ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Efficient Approximate Representations of Computationally Expensive Features

Raul Santos-Rodriguez and Niall Twomey*

Intelligent Systems Laboratory, University of Bristol, UK

Abstract. High computational complexity is often a barrier to achieving desired representations in resource-constrained settings. This paper introduces a simple and computationally cheap method of approximating complex features. We do so by carefully constraining the architecture of Neural Networks (NNs) and regress from raw data to the intended feature representation. Our analysis focuses on spectral features, and demonstrates how low-capacity networks can capture the end-to-end dynamics of cascaded composite functions. Not only do approximating NNs simplify the analysis pipeline, but our approach produces feature representations up to 20 times more quickly. Excellent feature fidelity is achieved in our experimental analysis with feature approximations, but we also report nearly indistinguishable predictive performance when comparing between exact and approximate representations.

1 Introduction

The objective of supervised Machine Learning (ML) is to accurately learn models that map from data 'features' to target 'labels'. Although much recent work has gone into automatically learning feature representations directly from data, hand-crafted representations are still required for some applications because interpretation and criticism of the model and its predictions become more accessible to the domain experts. Since unrepresentative features will beget unreliable predictions, it is critical that 'good' feature representations are selected, and, in most practical applications, we must resort to employing computationally expensive functions to capture discriminating aspects from the data. However, little consideration is afforded to the cost of analysis once it has moved to deployment machines where the cost of feature extraction may be comparable to the computational capacity of the machine, *e.g.* on wearable embedded devices and Internet of Things (IoT) equipment [1]. It is here where we focus our attention.

The key idea of this paper is to exploit the ability to learn *low-complexity* estimators of *complex* functions (both in terms of time and space requirements). This insight offers the opportunity to preserve the statistical properties of the features but at a significantly reduced computational cost. In other words, we make sacrifices only on the representational accuracy but gain faithful representations of the full feature space and lose little to no predictive performance. The successful realisation of the proposed approach should accommodate the extraction of expressive features and predictions on resource-constrained devices, *e.g.* in IoT [2, 3]. A convenient choice of function estimators are NNs since they

*Email: enrsr@bristol.ac.uk and niall.twomey@bristol.ac.uk

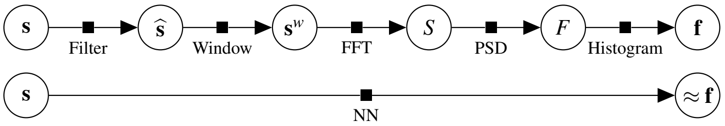


Fig. 1: Pipeline for computing PSD features from raw data (top). The proposed approach (lower) estimates all of the steps above with just one NN.

are universal function approximators [4] and their complexity is a deterministic function of their architecture and thus can be controlled.

Two sub-problems are crucial to the main idea of this paper: 1. feature learning; and 2. parsimonious function approximation. Feature learning has received some attention recently for extraction of chroma features in music information retrieval. In [5] the authors consider a regression framework from spectrograms to chromagrams. Our proposed technique is similar, but instead of operating on already processed data our analysis operates directly on raw and unprocessed data. In so doing we expect to report performance gains. Optimisation has also been studied in the NN field, where 'network sparsification' can reduce the computational burden required for prediction [6].

The main contributions of this paper are as follows: 1. we show that complex features can be accurately and efficiently approximated by networks; 2. we demonstrate that greater performance gains are obtained by jointly approximating several functions concurrently; and 3. we show that no loss of classification performance is obtained by the approximation.

2 Methods

This section outlines the experimental methodology that we followed so that the results can be replicated. Our code will become available after publication, and it will be found at `https://github.com/IRC-SPHERE/`.

2.1 Proposed Functions and Evaluation

Features derived from the Power Spectral density (PSD) of a time series signal quantify periodic regularity and periodicity in data. These features are often aggregated (with histograms) and used in many application areas including activity recognition [7], seizure detection [8], music information retrieval [9], and allergy detection [10]. The pipeline for calculating PSD histograms (referred to simply as PSD henceforth) is shown in the top row of Fig. 1. The raw signal ($\mathbf{s}$) undergoes several deterministic transformations: filtering, windowing (*e.g.* Hamming), Fast Fourier Transform (FFT), PSD, and binning. In contrast, we show how these features will be approximated by NNs in the lower section of Fig. 1.

The NN will be trained with the 'raw' signals as input data and the 'ground truth' (or exact) feature values as targets. We are interested in analysing the performance of the NN in Fig. 1 on several dimensions: 1. Mean Absolute Error

(MAE) between exact and approximate PSDs; 2. Time Gain (TG) achieved when computing approximate PSDs *v.s.* true PSDs; and 3. changes to predictive error rates when approximate PSDs are used with classifiers. Due to space limitations, we focus only on PSD histogram estimates in this paper. We have also experimented with Mel-Frequency Cepstrum Coefficients (MFCC) features and our main results (Section 3) are mirrored on the MFCC setting.

2.2 Data

We use the accelerometer channels from the SPHERE Challenge dataset [11, 12] in our analysis. This dataset was the focus of a public competition for the purposes of activity recognition. To extract the complete features, the acceleration data was filtered with high-pass filters with a cutoff frequency of 1 Hz and partitioned into sliding windows of 64 samples (≈ 3.2 seconds) before computing the FFT (with Hanning window). The PSD was then calculated, and aggregated with non-overlapping and logarithmically distributed bins, *i.e.* the i-th bin consists of 2^{i-1} elements.

2.3 Network Architecture

The architecture of the network (*i.e.* the number of layers, and number of hidden units per layer) can be selected to trade off computational complexity and feature accuracy. On highly resource-constrained devices, for example, the practitioner may target networks with little capacity. All experiments in this paper involve two hidden layers with $M = \lceil D/d \rceil$ hidden units, where D is the dimensionality of the data, and the scaling parameter $d > 0$. Larger values d will produce 'simpler' networks with fewer learnable parameters. We analyse several configurations of d in our experimental analysis. Hence, with activation functions σ_l, the output of a two-layer NN is compactly written as:

$$f(\mathbf{X}) = \sigma_3\left(\sigma_2\left(\sigma_1\left(\mathbf{X}\mathbf{w}_1 + \boldsymbol{b}_1\right)\mathbf{w}_2 + \boldsymbol{b}_2\right)\mathbf{w}_3 + \boldsymbol{b}_3\right)$$

where $\mathbf{w}_1 \in \mathcal{R}^{D\times M}$, $\boldsymbol{b}_1 \in \mathcal{R}^{M}$, $\mathbf{w}_2 \in \mathcal{R}^{M\times M}$, $\boldsymbol{b}_2 \in \mathcal{R}^{M}$, $\mathbf{w}_3 \in \mathcal{R}^{M\times K}$, $\boldsymbol{b}_3 \in \mathcal{R}^{D'}$, D' is the number of outputs of the network. We selected Rectified Linear Units (ReLU) as activation functions in all layers and all experiments in this paper since they are computationally efficient when contrasted to sigmoid, tanh, and softplus. ReLU is defined as $\sigma(z) = \max(0, z)$.

In our experiments, we have $D = 3 \times 64 = 192$ and $d \in K = \{1, 2, 3, 4, 5, 10\}$, yielding six model architectures (one for each element of K). Parameters are optimised on the training set with Stochastic Gradient Descent (SGD). We have not regularised the NN since the network capacities are low.

3 Results and Discussion

Table 1 presents the results of our method for a variety of network architectures. The PSD targets span a wide range of values, from approximately 1.5 to 330 with a median value of $\approx$ 51. MAE values are lower than 1 in most cases indicating that the NN is faithfully representing the characteristics of the PSDs. As expected, the MAE increases for NNs of lesser capacity.

Table 1: Test results.

d	MAE	TG	Δerr
1	0.165	2.395	0.002
2	0.277	4.259	0.013
3	0.355	6.460	0.040
4	0.431	8.098	-0.002
5	0.497	10.502	0.002
10	0.867	22.115	-0.002

The TG column tabulates the ratio in time of calculating the exact features *v.s.* approximating these with a NN. Time gains are made in all configurations of NN estimation considered in this paper, and our approach can estimate features maximally over 22 times faster than exact computation. Note that TG is linearly related to energy costs, which is of interest in wireless devices. Incorporating dedicated hardware (*e.g.* Digital Signal Processor (DSP) chips) to mobile/IoT devices will also greatly increase feature extraction throughput. However, while it is difficult to compare against these methods, we note that our approximation methods can also be deployed on DSPs and enjoy similar reductions in computation time.

The final column presents the difference in accuracy from Logistic Regression classifiers trained on exact and approximate features. The values are percentages calculated with $100(Acc_n - Acc_b)$, where Acc_n is the accuracy achieved by the NN model, and Acc_b is the accuracy achieved with exact feature extraction. Since all differences are very close to 0, we can conclude that no noticeable adverse effects are introduced by approximations in this application. Although these are promising results, further investigation on other applications is warranted. It is interesting that exact and approximate performance differences are not statistically significant, even for the setting $d = 10$ which produces features 22 times faster than exact computation.

The distribution of MAE is illustrated in Fig. 2 for the train and test data. The PSD were normalised before computing the MAE since the range of magnitudes of the PSD is large. However, this also allows us to interpret the figures as a percentage deviations from the ground truth data. With this interpretation, we can see that the majority of the approximate spectra are within 1% of the ground truth values. However, we can also see that the test distribution is skewed slightly more to the right on the test data, which might indicate overfitting on the training set. We did not add any regularisation parameters in our experiments since the capacity of the network is limited.

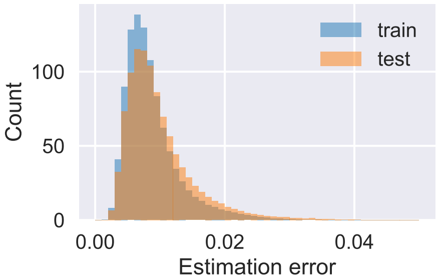
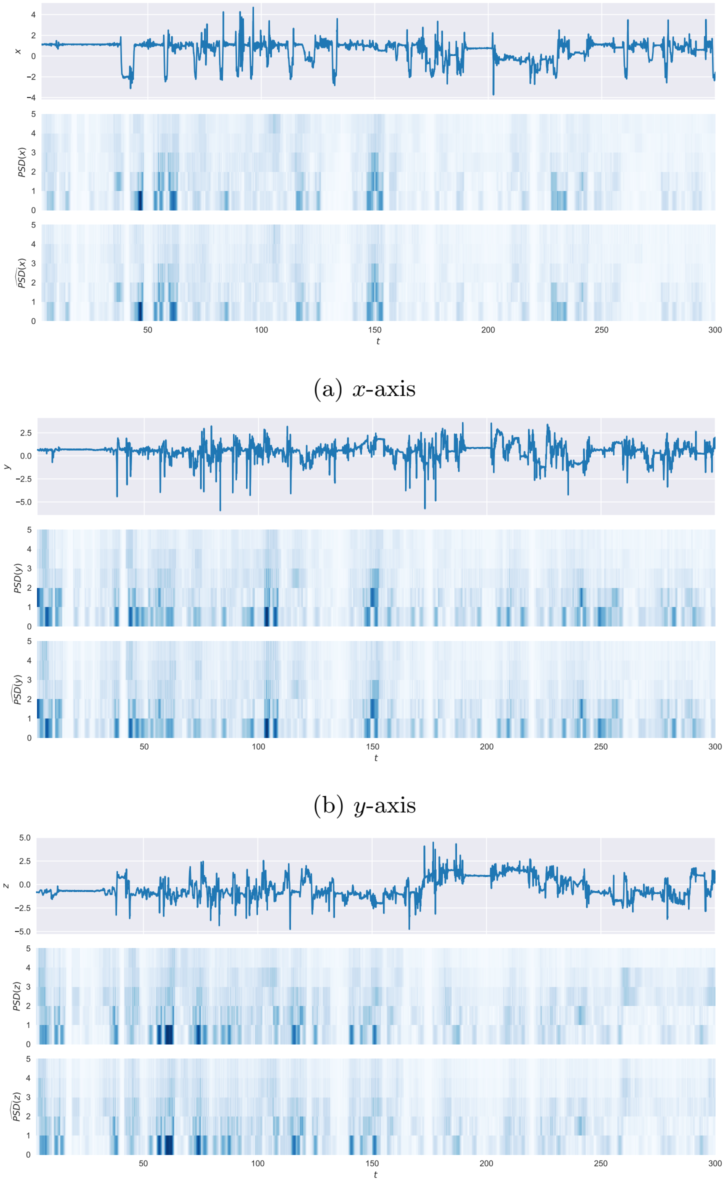


Fig. 2: Distribution over MAE.

Finally Fig. 3 presents visual illustrations of the similarity between ground truth and estimated PSD features for $d = 4$. We can observe very close similarity between the ground truth (middle) and estimated (bottom) spectrograms.



(a) x-axis

(b) y-axis

(c) z-axis

Fig. 3: Five minutes of synchronised triaxial acceleration (top), exact spectrograms (middle), and estimated spectrograms (bottom).

4 Conclusions

This ambition of this study was to investigate opportunities for efficiently approximating computationally intensive features and to quantify the effect of feature estimation on classification performance. Our results have verified that low-capacity neural networks can faithfully approximate the statistical representation of features (which we demonstrate with a case study on power spectral density features) but that these can be estimated with up to 20 times less time than traditional methods. An important finding was that even though smaller approximation networks produce larger feature approximation errors, these errors have negligible effect on predictive performance. The outlined approach is general and can be applied to new features on a variety of computing architectures. Future work will begin to model temporal dependencies in the spectrogram since neighbouring features will not be independent, and we will also extend our experimental analysis to optimise the network for integer-valued data and weights for deployment experimentation on resource-constrained devices.

References

[1] Atis Elsts, Ryan McConville, Xenofon Fafoutis, Niall Twomey, Robert Piechocki, Raul Santos-Rodriguez, and Ian Craddock. *On-board feature extraction from acceleration data for activity recognition.* Association for Computing Machinery, United States, 02 2018.

[2] Ni Zhu, Tom Diethe, et al. Bridging e-health and the internet of things: The sphere project. *IEEE Intelligent Systems*, 30(4):39–46, 2015.

[3] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[4] Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

[5] Filip Korzeniowski and Gerhard Widmer. Feature learning for chord recognition: the deep chroma extractor. *arXiv preprint arXiv:1612.05065*, 2016.

[6] Sourav Bhattacharya and Nicholas D. Lane. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Embedded Network Sensor Systems*, SenSys '16, pages 176–189, New York, NY, USA, 2016. ACM.

[7] Tom Diethe, Niall Twomey, and Peter Flach. Active transfer learning for activity recognition. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2016.

[8] A Temko, E Thomas, W Marnane, et al. Eeg-based neonatal seizure detection with support vector machines. *Clinical Neurophysiology*, 122(3):464–473, 2011.

[9] Rainer Typke, Frans Wiering, and Remco C Veltkamp. A survey of music information retrieval systems. In *Proc. 6th International Conference on Music Information Retrieval*, pages 153–160. Queen Mary, University of London, 2005.

[10] Niall Twomey, Andrey Temko, J Ob Hourihane, and William P Marnane. Automated detection of perturbed cardiac physiology during oral food allergen challenge in children. *IEEE journal of biomedical and health informatics*, 18(3):1051–1057, 2014.

[11] Niall Twomey, Tom Diethe, et al. The sphere challenge: Activity recognition with multimodal sensor data. *arXiv preprint arXiv:1603.00797*, 2016.

[12] Przemyslaw Woznowski, Alison Burrows, et al. Sphere: A sensor platform for healthcare in a residential environment. In *Designing, Developing, and Facilitating Smart Cities*, pages 315–333. Springer, 2017.