ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

Few-shot similarity learning for motion classification via electromyography

Rui Liu[1] and Benjamin Paassen[1]

1- Faculty of Technology
Bielefeld University, Germany
bpaassen@techfak.uni-bielefeld.de

Abstract. Accurate motion classification from surface electromyography signals is crucial for controlling bionic prostheses. Unfortunately, most state-of-the-art classifiers need to be re-trained with lots of data to recognize any new motion. Therefore, we propose a few-shot similarity learning approach that can be applied to new classes without any re-training, just using one to five reference points per new class. In experiments on two real-world data sets, we find that our proposed approach outperforms two state-of-the-art approaches for few-shot learning on sEMG signals, namely a transfer learning and a contrastive learning approach. Our experiments also reveal that the choice of loss function is crucial for performance whereas the choice of similarity function has less effect.

1 Introduction

Tens of thousand of patients lose their hand every year, not least due to injury in the Ukraine war [1]. Bionic prostheses enable such patients to regain some of their hand function [2]. Such prostheses are typically controlled by recording surface electromyography (sEMG) signals of the patient and infer the intended prosthesis motion from these signals [2]. Machine learning-wise, this is typically treated as an on-line motion classification task: every time window of sEMG data is classified and the classification yields a motion command for the prosthesis.

Unfortunately, current sEMG motion classification approaches suffer from several drawbacks, such as lack of robustness to everyday disturbances [3] or the need to re-train the model for every new motion not in the training data set. We address the latter problem. In particular, we propose a few-shot learning approach based on Siamese networks: we train a representation network which maps input sEMG signals to representations that are more conducive to motion classification, then we use a similarity function to perform the actual classification. To classify a new motion, we only need very few examples (about five time windows, corresponding to about one second of recording time) and no re-training of the model.

Some prior work has already addressed the issue of few-shot learning for motion classification from sEMG data. In particular, [4] proposes a meta-learning approach in which each classification decision is made by feeding the test point alongside reference points for each class into a deep network and predicting to which class the test point belongs. While this approach appears to perform well, it is limited to five classes during inference, which is not compatible with our

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

setting, where we wish to extend the number of possible motions a patient can perform. More applicable to our setting is prior work by [5]. They propose to first train a standard deep neural net classifier on the training set of motions, then removing the last two layers of the network, freezing the previous layers, and attaching two new layers which are compatible with few-shot learning and are adjusted to the new classes. We argue that this scheme can be improved by setting up the network as a few-shot learning approach right away.

In more detail, our contributions are 1) we simplify the approach of [5] by applying a few-shot learning scheme instead of a transfer learning + few-shot learning scheme; 2) we investigate four different notions of distance in the final layers of the architecture and demonstrate that the choice does not matter for accuracy (but for convergence behavior); and 3) we evaluate the proposed approach on two real-world motion classification data sets of sEMG data, demonstrating that our proposed scheme outperforms both the state-of-the-art of [5] as well as a variant of contrastive learning.

2 Method

The goal of our method is to perform few-shot motion classification on sEMG data, meaning: we wish to train a motion classifier for sEMG data on a basic set of motions and apply it to new motions without any re-training, using only a few (one to five) reference points for each new motion. We start from the concept of Siamese networks [6], meaning that we first pre-process the input signal, then feed it through several representation layers and finally perform classification based on similarity. However, while Siamese networks are traditionally trained via contrastive learning, we follow [5] who rather learn a similarity measure via binary classification (same-class versus different-class). Hence, we call our method few-shot similarity learning. We now describe representation layers, and similarity computation in turn before we go into more detail regarding training and inference.

Pre-processing: The input is a surface electromyography (sEMG) signal with multiple channels (typically 2-20 electrode channels). We preprocess such a signal using standard filters, and segment the signal into time windows with overlap. For example, Nearlab [5] segments the signal into time windows of 250 ms length where each window is shifted by 62.5 ms to the previous one. After pre-processing, we obtain time windows $\boldsymbol{x}_t$, each a matrix with samples as rows and channels as columns.

Representation layers: Each data point $\boldsymbol{x}_t$ is processed with representation layers to obtain a representation $\phi(\boldsymbol{x}_t)$ that is more suited to motion classification. Similar to [5], we apply three convolutional blocks, each consisting of a 1D convolution, batch normalization, a random rectified linear unit (RRELU) [7], max-pooling, and dropout (refer to Fig. 1). Finally, the resulting matrix representation of the input time window is flattened into a vector $\phi(\boldsymbol{x}_t)$.

Similarity layers: In order to classify an input data point x, we determine how similar it is to reference points x' with known label. As similarity measure,

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

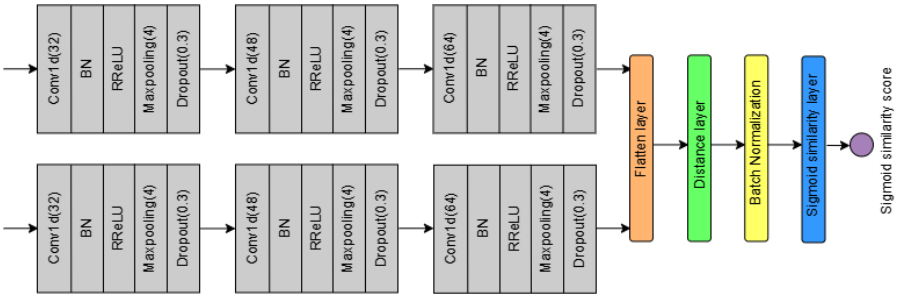

Figure 1: The architecture of the proposed Siamese network.

[5] suggest a rather unusual variant of the L1 distance, namely the absolute difference between the representations $|\phi(x) - \phi(x')|$ followed by a linear layer and a sigmoid. [5] call this an L1 layer. The sigmoid ensures that we can interpret the similarity as a confidence that x and x' belong to the same class. This confidence value can, then, be plugged into a binary classification loss.

Mathematically speaking, the linear transform could also be integrated into the representation layers before. Therefore, we investigate not only the L1 layer proposed by [5] but also a simplified L1-sum layer which uses the regular L1 distance between representations followed by a 1-dimensional linear layer that converts from distance to similarity. More precisely, we obtain the variants:

$$\mathrm{sim}_{\mathrm{L1}}(x, x') = \sigma\Big(b - \sum_{j=1}^{n} w_j \cdot |\phi(x)_j - \phi(x')_j|\Big), \qquad \text{and} \qquad (1)$$

$$\mathrm{sim}_{\mathrm{L1\text{-}sum}}(x, x') = \sigma\Big(b - w \cdot \sum_{j=1}^{n} |\phi(x)_j - \phi(x')_j|\Big), \qquad (2)$$

where w and b are the parameters of the linear layer and σ is the logistic function. To further study the impact of different similarity functions, we also compare to the squared Euclidean distance (L2) $(\phi(x) - \phi(x'))^2$ and its sum version (L2-sum).

Training: Similar to [5], we select random pairs from the training data that either belong to the same or to different classes. Same-class pairs receive the label $y = 1$, different-class pairs the label $y = 0$. Then, we employ the binary crossentropy loss $\ell(x, x', y) = -y \cdot \log[\mathrm{sim}(x, x')] - (1 - y) \cdot \log[1 - \mathrm{sim}(x, x')]$. We optimize the loss via standard, gradient-based optimizers (refer to the next section for more details). We also compare to a variant of the contrastive loss, $\ell(x, x', y) = y \cdot d(x, x')^2 + (1 - y)[1 - d(x, x')]^2$, where $d(x, x')$ is the Euclidean distance between $\phi(x)$ and $\phi(x')$.

Inference: For the inference process, we require reference points $x_{c,k}$ for each class c. For each test point x, we compute the similarities $\mathrm{sim}(x, x_{c,k})$ and assign x to the class with the most similar reference point, i.e. $\arg\max_c \max_k \mathrm{sim}(x, x_{c,k})$.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

3 Results

We compare the few-shot classification accuracy of the L1, L2, L1-sum, and L2-sum variants of our proposed approach with the transfer learning approach proposed by [5], as well as a variant of contrastive learning (see above for the loss).

We evaluate on two data sets, Nearlab and NinaPro DB2. The Nearlab dataset [5] includes surface electromyography (sEMG) data of 11 able-bodied participants performing 8 basic motions (flexion, extension, supination, pronation, hand open, pinch, lateral pinch, and grip) and 6 combined motions (supination or pronation combined with pinch, lateral pinch, or grip). Each basic motion is repeated 5 times in 3 different positions of the hand (sideways, palm up, palm down). The 6 combined motions are repeated 5 times in a sideways hand position. 10 differential sEMG channels are recorded using electrode pairs places around the forearm at 2048 Hz sampling frequency. The data are pre-processed with a 10-500 Hz bandpass filter and a 50 Hz Notch filter and segmented into time-windows of 250ms with a stride of 62.5 ms. The 8 basic motions are treated as training data and the 6 combined motions as test data.

The NinaPro DB2 [8] contains sEMG data of 40 able-bodied participants performing three exercises: exercise B involves 9 hand configurations (such as pointing or fist) as well as 8 wrist motions (such as pronation or supination); exercise C involves 23 grasps; and exercise D involves flexions of single fingers or two fingers combined. Each motion is repeated 6 times. 8 electrodes are placed in a ring around the forearm, 2 additional electrodes at tricep and bicep, and two final electrodes close to the wrist. All electrodes were wireless with 2kHz sampling rate. The sEMG data are pre-processed with a 10-450 Hz bandpass filter and segmented into time windows of 200 ms with a stride of 30ms. Exercises B and C form the training dataset, while exercise D is used as test dataset.

The hyperpaprameters are chosen in alignment with [5]. For all models, the representation layers are three 1D convolutional layers with 32, 48, and 64 neurons, respectively. The filter sizes of the 1D convolutions are 13, 9, and 5. Each convolutional layer is followed by 1D BatchNorm, a random ReLU, max-pooling with size of 4, and a dropout layer with rate at 0.3 (refer to Fig. 1). We train all models using an Adam optimizer with a batchsize of 32 for Nearlab (as in [5]) and 64 for NinaPro. The training lasts 100 epochs, and the learning rate decays from 0.0002 to 0.00008 for Nearlab and from 0.0002 to 0.00001 for NinaPro. For L1-sum and L2-sum, the learning rate decays from 0.0002 to 0.00001; the low learning rates were needed to make L1-sum and L2-sum converge somewhat reliably.

All models are implemented in pytorch with cuda. The experiments were run on a desktop PC with Intel Core i9-13900 CPU, 32 GB RAM, and NVIDIA RTX 4000 Ada GPU with 20GB VRAM. The code of this paper can be found at `https://gitlab.com/fewshotsiamesenetwork/ESANN/`.

Table 1 lists the accuracy and standard deviation of all methods on the Nearlab and NinaPro dataset. We observe that L1 performs best with an accuracy

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Table 1: Mean 5-shot accuracy (± std.) across participants of all methods on Nearlab (left) and NinaPro (right).

method	**Nearlab**	**NinaPro**
transfer learning[5]	0.75 ± 0.09	0.35 ± 0.09
contrastive loss	0.80 ± 0.07	0.39 ± 0.08
L1	0.85 ± 0.06	0.44 ± 0.09
L2	0.83 ± 0.06	0.43 ± 0.09
L1-sum	0.84 ± 0.04	0.45 ± 0.09
L2-sum	0.84 ± 0.06	0.43 ± 0.09

Table 2: Mean n-shot accuracy (± std.) of L1 on Nearlab.

n	1	2	3	4	5
acc.	0.73 ± 0.08	0.79 ± 0.08	0.82 ± 0.06	0.84 ± 0.06	0.85 ± 0.06

of 0.85 ± 0.06. The inference speed of each sample is 13ms. However, the other similarity learning methods(L2, L1-sum, L2-sum) perform similarly well and no significant differences are detected. Both contrastive learning and transfer learning performed significantly worse ($p < 0.01$ for both in a Wilcoxon signed-rank test). This indicates that the choice of distance has only minor effects on accuracy, but the loss function (binary crossentropy instead of our contrastive loss variant) and training procedure (few-shot learning instead of transfer learning) does matter. During the experiments, we note that the L1-sum, and L2-sum methods sometimes did not converge and had to be restarted. We suspect that the convergence issues for L1-sum and L2-sum are due to an ill-conditioned optimization problem, whereas the over-parametrization in L1 and L2 may support optimization speed and reliability [9]. As such, in practice, L1 or L2 are recommended. We also notice that accuracies on NinaPro are much lower compared to Nearlab, indicating that this data set is considerably harder, likely because the test motions are quite different from the training motions.

Table 2 lists the accuracy of the L1 model on NearLab for varying number of reference points. The accuracy of one-shot learning is 0.73 ± 0.08 and is similar to the transfer learning scheme from Table 1. The accuracy improves by 11% to 0.84 ± 0.06 when using 4 data points. The additional improvement to 5-shot learning is only one percent (non-significant), indicating saturation.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

4 Conclusion

We investigated four variants of few-shot similarity learning, differentiated by the similarity measure they use in the final layer. In experiments on two real-world data sets (Nearlab and NinaPro DB2), we found that the choice of similarity measure has no significant effect on accuracy, but that it does influence convergence properties during training. In particular, variants which weighted single dimensions of the distance converged more reliably. We also observed that training the few-shot learning model from scratch outperforms the transfer learning scheme of [5] by about 10% and contrastive learning by about 5%. This is an interesting finding for few-shot learning, where the Euclidean distance in the last layer has been standard practice. Finally, we observed that the accuracy already appears to saturate at four reference points for few-shot learning, indicating that – given an appropriate representation – very few data points are sufficient for this approach.

Several limitations remain before our approach can be applied in practice, though: Future work should evaluate the approach in an online study with actual amputees and check the the transferability of models across patients.

References

[1] I. Trutyak, V. Malickii, M. Samotowka, V. Trunkvalter, R. Trutyak, and V. Ivaschenko. Problematic issues of limb amputation in wounded with combat trauma. *Proceeding of the Shevchenko Scientific Society. Medical Sciences*, 72(2), 2023.

[2] D. Farina, I. Vujaklija, R. Brånemark, A.M.J. Bull, H. Dietl, B. Graimann, L.J. Hargrove, K.-P. Hoffmann, H. Huang, T. Ingvarsson, et al. Toward higher-performance bionic limbs for wider clinical use. *Nature biomedical engineering*, 7(4):473–485, 2023.

[3] C. Prahm, A. Schulz, B. Paaßen, J. Schoisswohl, E. Kaniusas, G. Dorffner, B. Hammer, and O. Aszmann. Counteracting electrode shifts in upper-limb prosthesis control via transfer learning. *IEEE TNSRE*, 27(5):956–962, 2019.

[4] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S.F. Atashzar, and A. Mohammadi. FS-HGR: Few-shot learning for hand gesture recognition via electromyography. *IEEE TNSRE*, 29:1004–1015, 2021.

[5] R. Soroushmojdehi, S. Javadzadeh, A. Pedrocchi, and M. Gandolla. Transfer learning in hand movement intention detection based on surface electromyography signals. *Frontiers in Neuroscience*, 16, 2022.

[6] G. Koch, R. Zemel, R. Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[7] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv*, 1505.00853, 2015.

[8] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G.M. Hager, S. Elsig, G. Giatsidis, R. Bassetto, and H. Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 16, 2014.

[9] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pages 244–253. PMLR, 2018.