BLEND-RES^2NET: Blended Representation Space by Transformation of Residual Mapping with Restrained Learning For Time Series Classification

[1]Arijit Ukil, [2]Antonio J. Jara, [3]Leandro Marin

[1]Tata Consultancy Services, [2]HOP Ubiquitous (HOPU), [3]University of Murcia

ICASSP 2021
Paper id# 1624

Summary of the work

Problem- Insufficient training instances in time series classification task demands novel deep neural network architecture to warrant consistent and accurate performance.

Solution-

- We propose Blend-Res^{2}Net that blends two different representation spaces: $\mathcal{H}^1(x) = \mathcal{F}(x) + Trans(x)$ and $\mathcal{H}^2(x) = \mathcal{F}\big(Trans(x)\big) + x$.
- Deep network complexity is adapted by proposed novel restrained learning, which introduces dynamic estimation of the network depth.

Results-

- Blend-Res^{2}Net is demonstrated by a series of ablation experiments over publicly available benchmark time series archive- UCR.
- Blend-Res^{2}Net outperforms baselines and state-of-the-art algorithms including 1-NN-DTW, HIVE-COTE, ResNet, InceptionTime, ROCKET, DMS-CNN, TS-Chief

Blend-Res2Net: Algorithms and architecture

Blend-Res^{2}Net consists of three distinct components-

1. *Transformation of the residual channel:* The transformation of residual channel incorporates spectral (Discrete Fourier co-efficients) signature in the residual mapping.
2. *Blending of learning channels:* Two parallelly constructed transformed residual networks are merged together to enable intricate learning model.
3. *Restrained learning:* Restrained learning dynamically limits the network complexity by incorporating elasticity of the residual block depth through dynamic layer configuration.

Transformation of the residual channel

Case 1: Construction of $\mathcal{H}^1(x)$

For the simplicity of explanation, we consider $L = 2$.

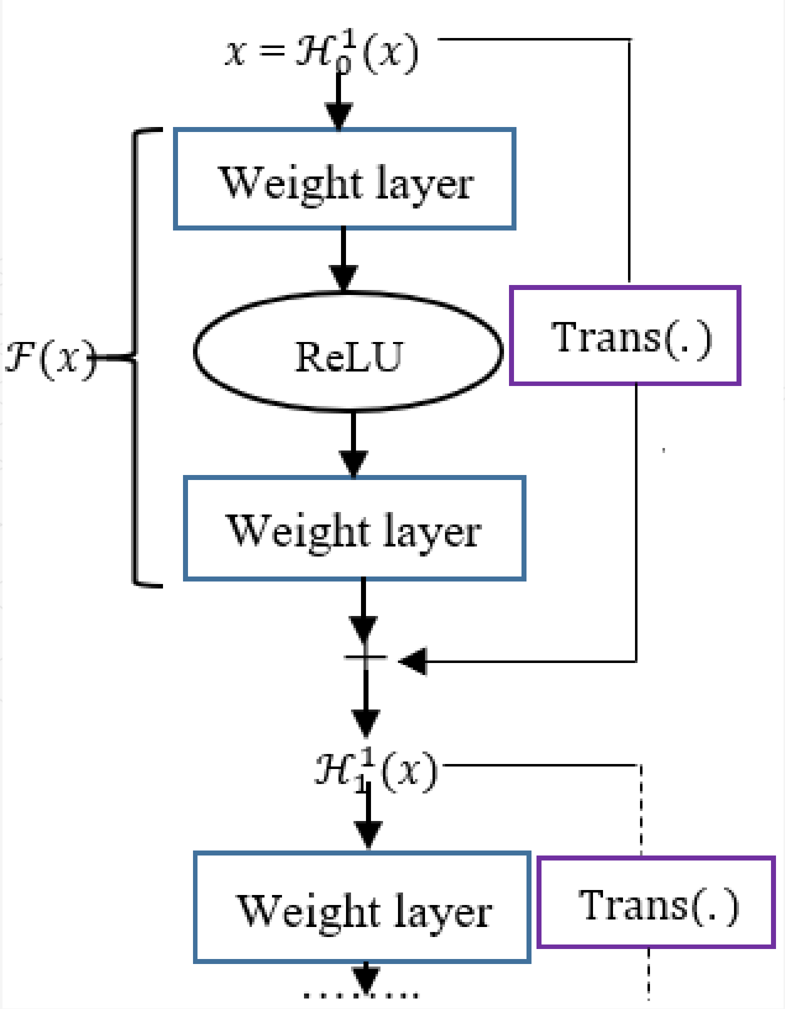
$$\mathcal{H}_1^1(x) = Trans(x) + \mathcal{F}(x)$$

$$\mathcal{H}_2^1(x) = \mathcal{H}_1^1(x) + \mathcal{F}\left(\mathcal{H}_1^1(x)\right)$$

$$= \underbrace{Trans(x)}_{\substack{Direct\\channel\\transformed\\representation}} + \underbrace{\mathcal{F}(x)}_{\substack{Non-linear\\channel\\temporal\\representation}} + \underbrace{\mathcal{F}\left(Trans(x) + \mathcal{F}(x)\right)}_{\substack{Non-linear\ channel\\mixed-representation}} \quad (2)$$

Let us assume X_D be the Discrete Fourier Transform of x, $x \leftrightarrow X_D$ and the real-part of X_D be $\Re(X_D)$. From equation (2), we get (where, $Trans(x) = \Re(X_D)$):

$$\mathcal{H}_2^1(x) = \Re(X_D) + \mathcal{F}(x) + \mathcal{F}\left(\Re(X_D) + \mathcal{F}(x)\right) \quad (3)$$



Transformation of the residual channel

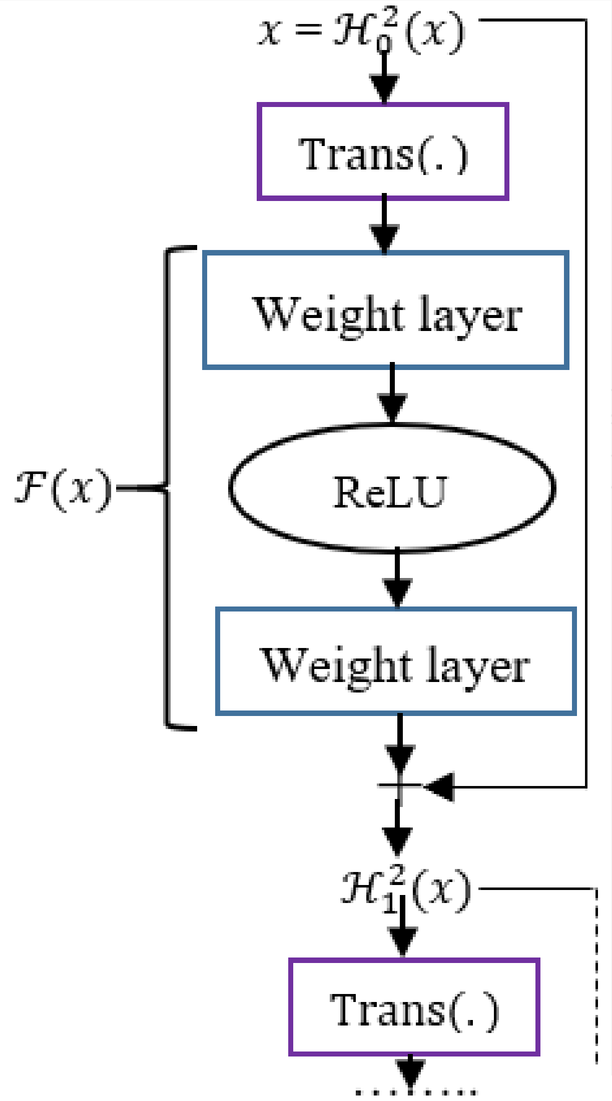
Case 2: Construction of $\mathcal{H}^2(x)$

$$\mathcal{H}_1^2(x) = x + \mathcal{F}(Trans(x))$$

$$\mathcal{H}_2^2(x) = \mathcal{H}_1^2(x) + \mathcal{F}\left(\mathcal{H}_1^2(x)\right)$$

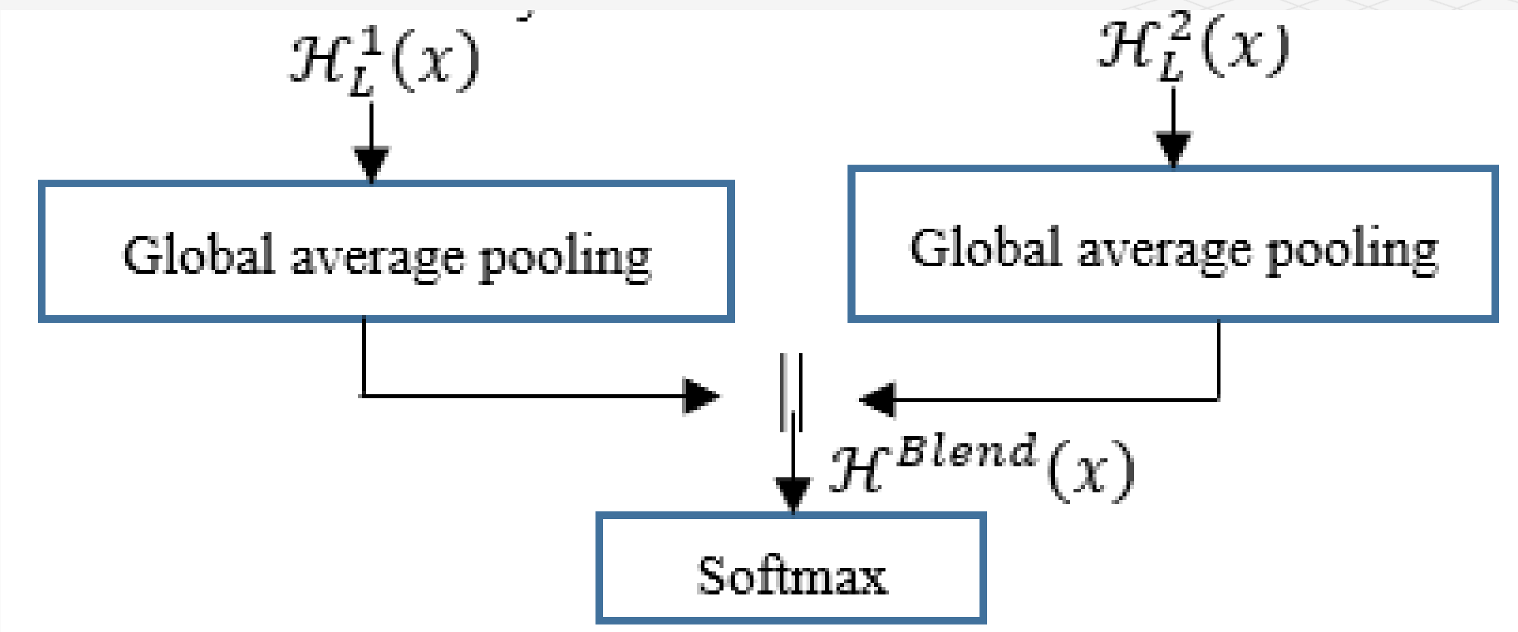
$$= \underbrace{x}_{\substack{Direct\\ channel\\ temporal\\ representation}} + \underbrace{\mathcal{F}(Trans(x))}_{\substack{Non-linear\\ channel\\ transformed\\ representation}} + \underbrace{\mathcal{F}\left(x + \mathcal{F}(Trans(x))\right)}_{\substack{Non-linear\ channel\\ mixed-representation}}$$

$$\mathcal{H}_2^2(x) = \underbrace{x}_{\substack{Direct\\ channel\\ temporal\\ representation}} + \underbrace{\mathcal{F}(\Re(X_D))}_{\substack{Non-linear\\ channel\\ transformed\\ representation}} + \underbrace{\mathcal{F}\left(x + \mathcal{F}\left(\Re(X_D)\right)\right)}_{\substack{Non-linear\ channel\\ mixed-representation}} \quad (4)$$



Blending of learning channels

We construct the blended representation $\mathcal{H}^{Blend}(x)$ by concatenating (||) $\mathcal{H}_L^1$ and $\mathcal{H}_L^2$



Restrained Learning by Estimating Network Depth

Algorithm: Estimating network depth (Demonstrating for $\mathcal{X}_{Train}^{1}$)

Input: $\mathcal{X}_{Train}^{Universe}$, $\mathcal{X}_{Train}^{1}$, maximum allowed residual blocks δ_{max} and minimum residual blocks δ_{min}

Output: $\delta_{Final}(\mathcal{X}_{Train}^{1})$

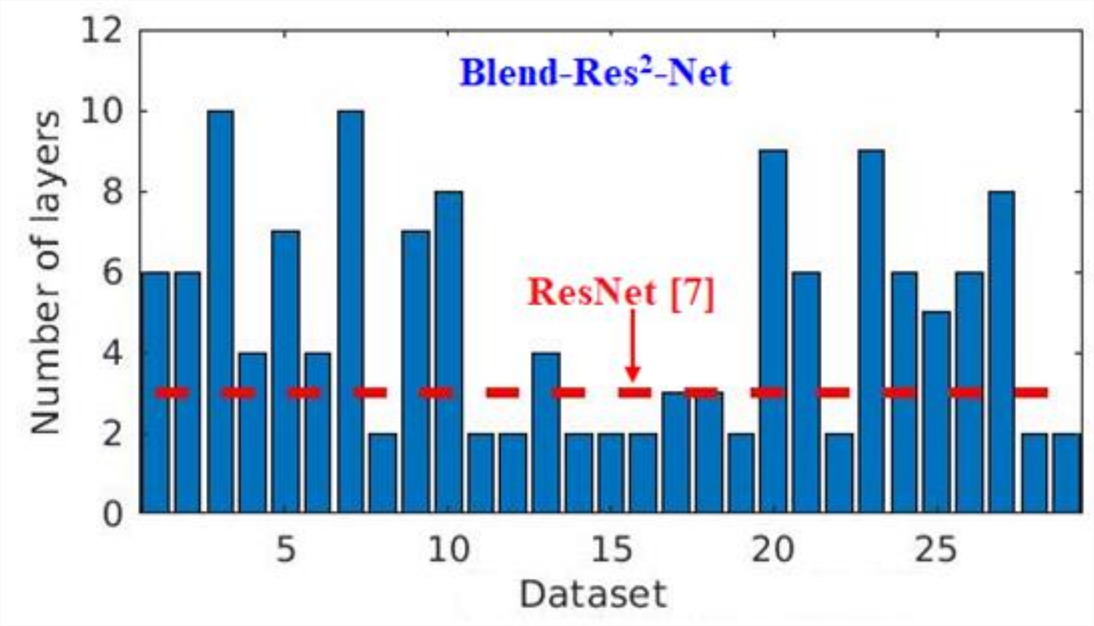
Method:

1. $\alpha^{Universe} = max\left(Entropy\left(\mathcal{X}_{Train}^{Universe}\right)\right)$
2. $\alpha^{1} = Entropy\left(\mathcal{X}_{Train}^{1}\right)$, $\alpha^{1\_normalized} = \frac{\alpha^{1}}{\alpha^{Universe}}$
3. $\beta^{Universe} = max\left(\left|\mathcal{X}_{Train}^{Universe}\right|\right)$
4. $\beta^{1} = \left|\mathcal{X}_{Train}^{1}\right|$, $\beta^{1\_normalized} = \frac{\beta^{1}}{\beta^{Universe}}$ (The normalization factors are set to 1 in the absence of other types of training datasets other than $\mathcal{X}_{Train}^{1}$ or when $\mathcal{X}_{Train}^{Universe} = \mathcal{X}_{Train}^{1}$).
5. $\delta_{Final}\left(\mathcal{X}_{Train}^{1}\right) = min\left(\delta_{max},\ max\left(\delta_{min}, \left\lceil \alpha^{1\_normalized} \times \beta^{1\_normalized} \right\rceil\right)\right)$

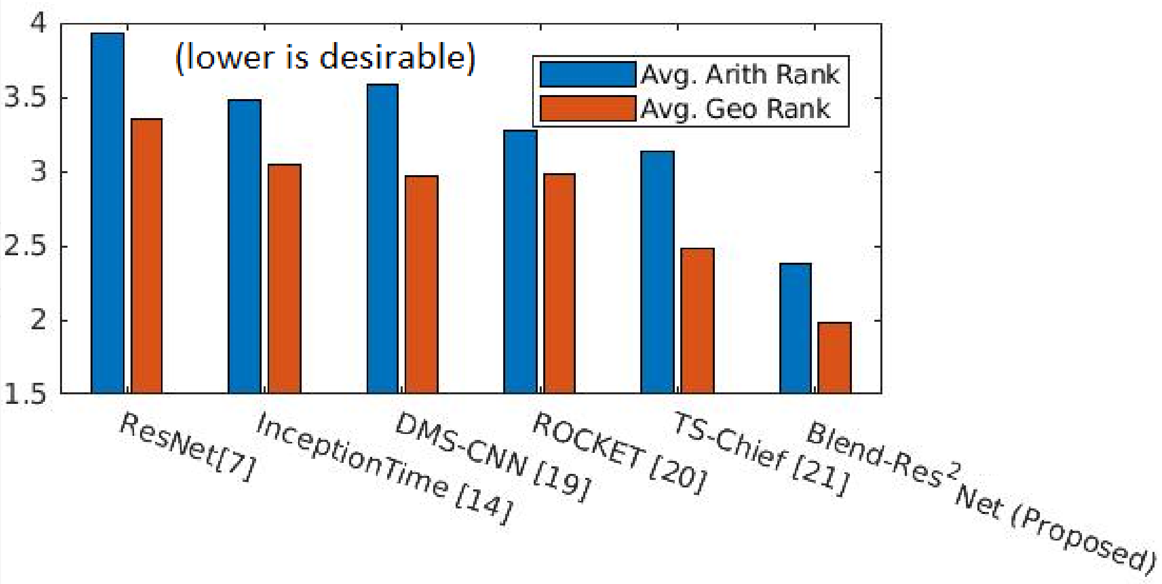
Results

Dataset	1-NN DTW [6, 8]	HIVE - COTE [11]	ResNet [7]	Inception Time [14]	~~Blend-~~ Res^{2}Net	Blend-Res^{2}Net	Blend-Res^{2}Net (Proposed)
Arrowhead	0.703	**0.863**	0.817	0.823	0.819	0.822	0.829
Beef	0.633	**0.933**	0.767	0.667	0.767	0.809	0.838
BirdChicken	0.750	0.850	0.900	**0.950**	0.912	0.920	**0.950**
Car	0.733	0.867	**0.933**	0.867	**0.933**	0.907	0.919
CinCTorso	0.651	**0.996**	0.771	0.865	0.769	0.752	0.848
Coffee	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
DiatomSizeRed	0.967	0.941	0.931	0.922	0.940	0.952	**0.973**
ECG200	0.770	0.850	0.870	**0.920**	0.865	0.859	0.870
ECGFivedays	0.768	1.000	0.931	**1.000**	0.982	**1.000**	**1.000**
FaceFour	0.830	0.954	0.932	0.955	0.927	0.938	**0.960**
FacesUCR	0.905	0.963	0.958	0.967	**0.969**	**0.969**	**0.969**
Fish	0.823	0.989	0.989	0.971	0.989	0.989	**0.995**
GunPoint	0.907	**1.000**	0.993	**1.000**	**1.000**	**1.000**	**1.000**
Ham	0.467	0.667	**0.781**	0.695	0.720	0.716	0.737
Haptics	0.377	0.519	0.506	0.526	0.522	0.510	**0.551**
InlineSkate	0.384	**0.500**	0.365	0.480	0.381	0.401	0.442
ItalyPower	0.950	0.963	0.960	0.955	0.960	0.955	**0.968**
Lightning2	**0.869**	0.820	0.746	0.803	0.781	0.784	0.811
Lightning7	0.726	0.740	0.836	0.808	0.827	**0.839**	**0.839**
MoteStrain	0.835	**0.933**	0.895	0.883	0.873	0.901	0.901
OliveOil	0.833	**0.900**	0.867	0.833	0.849	0.835	0.843
OSULeaves	0.591	**0.979**	**0.979**	0.926	0.933	0.921	0.933
Sony1	0.725	0.765	**0.985**	0.850	**0.985**	0.951	0.975
Sony2	0.831	0.923	0.962	0.938	**0.977**	**0.977**	**0.977**
ToeSeg1	0.772	0.982	0.965	0.969	0.958	0.979	**0.984**
ToeSeg2	0.838	0.954	0.862	**0.946**	0.863	0.856	0.876
TwoLeadECG	0.905	0.996	**1.000**	0.997	**1.000**	**1.000**	**1.000**
Worms	0.584	0.558	0.619	**0.779**	0.703	0.710	0.736
Wormtwoclass	0.623	0.779	0.735	0.766	0.754	**0.782**	**0.782**
Wins	2	9	6	7	7	8	**15**

Results



Blend-Res²-Net
ResNet [7]
Number of layers
Dataset



(lower is desirable)
Avg. Arith Rank
Avg. Geo Rank
ResNet[7]
InceptionTime [14]
DMS-CNN [19]
ROCKET [20]
TS-Chief [21]
Blend-Res²Net (Proposed)

Conclusion

- The proposed interplay of model complexity and relaxation in Blend-Res^{2}Net has played an important role to construct a substantially improved time-series classification model and outperformed the current benchmarks and state-of-the-art methods.
- Ablation studies further reveal efficacy of the proposed method.
- Our future direction is to introduce game theoretic equilibrium setting between model complexity and model relaxation.

Thank you