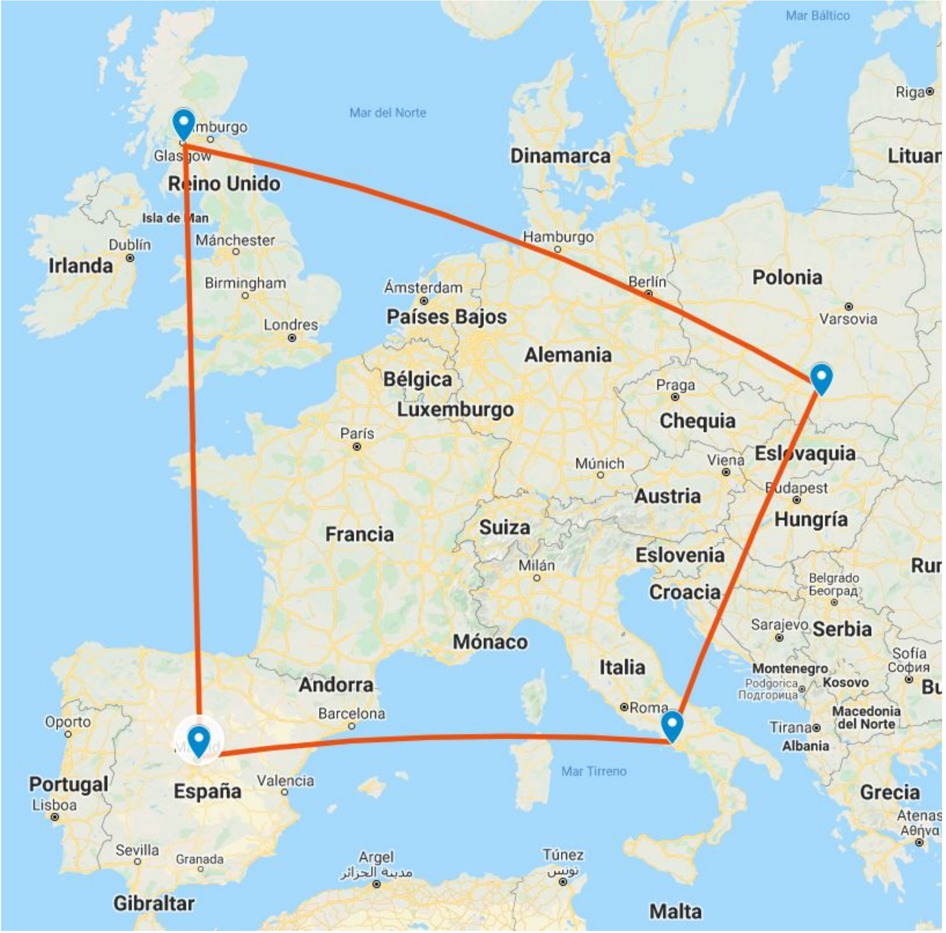
Modern deep learning approaches for time series

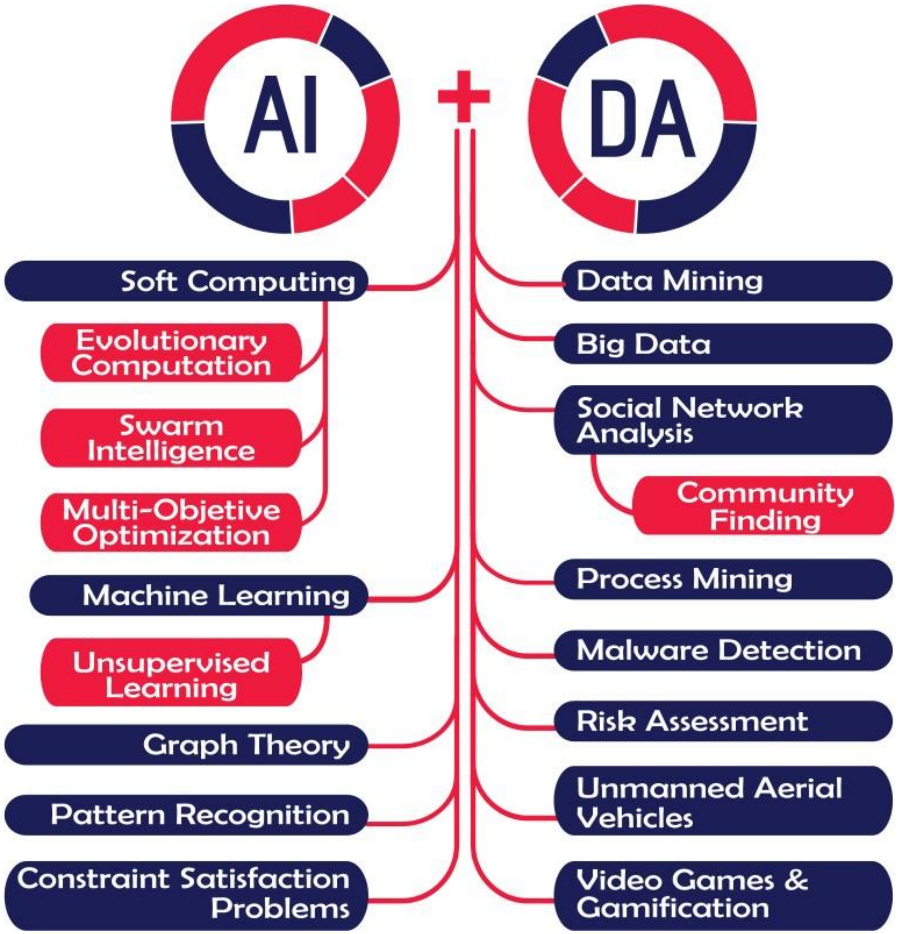
Victor Rodriguez-Fernandez
Universidad Politécnica de Madrid
victor.rfernandez@upm.es

vrodriguezf
@vrodriguezf90

About me

- Assistant professor at UPM (now)
- PhD in computer science (2019)
  - Airbus Defence & Space
  - Autonomous University of Madrid (UAM)
- Research stays
  - 2017 - University of Strathclyde (Glasgow, UK)
  - 2019 - AGH (Krakow, Poland)
  - 2021 - University of Naples Federico II (Naples, Italy)
- Interests:
  - Machine & Deep learning





http://aida.etsisi.upm.es/

Table of Contents

Backgrounds in time series

A time series X is an ordered sequence of t real values $X = \{x_1 \,, \ldots, \, x_t\}, x_i \in R$, $i \in N$.

Due to their natural temporal ordering, **time series data** are present in almost every task that requires some sort of human cognitive process.
Time series are encountered in many real-world applications ranging from:

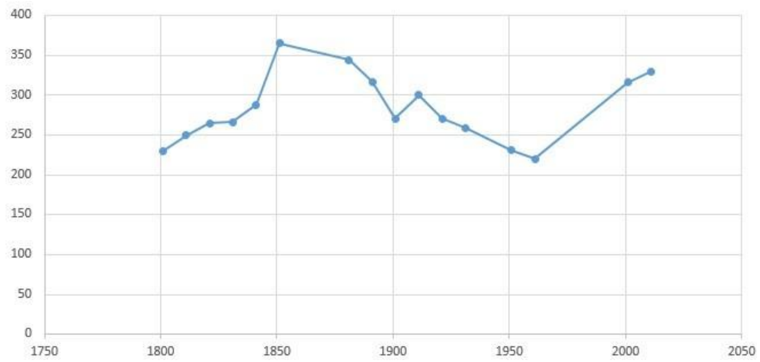
- Electronic health records
- Human activity recognition
- Cybersecurity
- Aerospace Engineering

When analysing a dataset of time series data, there are multiple characteristics that one must check before choosing a technique:

- Type of data contained in the time series
- Number of variables
- Time series length
- Spacing of observation times
- Presence of missing values
- Stationarity

Type of data

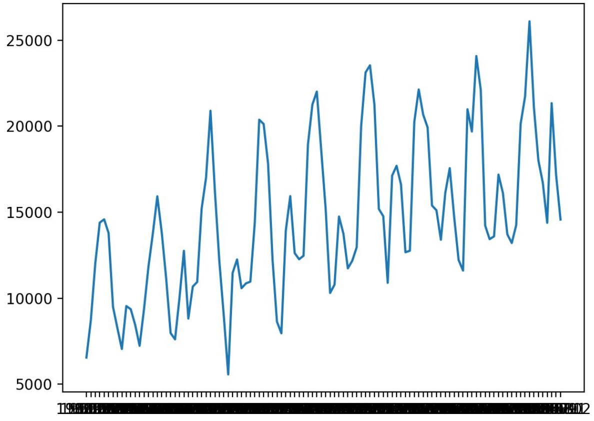
In the literature, the keyword time series usually refers to **countinuous** data, while the keyword sequence usually refers to **categorical** data (symbols).

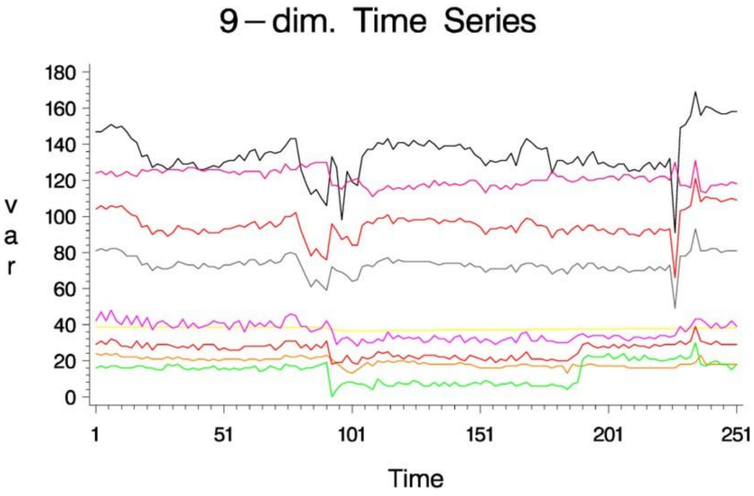


SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

Number of variables

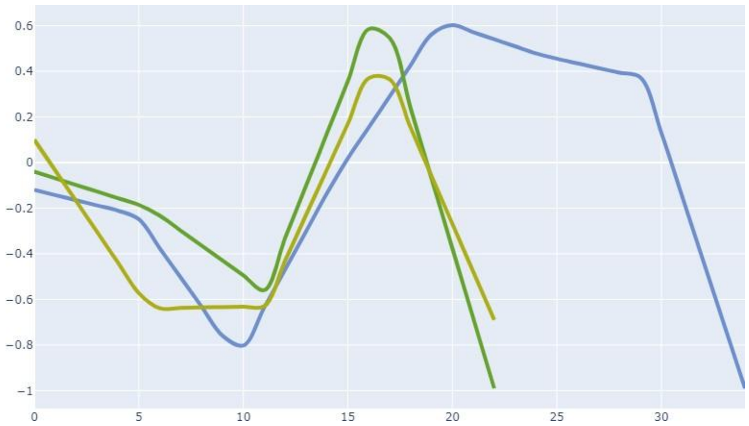
Classic datasets of time series are normally **univariate**. However, the increase amount of sensor data available in different domains is increasing the need of analysis of **multivariate** time series.





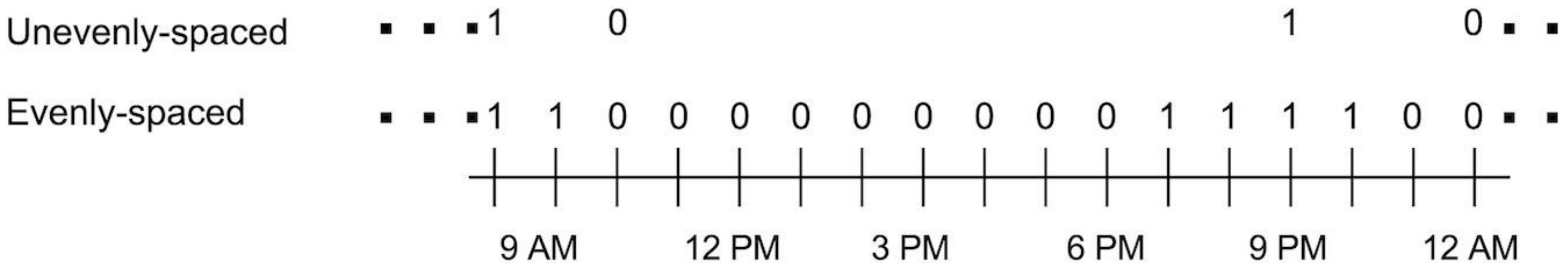
Time series length

Research into time series classification has tended to focus on the case of series of **uniform length**. However, it is common for real-world time series data to have **unequal lengths**.



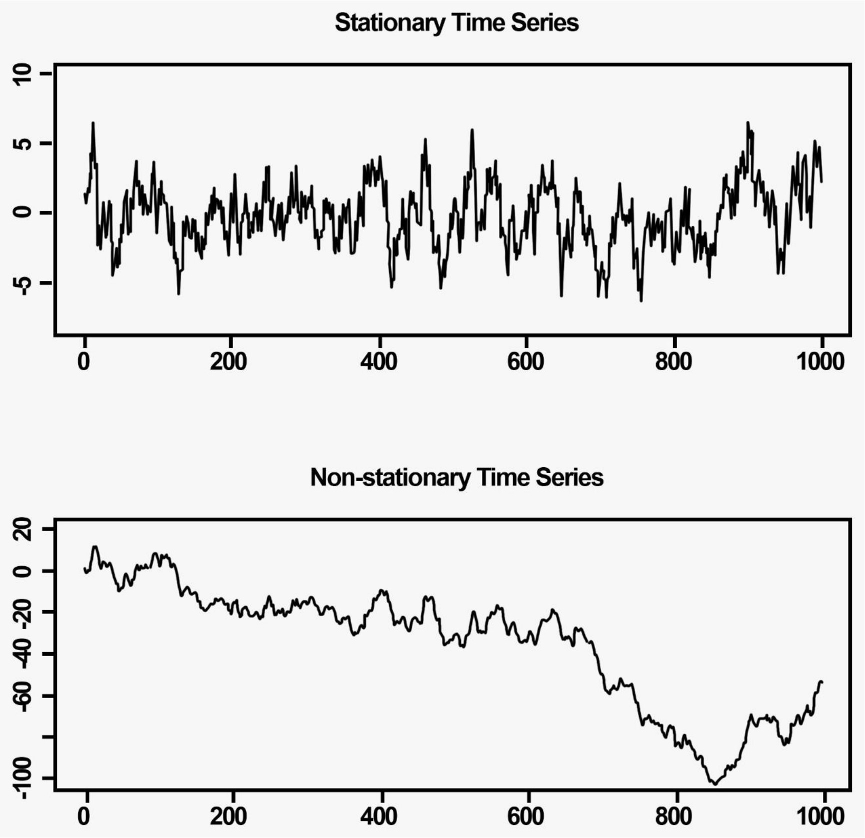
Spacing of observation times

As opposed to evenly (or **regular**) spaced time series, in (or **irregular**) the spacing of observation times is not constant.



Stationarity

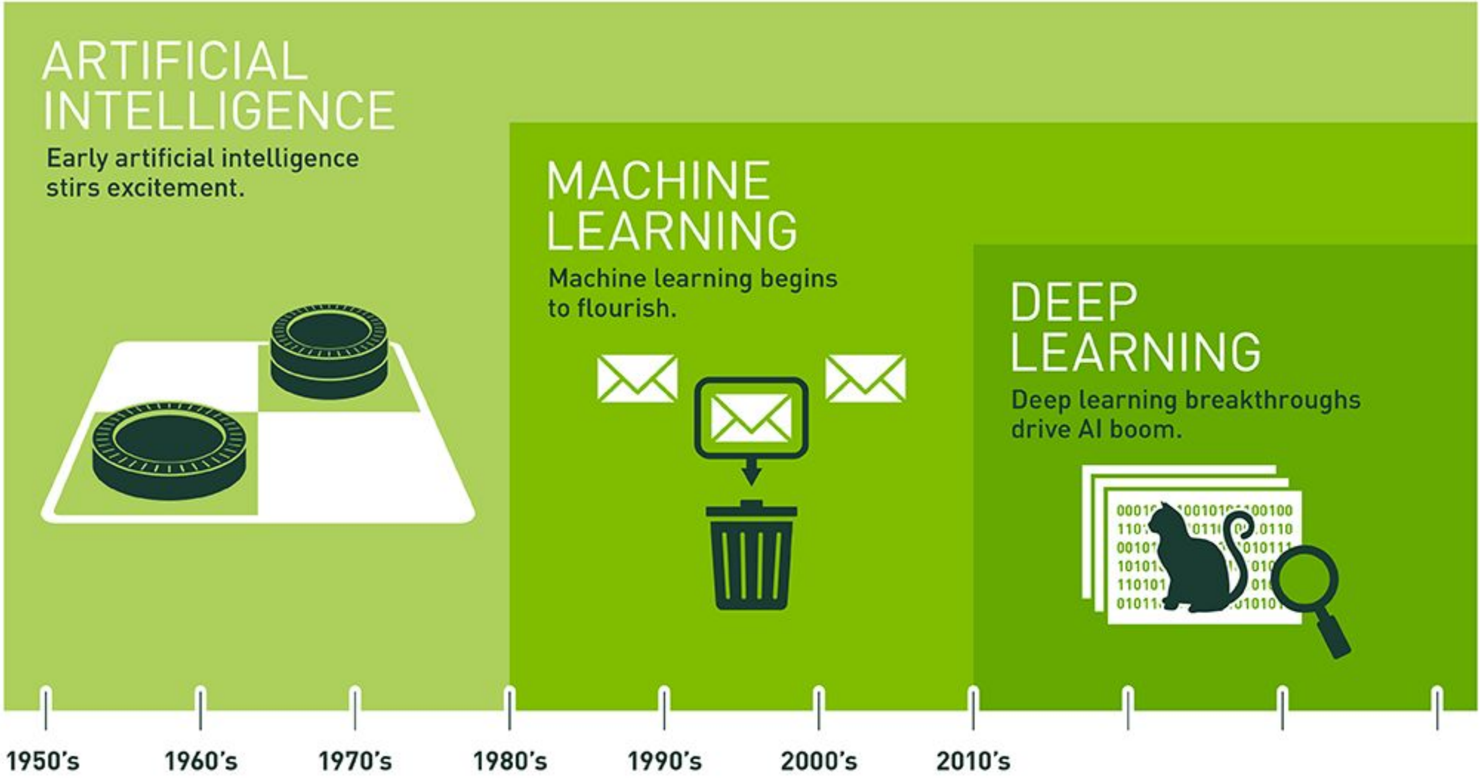
A stationary time series is one whose statistical properties such as the mean, variance and autocorrelation are all **constant over time**. Non-stationary data should be first converted into stationary data in some classic modelling techniques such as ARMA.



Backgrounds in deep learning

Jargon

- the functional form of a model -> architecture
- the weights -> parameters
- the results -> predictions
- the measure of performance -> loss
- the dependent variable -> targets, y (labels in the context of classification)

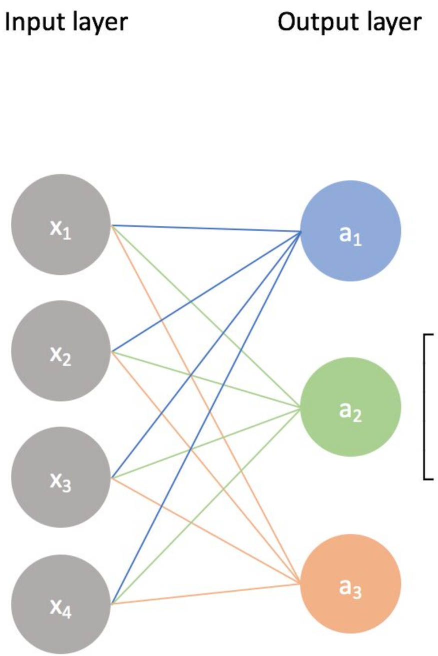


https://docs.paperspace.com/machine-learning/

Since the optimistic beginnings of AI in the 1950s, small subfields of AI - first Machine Learning and then Deep Learning (a subfield of Machine Learning) have made huge milestones.

Deep learning = Machine learning with Neural networks

Matrix multiplication is all you need

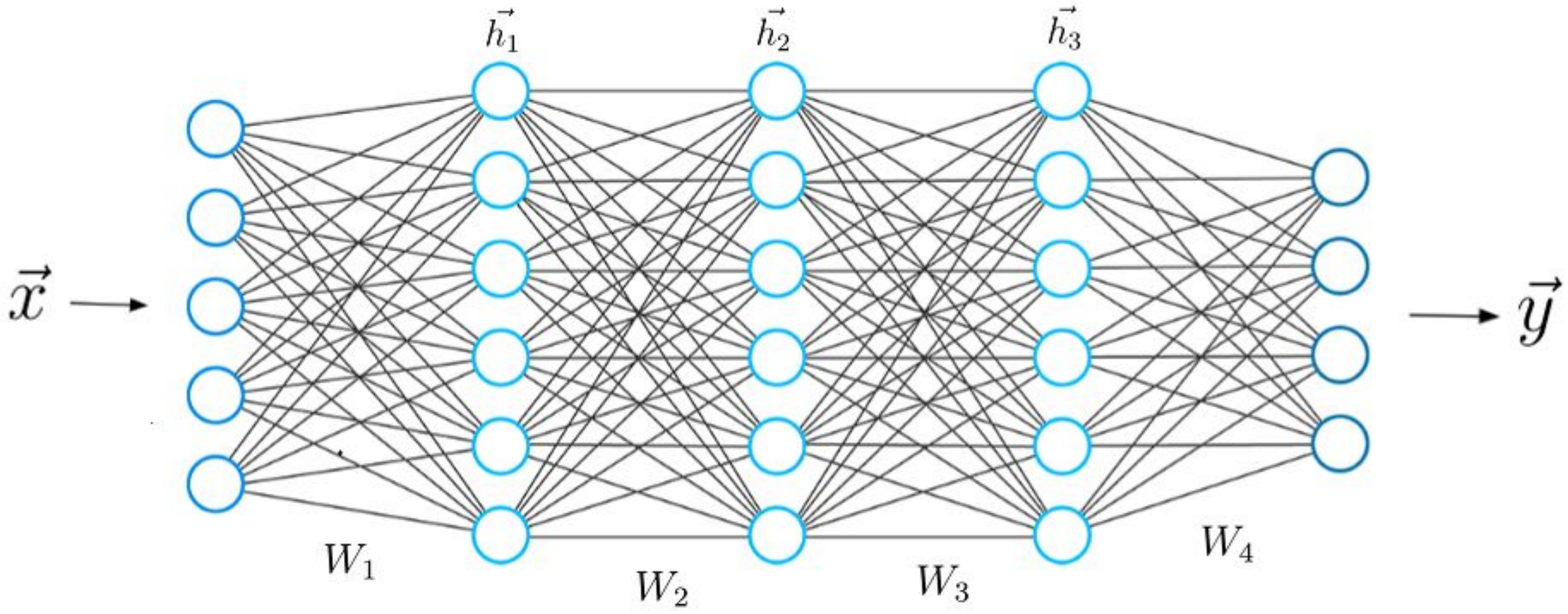


A simple neural network

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow[\text{activation}]{} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

https://www.jeremyjordan.me/intro-to-neural-networks/

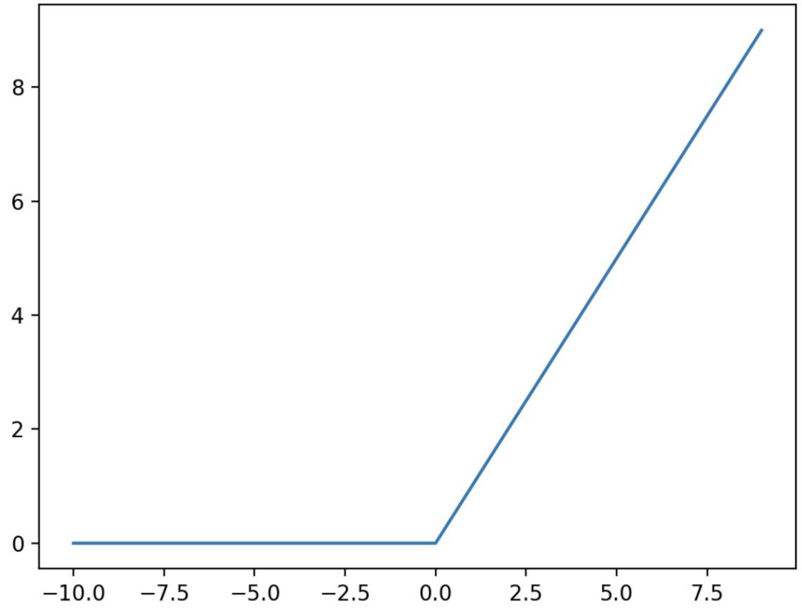
Going deep



https://www.deeplearning-academy.com/p/ai-wiki-machine-learning-vs-deep-learning

Activation functions

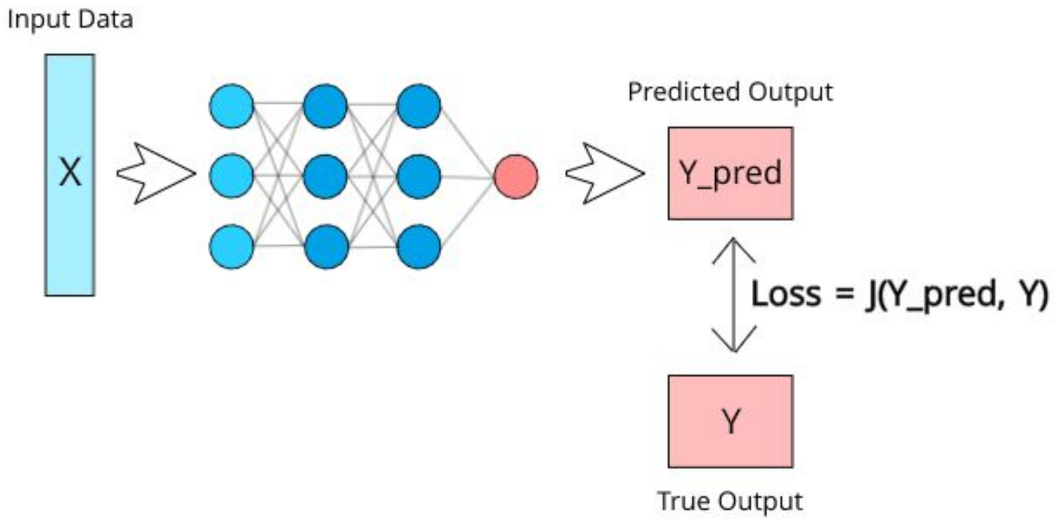
Non linearities applied after the matrix multiplication. The rectified linear unit, or ReLU, has been the most popular in the past decade.



Loss functions

Loss Functions are used to frame the problem to be optimized within deep learning. Most popular ones are:

- Cross Entropy Loss (classification)
- Mean Squared error (regression)

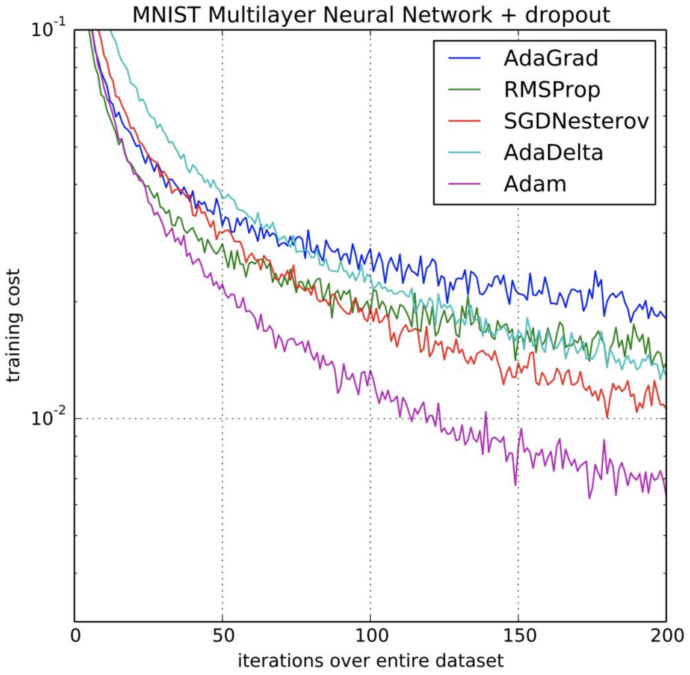


https://deeplearningdemystified.com/article/fdl-3

Stochastic optimization

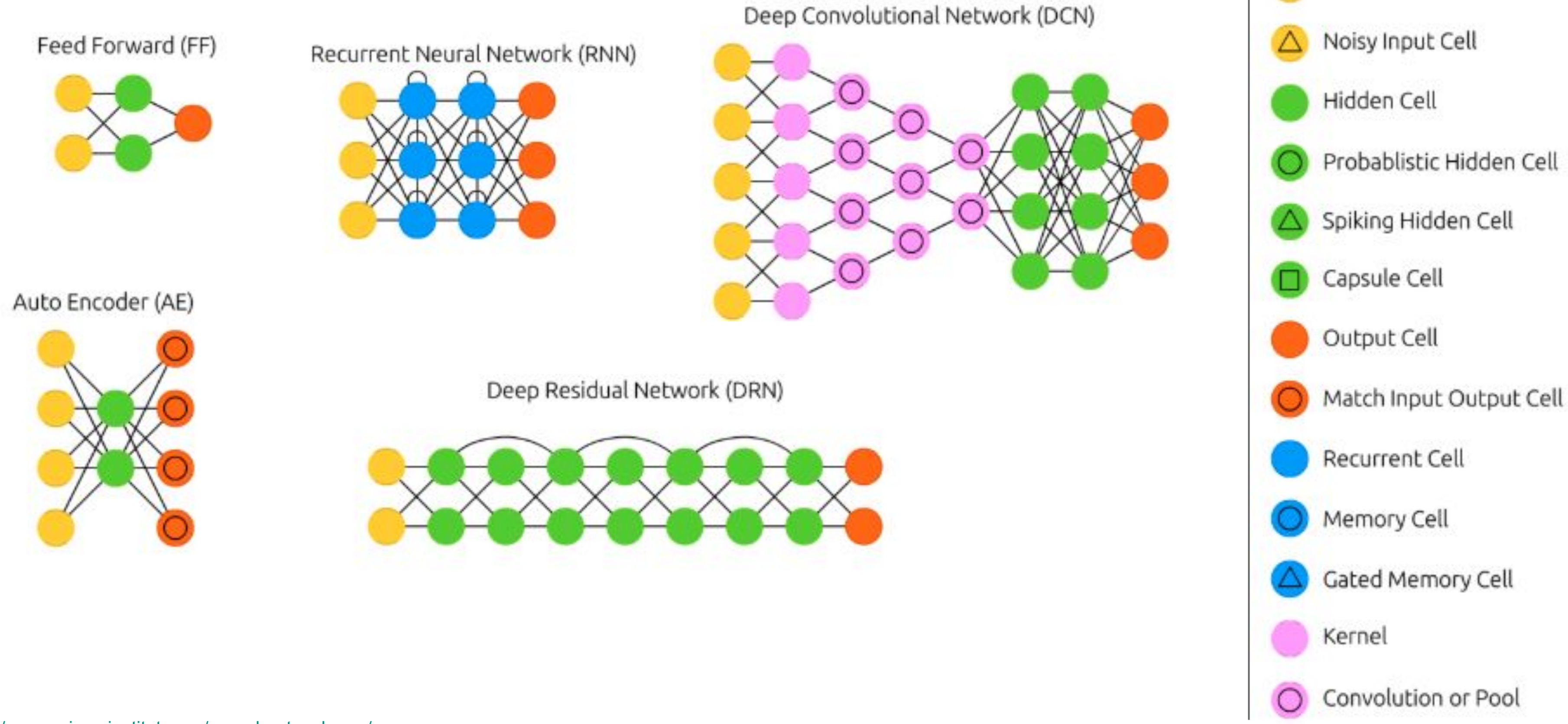
Used to train neural networks. They iteratively take a “mini-batch” of data, hence ‘stochastic’, and perform gradient descent on the loss function for that batch. Most popular methods are:

- SGD
- Adam



https://paperswithcode.com/method/adam

Neural architectures

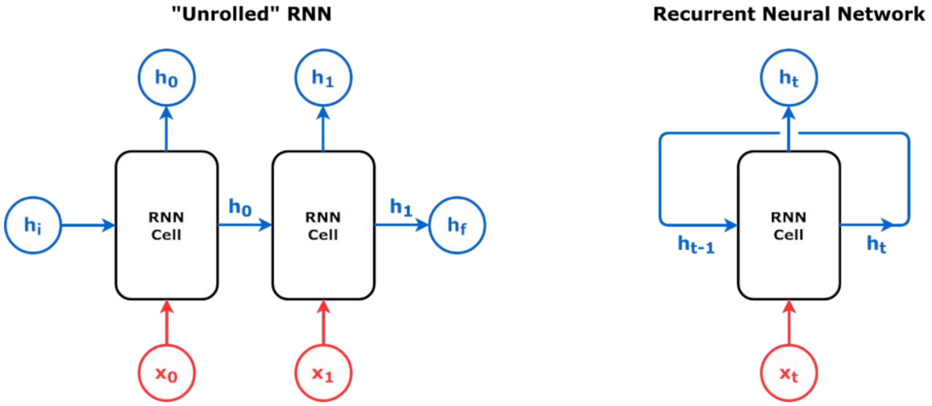


https://www.asimovinstitute.org/neural-network-zoo/

Recurrent Neural Network (RNN)

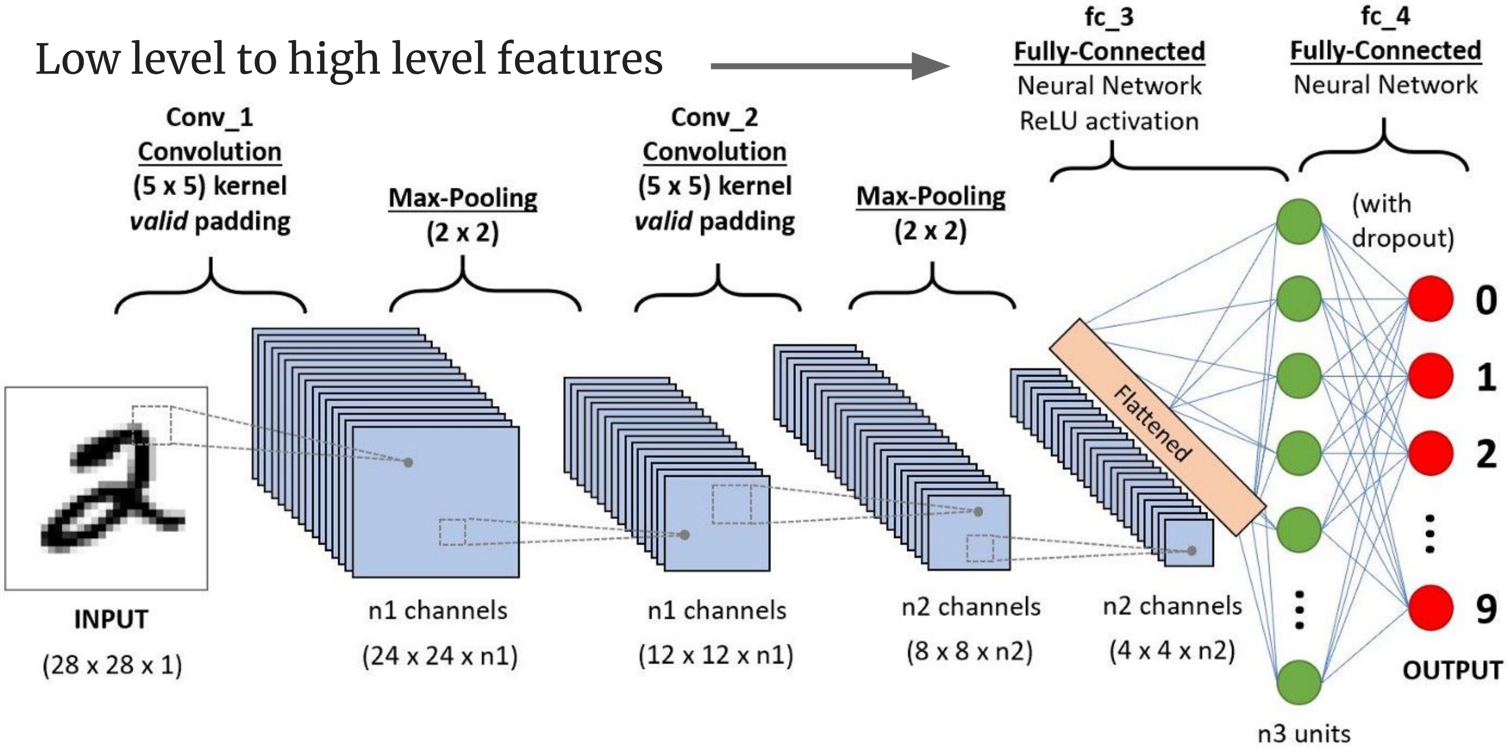
Extensions

- Stacked RNN
- Bidirectional RNN
- GRU
- LSTM



Convolutional Neural Networks

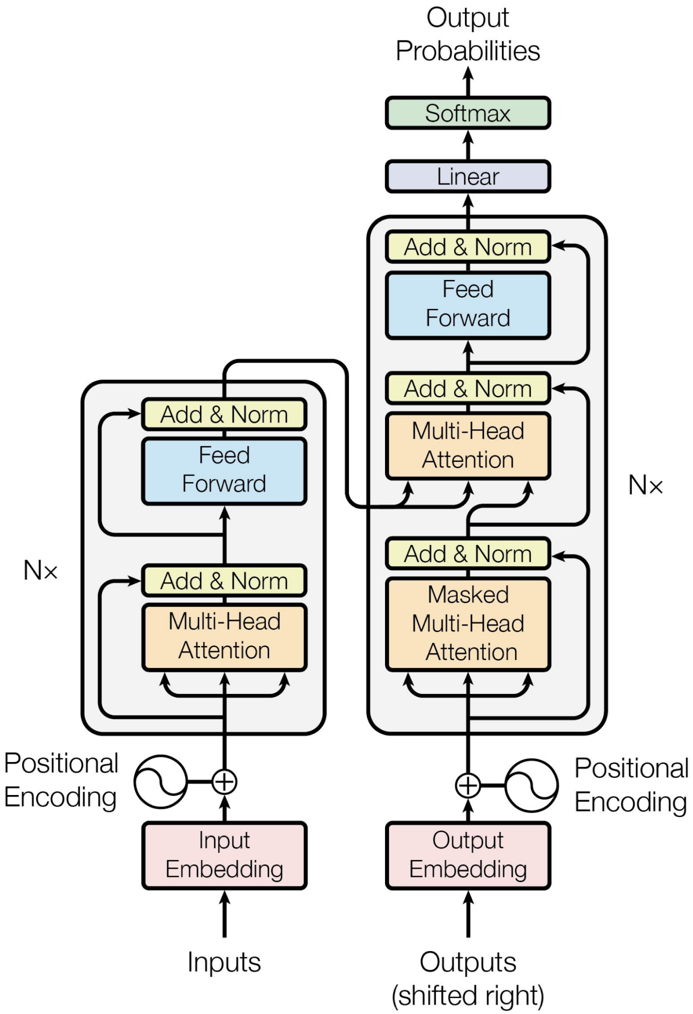
Low level to high level features



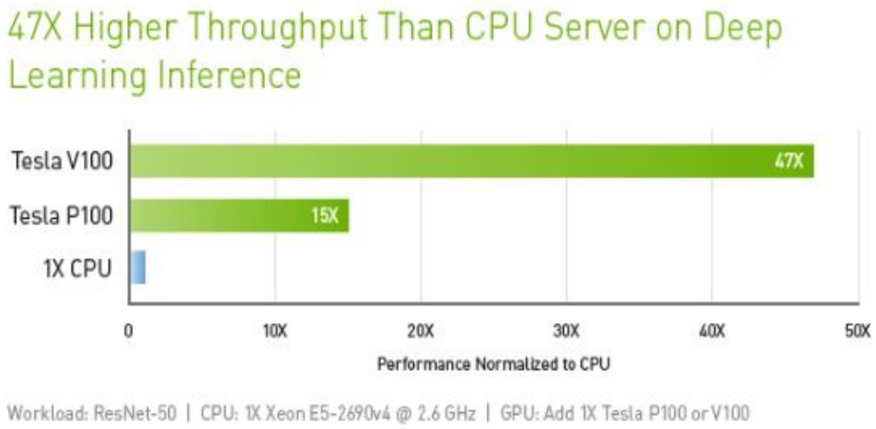
Transformers

State-of-the-art in many tasks involving sequence or sets

- BERT
- GPT-3
- ViT



GPU costs & availability



http://nvidia.ihuidian.com/product/detail_17.html

Deep learning frameworks

TensorFlow

Keras

fast.ai

PyTorch

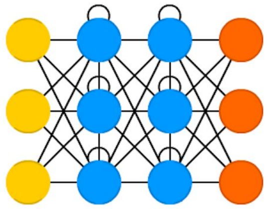
mxnet

https://docs.paperspace.com/machine-learning/wiki/comparison-of-ai-frameworks

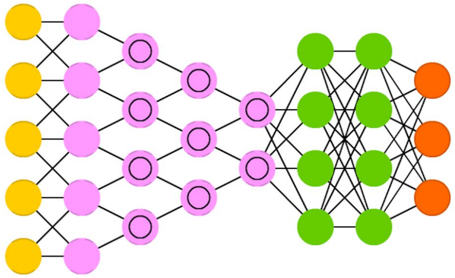
DL architectures for TSC

Since the recent success of deep learning techniques in supervised learning such as image recognition and natural language processing, researchers started investigating complex architectures for TSC:

- Recurrent Neural Networks (RNN)
- Convolutional Neural Networks (CNN)



Recurrent Neural Network

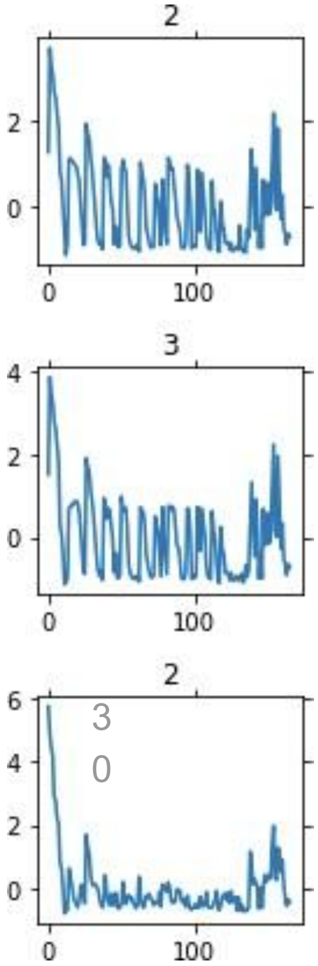


Convolutional Neural Network

Well-established tasks for deep learning

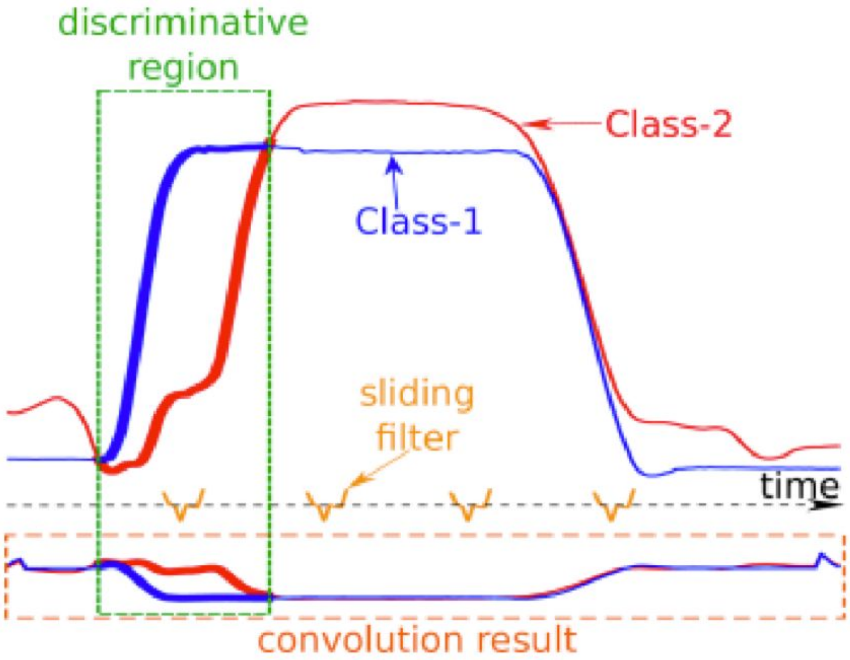
Time Series classification (TSC)

Time Series Classification (TSC) is a supervised learning problem that aims to predict a discrete label y ∈ {1, …, c} for an unlabeled time series, where c is the number of classes in the TSC task.



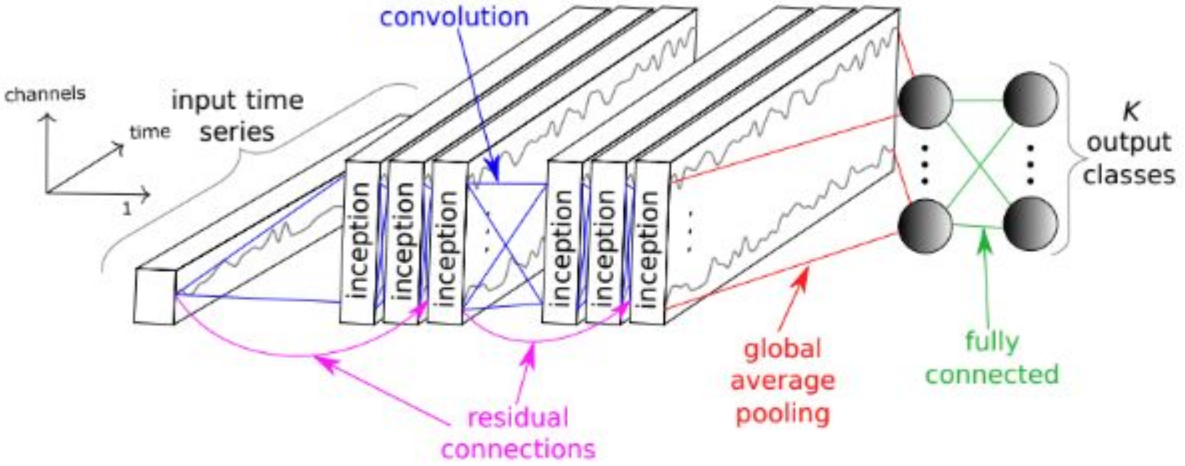
The convolution operation in a time series

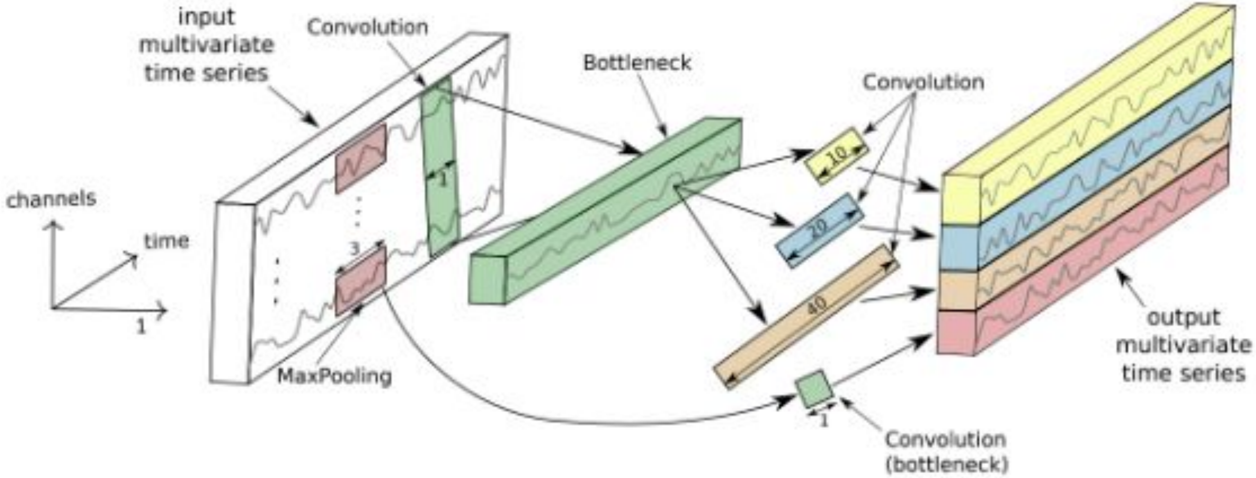
Given an input time series, a convolutional layer consists of sliding **one-dimensional filters** over the time series, thus enabling the network to extract non-linear discriminant features that are **time-invariant** and useful for classification. The filter can also be seen as a generic non-linear transformation of a time series



https://arxiv.org/pdf/1909.04939.pdf

Architectures for TSC: InceptionTime



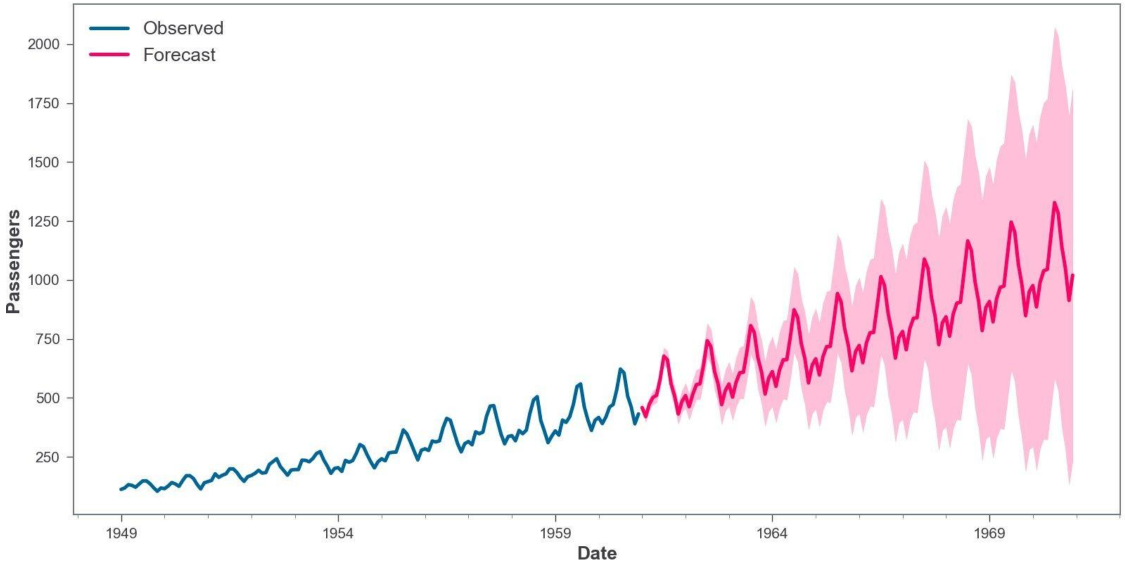


Each InceptionModule within InceptionTime is based on convolution and pooling layers.

Well-established tasks for deep learning

Time Series Forecasting

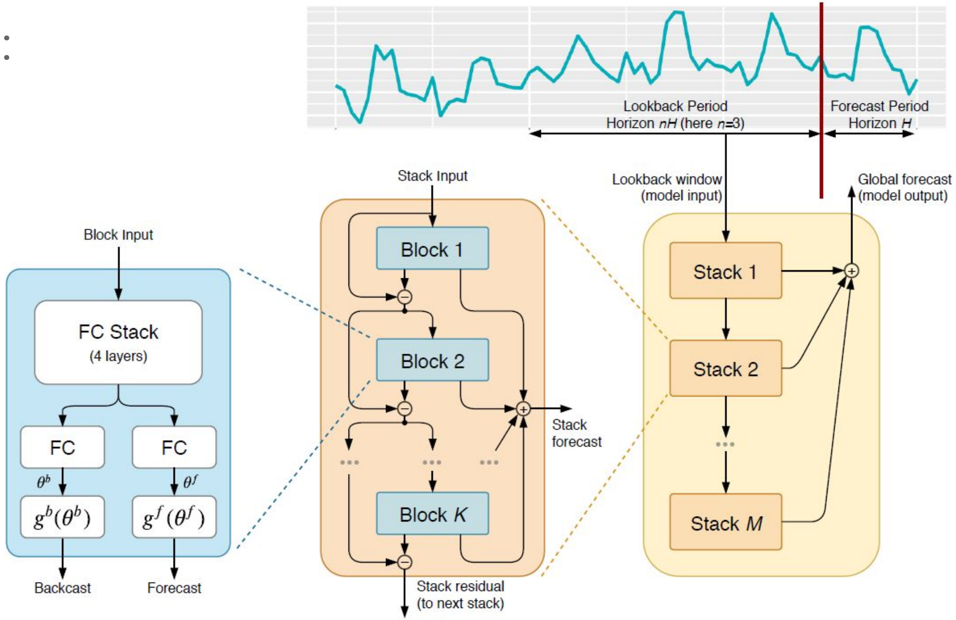
In forecasting, the machine **predicts future** time series based on past observed data. The better the interdependencies among different series are modeled, the more accurate the forecasting can be.



Forecasting Architecture: N-BEATS

Novel deep residual **N-BEATS** architecture:

- **Competition-winning accuracies**
  (Kaggle datasets - finance & business)
- **Agnostic & general**
  (No prior feature knowledge required)
- **Deeper** than typical TS architectures
- Univariate, single-point forecasting



kaggle.com/m4-forecasting-competition-dataset

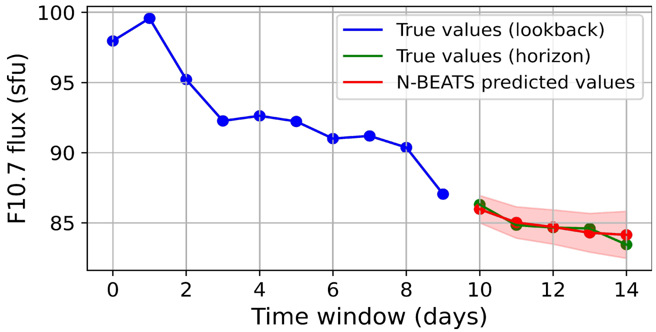
B.N. Oreshkin et al. (2020) “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting”, ICLR 2020

Applications of N-BEATS

Intelligent Atmospheric Density Modelling for Space Operations

Space Weather Proxies → Atmospheric Density → Collision & Casualty Risk

Space weather forecasting using novel deep learning architectures



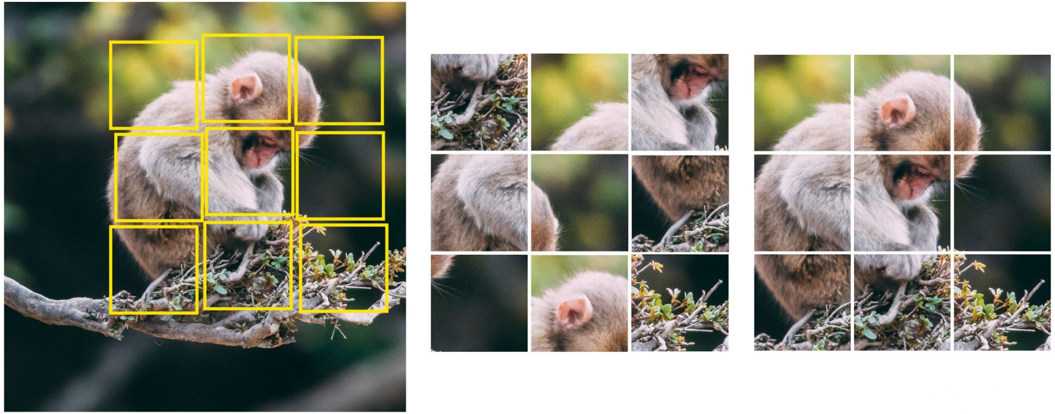
Stevenson, E., Rodriguez-Fernandez, V., Minisci, E., Camacho, D. (2020). A Deep Learning Approach to Space Weather Proxy Forecasting for Orbital Prediction. In Proceedings of the 71st International Astronautical Congress (IAC), The CyberSpace Edition, 12-14 October 2020.

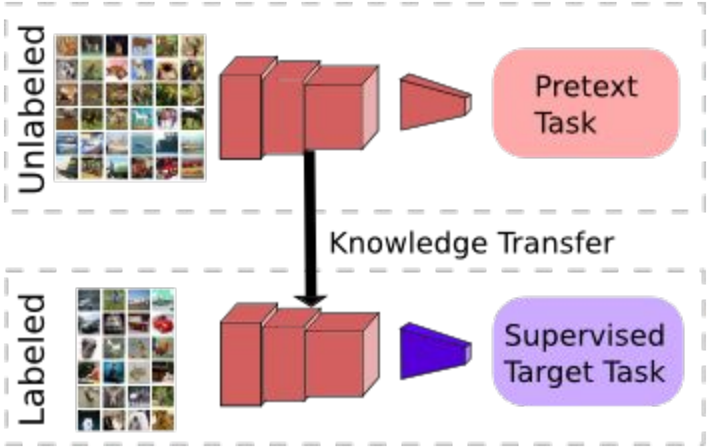
Modern topics in DL and time series

Self-supervised learning

Self-supervised learning

The main idea of Self-Supervised Learning is to generate the labels from unlabeled data, according to the structure or characteristics of the data itself, and then train on this unsupervised data in a supervised manner.





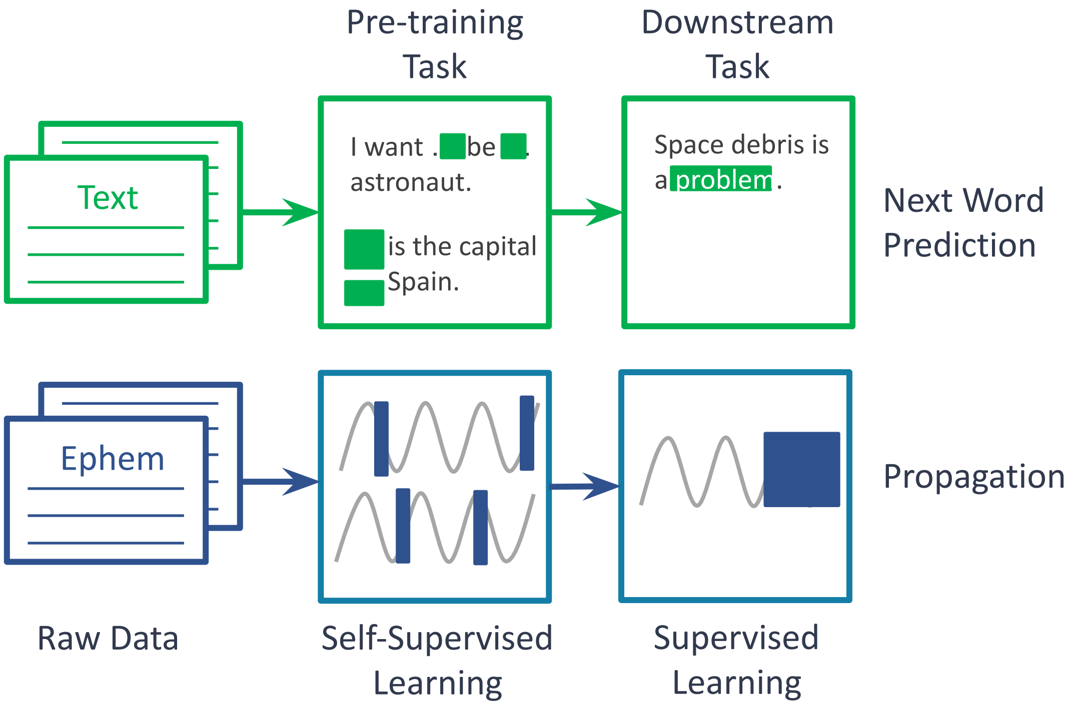
https://rl.uni-freiburg.de/img/teaching/selfsup-seminar

https://perfectial.com/wp-content/uploads/2020/03/SSL-02-scaled.jpg

Self-supervised learning in NLP

BERT-like **self-supervised language** models: [2]

- **Mask** & reconstruct words
- Labels embedded in data



[2] J. Devlin et al. (2019) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

Use case: Orbit modelling for Space Traffic Management (STM)

Self-supervised learning for STM?

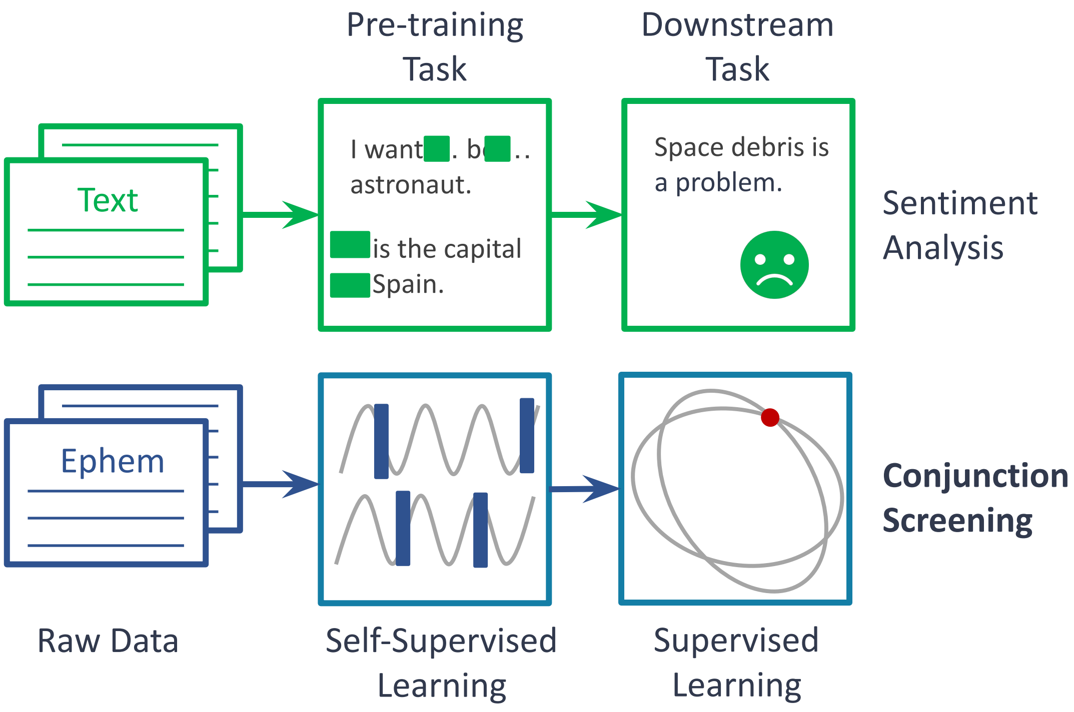
Translate BERT to time series domain using orbit data

- **ORBERT**

Learnt knowledge can improve performance of variety of supervised **downstream** tasks

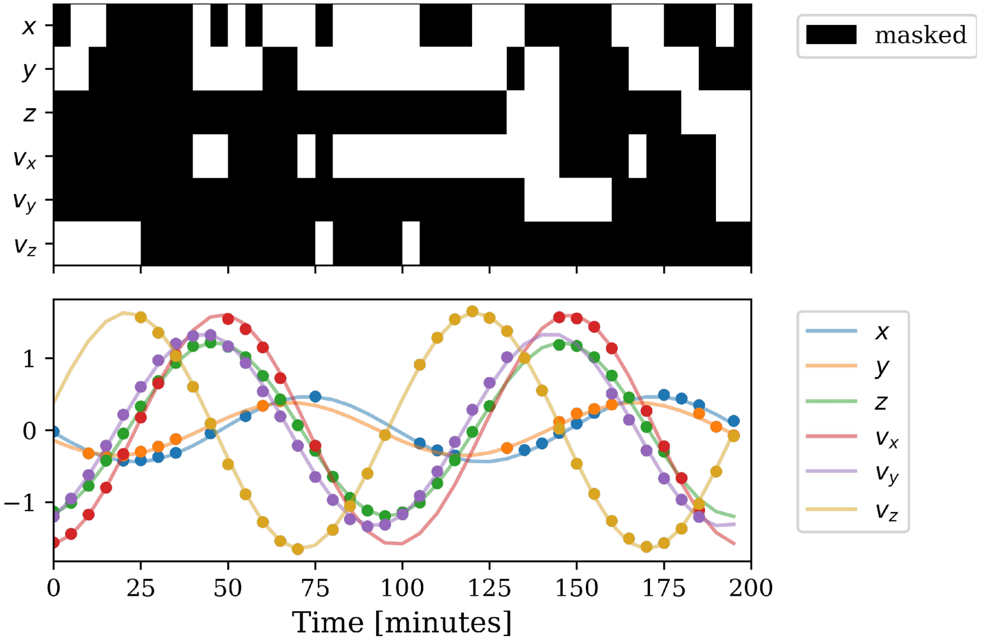
Labelling data expensive:

- **Leverages unlabelled data**



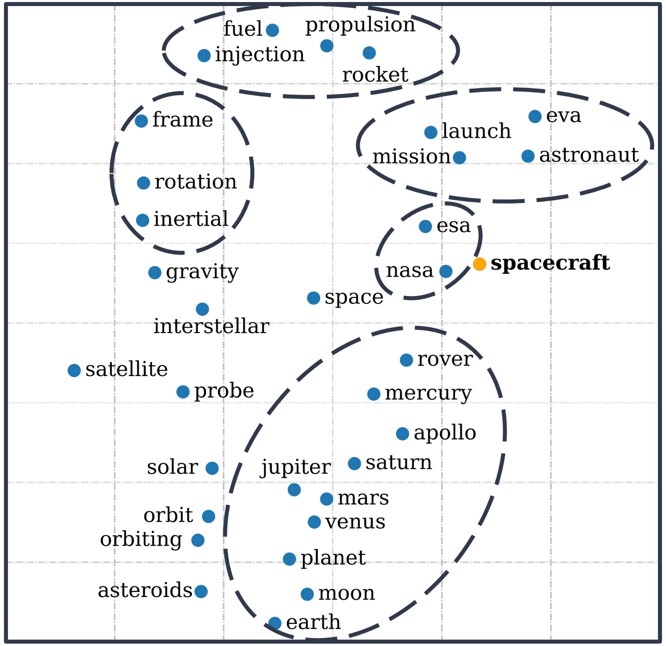
Self-supervised training approach

- Stochastically **mask** sections of orbit ephemerides (6-channel multivariate time series)
- Task model with **reconstruction**
- Choice of architecture open: convolution-based **InceptionTime** [4]



[4] H. Fawaz (2019) "InceptionTime: Finding AlexNet for Time Series Classification"
[5] G. Zerveas (2020) "A Transformer-based Framework for Multivariate Time Series Representation Learning"

Model Insights: Extracted Representations



Can the learnt representations carry **meaningful, valuable information** downstream?

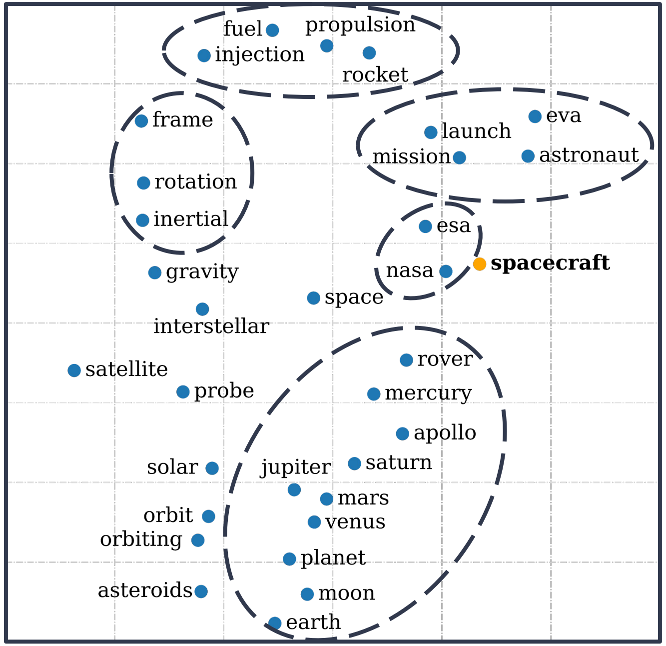
- Reveal knowledge extracted by model by analysing **neuron response**

NLP analogy:

- *Semantic similarity* – similar words clustered
- Expect similar orbits to be clustered

[6] T. Mikolov (2013) "Efficient Estimation of Word Representations in Vector Space"

Model Insights: Extracted Representations



Neuron activations

+

Dimensionality reduction (UMAP)

+

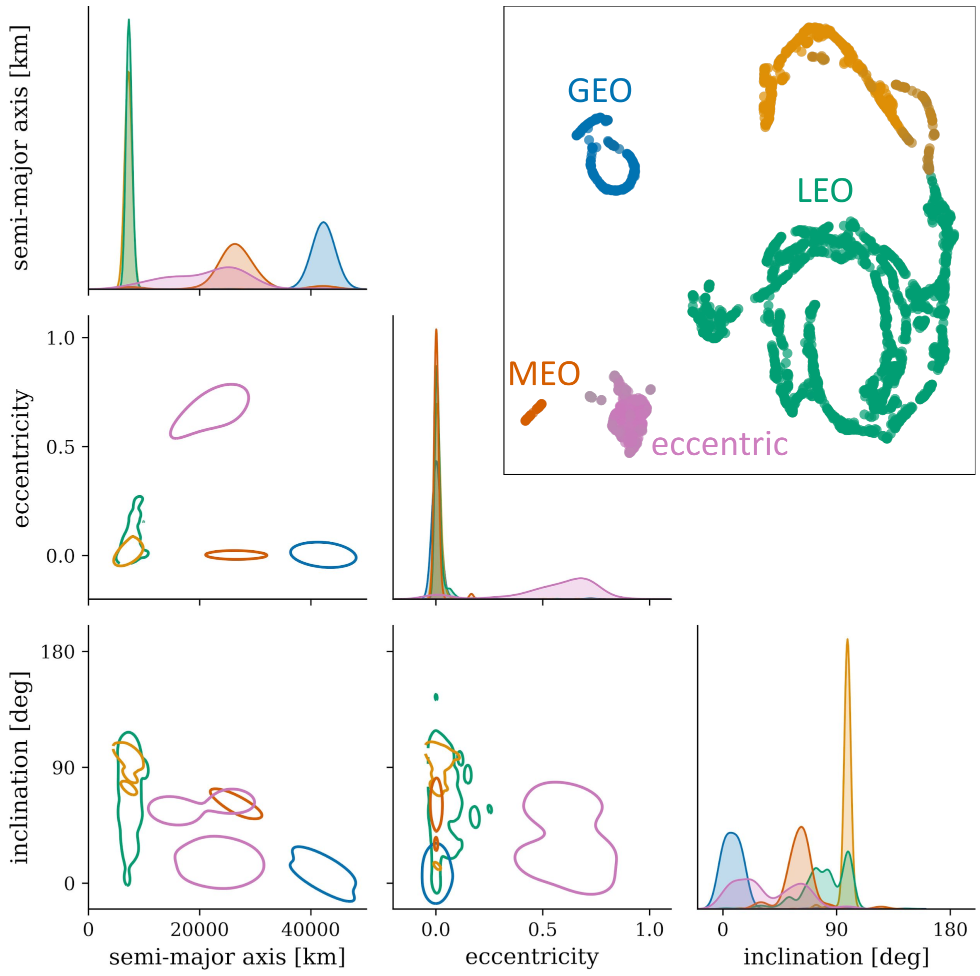
Clustering (HDBSCAN)

Model Insights

- Similar orbits clustered together
- Learning from Cartesian data for an imputation task, the model can successfully classify orbits in Keplerian space

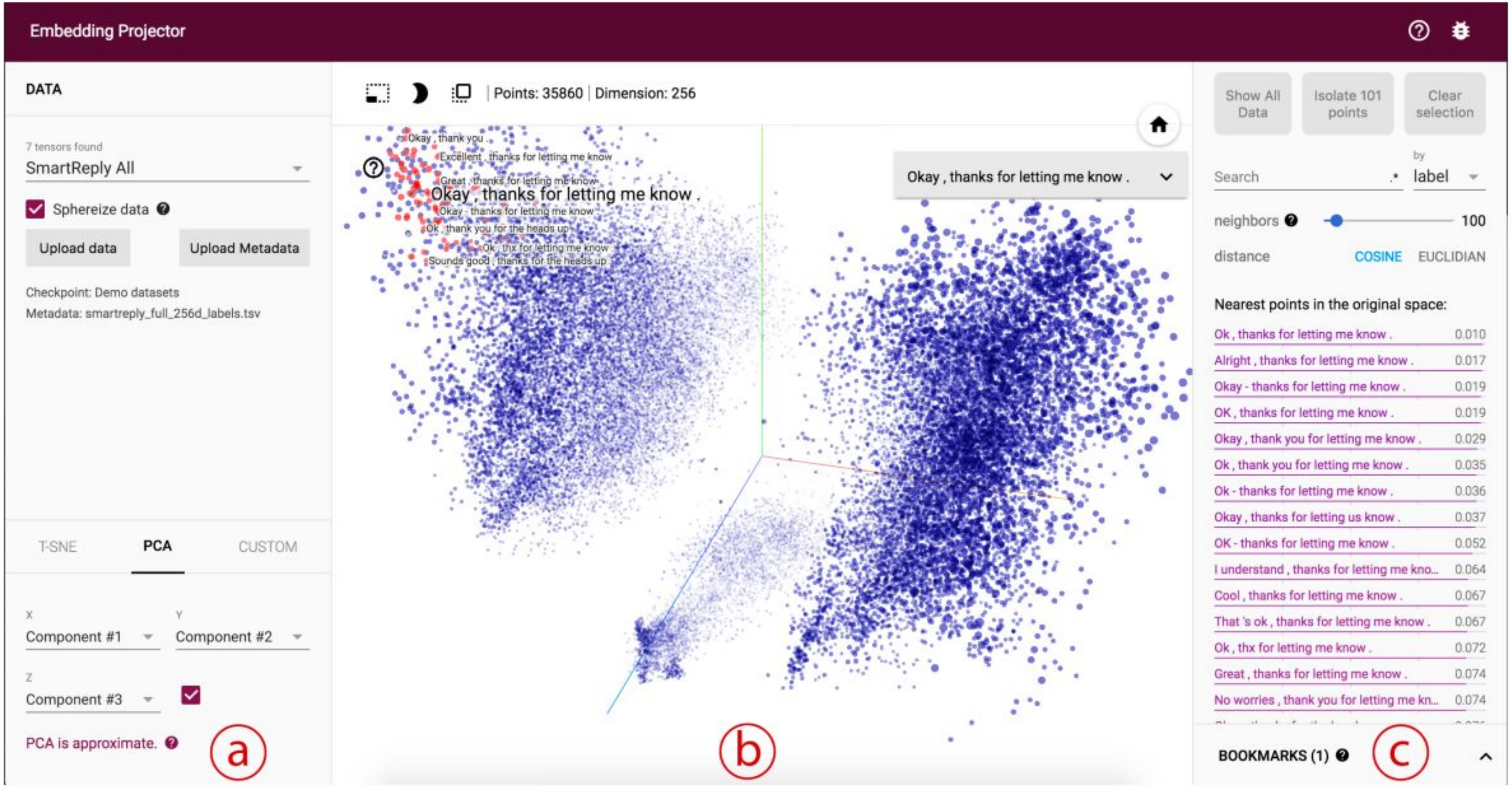
Power of transfer learning:

- Hidden physical knowledge unveiled from a simple different task



Modern topics in DL and time series

Deep Visual Analytics

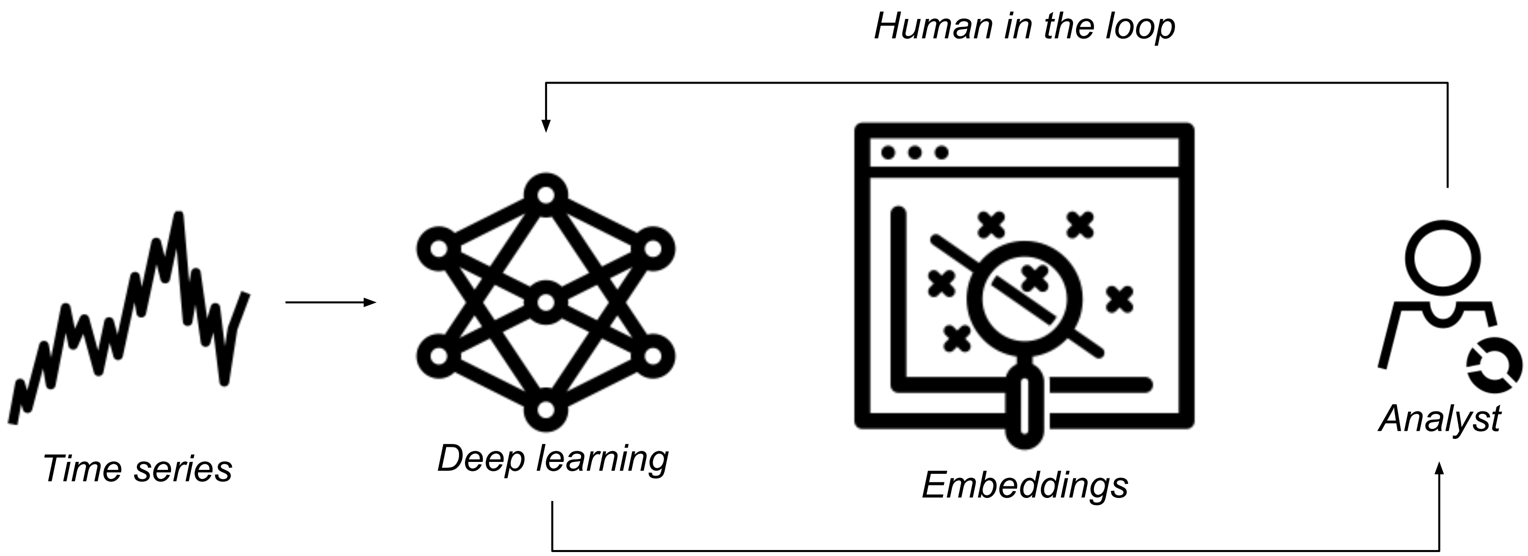


Tensorflow’s Embedding Projector: Deep Visual Analytics in NLP

https://arxiv.org/pdf/1611.05469.pdf

Use case: DeepVATS, Deep learning Visual analytics for Time Series

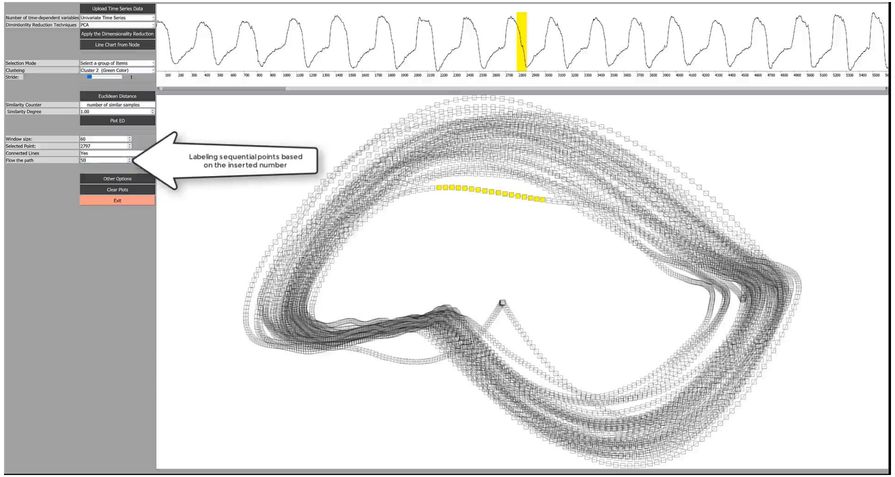
https://github.com/vrodriguezf/deepvats



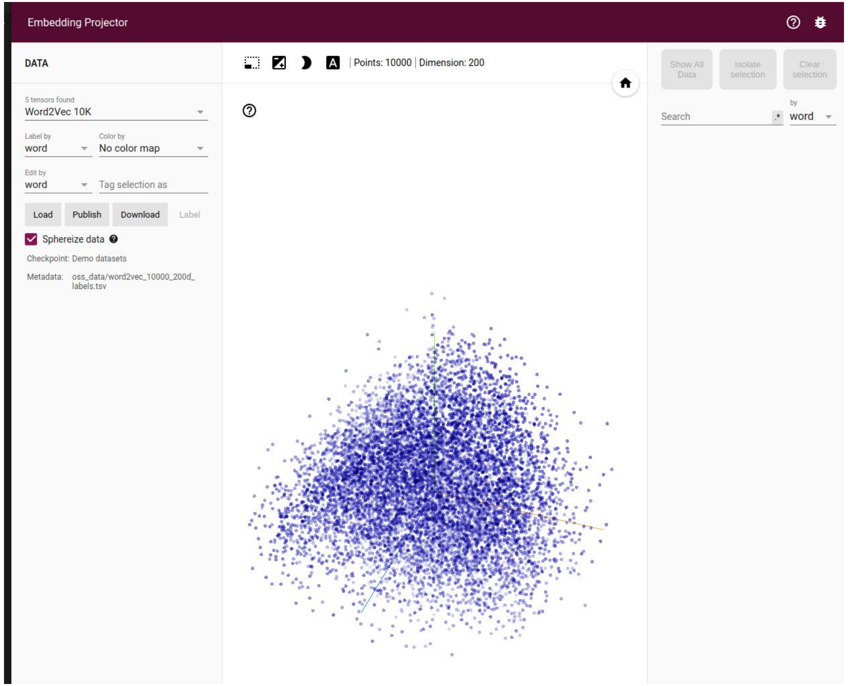
General scheme of DeepVATS. Visualizing the embeddings can help in easily **detecting outliers, change points, and regimes.**

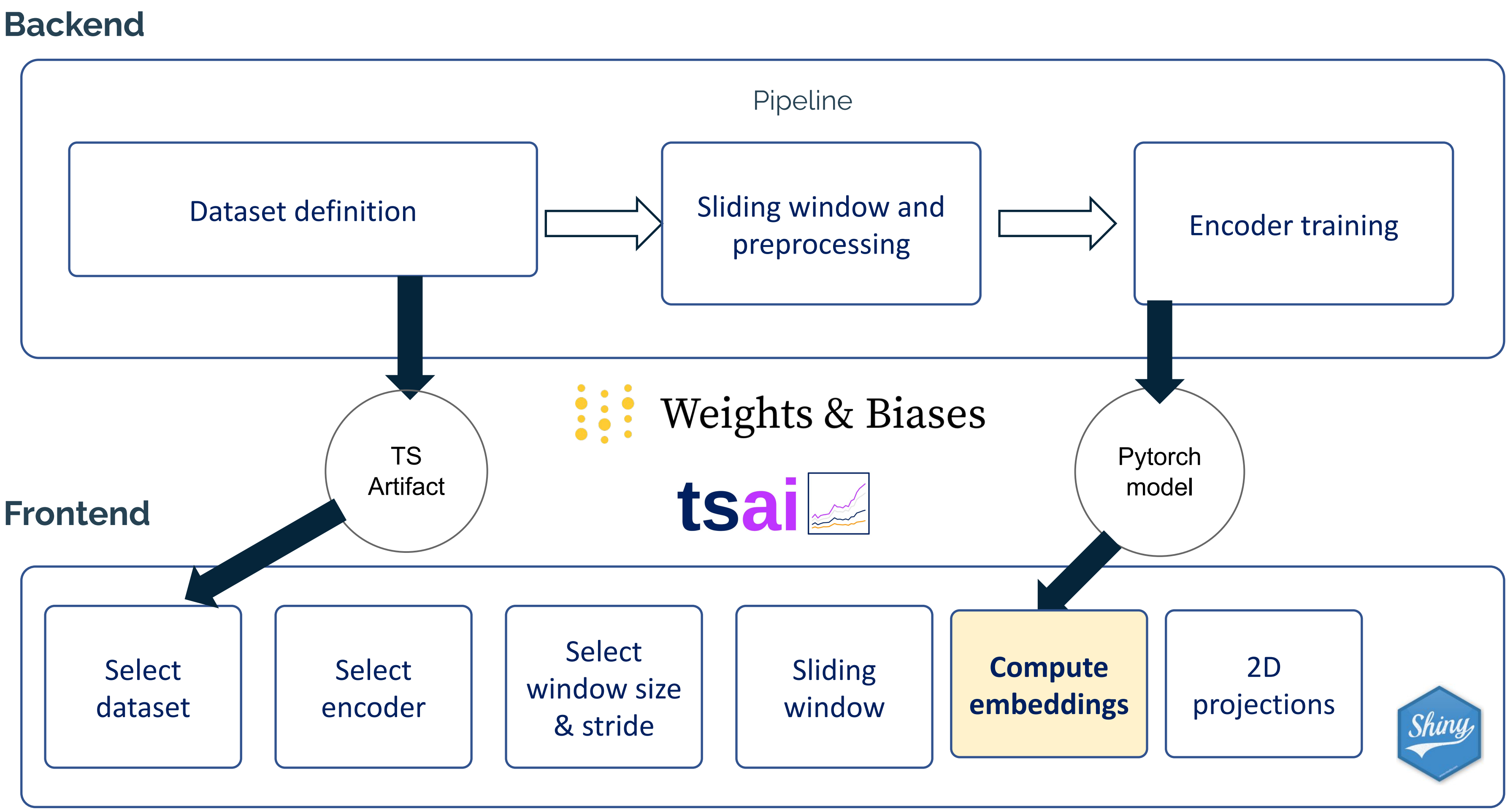
Related work

Ali, M., Jones, M.W., Xie, X. et al. TimeCluster: dimension reduction applied to temporal data for visual analytics. Vis Comput 35, 1013–1026 (2019). https://doi.org/10.1007/s00371-019-01673-y



TensorFlow’s embeddings projector




Backend
Pipeline
Dataset definition
Sliding window and preprocessing
Encoder training
TS Artifact
Weights & Biases
tsai
Pytorch model
Frontend
Select dataset
Select encoder
Select window size & stride
Sliding window
Compute embeddings
2D projections
Shiny

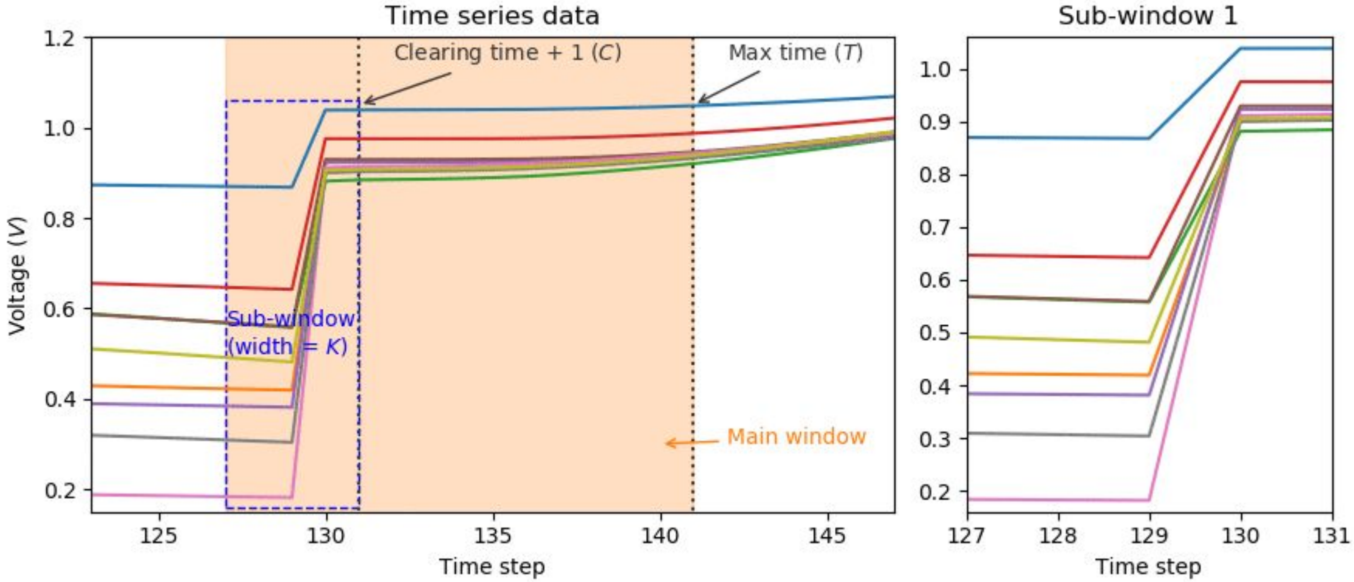
Backend

Dataset definition

- Univariate & Multivariate time series
- With or without natural timesteps
    - For now the time is not encoded as part of the data that goes through the network
- Regular timestamps
- 1 single series at a time
- Suitable for **long time series that present cyclical patterns**
- Once defined, the script logs the dataset as a **wandb artifact**.

Sliding window & preprocessing

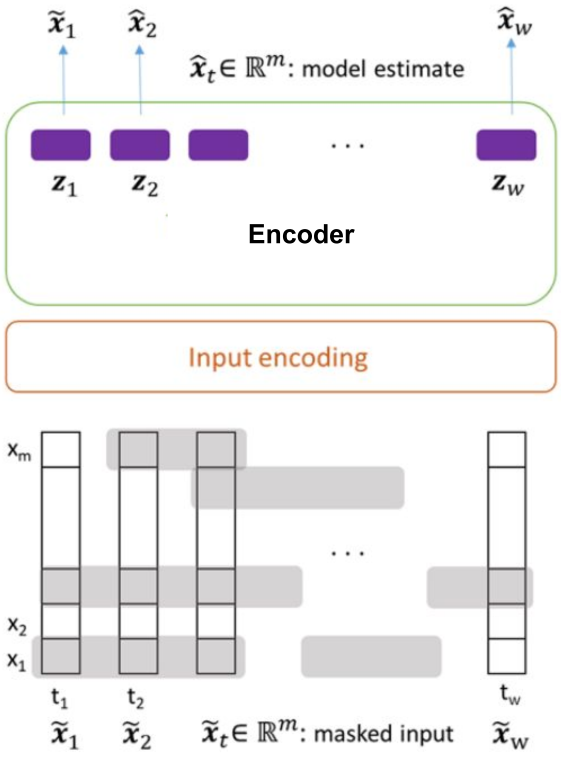
- In training, we normally use a **stride of 1** to achieve time smoothness in the embedding space
- The data normalization is done at **batch time.** Different datasets may require different datasets configuration for getting better encodings (by sample, by variable, by all the dataset…)



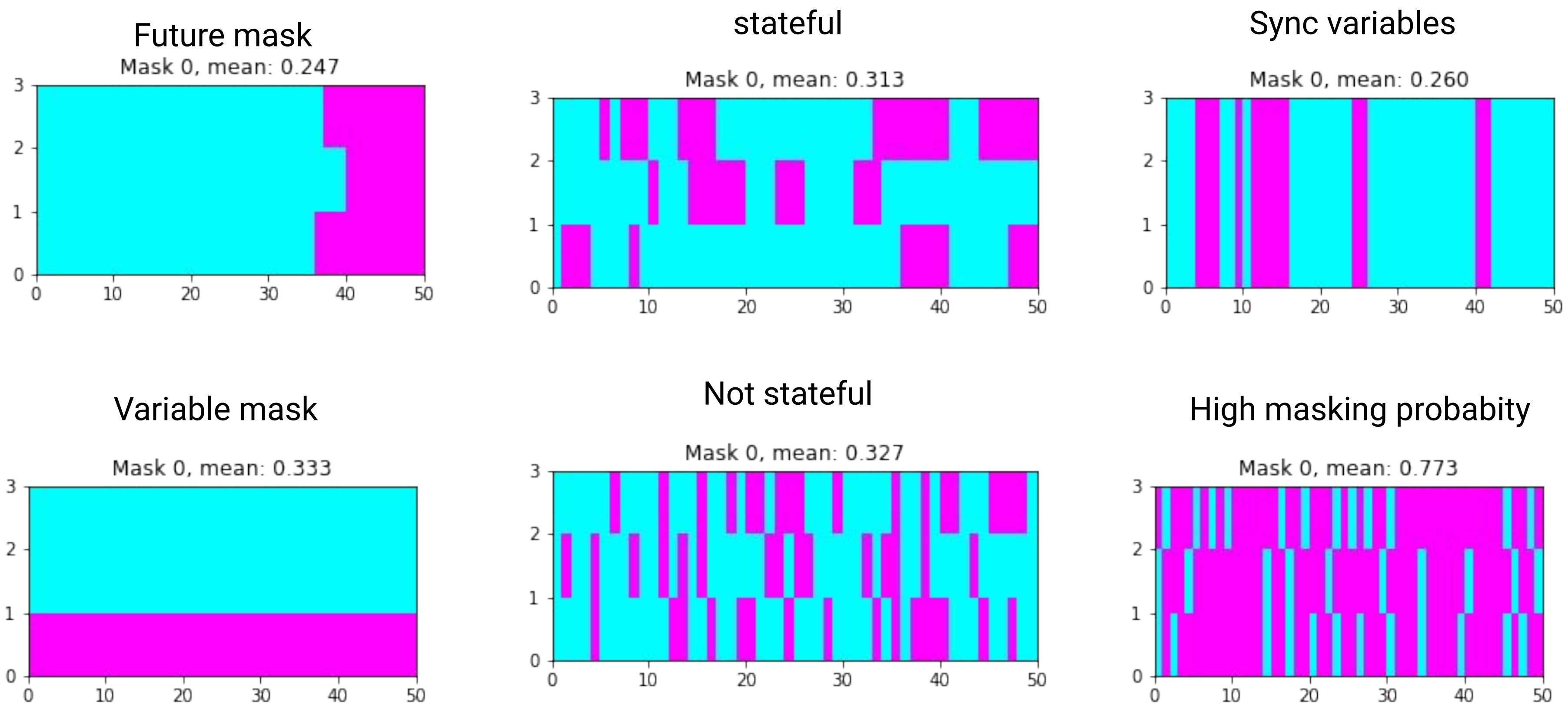
https://towardsdatascience.com/fast-and-robust-sliding-window-vectorization-with-numpy-3ad950ed62f5

Encoder: Masked autoencoder

- In the literature, the autoencoder used is a classic Deep Convolutional AutoEncoder
- Here, we learn only to reconstruct masked points in a time window, not the whole window
    - This is useful at detecting **point-based outliers**

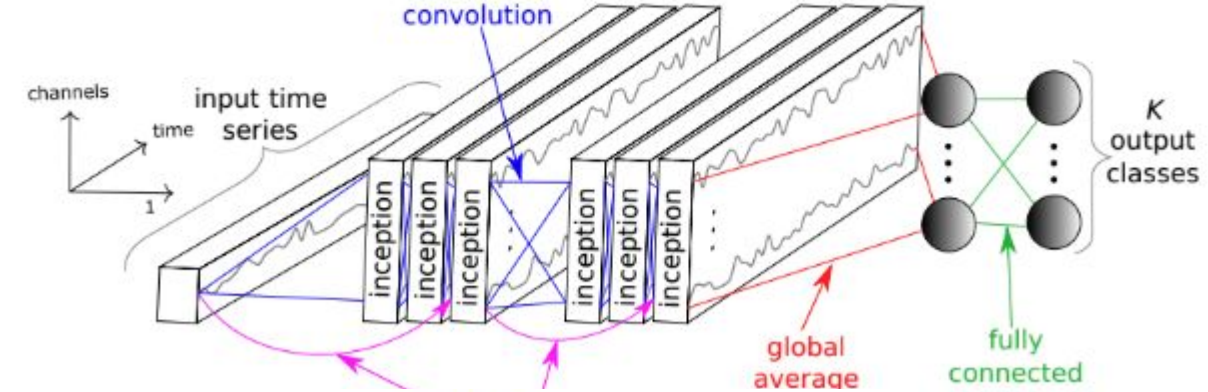


https://arxiv.org/pdf/2010.02803.pdf

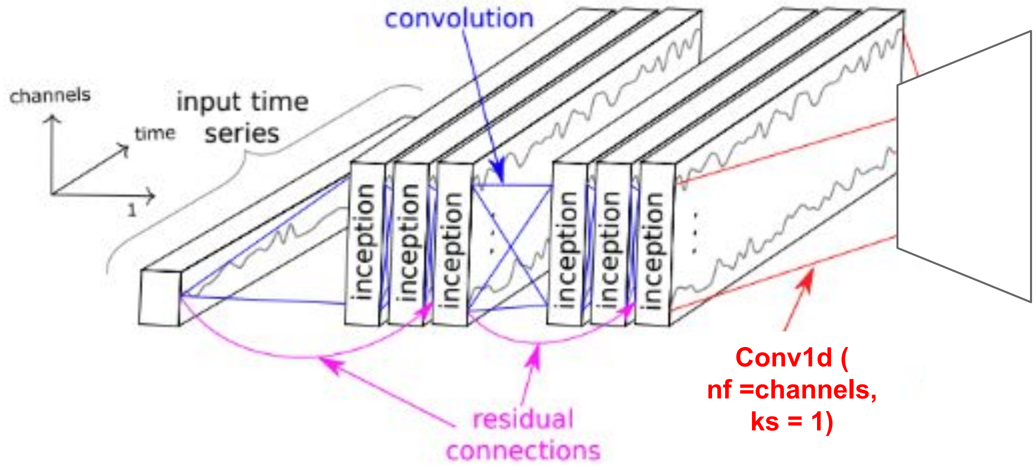


Masked encoders provide flexibility for training with different mask configurations. In all the examples a random time series of 3 variables and 50 time steps is used.

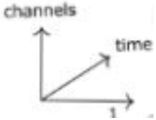
Original InceptionTime architecture



Masked Encoder

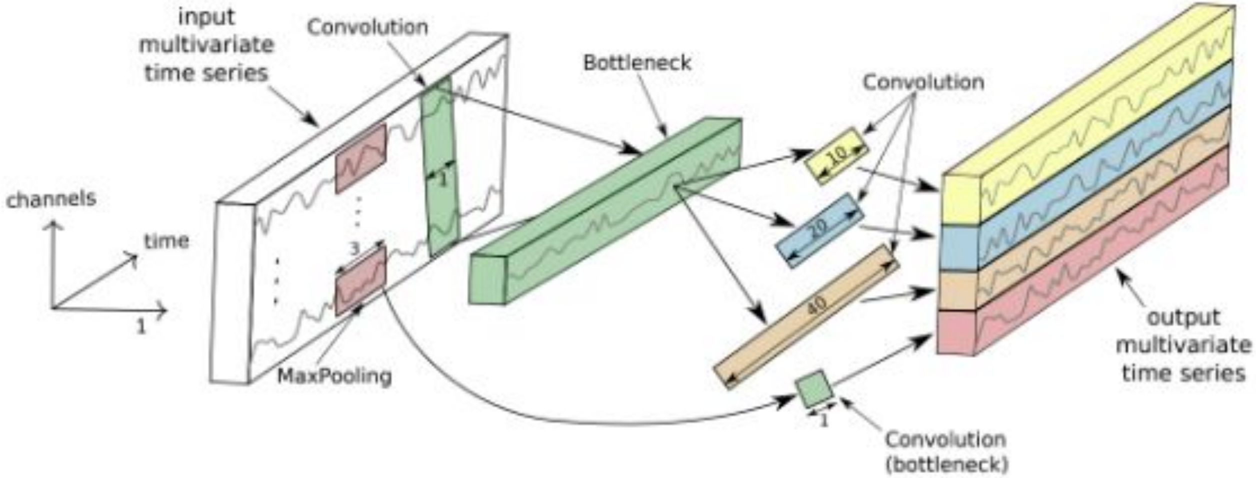


Output time series



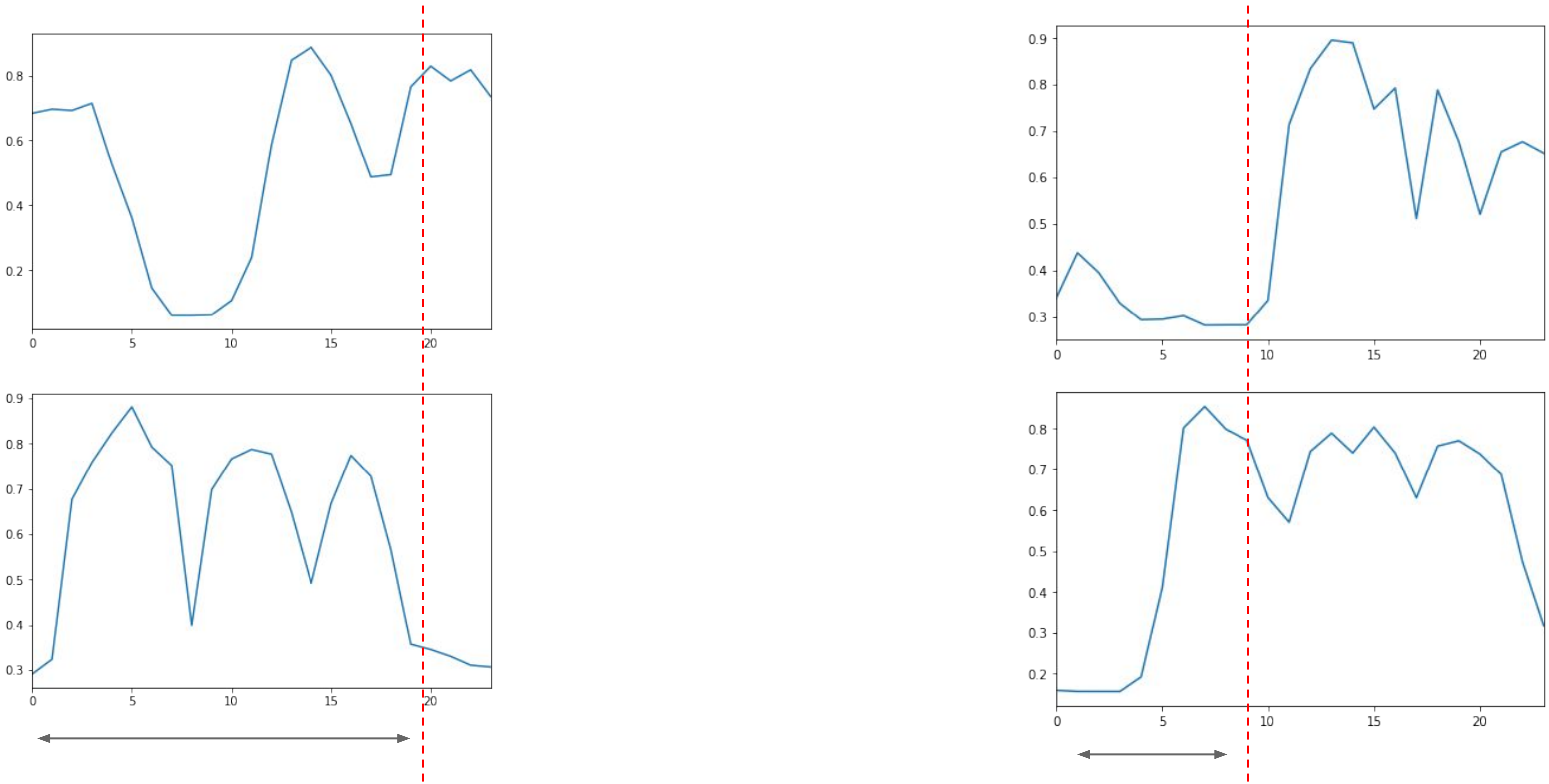
The masked autoencoder employs the architecture *InceptionTime* for time series classification, but changing the head (GAP + fully connected) for a ConvLayer with

https://arxiv.org/abs/1909.04939



Each InceptionModule within InceptionTime is based on convolution and pooling layers. This allows for the use of **variable window sizes** during training

Batch 1 Batch 2



Example of training the masked autoencoder with variable window sizes in ToIT dataset.

Additional encoder configurations

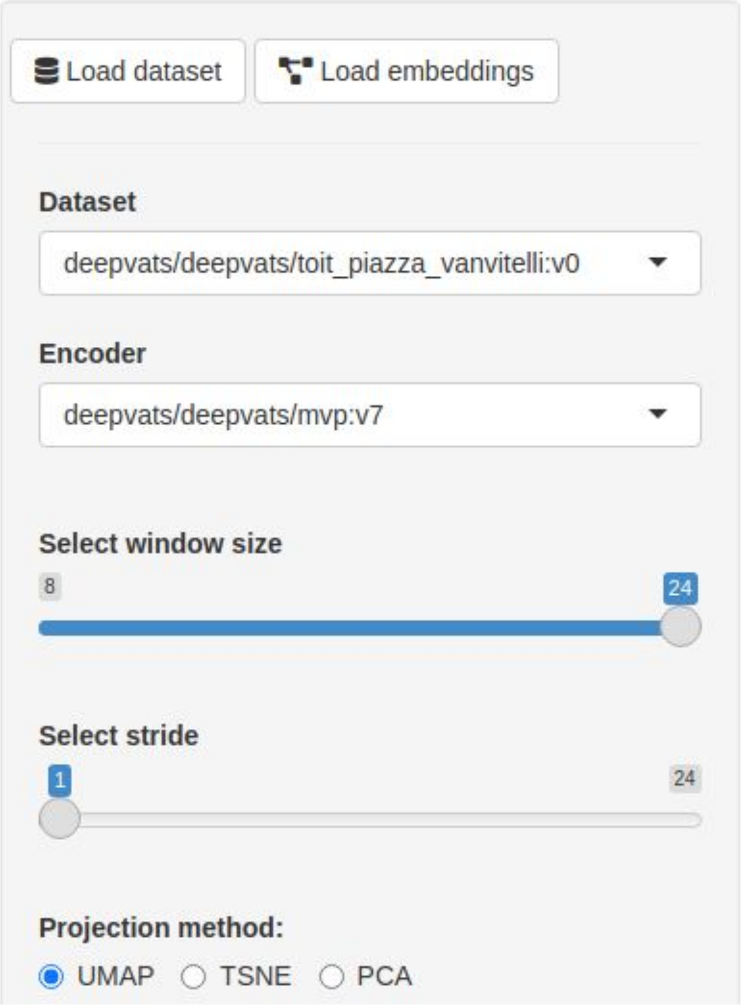
- Architecture
    - Number of Inception modules (depth)
    - Number of filters and size of filters in each of the IncepTion modules

- The learning rate comes from Pytorch/fastai's learning rate finder.
- The probability of masking (r)

Frontend

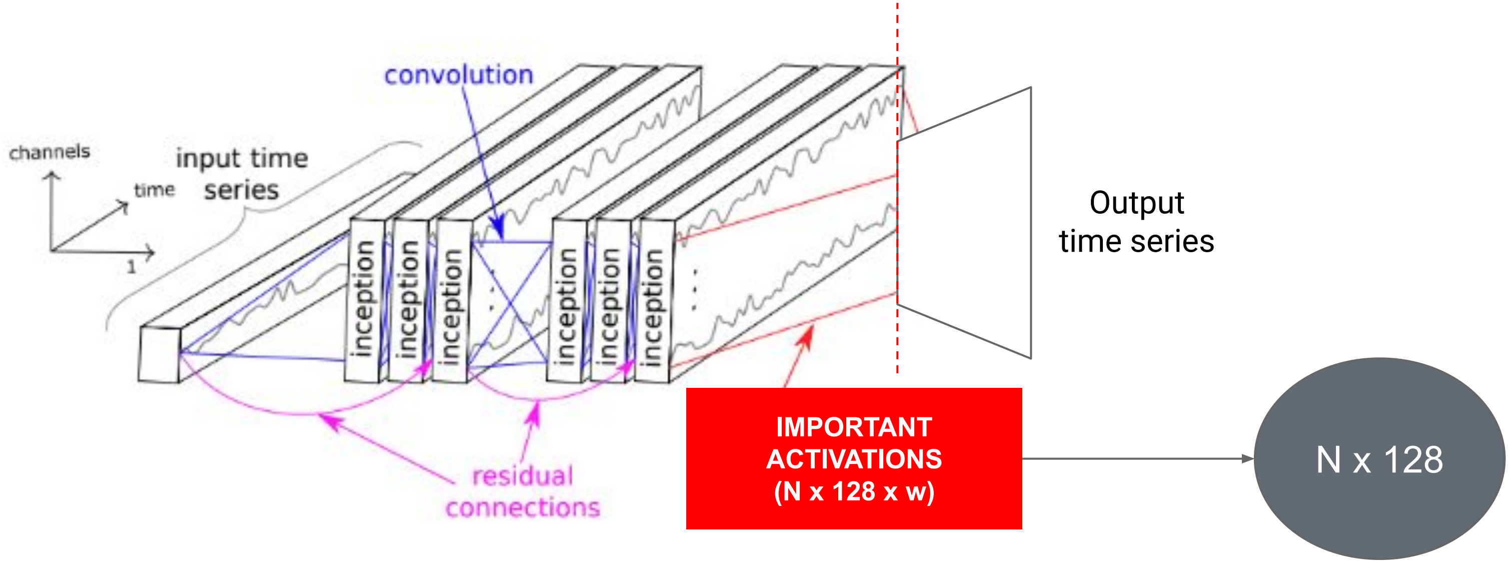
Selectors

- Dataset
  - Got from the TSArtifacts logged in wandb
  - Soon it will be possible to upload datasets directly here
- Encoder
- Window size
- Stride
- Projection algorithm
  - UMAP (default)
  - T-SNE
  - PCA

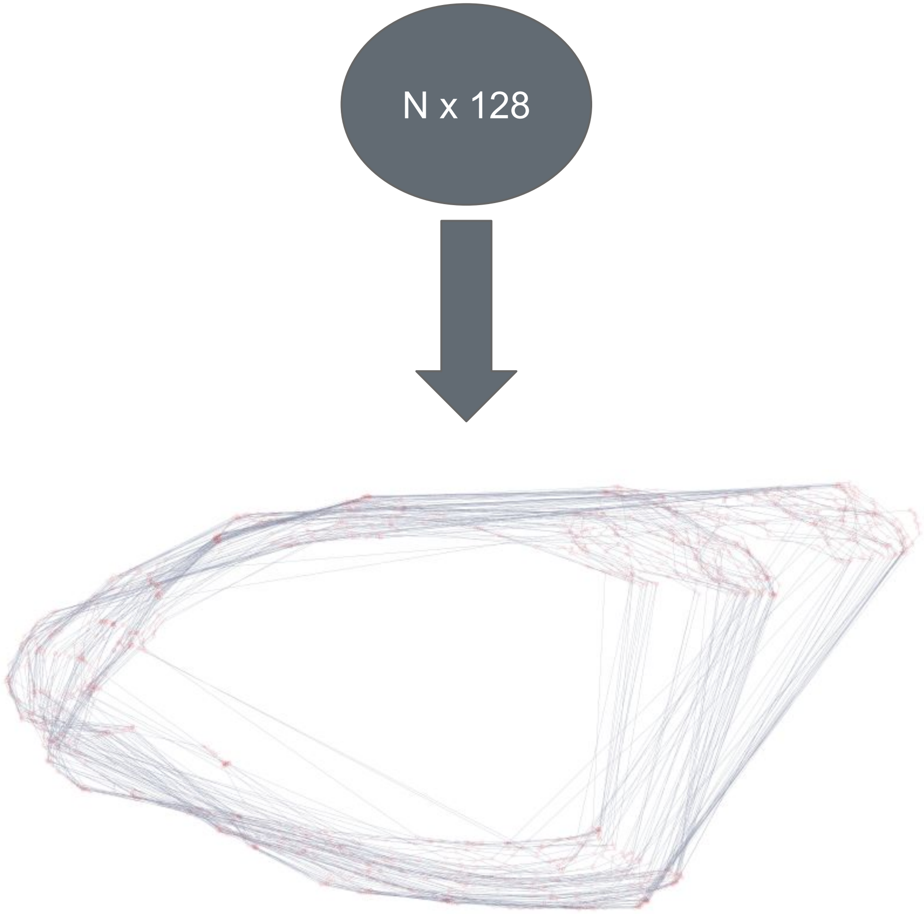


Compute embeddings

- Inference on the trained encoder. We take the activations **of the last layer before the head** (last Inception Module)
- For a [n x 1 x w] tensor, the activations will be [n x 128 x w]
- In order to get one activation vector per time window, we average on the time dimension (w)



2D projections

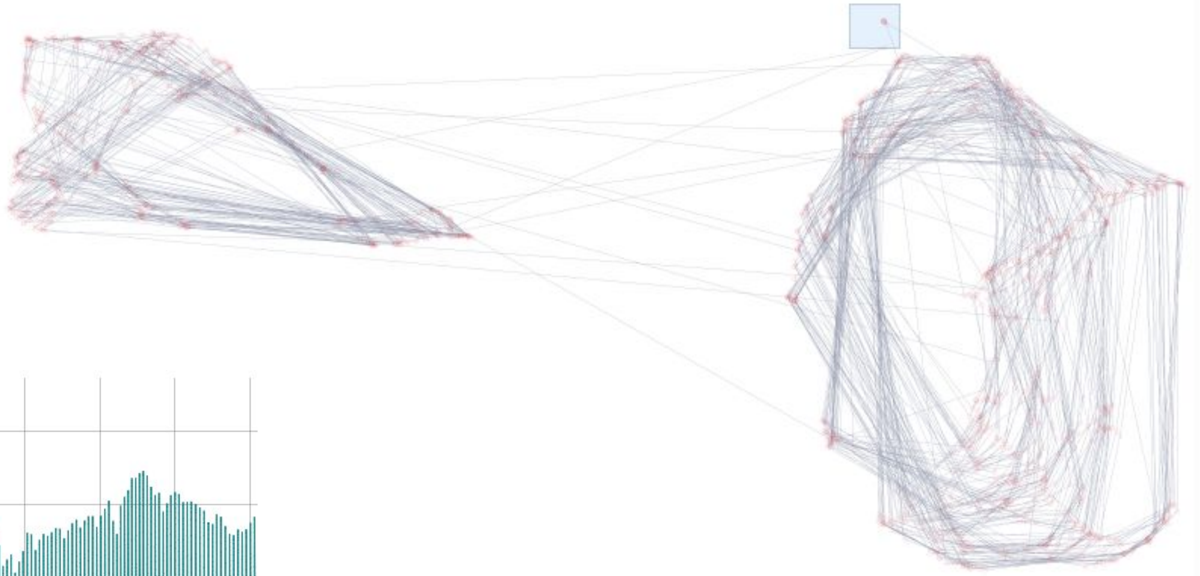


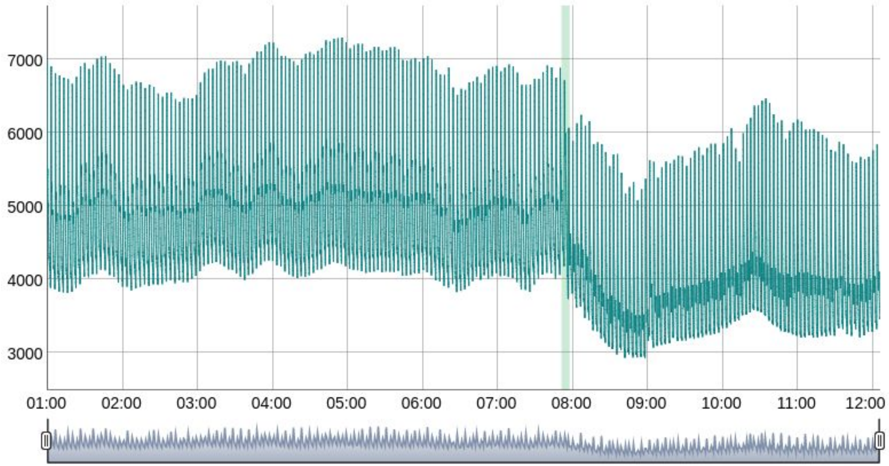
- **MADE ON GPU**
- 3 possible algorithms
  - UMAP
  - T-SNE
  - PCA

Only UMAP and T-SNE preserves temporal coherence in the embeddings

Experiments

Segmentation in Arterial Blood Pressure Data (ABP)

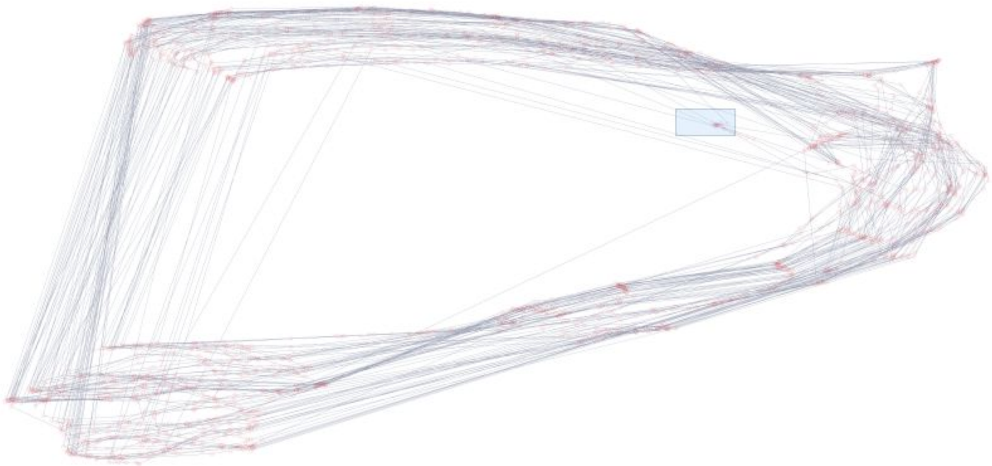


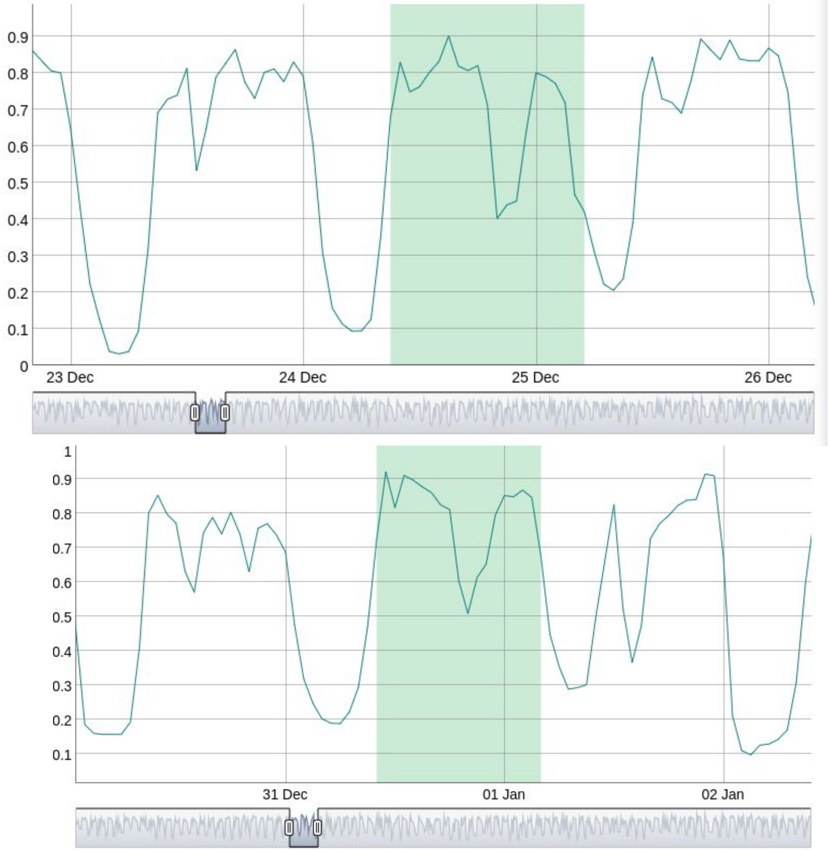


https://stumpy.readthedocs.io/en/latest/Tutorial_Semantic_Segmentation.html

Backend: Subsequence stateful masks, r=0.5, w = 210
Frontend: w=210, stride = 23, UMAP.

Outlier detection in ToiT dataset (piazza Vanvitelli)



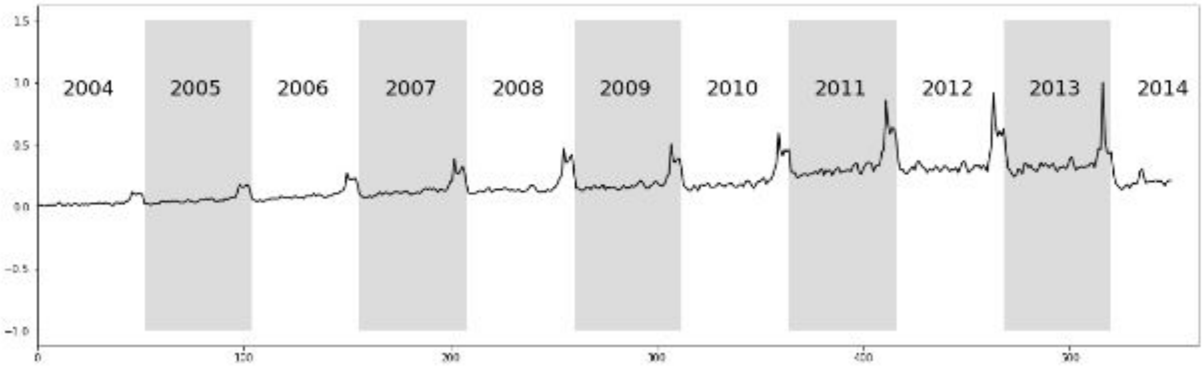


Backend: Subsequence NOT stateful masks, r=0.5, w = (8 hours, 24 hours)
Frontend: w=16 hours, stride = 1 hour, UMAP.

Other findings in initial experimentation

- Compared to the related work based on classic Deep Convolutional AutoEncoders, DeepVATS with **Masked Autoencoders detect better the outliers** and anomalies when these occurs in short intervals of time.

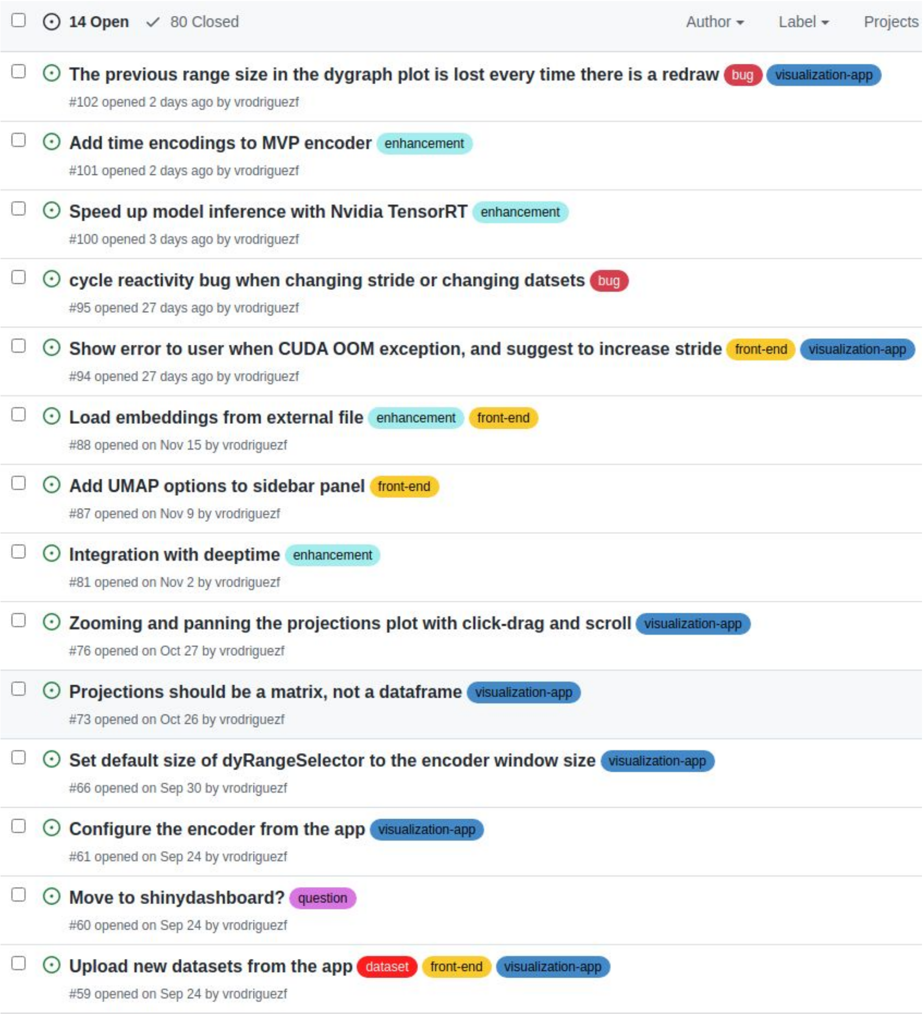
- When the data is not stationary (e.g. kohl dataset below), the projections do not show meaningful information



Future developments

TODO in the next 1-2 months (or so)

- Performance issues
- Expand experimentation
- **Write & submit the paper**
    - Ideas for appropriate journals/special issues?
- Project proposal around this topic



Software resources for time series

Python libraries/repositories for deep learning & time series

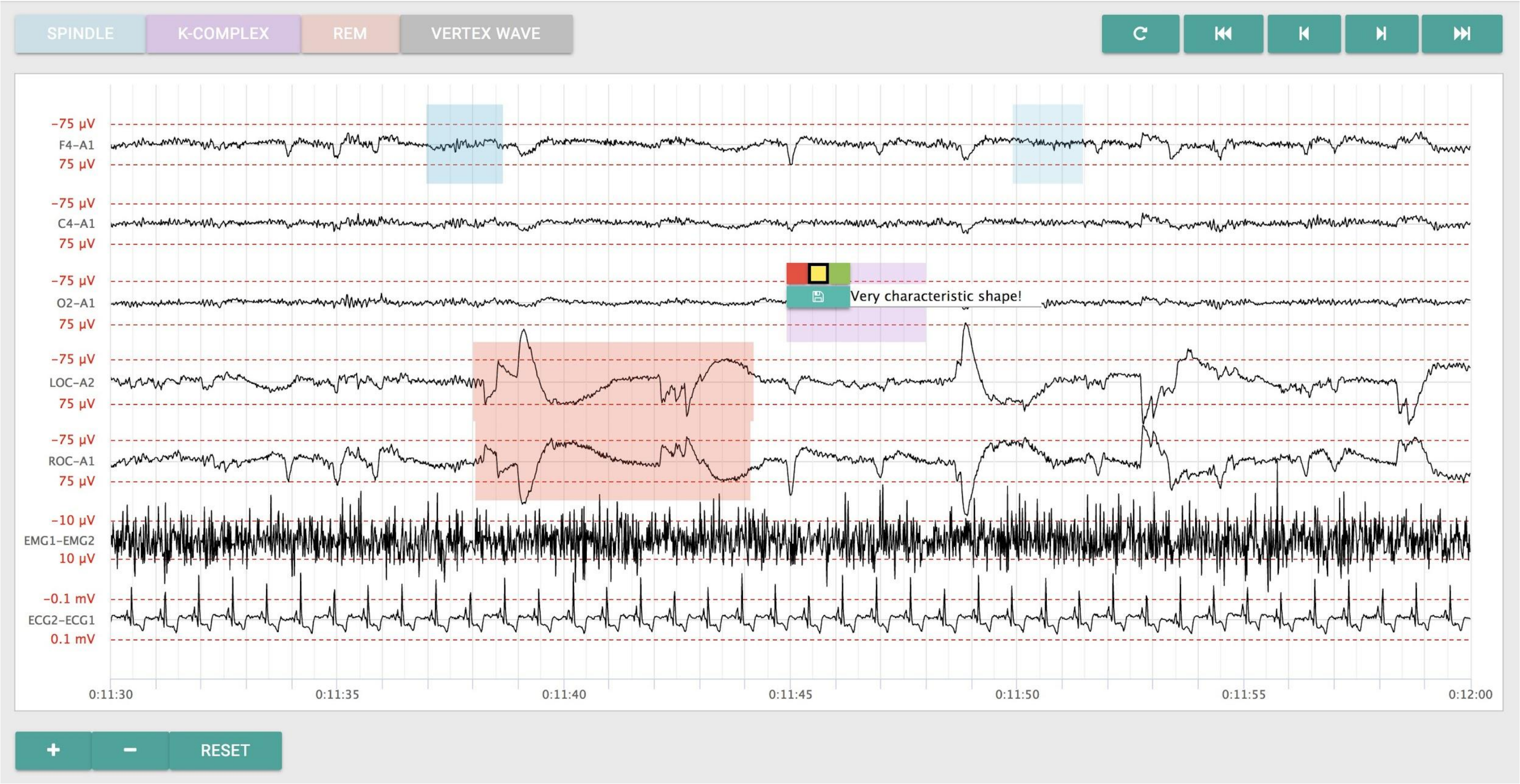
- **tsai**: This repository is focused on deep learning for time series classification using the fastai deep learning library.
- **NeuralProphet**: a Facebook Prophet extension that can quickly and interpretably make short to medium term forecasts
- **Pytorch-forecasting**: Implements state-of-the-art forecasting architectures (N-BEATS, TFT, Informer,...)
- **GluonTS**: Probabilistic time series forecasting in MXNet (by Amazon)
- **flow-forecast:** Time series forecasting in Pytorch (originally for flood forecasting)
- **deeptime:** Autoencoders and dimensionality reduction with neural networks

Annotation/labeling of time series

Labeled data (also known as the ground truth) is necessary for evaluating time series machine learning. Otherwise, one can not easily choose a detection method, or say method A is better than method B. The labeled data can also be used as the training set if one wants to develop supervised learning methods for detection.

Annotation tools provide visual ways to:

- Mark time windows with the presence of anomalies in time series
- Give categories (classes) to time series



SPINDLE
K-COMPLEX
REM
VERTEX WAVE
-75 µV
F4-A1
75 µV
-75 µV
C4-A1
75 µV
-75 µV
O2-A1
75 µV
Very characteristic shape!
-75 µV
LOC-A2
75 µV
-75 µV
ROC-A1
75 µV
-10 µV
EMG1-EMG2
10 µV
-0.1 mV
ECG2-ECG1
0.1 mV
0:11:30
0:11:35
0:11:40
0:11:45
0:11:50
0:11:55
0:12:00
RESET

Annotation tools

Curve - Curve is an open-source tool to help label anomalies on time-series data

TagAnomaly - Anomaly detection analysis and labeling tool, specifically for multiple time series (one time series per category)

time-series-annotator - The CrowdCurio Time Series Annotation Library implements classification tasks for time series.

WDK - The Wearables Development Toolkit (WDK) is a set of tools to facilitate the development of activity recognition applications with wearable devices.

Label Studio - Label Studio is a configurable data annotation tool that works with different data types

Modern deep learning approaches for time series

Victor Rodriguez-Fernandez
Polytechnic University of Madrid
victor.rfernandez@upm.es

vrodriguezf
@vrodriguezf90