displayed in Table 5. For individual results, have a look at the appendix. We see that MAML pretraining followed by fine-tuning with SGD and ADAM substantially outperforms all the other approaches. These results are statistically significant ($p = 0.0055$ for no pretraining and $p = 0.0003$ for conventional pretraining with fine-tuning).We can also see that meta-learning without rotations outperforms conventional pretraining without fine-tuning proving that it is possible to recalibrate the prosthesis even when we do not have examples of all the movement types. This result is statistically significant as well ($p = 4.56 \times 10^{-7}$).

5. Conclusion

In this paper we presented a method for recalibrating EMG-based upper limb prostheses based on meta-learning. The method substantially outperformed the others based on no pretraining and conventional pretraining even though the number of source sessions was relatively small. Furthermore, we believe that the benefits of meta-learning are going to increase with the number of source session used for pretraining. While we use R^2 score to measure how well the prosthesis will work, it is unknown whether this is the case in practice. However, the methods used in the paper should work for other performance metrics, as long as they are differentiable. Besides MAML, we tried using CAVIA (Zintgraf et al., 2019), but we got no improvements in terms of R^2 score. In future, we might use MT-nets (Lee & Choi, 2018) which also learn which model parameters need to be updated on the target session.

Acknowledgments

This research was partially supported by the H2020 project INPUT (grant #687795). We thank University of Göttingen for providing us with the data for our experiments.

References

Amsüss, S., Goebel, P. M., Jiang, N., Graimann, B., Paredes, L., and Farina, D. Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control. *IEEE Transactions on Biomedical Engineering*, 61(4):1167–1176, 2014.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pp. 3981–3989, 2016.

Bengio, Y., Bengio, S., and Cloutier, J. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, 1991.

Englehart, K. and Hudgins, B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE transactions on biomedical engineering*, 50(7):848–854, 2003.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Hargrove, L. J., Scheme, E. J., Englehart, K. B., and Hudgins, B. S. Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(1):49–57, 2010.

Hudgins, B., Parker, P., and Scott, R. N. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Lee, Y. and Choi, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2927–2936, 2018.

Merletti, R. and Parker, P. J. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.

Merletti, R., Rainoldi, A., and Farina, D. Surface electromyography for noninvasive characterization of muscle. *Exercise and sport sciences reviews*, 29(1):20–25, 2001.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, pp. 113–124, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*, 2017.

Scheme, E. and Englehart, K. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.

Schmidhuber, J. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

Schmidhuber, J. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Schmidhuber, J. Reducing the ratio between learning complexity and number of time varying variables in fully recurrent nets. In *International Conference on Artificial Neural Networks*, pp. 460–463. Springer, 1993a.

Schmidhuber, J. A ‘self-referential’weight matrix. In *International Conference on Artificial Neural Networks*, pp. 446–450. Springer, 1993b.

Vidovic, M. M.-C., Hwang, H.-J., Amsüss, S., Hahne, J. M., Farina, D., and Müller, K.-R. Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(9): 961–970, 2015.

Younger, A. S., Hochreiter, S., and Conwell, P. R. Meta-learning with backpropagation. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 3. IEEE, 2001.

Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. 2019.

Results on the Evaluation Set

Method	AS 1	AS 2
No pretraining	0.6556	0.5730
Conventional Pretraining w/o Fine-Tuning	0.6791	0.4664
Conventional Pretraining w/ Fine-Tuning	0.7070	0.5742
MAML with 15 SGD Steps	0.7269	0.5572
MAML with SGD	0.7270	0.5898
MAML with SGD + ADAM	0.7259	0.5975
MAML without Rotations	0.7104	0.5238

Method	ABS 1	ABS 2	ABS 3	ABS 4	ABS 5
No pretraining	0.9223	0.9448	0.9389	0.8621	0.8663
Conventional Pretraining w/o Fine-Tuning	0.8525	0.9010	0.7908	0.7950	0.8031
Conventional Pretraining w/ Fine-Tuning	0.9248	0.9392	0.9340	0.8626	0.8773
MAML with 15 SGD Steps	0.9110	0.9479	0.8912	0.8578	0.8665
MAML with SGD	0.9204	0.9495	0.9310	0.8706	0.8747
MAML with SGD + ADAM	0.9287	0.9508	0.9403	0.8758	0.8799
MAML without Rotations	0.8965	0.9347	0.8070	0.8412	0.8572

Method	ABS 6	ABS 7	ABS 8	ABS 9	ABS 10
No pretraining	0.8289	0.9186	0.7516	0.9414	0.8278
Conventional Pretraining w/o Fine-Tuning	0.7366	0.8946	0.7671	0.9134	0.7296
Conventional Pretraining w/ Fine-Tuning	0.8332	0.9277	0.7935	0.9378	0.8367
MAML with 15 SGD Steps	0.7959	0.9234	0.8179	0.9447	0.8025
MAML with SGD	0.8252	0.9280	0.8172	0.9464	0.8260
MAML with SGD + ADAM	0.8362	0.9332	0.8174	0.9478	0.8416
MAML without Rotations	0.7737	0.9124	0.8144	0.9396	0.7782