NTNU | Norwegian University of Science and Technology

TOPIC 6: FEW-SHOT LEARNING AND TRANSFER LEARNING FOR DATA SCARCITY IN TS ANALYSIS

Elen Ekeberg Klippen

Introduction - Paper

Towards a Universal Neural Network Encoder for Time Series

Joan Serrà [a,1], Santiago Pascual [b] and Alexandros Karatzoglou [a]
[a] *Telefónica Research, Barcelona*
[b] *Universitat Politècnica de Catalunya, Barcelona*

Introduction - Motivation

Processing

- **variable lengths**
- **high dimensional inputs**
- **few labeled data.**

Generalization

Generalization of learned representations to unseen data types.

Introduction - Related Work

Multi-task learning

- uses shared representation that is learnt in parallell across several tasks, including the target task
- requires that target data is available and labeled. Can also be unfeasible to re-train in parallell with all new datasets every time we find a new target task

Transfer learning

- a pre-trained model is adapted to a new target task with less effort and better results than training from scratch
- assumes labeled data for the target task

Introduction - Related work

- seq2seq autoencoders
- Deep Neural Networks
- Ensemble approaches with multiple classifiers, features and distances
  - COTE
  - HIVE-COTE

Objective - What do they try to do?

To develop and train an encoder network that converts variable-length time series to a fixed length, low-dimensional representation. Moreover (and importantly), they want the learned representations to generalise to unseen data types.

Methods - Encoder

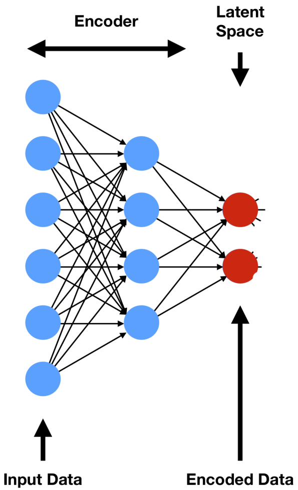

Figure: Illustration of an example encoder.

Proposed Solution - Architecture

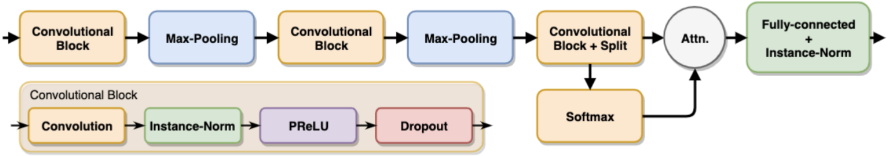

Figure: Architecture diagram of the proposed encoder (top) and the convolutional block (bottom left).

Proposed Solution - CNN

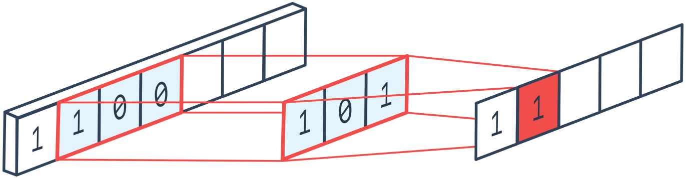

Figure: Example of 1-dimensional convolution block illustration.

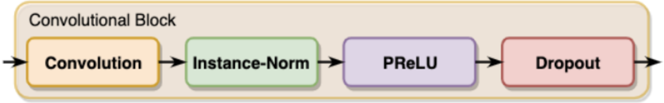

Figure: The convolution block used in the proposed solution.

Proposed Solution - Architecture

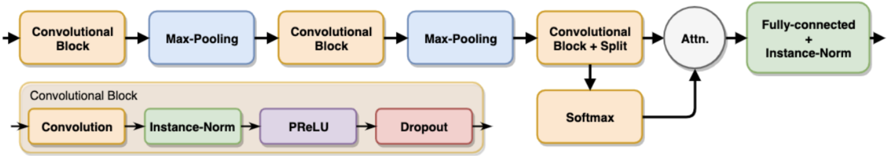

Figure: Architecture diagram of the proposed encoder (top) and the convolutional block (bottom left).

Proposed Solution - Implementation details

Encoder:

- Number of filters: respectively 128, 256 and 512 for the three convolutional layers.
- Dropout: 0.2 in all layers
- Half of the 512 filters in the last convolutional layer are input to the softmax layer and later used to compute the filterwise dot-product with the remaining half

Training:

- using PyTorch version 0.3.1
- Titan Xp GPU
- Update weights of network: stochastic gradient descent with lr=0.005
- batch size = 12

Experimental setup - Training

- 85 datasets from UEA/UCR time series classification repository
- split into train/test according to data type
- There are 7 datatypes. Hence, they follow a 7-fold training procedure where they leave out all data sets corresponding to one data type for testing and split the rest into train/validation (80/20%).

Experimental setup - Encoder adaption

To check the goodness of the learned representation, they have three approaches of encoder adaption.

1. No adaption: consider performance of a 1NN classifier.
2. Partial adaption: consider the performance of two classifiers: a logistic regression classifier and a SVM classifier.
3. Full adaption: for the third approach they adapt both the encoder and the mapping to the new target data. They fine-tune the pre-trained encoder together with a fully connected layer with softmax activation.

Result - Result

Approach	$\bar{A}$	$\bar{R}$	Rank	Wins
Euclidean-1NN	70.9	0.504	29.7	1
DTW-Rn-1NN	75.9	0.580	23.4	2
TWE-1NN	76.4	0.580	22.4	3
Encoder-1NN	**76.5**	**0.599**	**22.7**	**2**
MSM-1NN	77.3	0.593	20.1	2
RotF	77.6	0.608	17.8	6
Encoder-LR	**79.8**	**0.650**	**17.3**	**5**
Encoder-SVM	**80.3**	**0.667**	**15.6**	**5**
BOSS	81.0	0.676	14.3	15
Encoder-NEW	**81.3**	**0.682**	**11.9**	**16**
ST	82.2	0.694	11.9	17
Encoder-ADAPT	**82.9**	**0.708**	**8.7**	**26**
COTE	83.8	0.715	7.7	18

Table 1. Average performance of selected approaches. Values are computed by considering the original single splits of all the 85 data sets and 36 baselines of the UCR/UEA repository, together with the encoder-based approaches. However, due to space constrains, we do not show all baselines and individual data set values. The encoder-based classifiers use $k = 256$.

Conclusion - Conclusion

- The use of an univeral encoder for time series classification of out-of-sample data gives competitive results for all three cases (no adaption, mapping adaption and full adaption).

Thank you for your attention

NTNU | Norwegian University of Science and Technology