

Meta-Learning for Few-Shot Time Series Classification

Presented by Erik Olsvik Dengerud

arXiv:1909.07155v2 [cs.LG] 25 Sep 2019

Meta-Learning for Few-Shot Time Series Classification

Jyoti Narwariya, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Vishnu TV
jyoti.narwariya@tcs.com, malhotra.pankaj@tcs.com, lovekesh.vig@tcs.com, gautam.shroff@tcs.com, vishnu.tv@tcs.com
TCS Research
New Delhi, India

ABSTRACT

Deep neural networks (DNNs) have achieved state-of-the-art results on time series classification (TSC) tasks. In this work, we focus on leveraging DNNs in the often-encountered practical scenario where access to labeled training data is difficult, and where DNNs would be prone to overfitting. We leverage recent advancements in gradient-based meta-learning, and propose an approach to train a residual neural network with convolutional layers as a meta-learning agent for few-shot TSC. The network is trained on a diverse set of few-shot tasks sampled from various domains (e.g. healthcare, activity recognition, etc.) such that it can solve a target task from another domain using only a small number of training samples from the target task. Most existing meta-learning approaches are limited in practice as they assume a fixed number of target classes across tasks. We overcome this limitation in order to train a common agent across domains with each domain having different number of target classes, we utilize a triplet-loss based learning procedure that does not require any constraints to be enforced on the number of classes for the few-shot TSC tasks. To the best of our knowledge, we are the first to use meta-learning based pre-training for TSC. Our approach sets a new benchmark for few-shot TSC, outperforming several strong baselines on few-shot tasks sampled from 41 datasets in UCR TSC Archive. We observe that pre-training under the meta-learning paradigm allows the network to quickly adapt to new unseen tasks with small number of labeled instances.

KEYWORDS

Time Series Classification, Meta-Learning, Few-Shot Learning, Convolutional Neural Networks

1 INTRODUCTION

Time series data is ubiquitous in the current digital era with several applications across domains such as forecasting, healthcare, equipment health monitoring, and meteorology among others. Time series classification (TSC) has several practical applications such as disease diagnosis from time series of physiological parameters [4], classifying heart arrhythmias from ECG signals [28], and human activity recognition [43]. Recently, deep neural networks (DNNs) such as those based on long short term memory networks (LSTMs) [17] and 1-dimensional convolution neural networks (CNNs) [9, 18, 40] have achieved state-of-the-art results on TSC tasks. However, it is well-known that DNNs are prone to overfitting, especially when access to a large labeled training dataset is not available. [10, 18].

Few recent attempts aim to address the issue of scarce labeled data for univariate TSC (UTSC) by leveraging transfer learning

[44] via DNNs, e.g. [10, 18, 22, 36]. These approaches consider pre-training a deep network in an unsupervised [22] or supervised [10, 18, 36] manner using a large number of time series from diverse domains, and then fine-tune the pre-trained model for the target task using labeled data from target domain. However, these transfer learning approaches for TSC based on pre-training a network on large number of diverse time series tasks do not necessarily guarantee a pre-trained model (or network initialization) that can be quickly fine-tuned with a very small number of labeled training instances, and rather rely on ad-hoc fine-tuning procedures.

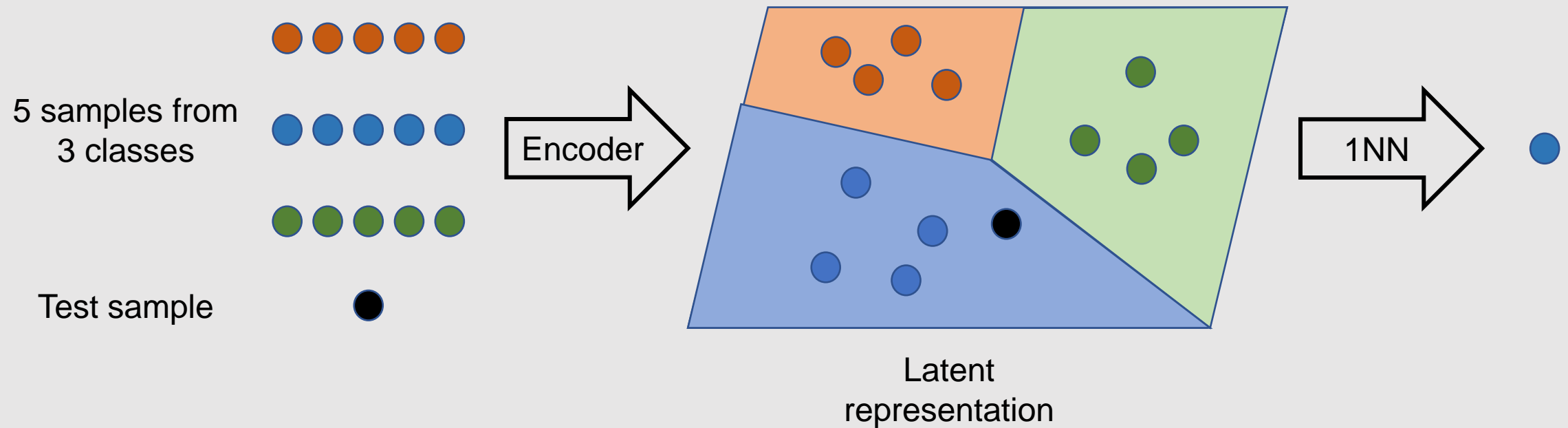
Rather than learning a new task from scratch, humans leverage their pre-existing skills by fine-tuning and recombining them, and hence are highly data-efficient, i.e. can learn from as little as one example per category [27]. Meta-learning [34] approaches intend to take a similar approach for *few-shot learning*, i.e. learning a task from few examples. More recently, several approaches for few-shot learning for regression, image classification, and reinforcement learning domains have been proposed under the gradient-based meta-learning or the “learning to learn” framework, e.g. in [11, 24, 30]. A neural network-based meta-learning model is explicitly trained to quickly learn a new task from a small amount of data. The model learns to solve several tasks sampled from a given distribution where each task is, for example, an image classification problem with few labeled examples. Since each task corresponds to a learning problem, performing well on a task corresponds to learning quickly.

Despite the advent of aforementioned pre-trained models for time series, *few-shot learning* (i.e. learning from few, say five, examples per class) for TSC remains an important and unaddressed research problem. The goal of few-shot TSC is to train a model on large number of diverse few-shot TSC tasks such that it can leverage this experience through the learned parameters, and quickly generalize to new tasks with small number of labeled instances. More specifically, we train a residual network (ResNet) [9, 40] on several few-shot TSC tasks such that the ResNet thus obtained generalizes to solve new few-shot learning tasks. In contrast to existing methods for data-efficient transfer learning, our method provides a way to directly optimize the embedding itself for classification, rather than an intermediate bottleneck layer such as the ones proposed in [22, 36].

Key contributions of this work are:

- We define the problem of few-shot learning for univariate TSC (UTSC), and propose a training and evaluation protocol for the same.
- We propose a few-shot UTSC approach by training a ResNet to solve diverse few-shot UTSC tasks using a meta-learning procedure [24]. The ResNet thus obtained can be quickly adjusted (fine-tuned) on a new, previously unseen, UTSC task with few labeled examples per class.

The **goal** is to classify time series with **limited** access to **labeled** data



Towards a Universal Neural Network Encoder for Time Series

Juan Serra^{*1}, Santiago Pascual[†] and Alexandros Karatzoglou^{*}
^{*}Telefonica Research, Barcelona
[†]Universitat Politècnica de Catalunya, Barcelona

Abstract. We study the use of a time series encoder to learn useful on data set types with which it has not been trained. We propose a convolutional neural network architecture that can learn to represent time series data from a wide range of domains.

FaceNet: A Unified Embedding for Face Recognition and Clustering

Florian Schroff
fschroff@google.com
Google Inc.

Dmitry Kalenichenko
dkalenichenko@google.com
Google Inc.

James Philbin
jphilbin@google.com
Google Inc.

Abstract

Despite significant recent advances in the field of face recognition [10, 14, 15, 17], implementing face verification and recognition efficiently at scale presents serious challenges to current approaches. In this paper we present a new system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification, and clustering can be easily implemented using standard distance metrics.



On First-Order Meta-Learning Algorithms

Alex Nichol and Joshua Achiam and John Schulman
OpenAI
{alex, jachiam, joschu}@openai.com

Abstract

This paper considers meta-learning problems, where there is a distribution of tasks, and we would like to obtain an agent that performs well (i.e., learns quickly) when presented with a previously unseen task sampled from this distribution. We analyze a family of algorithms for learning a parameter initialization that can be fine-tuned quickly on a new task, using only first-order derivatives for the meta-learning updates. This family includes and generalizes first-order MAML, an approximation to MAML obtained by ignoring second-order derivatives. It also includes Reptile, a new algorithm that we introduce here, which works by repeatedly sampling a task, training on it, and moving the initialization towards the trained weights on that task. We expand on the results from Finn et al. showing that first-order meta-learning algorithms perform well on some well-established benchmarks for few-shot classification, and we provide theoretical analysis aimed at understanding why these algorithms work.

1 Introduction

While machine learning systems have surpassed humans at many tasks, they generally need far more data to reach the same level of performance. For example, Schmidt et al. [17, 15] showed that human subjects can recognize new object categories based on a few example images. Lake et al. [12] noted that on the Atari game of Frostbite, human novices were able to make significant progress on the game after 15 minutes, but double-dueling-DQN [19] required more than 1000 times more experience to attain the same score.

It is not completely fair to compare humans to algorithms learning from scratch, since humans enter the task with a large amount of prior knowledge, encoded in their brains and DNA. Rather than learning from scratch, they are fine-tuning and recombining a set of pre-existing skills. The work cited above, by Tenenbaum and collaborators, argues that humans' fast-learning abilities can be explained as Bayesian inference, and that the key to developing algorithms with human-level learning speed is to make our algorithms more Bayesian. However, in practice, it is challenging to develop (from first principles) Bayesian machine learning algorithms that make use of deep neural networks and are computationally feasible.

Meta-learning has emerged recently as an approach for learning from small amounts of data.

Towards a Universal Neural Network Encoder for Time Series (2018)

Pre-training: Training an encoder on many datasets.

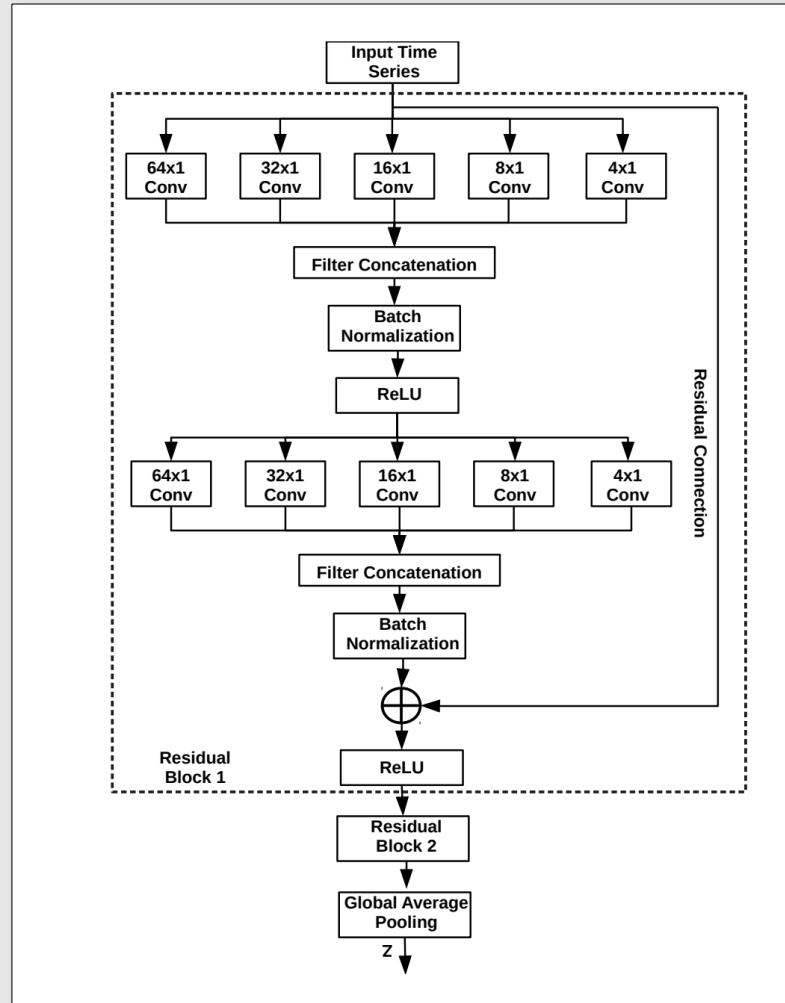
FaceNet (2015)

Triplet loss: Optimizing the latent embeddings directly.

On First-Order Meta-Learning Algorithms (2018)

Gradient descent based meta-learning: Learning to learn few shot tasks.

ResNet is a **successful** architecture for time series classification



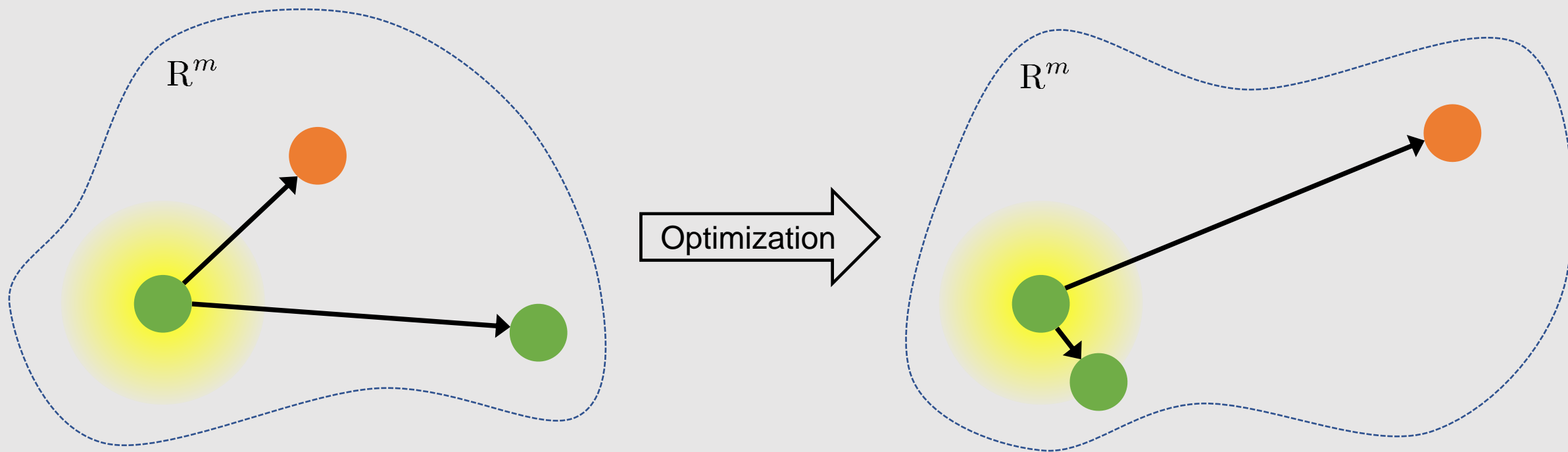
Time series classification from scratch with deep neural networks: A strong baseline (Wang, Z., Yan, W., & Oates, T. , 2017)

Towards a Universal Neural Network Encoder for Time Series (Serrà, J., Pascual, S., & Karatzoglou, A. , 2018).

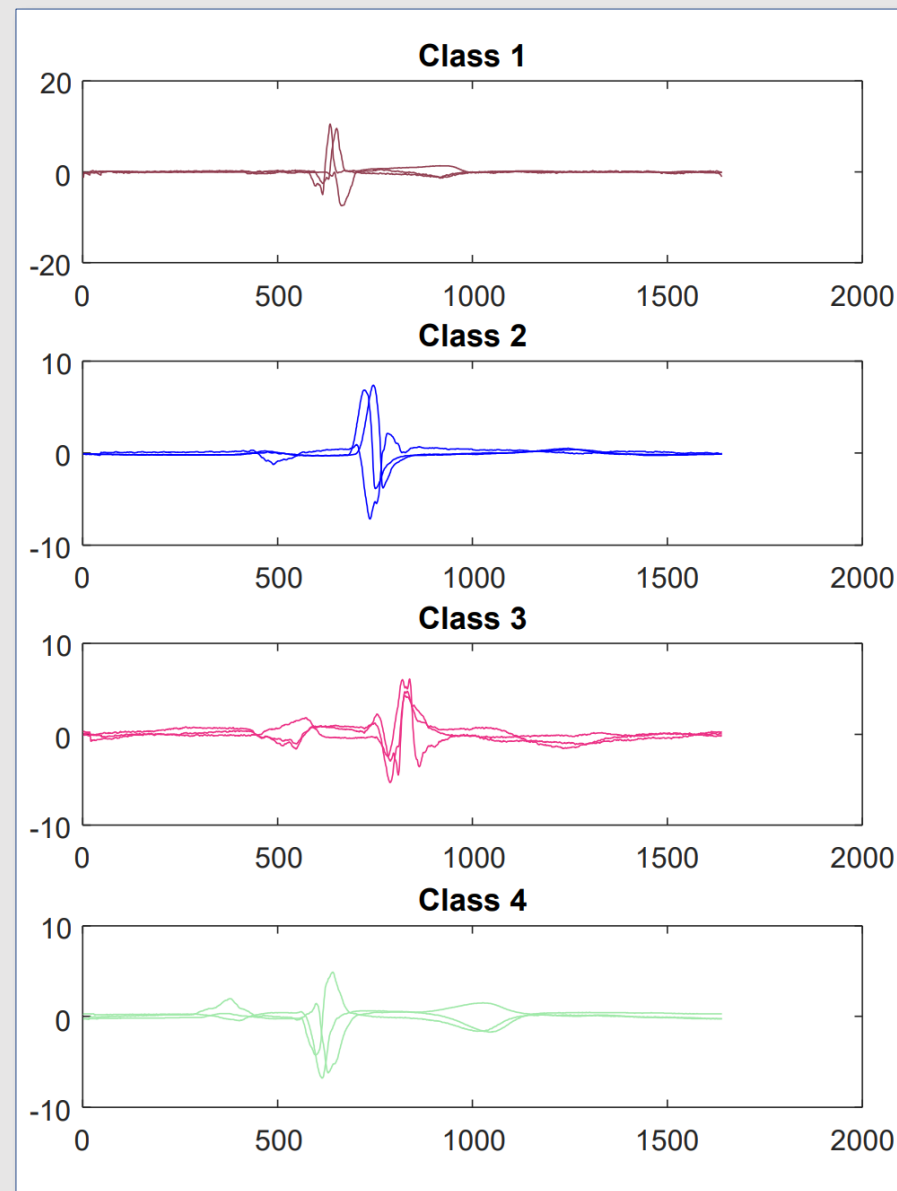
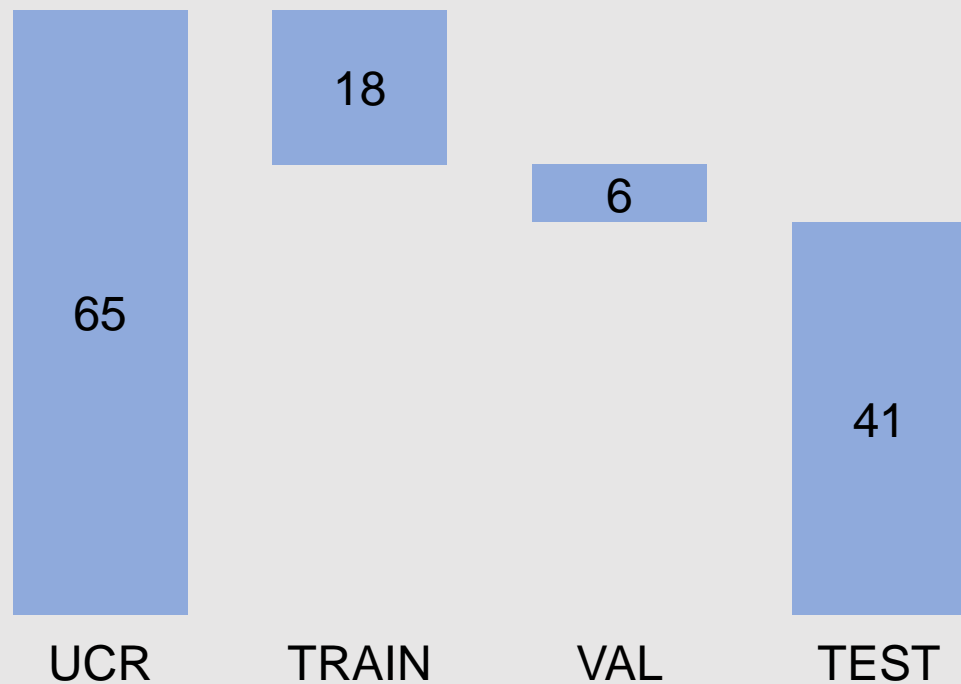
ConvTimeNet: A pre-trained deep convolutional neural network for time series classification (Kashiparekh, K., Narwariya, J., Malhotra, P., Vig, L., & Shroff, G. ,2019)

The triplet loss **optimize** the latent **embedding** directly

Triples of one anchor , one of same class , and one of different class .

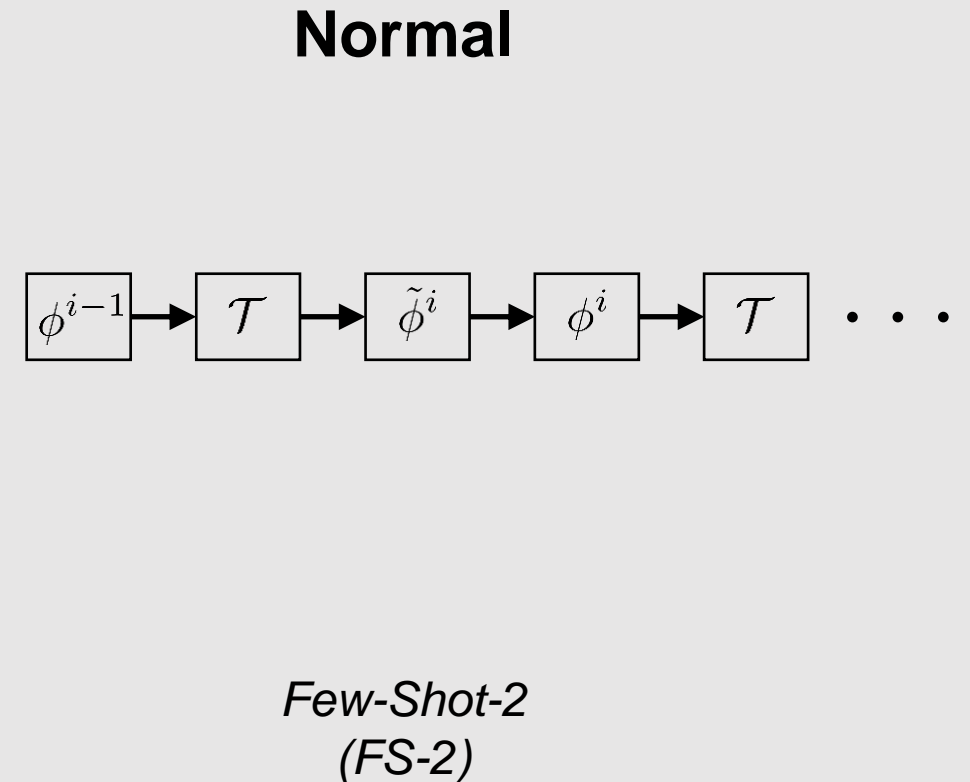
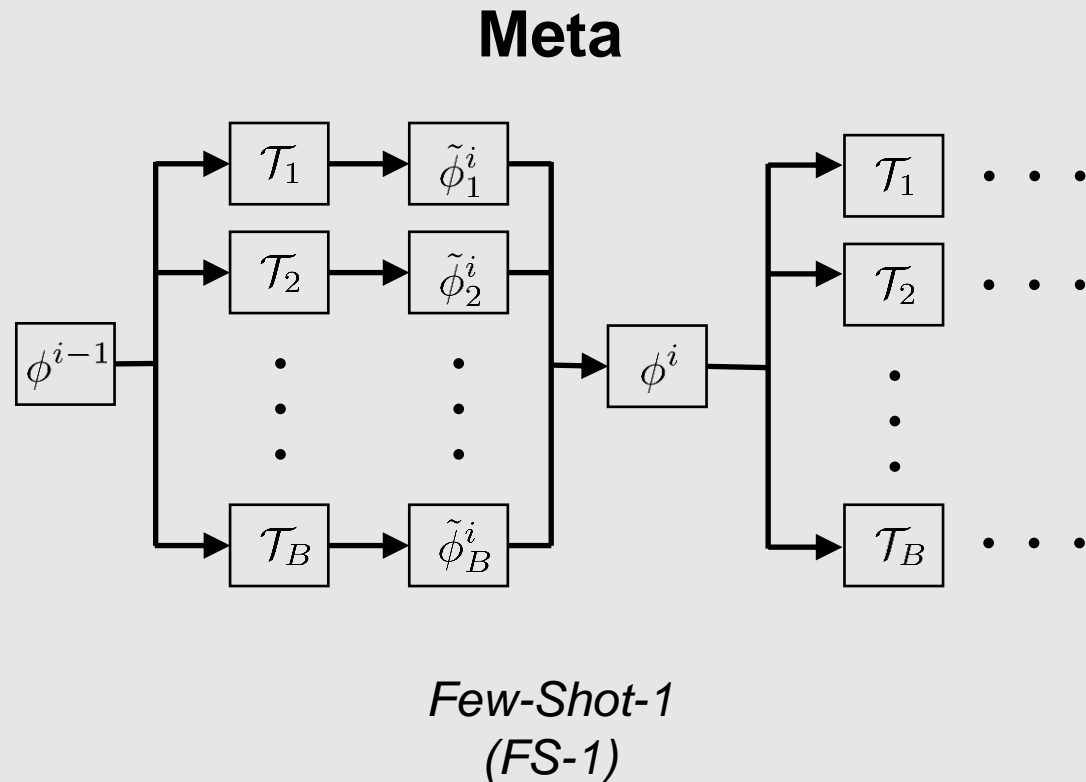


The UCR archive



Gradient descent based meta-learning

Optimization: $\tilde{\phi}_j^i = U_{\mathcal{T}_j}^k(\phi_j^i)$



Fine-tuning and inference on specific datasets

Fine-tuning

A new ResNet is initialized with the weights from pre-training.

Fine-tuning on the train split of the specific dataset.

Inference

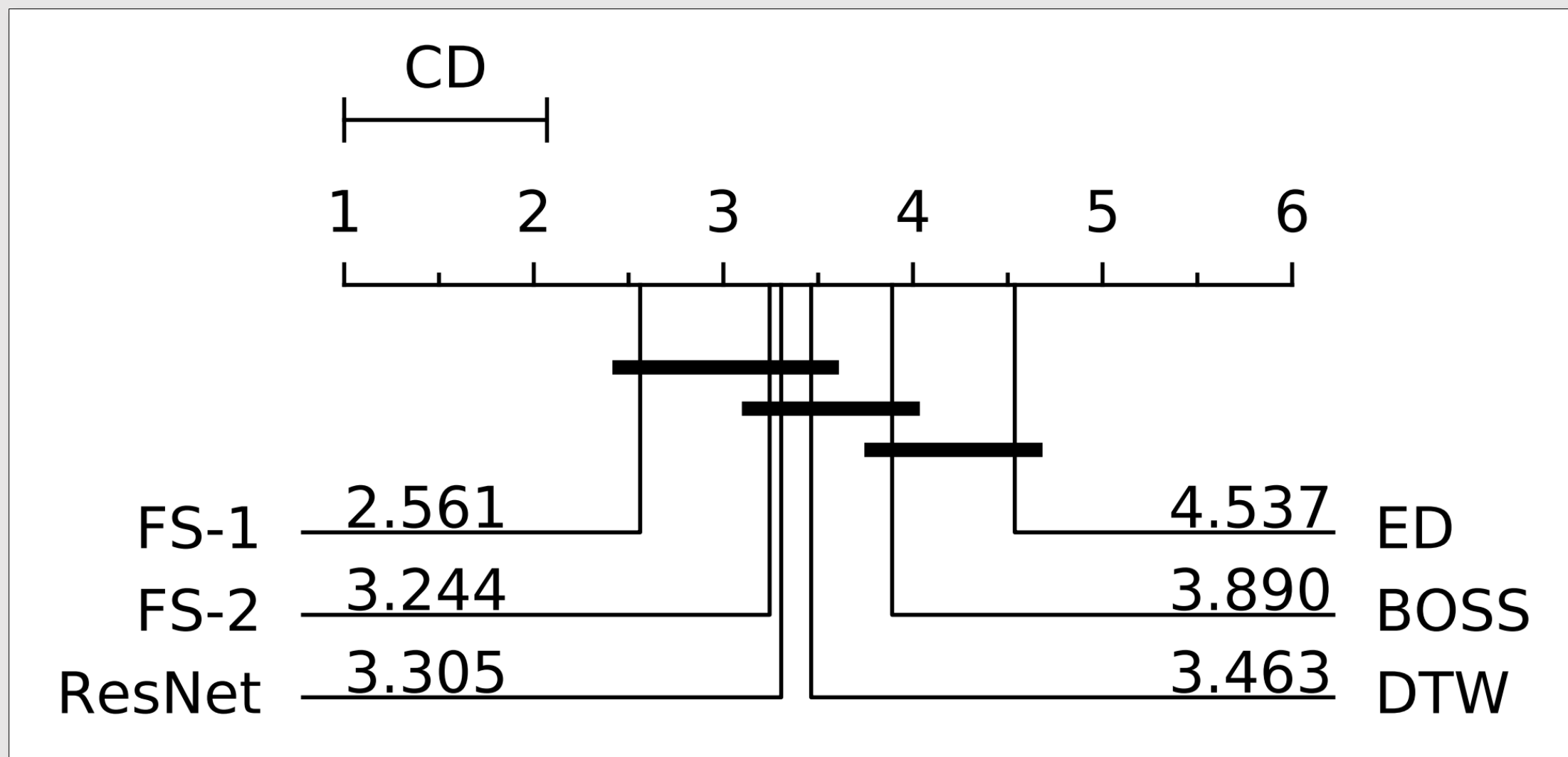
Encode the K time series from the train split of the dataset.

Use an 1NN classifier for each new embedded time series in the test split.

The method outperforms other typical representations

K	ED	DTW	BOSS	ResNet	FS-2	FS-1
2	4.232	<u>2.976</u>	3.902	3.805	3.207	2.878
5	4.537	3.463	3.890	3.305	<u>3.244</u>	2.561
10	4.573	3.476	3.646	3.683	<u>3.427</u>	2.195
20	4.439	3.354	<u>2.927</u>	3.902	3.793	2.585

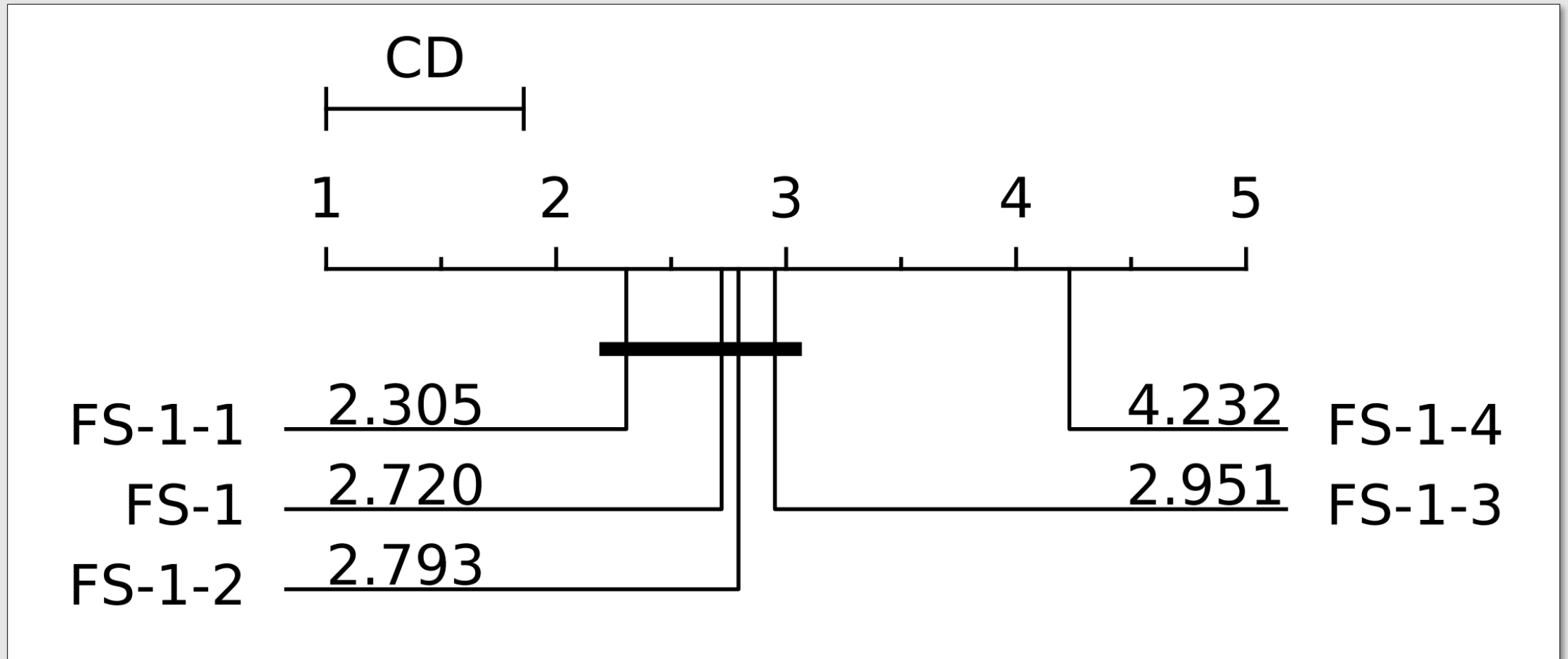
It is not significantly better than DTW



Performance is dependent on the **number** of classes

N	n	ED	DTW	BOSS	ResNet	FS-2	FS-1
2-5	24	4.167	4.083	3.375	3.458	<u>3.042</u>	2.875
6-10	9	4.778	2.333	5.333	<u>2.389</u>	3.778	<u>2.389</u>
>10	8	5.375	2.875	3.812	3.902	3.875	1.812
Overall	41	4.537	3.463	3.890	3.305	<u>3.244</u>	2.561

Freezing the first **layer** during fine tuning increase the performance



Important aspects of the paper

1. Triplet loss optimizes the latent embeddings directly and removes the need for a multi-head network.
2. Gradient descent based meta-learning learns to learn tasks and is different from “normal” learning.
3. Pre-training makes the network generalize better.

