Few-Shot Time Series Classification

Ninduwara K.G.M.
Department of Computer Science and Engineering
University of Moratuwa
Colombo, Sri Lanka
maneth.21@cse.mrt.ac.lk

I. METHODOLOGY

The proposed approach extends the Temporal Neighborhood Coding (TNC) framework by incorporating few-shot classification strategies. The methodology comprises two principal phases: self-supervised representation learning through pre-training, followed by few-shot classification using both baseline and enhanced methods. This two-stage pipeline enables the evaluation of learned representations under realistic low-data scenarios.

A. *Self-Supervised Representation Learning*

The first phase focuses on learning robust temporal representations without supervision through contrastive learning. This pre-training stage establishes the foundation for subsequent few-shot classification tasks.

*1) **Dataset Generation**:* The dataset was synthetically generated to emulate long, non-stationary multivariate time series commonly encountered in real-world applications. Each sequence contained 2000 measurements with three features, governed by four latent states sampled from a Hidden Markov Model (HMM). Within each state, data were generated using either a Gaussian Process with different kernels or a Nonlinear Auto-Regressive Moving Average (NARMA) model. To better resemble realistic signals, two features were maintained as correlated throughout the sequence, introducing dependencies that challenge representation learning.

*2) **Encoder Training**:* The encoder was optimized using the contrastive objective from the TNC framework. The model learned to distinguish temporally neighboring segments (positive pairs) from distant segments (negative pairs), where neighborhoods were dynamically defined using the Augmented Dickey-Fuller (ADF) test for stationarity detection. A bi-directional single-layer RNN encoder was employed to capture temporal dependencies in both forward and backward directions. The architecture encoded windows of size 50 into 10-dimensional representation vectors, carefully balancing the trade-off between capturing sufficient state information and maintaining temporal locality.

*3) **Pre-training Outcome**:* After training convergence, the encoder produced embeddings that clearly separated the underlying signal types, achieving a class separation ratio of 1.648. This metric indicates strong discriminative power in the learned representation space. The encoder weights were subsequently frozen and utilized for all downstream few-shot classification experiments, ensuring consistent feature extraction across different methods.

B. *Few-Shot Classification*

To evaluate the quality and transferability of the learned TNC representations, a comprehensive few-shot classification framework was designed. The primary objective was to assess the encoder's ability to generalize to new classification tasks with minimal labeled data, which is central to the promise of self-supervised representation learning. Seven distinct classification approaches were implemented, ranging from simple baselines to sophisticated meta-learning techniques.

*1) **Baseline Methods**:* Three baseline methods were employed to establish performance benchmarks and evaluate fundamental properties of the learned embedding space.

*a) **Linear Classifier**:* A logistic regression model was trained directly on the frozen TNC feature representations, serving as a baseline for assessing linear separability of the embeddings. The model employed scikit-learn's default L2 regularization with regularization strength C=1.0 to prevent overfitting in low-data regimes. Training was performed using the Limited-memory BFGS (L-BFGS) optimizer with a maximum of 1000 iterations, which ensures stable convergence with small datasets and provides insights into the linear structure of the learned embedding space. Each trial used a different random seed to ensure reproducible yet varied sampling of the few-shot support sets.

*b) **k-Nearest Neighbors**:* A non-parametric classifier was implemented where labels are assigned based on the majority vote of the k nearest neighbors in the TNC feature space. Euclidean distance in the 10-dimensional embedding space was used as the similarity metric, with the optimal value of k determined through cross-validation. This approach evaluates whether semantically similar time series naturally cluster together in the learned representation space without requiring any model training.

*c) **Standard Prototypical Networks**:* This baseline few-shot method computes class prototypes as the mean of support set embeddings for each class. Query samples are then classified based on their Euclidean distance to the nearest prototype. This approach directly tests the natural clustering properties of the TNC feature space without requiring any task-specific adaptation or fine-tuning.

*2) **Enhanced Prototypical Methods**:* To leverage the characteristics of TNC representations more effectively, four en-

hanced prototypical methods were developed. These methods introduce learnable components that adapt to the specific properties of temporal embeddings.

*a) **Linear Prototypical Networks**:* This method introduces a learnable two-layer neural transformation that projects TNC features into an optimized prototype space before computing class centroids. The transformation consists of sequential linear layers with ReLU activation and dropout regularization, using Xavier normal initialization for stable training. The network learns to transform features into a representation space where Euclidean distances between query examples and class prototypes are more discriminative, exploiting the linear separability of TNC embeddings through learned feature transformation with a fixed temperature parameter for logit conversion.

*b) **Metric Prototypical Networks**:* Rather than relying on fixed Euclidean distance, this approach employs a learnable neural network that computes distance scores between query samples and class prototypes. The network comprises three fully connected layers with ReLU activation and dropout regularization, followed by a sigmoid activation producing distance values between 0 and 1. The architecture concatenates query and prototype features as input, enabling task-specific distance modeling that can capture non-linear relationships in the embedding space through learned distance metrics rather than similarity scores.

*c) **Hybrid Few-Shot Classifier**:* This method combines prototypical and linear classifiers through a learnable parameter-based fusion mechanism. The fusion uses a single sigmoid-activated parameter for static weighted combination of both approaches. The linear branch employs a two-layer neural network with ReLU activation and dropout, while the prototypical branch uses feature transformation before computing distance-based logits with learnable temperature. This hybrid approach leverages complementary strengths through fixed weighted fusion.

*d) **Adaptive Prototypical Networks**:* This method dynamically adjusts prototype computation and feature transformation based on the number of available shots. The approach incorporates multiple learnable linear transformations with ReLU activation and dropout regularization, using different transformation pathways for varying shot numbers. Xavier normal weight initialization ensures stable training, and the method adapts its architectural complexity based on shot availability rather than modulating reliance on individual examples versus class statistics during prototype formation.

*3) **Experimental Protocol**:* All methods were evaluated using the standard N-way, K-shot episodic protocol, which simulates realistic few-shot learning scenarios. The experimental design ensures rigorous and reproducible evaluation across all classification approaches.

*a) **Task Configuration**:* The focus was placed on 4-way classification tasks, distinguishing between four distinct signal types: Periodic Gaussian Process, NARMA-5, Squared Exponential Gaussian Process, and NARMA-3. This configuration represents a challenging multi-class scenario that tests the discriminative power of learned representations.

*b) **Episode Structure**:* Each evaluation episode randomly selected 4 classes from the available signal types. For each selected class, K support examples were sampled for training and 15 query examples were sampled for testing. This structure ensures consistent evaluation across all methods while maintaining independence between support and query sets.

*c) **Data Availability Scenarios**:* Experiments were conducted across five different shot settings: 1-shot, 3-shot, 5-shot, 10-shot, and 20-shot. These variations simulate realistic scenarios with varying levels of labeled data availability. The 1-shot scenario represents the most challenging extreme low-data condition, while the 20-shot setting provides more substantial adaptation data for methods that can leverage larger support sets.

*d) **Statistical Validation**:* Each configuration was evaluated over 50 independent episodes with different random class and sample selections.

*e) **Feature Consistency**:* All methods utilized identical 10-dimensional embeddings extracted from the frozen TNC encoder. This design choice ensures that performance differences are attributable solely to the classification methods rather than variations in feature quality or extraction procedures.

*f) **Evaluation Metrics**:* Classification accuracy served as the primary performance metric, computed as the proportion of correctly classified query samples. This was complemented by confusion matrix analysis and per-class accuracy breakdowns to identify potential biases across different signal types and to understand method-specific strengths and weaknesses.

This comprehensive methodology provides a rigorous framework for assessing both the intrinsic quality of TNC representations and the effectiveness of diverse few-shot classification strategies for temporal data analysis.

II. Experimental Results

Table I presents the few-shot classification accuracy across all evaluated methods and shot configurations. The results demonstrate the effectiveness of learned TNC representations and reveal interesting trade-offs between different classification approaches.

TABLE I
FEW-SHOT CLASSIFICATION ACCURACY ON SIMULATED DATASET

Method	1-shot	3-shot	5-shot	10-shot	20-shot
Prototypical Networks	0.566	0.699	0.718	0.733	0.765
k-NN Baseline	0.357	0.621	0.659	0.704	0.725
Linear Baseline	0.555	0.691	0.721	0.751	0.778
Linear-Prototypical	0.578	0.727	0.724	0.743	0.752
Metric-Prototypical	0.587	0.729	0.774	0.769	0.772
Hybrid-Prototypical	0.580	0.691	0.740	0.727	0.756
Adaptive-Prototypical	0.531	0.672	0.705	0.719	0.648

Metric-Prototypical Networks achieved the best overall performance across most configurations, with particularly strong results in the 5-shot (0.774) and 10-shot (0.769) scenarios. The learnable similarity metric effectively captured task-specific relationships in the embedding space.