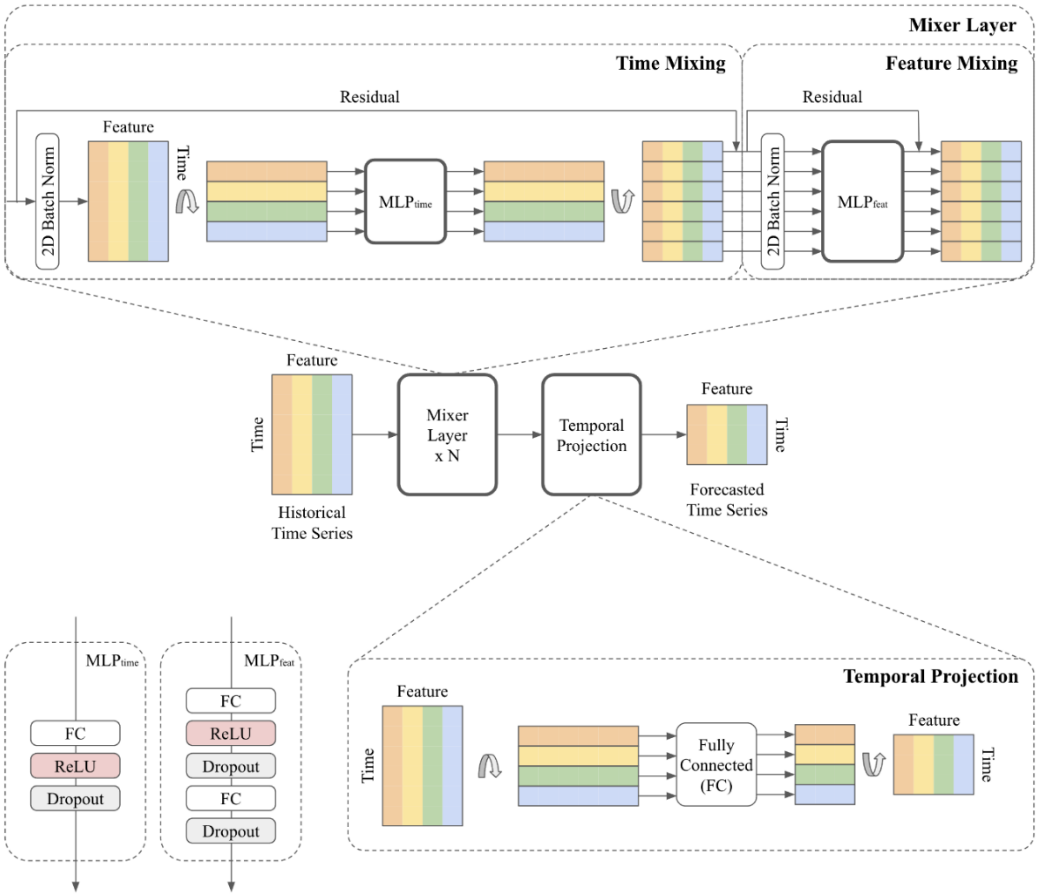
TSMixer

First of all I would like to mention that this paper gives some answers on a previows project that I did 1 year ago on time series. My previows work consisted in using statistichal and Machine Learning methods to analyse and predict time series. In that time most of ML algorithms failed to outperform the "simpler" statictichal methods like ARIMA, SARIMA and so on. The TSMixer seems to be the best method so far in dealing with time series.

Time-Series Mixer (TSMixer) is an innovative architecture created by layering multiple multi-layer perceptrons (MLPs). The foundational concept of TSMixer revolves the use of **mixing processes across both time and feature** axes, enabling it to extract information in a highly efficient manner. This design approach allows TSMixer to blend elements from different time points and features within a time series dataset, facilitating a more effective analysis and understanding of the underlying patterns and relationships.



Architecture of TSMixer. Image by S. Chen, C. Li, N. Yoder, S. Arik and T. Pfister from TSMixer: An All-MLP Architecture for Time Series Forecasting

Since TSMixer is simply extending linear models, its architecture is fairly straightforward, since it is entirely MLP-based.

From the figure above, we can see that the model mainly consists of two steps: a mixer layer and a temporal projection.

We can see that for time mixing, the MLP consists of a fully connected layer, followed by the ReLU activation function, and a dropout layer.

1 - The input, where rows represent time and columns represent features, is transposed so the MLP is applied on the time domain and shared across all features. This unit is responsible for learning temporal patterns.

2 - Before leaving the time mixing unit, the matrix is transposed again, and it is sent to the feature mixing unit. The feature mixing unit, then consists of two MLPs. Since it is applied in the feature domain, it is shared across all time steps. Here, there is no need to transpose, since the features are already on the horizontal axis.

3 - The temporal projection step is what generates the predictions in TSMixer.Here, the matrix is again transposed and sent through a fully connected layer to generate predictions. The final step is then to transpose that matrix again to have the features on the horizontal axis, and the time steps on the vertical axis.

The most important aspect is the cross-variete information, so the correlation between variables where the statisctical methods lack as they perform only on univariate time series.

TSMixer exhibits a unique value as it is the only model that performs as well as univariate models when cross-variate information is not useful, and it is the best model to leverage cross-variate information when it is useful.

Future developments might focus on improving its interpretability and scaling it up for even larger datasets.

Implementation

Normalization: There are three types of normalizations used in the implementation: 1. Global normalization: Global normalization standardizes all variates of time series independently as a data pre-processing. The standardized data is then used for training and evaluation.

Local normalization: In contrast to global normalization, local normalization is applied on each batch as pre-processing or post-processing.

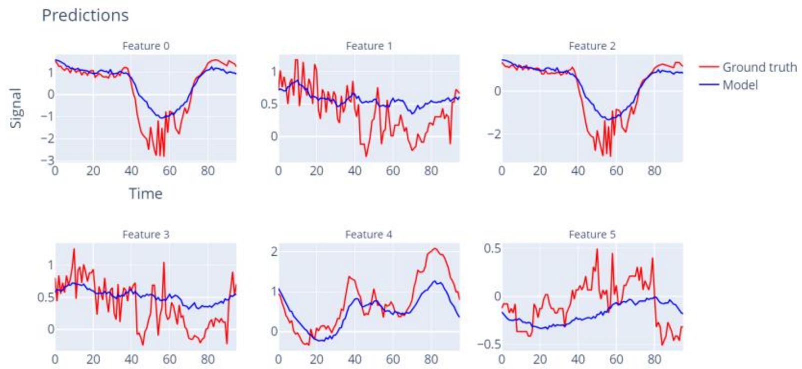
Application

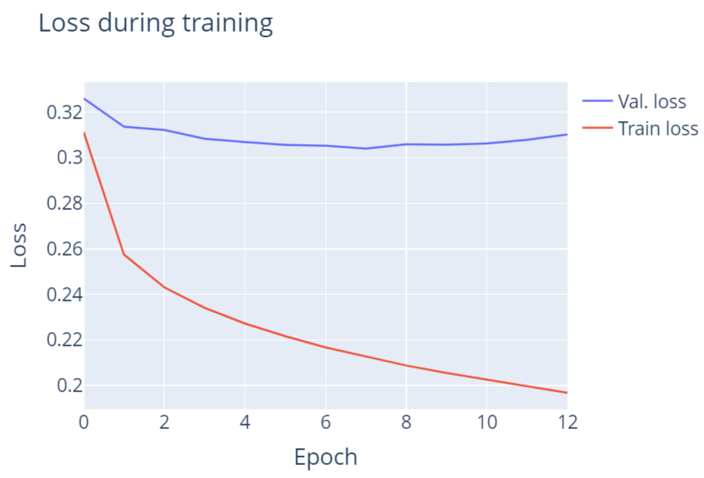
Field	date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
Description	The recorded date	High UseFul Load	High UseLess Load	Middle UseFul Load	Middle UseLess Load	Low UseFul Load	Low UseLess Load	Oil Temperature (target)

date	HUFL	HULL	MUFL	MULL	LUFL	LULL	OT
01/07/2016 00:00	5.827	2.009	1.599	0.462	4.203	1.34	30.531
01/07/2016 00:15	5.76	2.076	1.492	0.426	4.264	1.401	30.46
01/07/2016 00:30	5.76	1.942	1.492	0.391	4.234	1.31	30.038
01/07/2016 00:45	5.76	1.942	1.492	0.426	4.234	1.31	27.013

I applied TSMixer on ETTm1time serie dataset which has 70000 samples and 7 features.

Unfortunately I couldn't execute grid search till the end, but I will train the model with the values I received till the results I got after some hours of executation.





Models		TSMixer		TFT		FEDformer*		Autoformer*		Informer*	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.285**	**0.339**	0.752	0.626	0.326	0.390	0.510	0.492	0.626	0.560
	192	**0.327**	**0.365**	0.752	0.649	0.365	0.415	0.514	0.495	0.725	0.619
	336	**0.356**	**0.382**	0.810	0.674	0.392	0.425	0.510	0.492	1.005	0.741
	720	0.419	**0.414**	0.849	0.695	0.446	0.458	0.527	0.493	1.133	0.845

As seen from the figures above where we take an loss funscrtion around 0.3 which is close to the loss fynction obtained by the official paper, with prediction length 96. This value is less than any other method applied to this time serie. Till now the TSMixer seems to outperform everyother method in multivariate time series.