The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Deep Feature Extraction for Representing and Classifying Time Series Cases: Towards an Interpretable Approach in Haemodialysis

Giorgio Leonardi, Stefania Montani, Manuel Striani[1]
[1]DISIT, Computer Science Institute, Università del Piemonte Orientale
Viale Michel 11
Alessandria, Italy, 15121
giorgio.leonardi,stefania.montani,manuel.striani@uniupo.it

Abstract

Case-based retrieval and K-NN classification techniques are suitable for assessing haemodialysis treatment efficiency and for identifying risk situations. In this domain, cases involve time series data, that need to undergo a feature extraction phase in order to reduce dimensionality and to speed up similarity calculation. In this paper, we propose a deep learning architecture for time series feature extraction, based on the use of a convolutional autoencoder. *Deep* features provide a better time series representation with respect to features produced by the Discrete Cosine Transform (DCT). Indeed, in our experiments, K-NN classification based on *deep* features has outperformed the DCT-based one. We are also working in the direction of improving interpretability, by using case retrieval results obtained in a different feature space (defined on the basis of domain knowledge) to explain the outputs provided by the adoption of the deep learning technique.

Introduction

Haemodialysis is the most widely used treatment method for End Stage Renal Disease, a severe chronic condition that corresponds to the final stage of kidney failure. The procedure exploits an electromechanical device (haemodialyzer) able to clear the patient's blood from metabolites, to re-establish acid-base equilibrium and to remove water in excess. Every patient typically undergoes three haemodialysis sessions a week, each one lasting for about four hours. Haemodialysis is a critical procedure, as the patient may incur in both short-term complications (such as sickness during the session) and mid/long-term complications, due to an inefficiency of the treatment. Specifically, the efficiency of haemodialysis can be assessed on the basis of a few monitoring variables (Bellazzi et al. 2005), regularly sampled during each session, and thus recorded as time series. Among them, the behavior of the Haematic Volume (HV) variable is particularly important, because it is strictly related to the water reduction rate. Ideally, the HV should fit a model where, after a short period of exponential decrease, a linear decrease follows. Hypotension or haemodynamic instability of the patient under control may influence this kind of behavior; this may result in a different temporal pattern not fitting the model, showing, e.g., a linear decreasing trend since the beginning (thus leading to an insufficient water extraction), or sudden peaks and changes that should be identified, since they can be related to the onset of different cardiovascular problems, to be monitored over time and dealt with (Krepel et al. 2000). It is however worth noting that it is not always easy to distinguish between ideal and critical situations by a simple visual inspection of the HV time series plot, due to the presence of noise, outliers and individual variability. An automated support is therefore strongly desirable.

In this context, the application of case-based retrieval techniques (Aamodt and Plaza 1994) seems particularly suitable. Indeed, defining the HV collected over a haemodialysis session as a case, it is possible to look for similar cases, already labelled as critical or not. A K-Nearest Neighbor (K-NN) classification can thus be provided, allowing for a better management of a patient classified as critical, by means, e.g., of a personalization of the haemodialysis device settings, or of the introduction of corrective actions. Moreover, frequent similar criticalities repeated over time, experienced by the same patient or by different ones, can be provided as an input to (statistical) quality assessment systems, to support health care managers in optimizing the overall provided medical service.

K-NN retrieval and classification require that a proper case representation is available. When dealing with time series, in particular, one typically moves in the direction of converting the original n points measured at the different sampling times into m (with $m << n$) features, able to summarize the time series behavior, thus reducing dimensionality and allowing for a computationally efficient similarity calculation. Classical approaches to time series feature extraction are based on intensive hand-crafted feature engineering (not always practical/possible), or on the adoption of mathematical transforms, such as the Discrete Cosine Transform (DCT) (Strang 1999). However in recent years, as Artificial Intelligence progresses, we are assisting to the development of an alternative approach to feature learning, which is based on deep learning techniques (LeCun, Bengio, and Hinton 2015).

In this paper, we have tested the feasibility of a deep learn-

ing approach to feature extraction for K-NN classification. Specifically, we have compared a traditional DCT + K-NN framework, to an architecture composed by a convolutional autoencoder, adopted to learn *deep* features, followed by K-NN classification as well. In our experimental results in the field of haemodialysis, the deep learning solution has clearly outperformed the solution based on DCT. It is however worth noting that deep learning architectures operate as black boxes, as the meaning of *deep* features (in the sense of their correlation to the original input data) is typically difficult to understand, and a motivation for (mis)classified examples is not provided. In the paper, we also propose a first step in the direction of "opening the black box", by improving interpretability. Specifically, we aim at improving *post-hoc interpretability*, by using previous cases retrieval, conducted in a different feature space, to explain deep learning outputs. In the following, we present the details of our work.

Related work

Deep learning architectures are able to stack multiple layers of operations, in order to create a hierarchy of increasingly more abstract *deep* features (LeCun, Bengio, and Hinton 2015). These techniques have achieved a great success in computer vision, and also their application to time series data classification is gaining increasing attention (Längkvist, Karlsson, and Loutfi 2014), with proposals ranging from the application of Convolutional Neural Networks (CNNs) (Sani et al. 2017) to Long Short Term Memory Networks (LSTMs)(Mehdiyev et al. 2017). Within this research area, autoencoders (Wen and Zhang 2018) have also been proposed for feature extraction. The main idea behind autoencoders is to reduce the input into a latent space with fewer dimensions (encoding) and then try to reconstruct the input from this representation (decoding). By reducing the number of variables which represent the data, we force the model to learn how to keep only meaningful information, from which the input is reconstructable. It can thus be viewed as a dimensionality reduction/compression technique. In image and time series classification, convolutional autoencoders are often adopted (Wen and Zhang 2018). In this kind of architecture, encoding uses convolutional layers, followed by pooling layers, meant to further reduce dimensionality. A convolution is an operation which takes a filter and multiplies it over the entire area of the input. Convolution is particularly suitable for time series data, due to its ability to model local dependencies that may exist between adjacent data points, and to capture how the input evolves over time. The convolution+pooling modules can be stacked in the network, providing progressively deeper architectures. Decoding uses upsampling and convolutions. Once the model achieves a desired level of performance recreating the time series, the decoder part may be removed, leaving just the encoder model. This model can then be used to encode input time series to a fixed-length vector, that can be exploited as an input to K-NN retrieval and classification, as we have done in this paper.

Post-hoc interpretability, i.e., explaining why a given Artificial Intelligence (AI) tool reached its outputs by providing some after-the-fact rationale for the outputs themselves (Lipton 2018), is a line of research being investigated within the more general eXplainable AI (XAI) problem, nowadays addressed by many conferences, and considered by governments in their regulatory effort as well (see, e.g., the European GDPR). Case Based Reasoning (CBR) (Aamodt and Plaza 1994) appears particularly suitable to support post-hoc interpretability: by exploiting cases, it can be used quite naturally to explain "by example" the system output (Keane and Kenny 2019). In fact, CBR has already been coupled to Neural Network (NN) architectures, including deep learning ones, in order to make their output less "obscure" (see, e.g., (Li et al. 2018) and the survey in (Keane and Kenny 2019)). Most works in this area operate by retrieving the most similar cases to a given query case in the same feature space produced by the NN (by using neuron activations (Sani et al. 2017; Papernot and McDaniel 2018) and then deriving feature weights from the NN through different techniques, or by using contribution methods (Kenny and Keane 2019), that do not separate features from weights). On the other hand, we propose to perform case retrieval in a different feature space (defined on the basis of domain knowledge), to explain the outputs provided by the adoption of the deep learning technique. Some preliminary results are presented in the following.

Feature extraction and classification

We have adopted an autoencoder architecture with 4 one-dimension convolution layers with 16, 32, 64 and 128 filters respectively and a kernel size of 3, each one followed by a max pooling layer with a pool size of 2. All the convolution layers are activated by the Rectified Linear Unit function. Subsequently, a flatten layer and a fully connected layer with Rectified Linear Unit activation function are adopted in order to reduce dimensionality to 10 values. The decoder mirrors the architecture of the encoder: a fully connected layer followed by a reshape reorganizes the output of the decoder, furtherly processed by 4 convolutional layers with 64, 32, 16 and 1 filter respectively and a kernel size of 3. Each convolutional layer is preceded by a 2X upsampler. The parameter values were set experimentally. We then use the learned *deep* features as an input to a K-NN classifier using the Euclidean distance as a similarity measure, as provided by the open source tool Weka (Hall et al. 2009).

Experimental comparisons on classification

Our input HV time series cases were recordings of 240 samples on average, with a sampling time of 1 minute. We truncated longer series, and added zeros to extend shorter series. Our dataset was comprised of 5376 time series, belonging to 74 different patients (72 series per patient on average, varying from 1 to 280). Our classification was a binary one, where positive cases are related to an insufficient reduction of water and metabolites from the patient's blood, while negative cases are associated to HV time series whose behavior is closer the ideal model described in the Introduction. Our case base contained 3680 negative cases and 1696 positive cases. The autoencoder was defined and tested by resorting

Table 1: Results obtained by the autoencoder-based classifier vs. the DCT-based classifier

Method	Class	Precision	Recall	MCC	Accuracy
Auto	0 (positive)	0.82	0.81		
Auto	1 (negative)	0.91	0.92		
Auto	Weighted average	0.88	0.88	0.73	0.88
DCT	0 (positive)	0.77	0.10		
DCT	1 (negative)	0.70	0.98		
DCT	Weighted average	0.72	0.70	0.21	0.70

to the TensorFlow tool[1], and was run with 30 epochs for training. As regards DCT, it operates by decomposing the input into its constituent cosine waves, and returns an ordered sequence of coefficients where the most important information is concentrated at the lower indices of the sequence itself (energy compaction property) (Strang 1999). We extracted the first 10 DCT coefficient for each time series. All the experiments were performed with a 10-fold cross validation (90% of the time series for the training set, and 10% for the validation set), and we calculated the average classification performance. We realized a 9-NN classification (k=9 was the optimal parameter setting automatically calculated by Weka (Hall et al. 2009)).

Considering the autoencoder-based classification performance, we obtained an average accuracy of 88%, coupled with a Matthews Correlation Coefficient (MCC), a parameter which is particularly suitable to assess the quality of classification when dealing with unbalanced classes, very close to 1 (namely 0.73). The complete validation results are shown in table 1, which reports precision, recall, MCC and accuracy. The validation results are provided for each class and as the weighted average by class cardinality, according to the unbalanced distribution of positive and negative cases. Class 0 refers to the positive cases, while class 1 refers to the negative ones. Note that in haemodialysis false negative is more important than false positive. MCC and accuracy are not related to a single class, therefore we provide them only as overall results. As reported in the table, on the other hand, classification using the DCT coefficients provided poorer results. In particular, this model failed in identifying the positive cases, making it almost useless in a real environment. Furthermore, the very low value of the MCC suggests that this model is not far from a random predictor.

Post-hoc interpretability

To provide post-hoc interpretability of our retrieval/classification results in the *deep* feature space generated by the autoencoder, we conduct a second case based retrieval step, but working in a different feature space. Specifically, we have considered a set of features, able to describe the patient and/or the haemodialyzer settings, on the basis of medical knowledge (*medical* features henceforth). Such features are available in our database for every patient, and are "static": indeed, haemodialyzer settings tend to remain stable for many consecutive sessions, and characterize the patient in the mid-term (they may change just if haemodialysis treatment continues for many years).

[1]https://www.tensorflow.org/

Table 2: Medical features and weights

Feature	Type	Weight
Gender	symbolic	0.06
Age	numeric	0.06
Expected weight loss	numeric	1
Dialyzer preparation	symbolic	1
Arterious pressure (haemodyalizer input)	numeric	0.66
Venous pressure (haemodyalizer output)	numeric	0.66
Blood rate	numeric	0.66
Session duration	numeric	1
Medication 1	symbolic	0.66
Medication 2	symbolic	0.66

On the other hand, the *deep* features are "dynamic", as they capture the evolution of the patient's HV behavior in time, over the single session – but in a black box way. The relative importance of the different *medical* features has been elicited from medical knowledge as well, and has allowed us to pair each feature with a corresponding weight, to be used in similarity calculation. Table 2 reports the list of *medical* features and their weights.

Our basic idea for interpreting retrieval results in the *deep* feature space for a given query case consists in retrieving the most similar patients to the query patient (i.e., the patient associated to the query case) in the *medical* feature space[2]. We then verify how many time series retrieved in the *deep* feature space belong to the query patient, or to one of the retrieved patients. Obviously, the more they are, the more retrieval results in the *deep* features space can be justified (and the easier it is to justify classification results as well). Indeed, the retrieved patients define the *context for interpretation* of the query case: they share very similar clinical characteristics with respect to the query patient (as they were retrieved in the *medical* features space), and these characteristics can be shown to the end user to motivate the time series retrieval output. In order to verify the feasibility of the idea, we have conducted a first experiment on our 5376 cases, performing 9-NN retrieval in the *deep* feature space (as described in the previous section), and then 4-NN retrieval in the *medical* feature space, by adopting weighted Euclidean or overlap distance (depending on the feature type), resorting to the weights in table 2. Some of the 9 retrieved time series in the *deep* feature space could belong to the same patient; on average, distinct patients were 4 (and this motivated our choice of performing 4-NN retrieval in the *medical* feature space). We then estimated the quality of the approach on the basis of the number of retrieved time series belonging to the query patient, or to the patients retrieved in the *medical* feature space, as follows:

$$Q = \frac{\sum_{i=1}^{N} \frac{num(deep_i, medical_i)}{min(deep_i, medical_i)}}{N}$$

where N is the number of available cases (5376 in our ex-

[2]An alternative approach could be the one of adding medical features to the time series *deep* features, thus enriching the time series description. This strategy may be considered in our future work to improve classification performance, while in this paper we wanted to focus on the interpretability problem.

periment), $num(deep_i, medical_i)$ is the number of time series in the *deep* features retrieval set that belong to the query patient or to patients in the *medical* features retrieval set for query case i, and $min(deep_i, medical_i)$ is the minimum between the number of distinct retrieved patients in the *deep* features retrieval set and the number of retrieved patients (4 in our experiment) in the *medical* features retrieval set for query case i. Obviously, the larger is num, the more cases retrieved in the *deep* feature space can be easily explained on the basis of medical knowledge. In our experiments Q reached a value of 0.74, thus our method provides the possibility to explain 74% of the retrieval results obtained by exploiting the deep learning technique. It is a very encouraging result, since deep learning methods typically provide no explanation at all.

Conclusions

The contribution we have presented in this paper is twofold: (1) we have proposed a novel time series representation approach, based on *deep* features extraction by means of a convolutional autoencoder; this representation has outperformed a classical DCT-based representation in k-NN classification in the field of haemodialysis, and is domain-independent, therefore it could be possibly adopted in other applications as well; (2) we have moved some first steps towards post-hoc interpretability, in order to mitigate the well-known limit of deep learning in explaining its outputs. Specifically, we have proposed to exploit the results of a second k-NN retrieval step, conducted in the *medical* feature space, to justify the *deep* feature space retrieval results on the basis of medical knowledge.

Our experimental results have been encouraging, but there is space for improvement and further research. As regards deep learning techniques, we plan to study other architectures, in particular CNNs and LSTMs, as an alternative to autoencoders. As regards interpretability, we would like to verify whether a different value of retrieved patients in the *medical* feature space can provide a higher number of explainable time series. Moreover, the approach should be tested in different domains as well, in order to understand its generalizability, at least in the medical field, or in applications that can count on a strong and well-established domain knowledge. In all cases, a validation phase involving end users will be required, also to identify the best way to provide the explainability information. If generalizability holds, our technique could support the claim that the cooperation of data-driven AI (which includes deep learning) and knowledge-based AI represents a key direction for future AI research (Montani and Striani 2019).

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: foundational issues, methodological variations and systems approaches. *AI Communications* 7:39–59.

Bellazzi, R.; Larizza, C.; Magni, P.; and Bellazzi, R. 2005. Temporal data mining for the quality assessment of a hemodialysis service. *Artificial Intelligence in Medicine* 34:25–39.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *SIGKDD Explorations* 11(1):10–18.

Keane, M. T., and Kenny, E. 2019. How case-based reasoning explains neural networks. In Bach, K., and Marling, S., eds., *Proc. International Conference on Case Based Reasoning (ICCBR) 2019, in: Lecture Notes in Artificial Intelligence*. Springer-Verlag, Switzerland.

Kenny, E. M., and Keane, M. T. 2019. Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2708–2715. ijcai.org.

Krepel, H. P.; Nette, R. W.; Akcahuseyin, E.; Weimar, W.; and Zietse, R. 2000. Variability of relative blood volume during hemodialysis. *Nephrology, dialysis, transplantation : official publication of the European Dialysis and Transplant Association - European Renal Association* 15:673–9.

Längkvist, M.; Karlsson, L.; and Loutfi, A. 2014. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters* 42:11–24.

LeCun, Y.; Bengio, Y.; and Hinton, G. E. 2015. Deep learning. *Nature* 521(7553):436–444.

Li, O.; Liu, H.; Chen, C.; and Rudin, C. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI.

Lipton, Z. 2018. The mythos of model interpretability. *Queue* 16(3):30.

Mehdiyev, N.; Lahann, J.; Emrich, A.; Enke, D.; Fettke, P.; and Loos, P. 2017. Time series classification using deep learning for process planning: A case from the process industry. *Procedia Computer Science* 114:242 – 249.

Montani, S., and Striani, M. 2019. Artificial intelligence in clinical decision support: a focused literature survey. In Hollis, K. F.; Soualmia, L.; and Seroussi, B., eds., *IMIA Yearbook of Medical Informatics*. IMIA and Georg Thieme Verlag KG. 120–127.

Papernot, N., and McDaniel, P. D. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *CoRR* abs/1803.04765.

Sani, S.; Wiratunga, N.; Massie, S.; and Cooper, K. 2017. KNN sampling for personalised human activity recognition. In Aha, D. W., and Lieber, J., eds., *Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings*, volume 10339 of *Lecture Notes in Computer Science*, 330–344. Springer.

Strang, G. 1999. The discrete cosine transform. *SIAM Rev.* 41(1):135–147.

Wen, T., and Zhang, Z. 2018. Deep convolution neural network and autoencoders-based unsupervised feature learning of EEG signals. *IEEE Access* 6:25399–25410.