CRASH COURSE TIME SERIES

Lecture 1: Deep Learning for Time Series

25/01/2024

THE TEAM WITH WHOM YOU WILL SPEND YOUR SESSION

Cadmos KAHALE-ABDOU

Consultant Data Scientist

cadmos.kahale-abdou@capgemini.com

Charles BOY DE LA TOUR

Consultant Data Scientist

charles.boy-de-la-tour@capgemini.com

AGENDA

1. Reminder on Time Series Analysis
2. Reminder on Time Series forecasting: Feature engineering
3. Hands-On presentation: Energy Consumption in France
4. The advantages of Deep Learning: RNN
5. LSTM and GRU

RECORD

01 - REMINDER ON TIME SERIES ANALYSIS

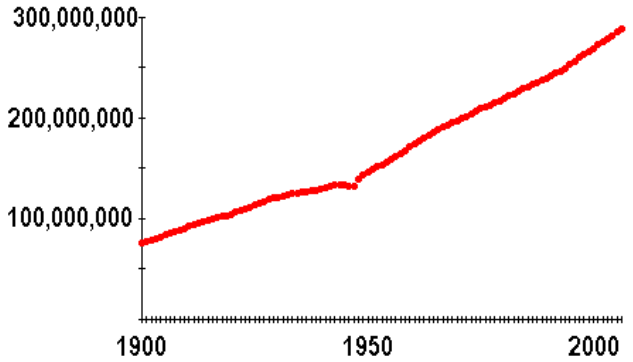
WHAT ARE TIME SERIES ?

Definitions

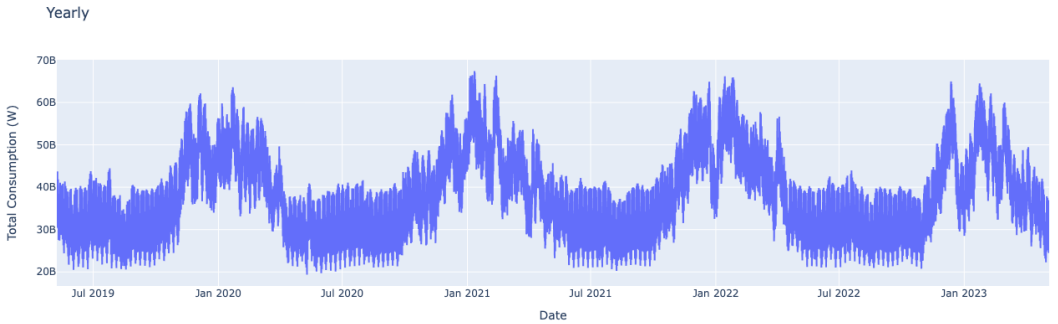
A time series is a **sequence** of **numerical data** points in **successive order**.
A stochastic process $\{\ldots, X_{t-1}, X_t, X_{t+1}, \ldots\}$ consisting of random variables indexed by time index t is a **time series**.

Time step (Δt): It is the interval between consecutive data points and is often consistent throughout the series such that : $y(t), y(t+\Delta t), y(t+2\Delta t), \ldots$

Examples



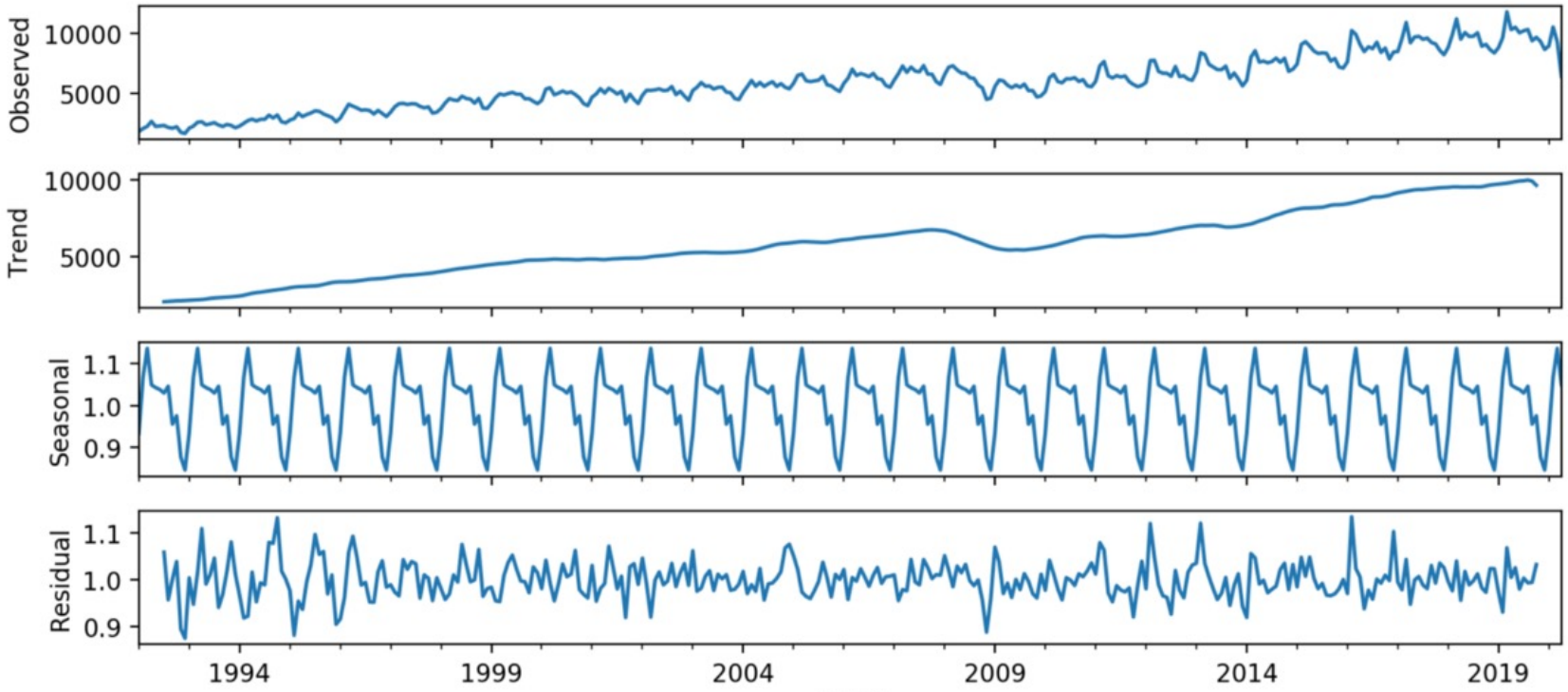
A time series graph of the population of the United States from 1900 to 2000.C.K. Taylor



Electricity consumption in France.
from july 2019 to january 2023. ENEDIS

ECG

DECOMPOSITION OF TIME SERIES: TREND, SEASONALITY AND CYCLE



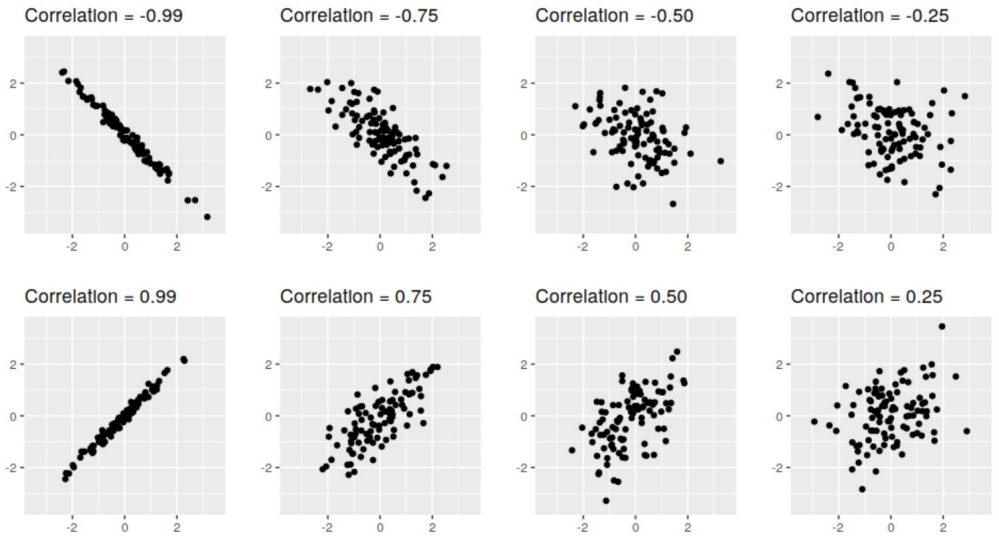
- **Trend**: long-term increase or decrease in the data.
- **Seasonality**: pattern occurs when a time series is affected by seasonal factors as the time of the year or the day of the week.
- **Cycle**: pattern occurs when the data exhibit rises and falls that are not of a fixed period, can be due to economic conditions and are often related to the « business cycle ». Without specific information, it is generally very difficult to dissociate trend and cycle and they are often treated together.
- **Noise:** random variability in the data that cannot be easily modeled or predicted.

MEASURING THE INFLUENCE OF THE PAST VALUES: CORRELATION VS AUTOCORRELATION

Definition

Correlation measures the strength of a linear relationship between two variables:

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}.$$



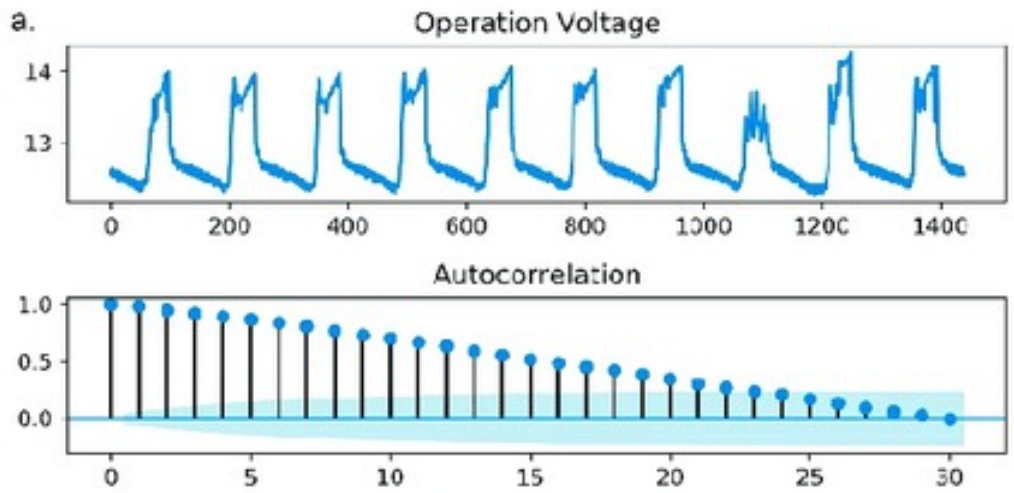
Example of the correlation plot between two variables

Definition

Autocorrelation is based on the same idea but instead of comparing two different variables it compares a time series and its **kth-lag**:

$$s_k = \frac{1}{n}\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x}) = \frac{1}{n}\sum_{i=k+1}^{n}(x_i - \bar{x})(x_{i-k} - \bar{x})$$

The ACF or correlogram is the representation of the autocorrelation for every lag: $\rho_k = \frac{s_k}{s_0}$



MEASURING THE INFLUENCE OF THE PAST VALUES: THE PACF – PARTIAL AUTO CORRELATION FUNCTION

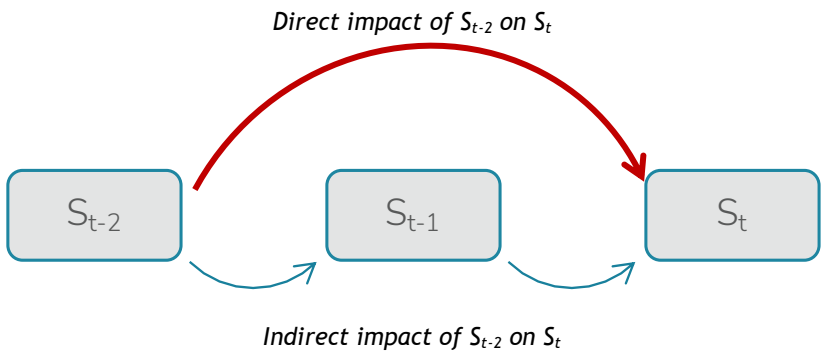
Definition

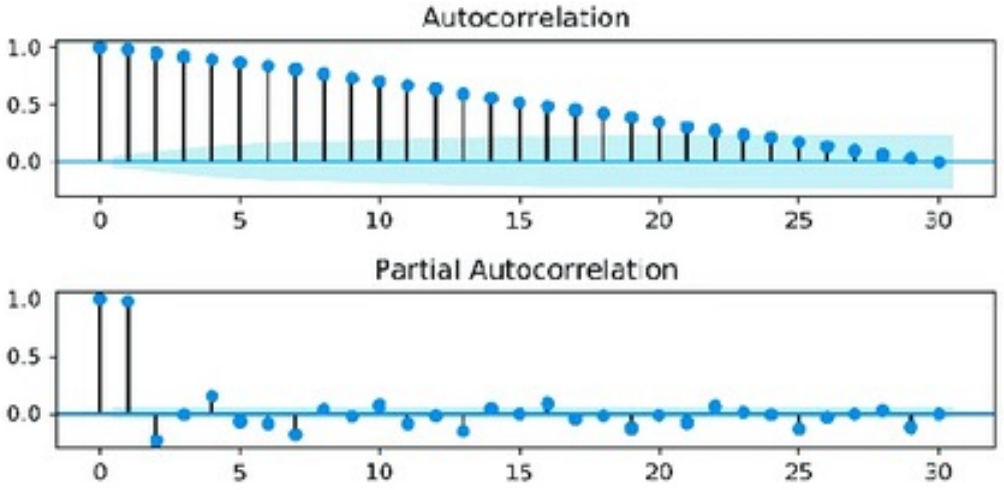
The partial autocorrelation function (PACF) measures the relationship in x_t and x_{t-k} after removing the effects of intermediate lags (lag 1, 2, …, k-1). Thus the correlations are isolated and the information already exploited is not reused.

- We denote r_k the k-ranked partial autocorrelation, which defines the coefficient of x_{i-k} in the linear regression on variables $1, x_{i-i}, \cdots$, such that: $x_i = \alpha_0 + \sum_{j=1}^{k-1} \alpha_j x_{j-1} + r_k x_{i-k} + \varepsilon_i$
- This implies that the partial autocorrelation is the correlation between x_i and x_{i-k} once all the ranks between $x_{i-k+1}, \cdots, x_{i-1}$ are defined and removed from the equation.
- k-ranked partial autocorrelation: $x_i = \alpha_1 x_{i-1} + \alpha_2 x_{i-2} + \cdots + \alpha_{k-1} x_{i-k+1} + U,$

$$x_{i-k} = \beta x_{i-1} + \beta_2 x_{i-2} + \cdots + \beta_{k-1} x_{i-k+1} + V,$$

$$r_k = corr(U, V)$$



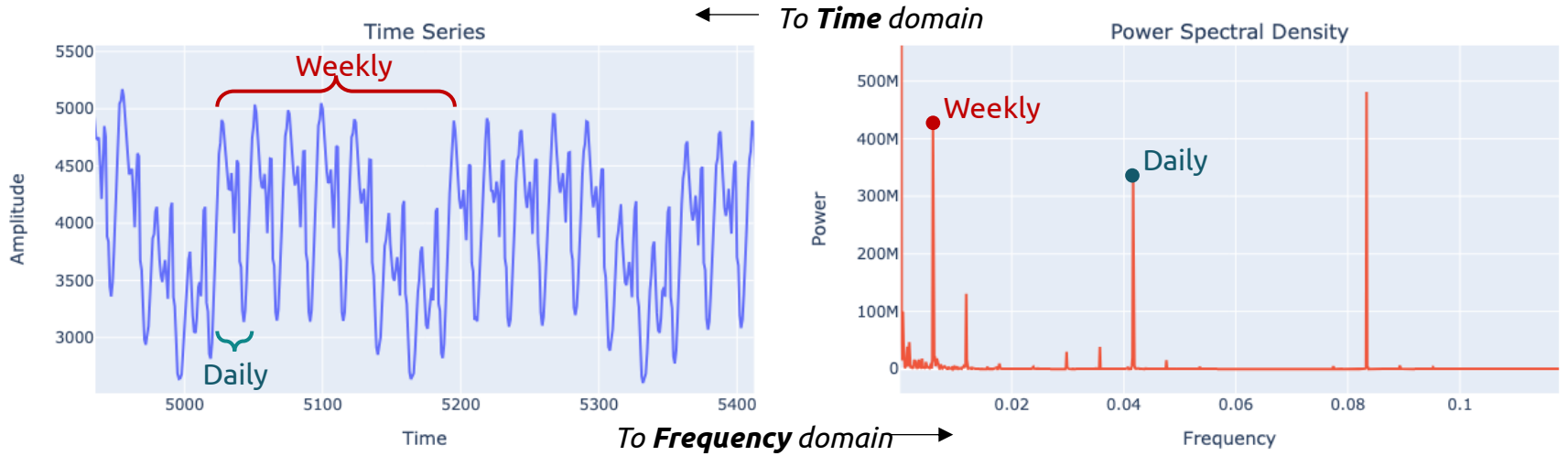


ANALYSES WITH TIME AND FREQUENCY

Frequency domain analyses, including FFT and spectrograms, are important for time series analysis as they reveal hidden periodicities and frequency components, crucial for understanding underlying patterns, especially in signals where time domain representations are less informative or more complex.

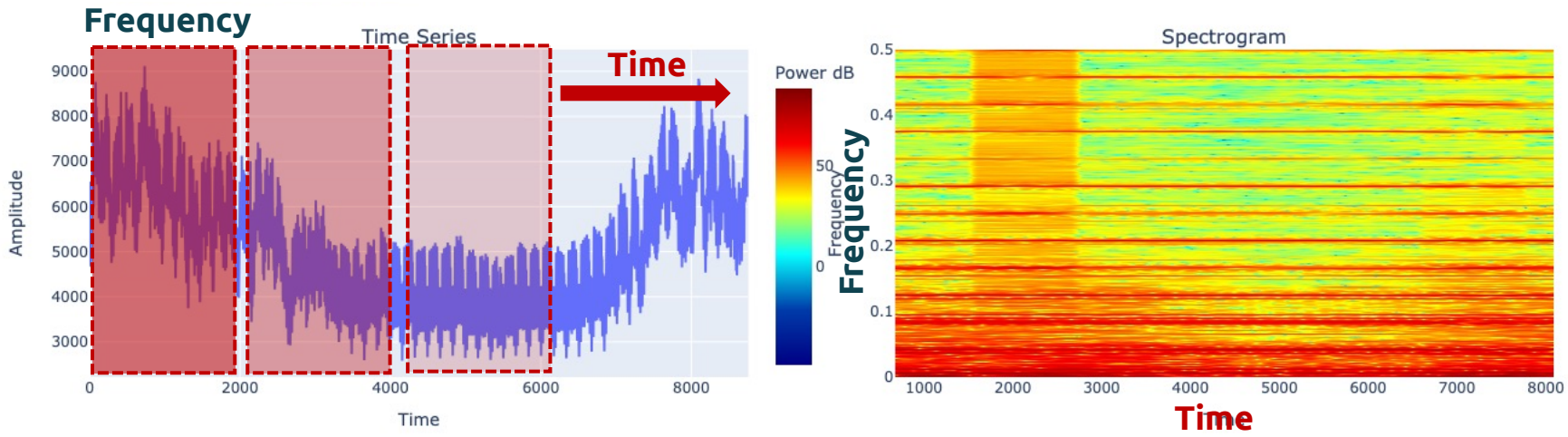
Frequancy domain

- *Reveal hidden periodicities*
- *Harmonic analysis*
- *Noise reduction & data compression*
- *Timeseries decomposition*



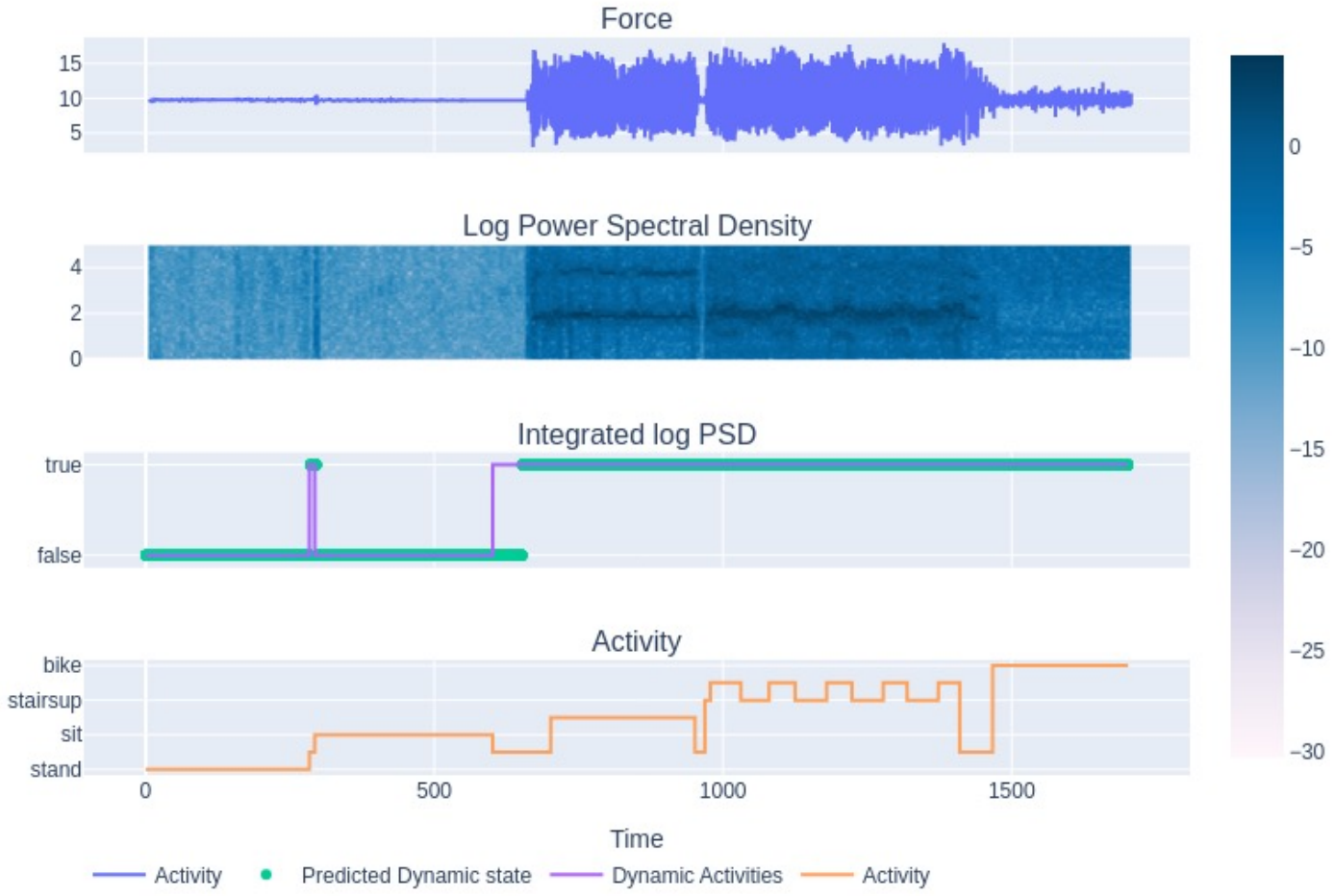
Time-Frequency analyses

- *Time-Localized Frequency Information*
- *Time-Localized Frequency Information*
- *Identifying Transient Features*
- *Detecting patterns*
- *Detecting anomalies*



FREQUENCY SPACE ANALYSES

Human Activity Recognition (HAR) through spectrogram characterization leverages the detailed time-frequency representation of sensor data to accurately identify and differentiate between various human movements and activities.

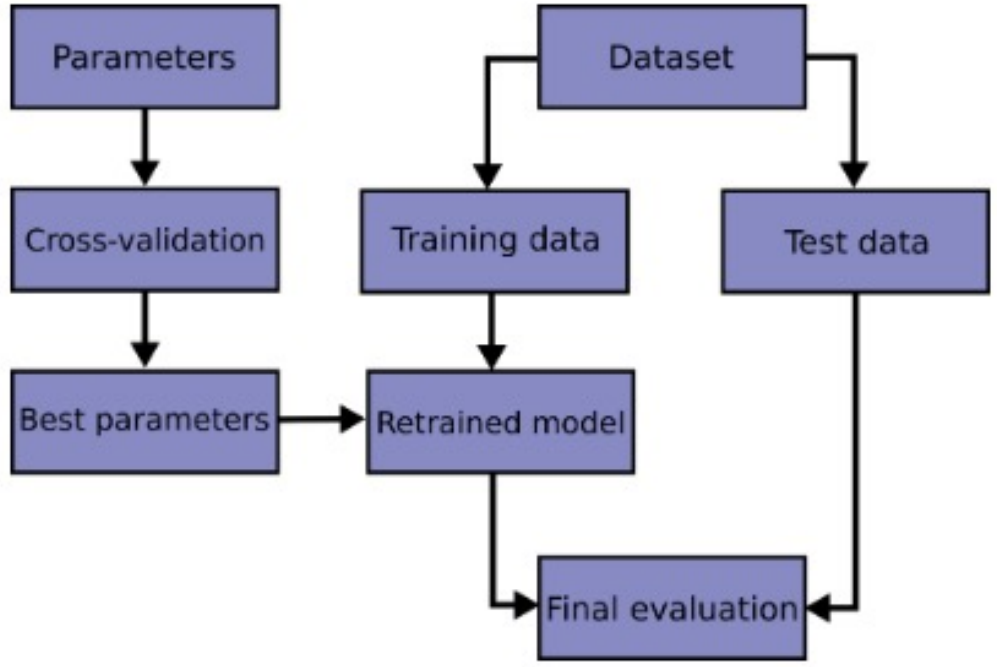


02 - REMINDER ON TIME SERIES FORECASTING: FEATURE ENGINEERING

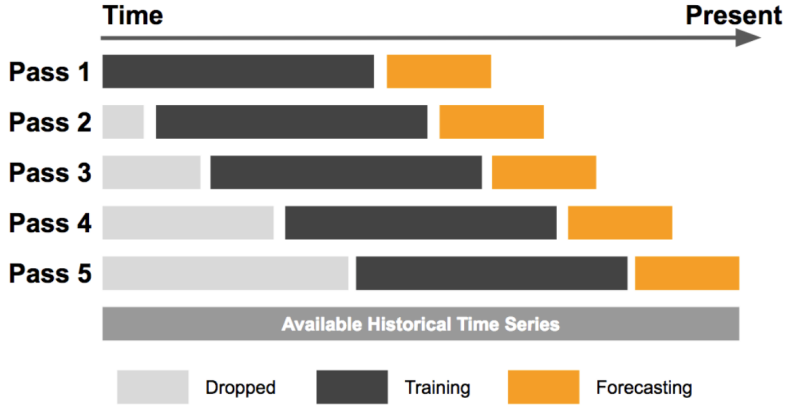
CROSS VALIDATION IN TIME SERIES

Why use cross validation?
- Optimise the model **hyperparameters**
- Avoid **overfitting**

Cross validation workflow



Sliding Window cross-validation



Increasing Window cross-validation

Période doit prendre en compte plus basse freq season

FEATURE ENGINEERING: USING LAGS AS FEATURES

Lag Features are **previous time points** in the serie used to predict future time points.

All lags are not equally informative: the lags to include as features can be determined based on **domain knowledge** and **autocorrelation** analysis.

	y	y_lag_1	y_lag_2
Date			
1954-07	5.8	NaN	NaN
1954-08	6.0	5.8	NaN
1954-09	6.1	6.0	5.8
1954-10	5.7	6.1	6.0
1954-11	5.3	5.7	6.1

RISKS:

Including too many lags can introduce **overfitting** :

- Alternatively, **rolling window statistics** such as mean, max, min can be informative.
- **Regularisation** methods such as l1 or l2.

Be careful to data leakage

FEATURE ENGINEERING: USING LAGS AS FEATURES

Lag Features are **previous time points** in the serie used to predict future time points.

All lags are not equally informative: the lags to include as features can be determined based on **domain knowledge** and **autocorrelation** analysis.

Some forecasting problems are framed as one-step ahead prediction. That is, predicting the next value of the series based on recent events. But, forecasting a single step is too narrow for many problems requiring to predict **n** steps ahead. **How to do so ?**

1.Recursive:
Train a single model for one-step ahead.
Predict one-step at a time and use the prediction you have made to predict the next step.

LIMITS: Iterating the same model with its own forecasts as input leads to propagation of errors.

2. Direct:
The Direct approach builds one model for each horizon.

LIMITS: requires more computational resources for the extra models, assumes that each horizon is independent.

3. DirectRecursive (chaining):
A model is built for each horizon (following Direct). But, at each step, the input data increases with the forecast of the previous model (following Recursive).

LIMITS: computational ressources.

4. Multi output
Some algorithms can naturally take multiple output variables (eg: LinearRegression, KNeighbors, Neural Networks, or Random Forests).

LIMITS: not all algorithms can predict multiple output.

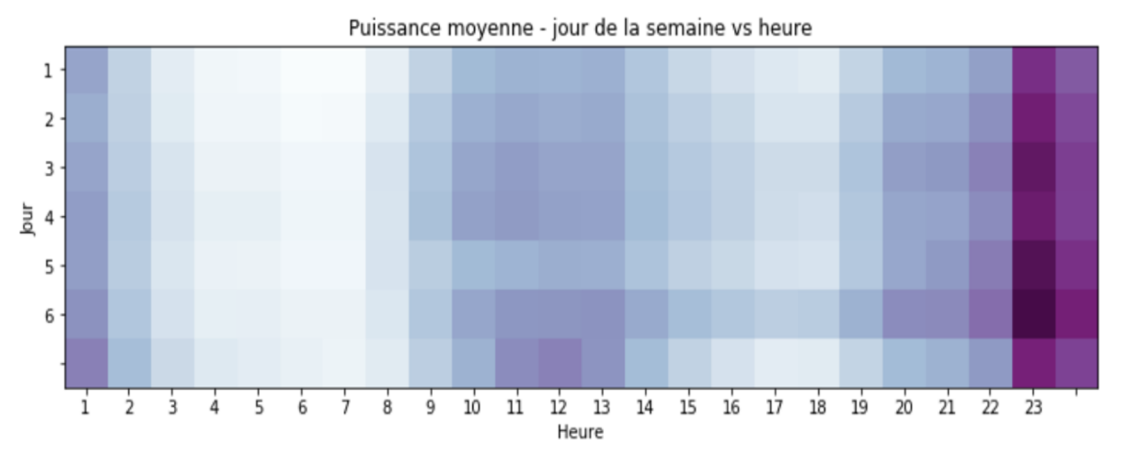
FEATURE ENGINEERING: CALENDAR FEATURES

The date often has a great impact on the prediction (**year**, **quarter**, **month**, or even sometimes the **day** of the week). They witness the possible seasonality in the data for example.

Thus it is often a great source of information that one should integrate in the model's features.
We often decompose the date first (year/month/day/hour) as each component brings its own piece of information.
Feature transformations can also be done (aggregation, encoding, etc.).

How to integrate seasonality?

- Insert lags with an offset equal to that seasonality
- Add columns with the season
- Add columns that group similar values together
- Etc.

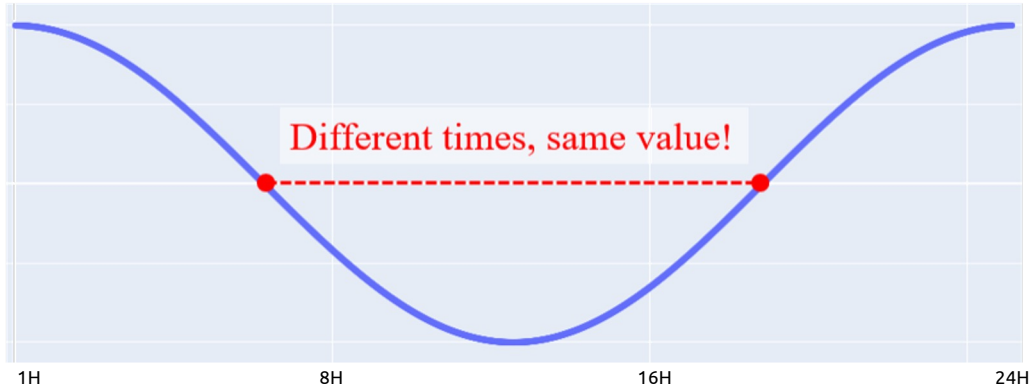


FEATURE ENGINEERING: CYCLICAL FEATURES ENCODING

Hours of the day, days of the week, months of the year occurs in cycles : proper encoding preserves the **continuity** and **natural order** of cycles (12am and 1am are close to each other).

1. Convert to 1 dimension with sin/cos functions:

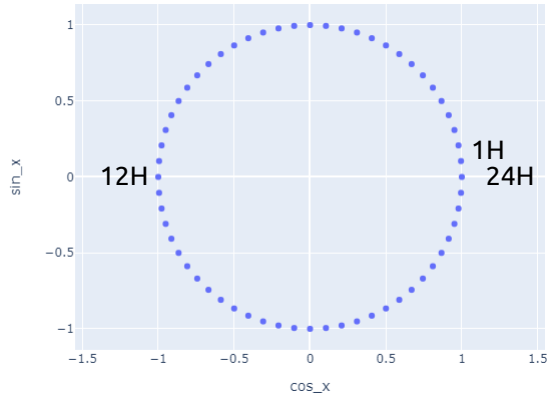
- **Methods**: run the cosine function after normalizing the variable between 0 to 2π.
- **Limits**: Two different times would get the same value



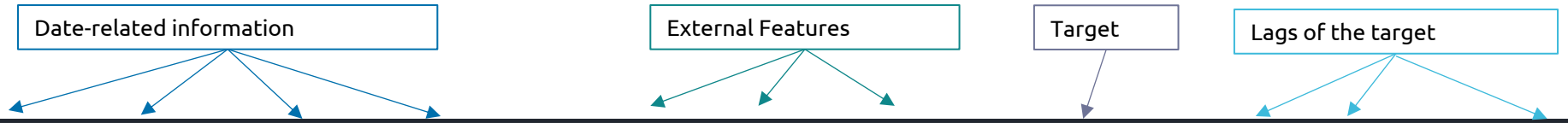
2. Convert to 2 dimensions – map cyclical variables to a unit cirlcle:

- **Methods**:

$$x = \sin\left(\frac{a \times 2\pi}{\max(a)}\right) \qquad y = \cos\left(\frac{a \times 2\pi}{\max(a)}\right)$$



EXOGENOUS DATA - UNIVARIATE VS MULTIVARIATE



timestamp	Jour de la semaine	Heure	Semaine/Weekend	Jour/Nuit	Direction du vent	Vitesse du vent	Température	Target	Target_lag_1	Target_lag_12	Target_lag_24
2016-05-31 01:00:00	1	1	0.0	1.0	340.0	5.1	8.0	230.4	NaN	NaN	NaN
2016-05-31 02:00:00	1	2	0.0	1.0	350.0	5.7	8.2	232.8	230.4	NaN	NaN
2016-05-31 03:00:00	1	3	0.0	1.0	340.0	5.7	8.3	233.6	232.8	NaN	NaN
2016-05-31 04:00:00	1	4	0.0	1.0	350.0	4.6	9.0	236.4	233.6	NaN	NaN
2016-05-31 05:00:00	1	5	0.0	1.0	350.0	6.2	8.8	285.2	236.4	NaN	NaN
2016-05-31 06:00:00	1	6	0.0	1.0	350.0	5.7	9.0	316.8	285.2	NaN	NaN
2016-05-31 07:00:00	1	7	0.0	0.0	350.0	6.7	9.3	376.4	316.8	NaN	NaN
2016-05-31 08:00:00	1	8	0.0	0.0	350.0	5.7	9.7	463.2	376.4	NaN	NaN
2016-05-31 09:00:00	1	9	0.0	0.0	350.0	6.2	10.5	543.0	463.2	NaN	NaN
2016-05-31 10:00:00	1	10	0.0	0.0	340.0	5.7	11.1	552.2	543.0	NaN	NaN
2016-05-31 11:00:00	1	11	0.0	0.0	350.0	4.1	11.2	554.2	552.2	NaN	NaN
2016-05-31 12:00:00	1	12	0.0	0.0	350.0	5.7	11.1	567.8	554.2	NaN	NaN
2016-05-31 13:00:00	1	13	0.0	0.0	350.0	5.1	11.5	554.6	567.8	230.4	NaN
2016-05-31 14:00:00	1	14	0.0	0.0	350.0	4.1	11.2	541.2	554.6	232.8	NaN
2016-05-31 15:00:00	1	15	0.0	0.0	360.0	3.6	11.0	525.0	541.2	233.6	NaN
2016-05-31 16:00:00	1	16	0.0	0.0	360.0	6.2	11.2	506.0	525.0	236.4	NaN
2016-05-31 17:00:00	1	17	0.0	0.0	360.0	6.2	11.0	468.6	506.0	285.2	NaN
2016-05-31 18:00:00	1	18	0.0	0.0	360.0	6.2	11.1	401.6	468.6	316.8	NaN
2016-05-31 19:00:00	1	19	0.0	0.0	360.0	5.7	11.3	330.2	401.6	376.4	NaN
2016-05-31 20:00:00	1	20	0.0	0.0	360.0	4.6	11.5	285.4	330.2	463.2	NaN

03 - HANDS-ON PRESENTATION

ENERGY CONSUMPTION IN FRANCE

ELECTRICITY MARKET CHALLENGES

The demand for electricity in the world continues to increase, this resource is used in all sectors. Any power failure has very significant economic consequences.

In recent months, the electricity market has experienced tensions in France for the following reasons:

- Low availability of the nuclear fleet
- Price increase due to gas price increase

For the first time the risk of power failure cuts was very high this winter.

To avoid that, demand must always equal production. This is where the forecast comes in.

WHY DO WE NEED TO FORECAST ENERGY DEMAND ?

Short-term planning of supply

Operational planning

- Real-time production planning
- Managing imports & exports

Multi-objective

1) Meet the demand
2) Manage energy mix (dispatchable vs. non-dispatchable sources)

Just-in-time production

Long-term planning of supply

Development strategy

- Building infrastructure
- Training and recruiting workers

Multi-objective

1) Environmental footprint trajectory
2) Costing strategy
3) National Sovereignty

Long term investments

MAIN PLAYERS AND THEIR ROLES

Production → Power Grid → Client supply

Electricity Producers

- ENGIE and EDF : 95% of the French production
- A producer can also be a supplier, but the two activities are distinct

EDF

ENGIE

TOTAL direct energie

Electricity Transmission System Operators

- Long distance transport : high-medium voltage
- One unique company

RTE

Electricity Distribution System Operators

- Local transport and distribution: medium-low voltage
- +150 local companies

ENEDIS

GÉRÉDIS Deux-Sèvres

Electricity Suppliers

- Free market: 36 suppliers in France
- EDF: major player

EDF ENGIE

*Lists of electricity producers, distribution system operators and suppliers are not exhaustive

DATASETS FEATURES :

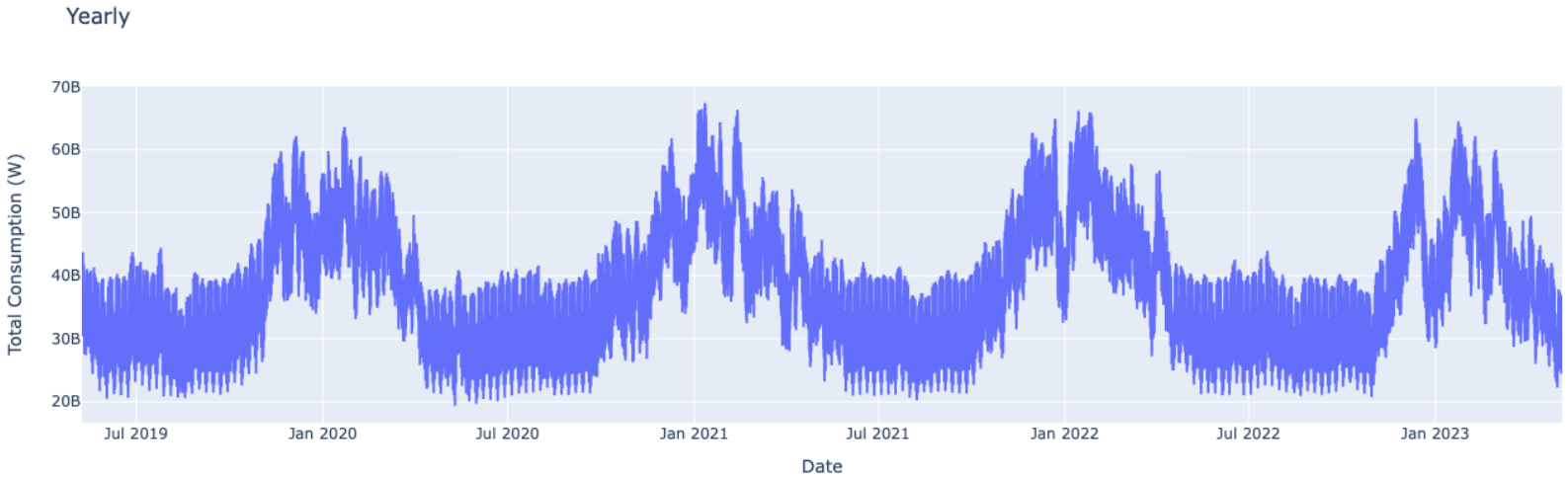
Consommation quotidienne brute régionale

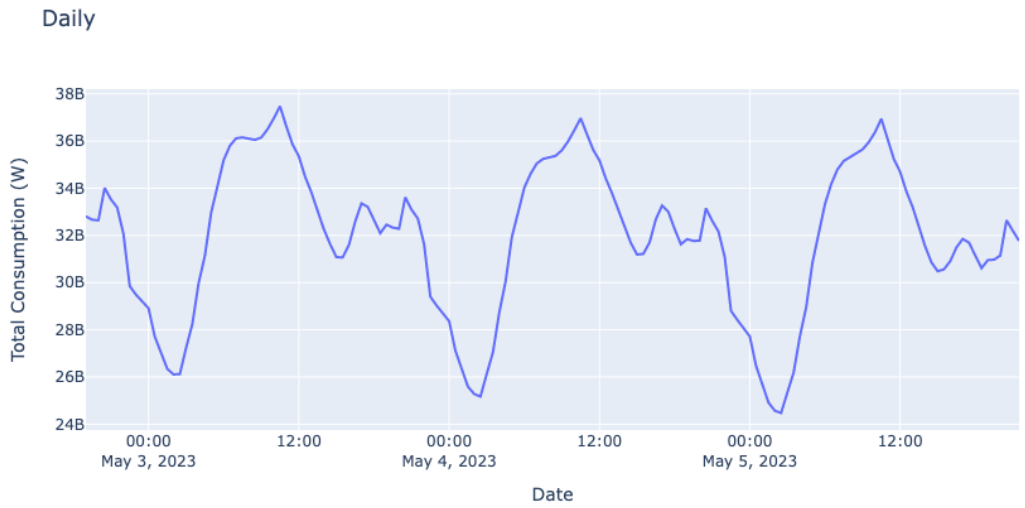
Group	Feature	Type
Time	Date - Heure	Datetime
	Heure	Datetime
	Date	Datetime
Région	Code INSEE région	str
	Région	str
Consumption	Consommations totale (W)	Decimal
	Consommation brute gaz (MW PCS 0°c) – GRTgaz	Decimal
	Statut - GRTgaz	Str
	Statut - Teréga	str
	Consommation brute gaz (MW PCS 0°c) – Teréga	Decimal
	Consommation brute gaz totale (MW PCS 0°c)	Decimal
	Consommation brute électricité (MW) - RTE	Decimal
	Statut - RTE	str
	Consommation brute totale (MW)	Decimal

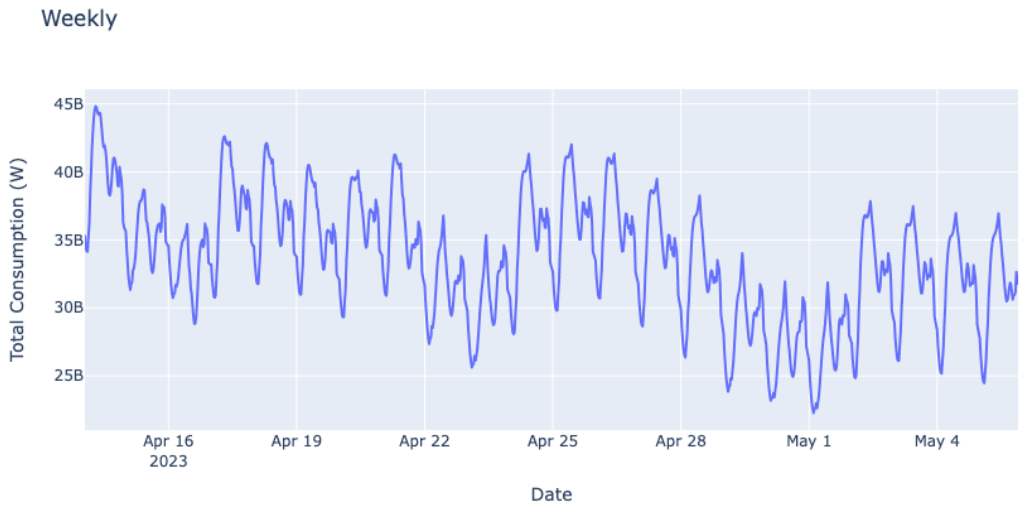
Observation météorologique historiques France (SYNOP)

	Date	region (name)	department (name)	communes (name)	Pression au niveau mer	Variation de pression en 3 he...	Direction du vent moyen 10 mn	Vitesse du vent moyen 10 mn	Température	Température (°C)	Point de rosée	Humidité	Visibilité horizontale	Nebulosité totale	Nébulosité des nuages de l' éta...	Hauteur de la base des nuage...	Type des nuages de l'étag
1	22 janvier 2024 19:00	Normandie	Manche	La Hague	102 080 Pa	360 Pa	260	14 m/s	283,95	10,800 °C	280,05	77 %					
2	22 janvier 2024 19:00	Bretagne	Côtes-d'Armor	Perros-Guirec	102 410 Pa	330 Pa	230	8,3 m/s	283,65	10,5 °C	280,35	80 %					
3	22 janvier 2024 19:00	Occitanie	Aveyron	Millau	103 070 Pa	90 Pa	220	4 m/s	280,65	7,5 °C	278,65	87 %	11 330 m				
4	22 janvier 2024 19:00	Bretagne	Ille-et-Vilaine	Saint-Jacques-de-la-Lande	102 590 Pa	380 Pa	230	3,3 m/s	282,95	9,800 °C	277,95	71 %	32 600 m		2	1 250 m	
5	22 janvier 2024 19:00	Centre-Val de Loire	Indre-et-Loire	Parçay-Meslay	102 620 Pa	370 Pa	240	4,8 m/s	282,85	9,700 °C	280,15	83 %	16 710 m		0		
6	22 janvier 2024 19:00	Nouvelle-Aquitaine	Gironde	Mérignac	103 250 Pa	360 Pa	250	3,4 m/s	284,15	11 °C	283,25	94 %	9 020 m		1	800 m	
7	22 janvier 2024 19:00	Guadeloupe	Guadeloupe	Les Abymes	101 940 Pa	-250 Pa	60	6,5 m/s	301,55	28,400 °C	294,05	64 %	58 250 m				
8	22 janvier 2024 19:00	Hauts-de-France	Somme	Abbeville	101 880 Pa	340 Pa	250	9,3 m/s	282,35	9,200 °C	278,85	79 %	17 730 m				
9	22 janvier 2024 19:00	Grand Est	Meurthe-et-Moselle	Thuilley-aux-Groseilles	102 230 Pa	390 Pa	230	6,6 m/s	280,95	7,800 °C	279,45	90 %	41 790 m	100 %	6	450 m	
10	22 janvier 2024 19:00	Occitanie	Haute-Garonne	Blagnac	103 260 Pa	230 Pa	220	1,7 m/s	284,55	11,400 °C	280,45	76 %	15 470 m	100 %	3	1 250 m	

FLUCTUATION OF ELECTRICTY CONSUMPTION







POWER CONSOMPTION IN FRANCE

Key Figures for France

25%

of energy used is electricity

474 TWh

Yearly power consumption

7 MWh

Yearly power consumption per capita

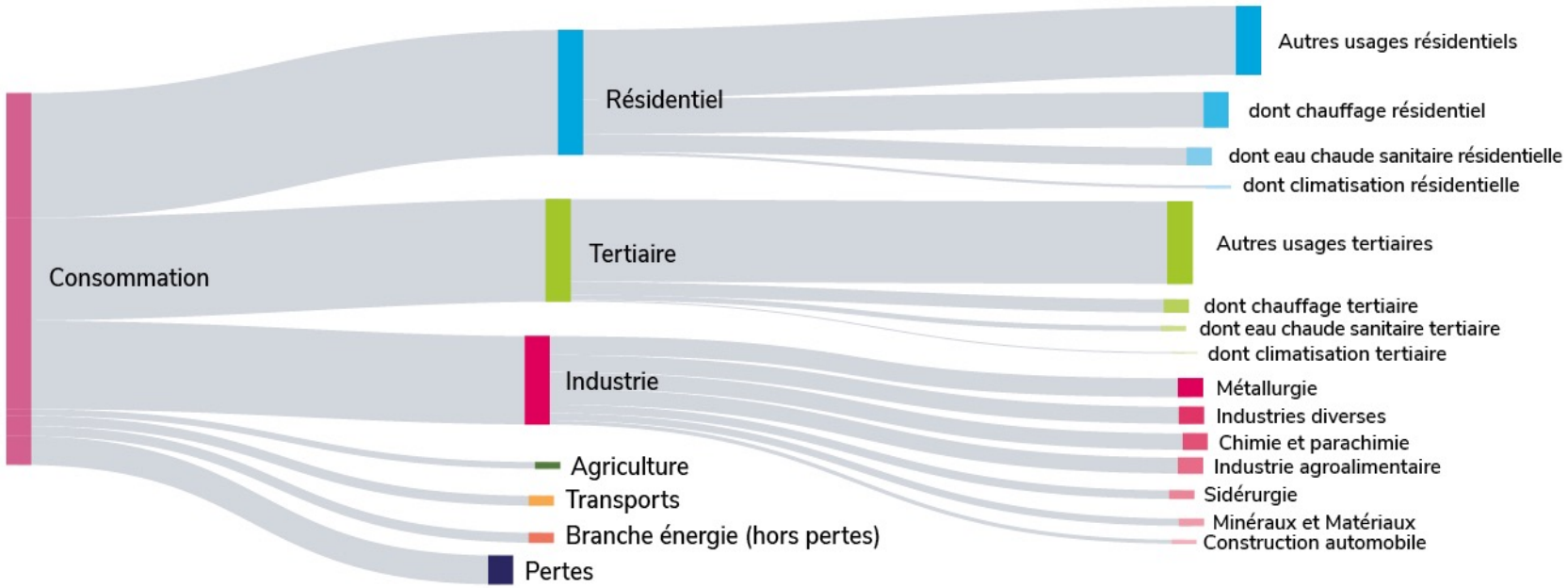
Source : IEA, 2021

Illustration

1 kWh =

- 1 diswasher or laundry machine cycle
- 3 days of wif
- 6 km in an electric car

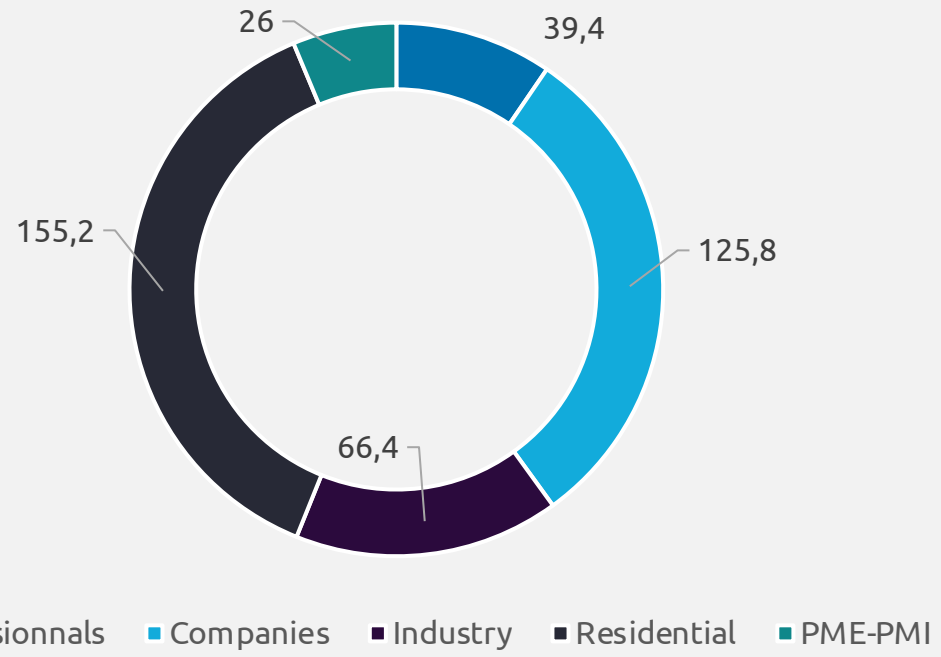
Power consumption in France decomposed by type of use (2019)



Source : Bilan électrique 2022, RTE
https://analysesetdonnees.rte-france.com/bilan-electrique-consommation#Repartitionsectorielle

POWER CONSOMPTION IN FRANCE

Sectorial decomposition

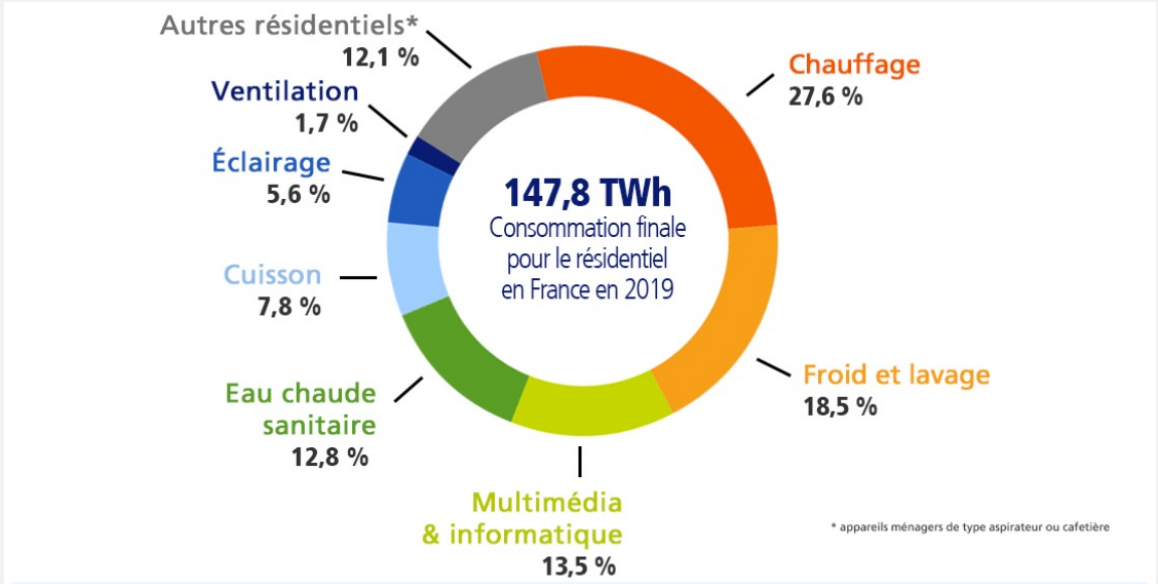


Sectorial decomposition of power consumption in France in 2021 (TWh)

Source : OpenData Réseau Energie, 2021

Zoom : Residential use

Residential use is the main sector of consumption in France (38% of the total power consumption)



Distribution of electrical consumption by residential uses

Source : ADEME – Pour agir 2019 © EDF

Find more information on power consumption in France :

- https://bilan-electrique-2020.rte-france.com/consommation-repartition-sectorielle-de-la-consommation-2/#3
- https://analysesetdonnees.rte-france.com/bilan-electrique-consommation
- https://analysesetdonnees.rte-france.com/consommation/synthese

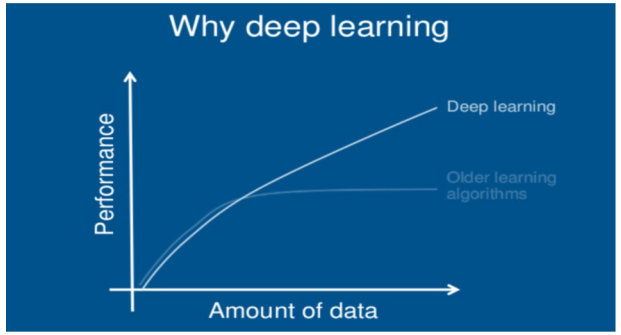
PRACTICAL SESSION

https://github.com/CharlesBoydelaTour/TS_XHEC_TP

04 – THE ADVANTAGES OF DEEP LEARNING: RNN

WHY DEEP LEARNING HYPE?

BIG DATA AVAILABILITY → NEW DL TECNHIQUES ← GPU/TPU ACCELERATIONS



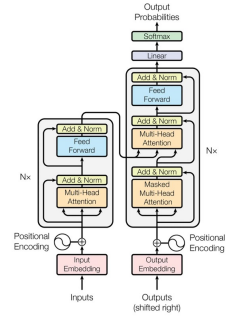
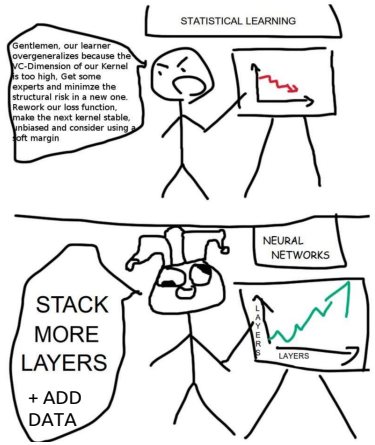
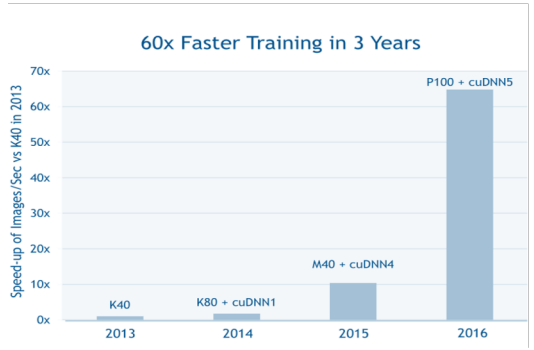
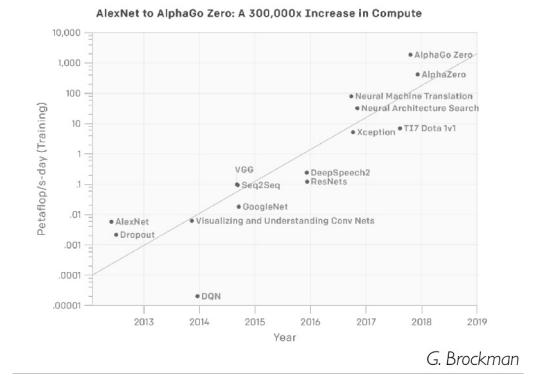


Figure 1: The Transformer - model architecture.

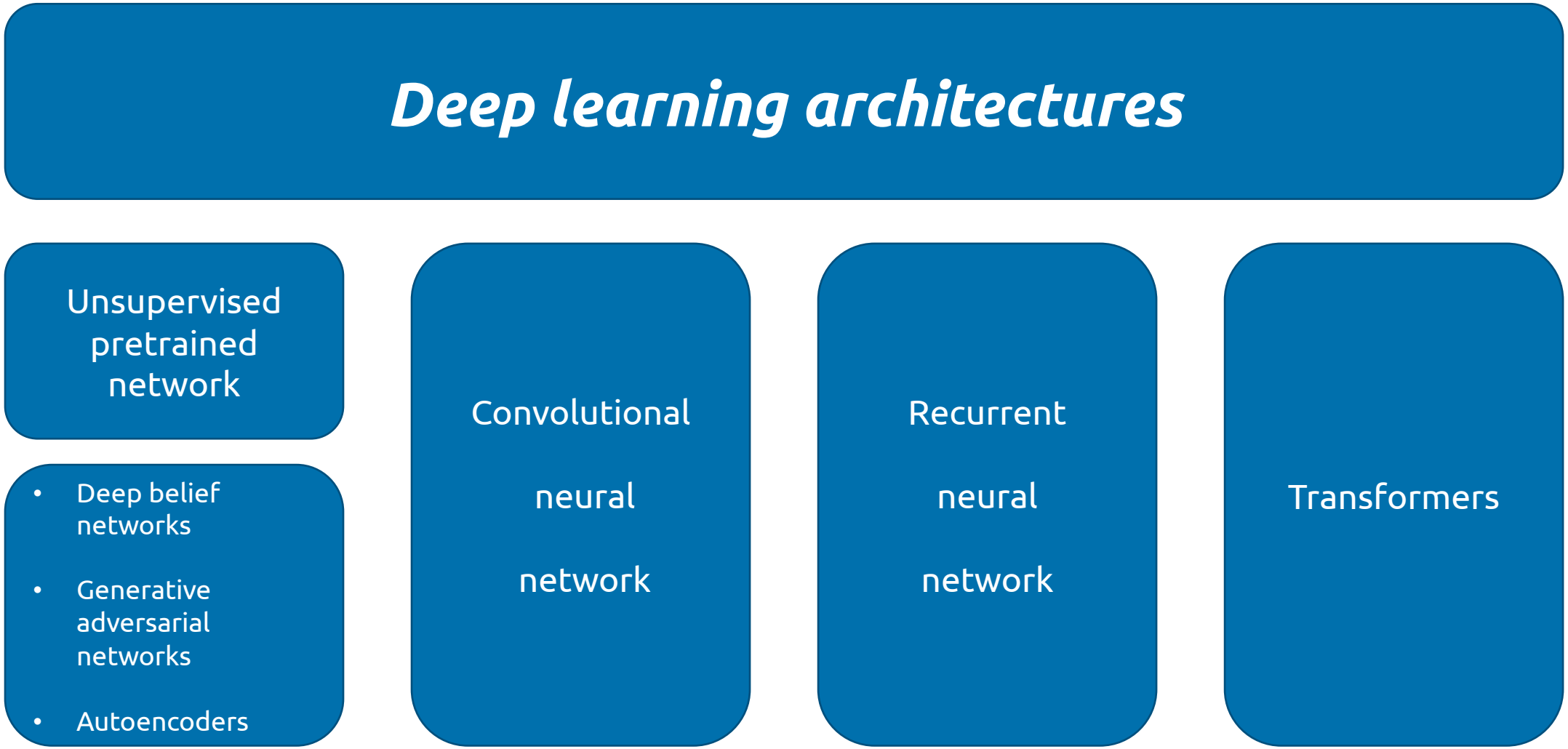




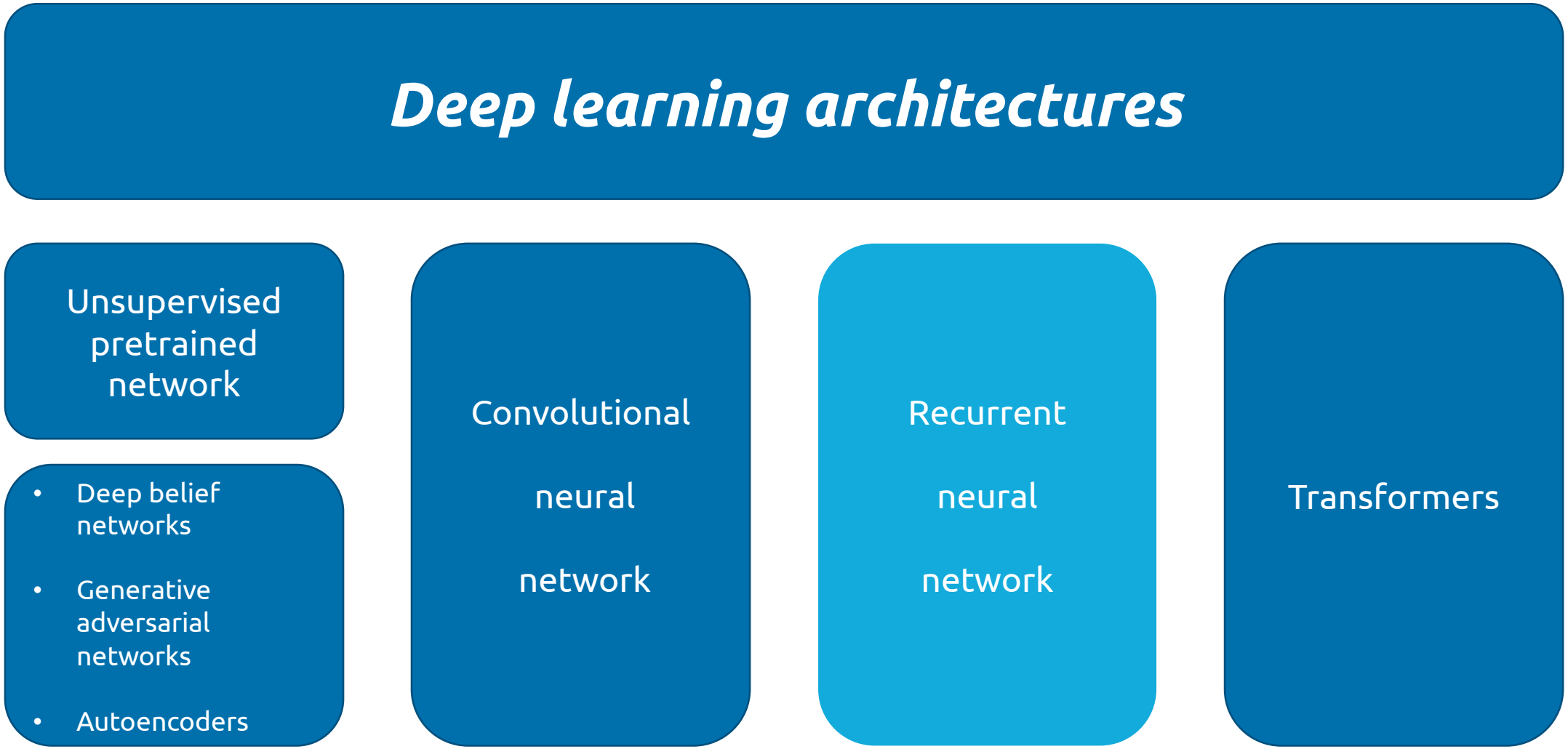


G. Brockman

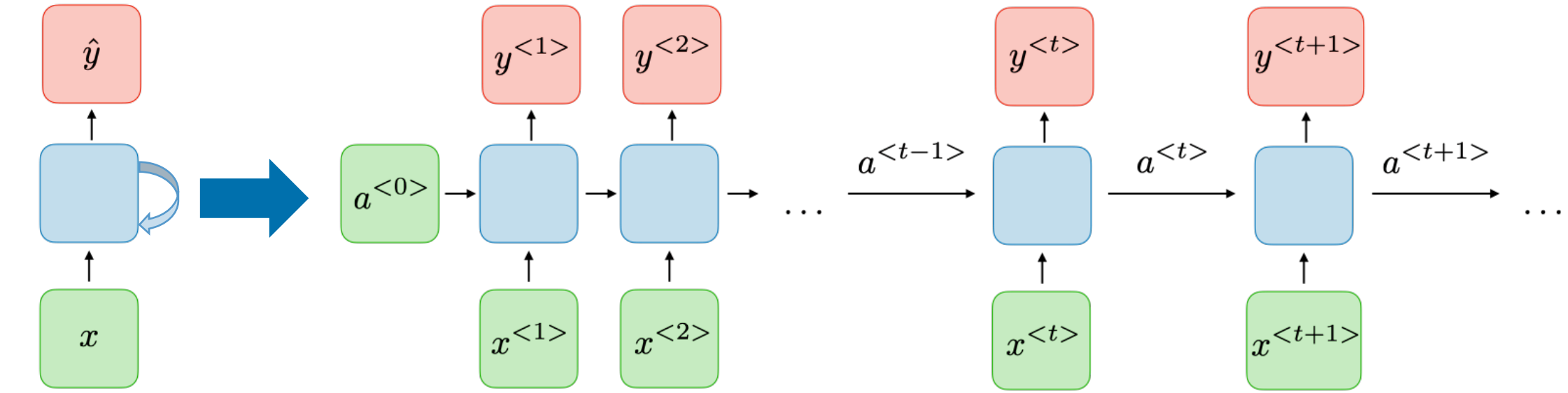
DEEP LEARNING ARCHITECTURES



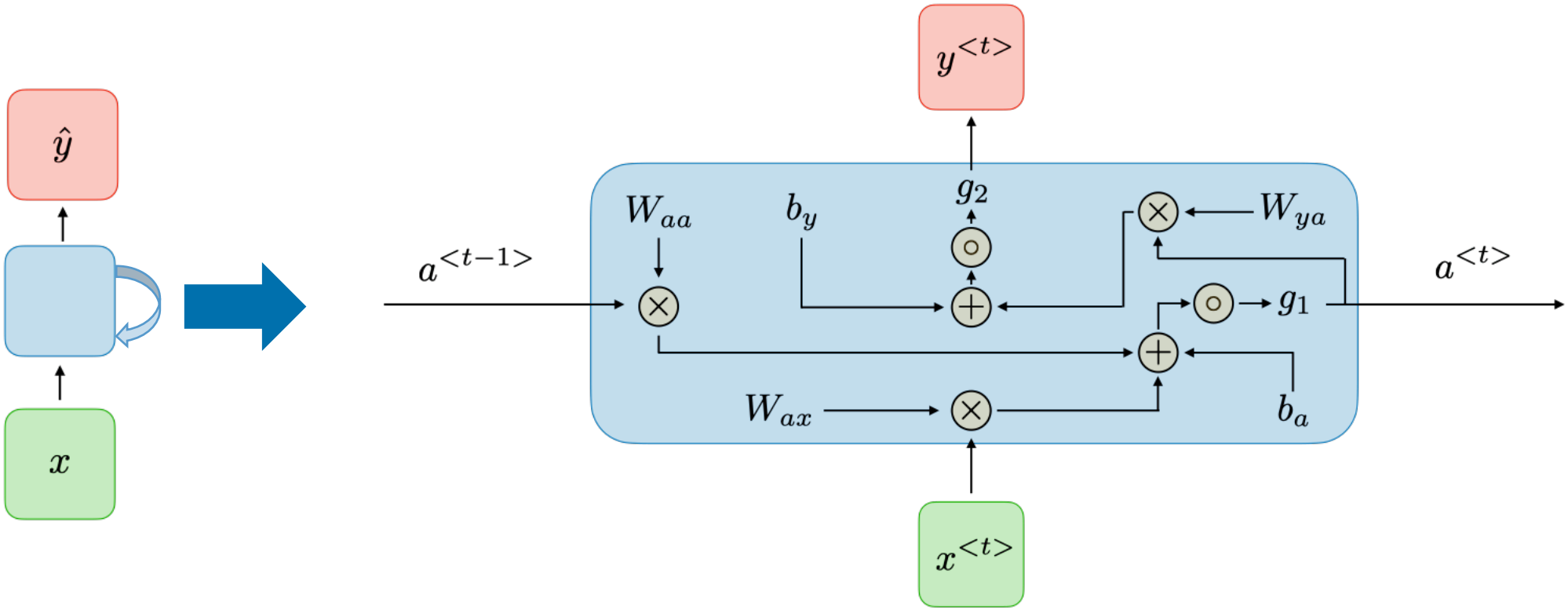
DEEP LEARNING ARCHITECTURES



RECURRENT NEURAL NETWORK ARCHITECTURE



RECURRENT NEURAL NETWORK IN DEPTH



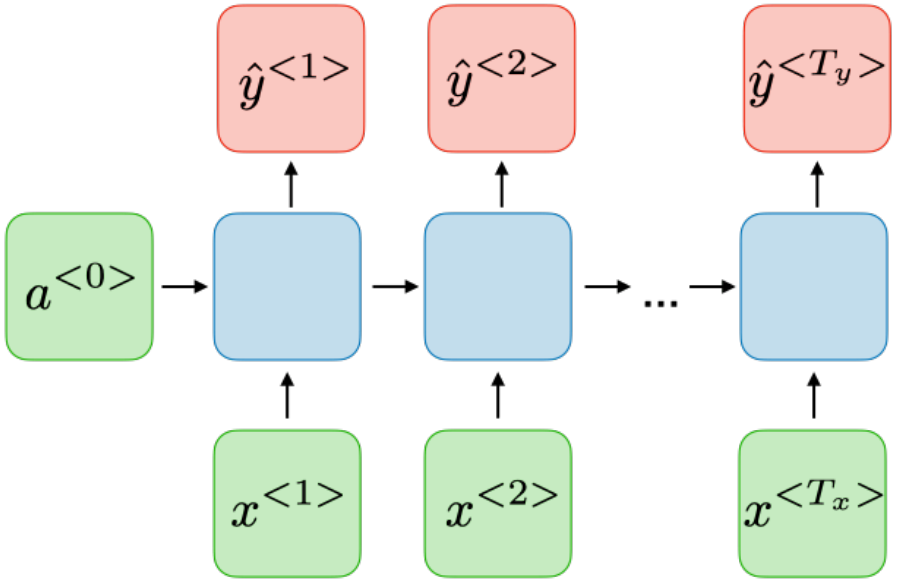
For each timestep t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \qquad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

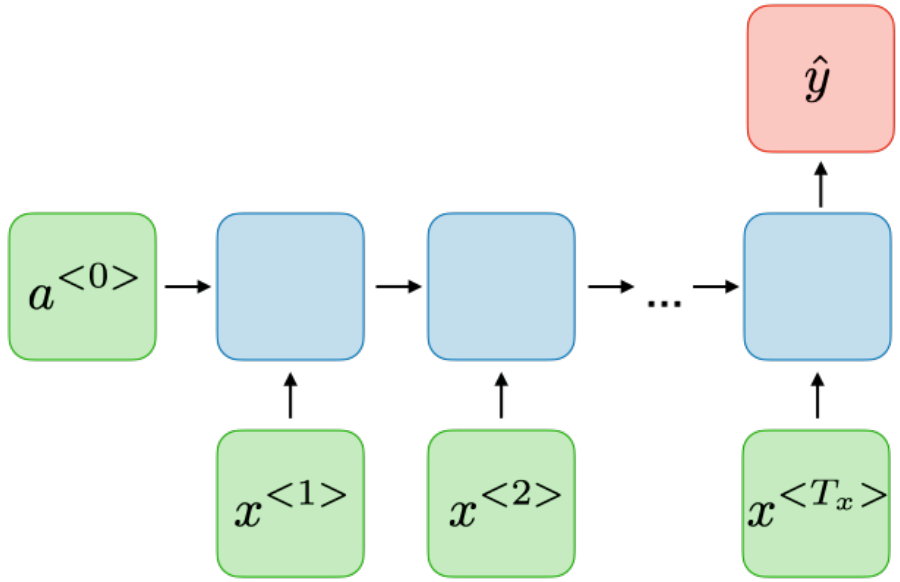
Where W and b are coefficients that are shared temporally and g are activation functions.

MANY-TO-MANY VS MANY-TO-ONE

Many-to-Many



Many-to-One



Example:

Let be the univariate time series of daily electricity consumption. By using RNNs, we can, among other things:

- Predict a day's consumption using the last 12 days' consumption. (**Many to One**)
- Predict consumption for the next 4 days using the consumption of the last 12 days. (**Many to Many**)

TRAINING A RNN: LOSS FUNCTION AND BACKPROPAGATION THROUGH TIME

Loss function

The loss function $\mathcal{L}$ of all time steps is defined based on the loss at **every time step** as follows:

$$\mathcal{L}(\hat{y}, y) = \frac{1}{\mathrm{T}}\sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

Backpropagation through time

Backpropagation is done at **each point in time**.
At timestep T, the derivative of the loss $\mathcal{L}^{(T)}$ with respect to weight matrix W is expressed as follows:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \frac{1}{\mathrm{T}}\sum_{t=1}^{T} \frac{\partial \mathcal{L}^{(t)}}{\partial W}\Bigg|_{(t)}$$

Analysis of Gradients in the output layer of an RNN

- The first and the second factors of the product are easy to compute. The third factor $\frac{\partial \mathrm{a}^{<t>}}{\partial W_a}$ is where things get tricky, since we need to recurrently compute the effect of the parameter $\mathrm{a}^{<t>}$ on W_a.

$$\frac{\partial \mathcal{L}}{\partial W_a} = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial \mathcal{L}(\hat{y}^{<t>}, y^{<t>})}{\partial y^{<t>}} \frac{\partial g\,(\mathrm{a}^{<t>}, W_y)}{\partial a^{<t>}} \boxed{\frac{\partial \mathrm{a}^{<t>}}{\partial W_a}}$$

- According to the recurrent computation $\mathrm{a}^{<t>}$ depends on both $\mathrm{a}^{<t-1>}$ and W_a, where computation of $\mathrm{a}^{<t-1>}$ also depends on W_a. Thus, evaluating the total derivate of $\mathrm{a}^{<t>}$ with respect to W_a using the chain rule yields :

$$\frac{\partial \mathrm{a}^{<t>}}{\partial W_a} = \frac{\partial g(x_t, \mathrm{a}^{<t-1>}, W_a)}{\partial W_a} + \frac{\partial g(x_t, \mathrm{a}^{<t-1>}, W_a)}{\partial \mathrm{a}^{<t-1>}} \frac{\partial \mathrm{a}^{<t-1>}}{\partial W_a}$$

- While we can use the chain rule to compute $\frac{\partial \mathrm{a}^{<t>}}{\partial W_a}$ recursively, this chain can get very long whenever t is large.

$$\frac{\partial \mathrm{a}^{<t>}}{\partial W_a} = \frac{\partial \mathrm{g}(x_t, \mathrm{a}^{<t-1>}, W_a)}{\partial W_a} + \sum_{i=1}^{t-1}\left(\prod_{j=i+1}^{t} \frac{\partial \mathrm{g}(x_j, \mathrm{a}^{<j-1>}, W_\mathrm{a})}{\partial \mathrm{a}^{<j-1>}}\right) \frac{\partial \mathrm{g}(x_i, \mathrm{a}^{<j-1>}, W_\mathrm{a})}{\partial W_\mathrm{a}}$$

VANISHING AND EXPLODING GRADIENTS

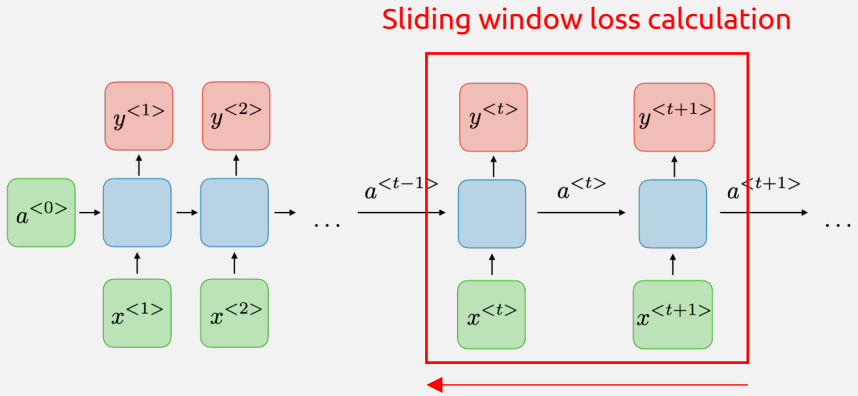
Vanishing gradients

Exploding gradients

$$0 < \left| \frac{\partial a^{<j>}}{\partial a^{<j-1>}} \right| < 1$$

Strategies to counter the vanishing/exploding gradients:

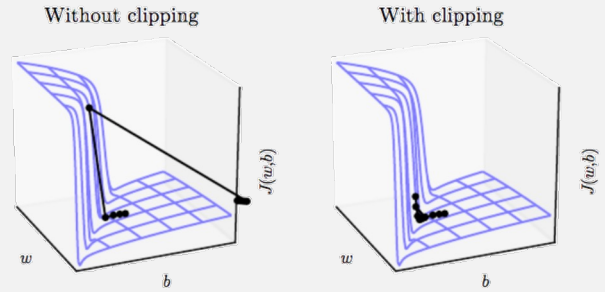
Truncated Backpropagation Through Time (Truncated BPTT)



Limit: The minus of this approach is that dependencies of longer than the chunk length, are not taught during the training process.

Gradient Clipping

Prevent gradients from blowing up by **rescaling** them so that their norm is **at most a particular** value **η**.



— Goodfellow et al., *Deep Learning*

FROM RECURRENT NEURAL NETWORK TO LSTM TO GRU AND BEYOND

RNN

- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

LSTM

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

GRU

- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*

ATTENTION

- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.Text

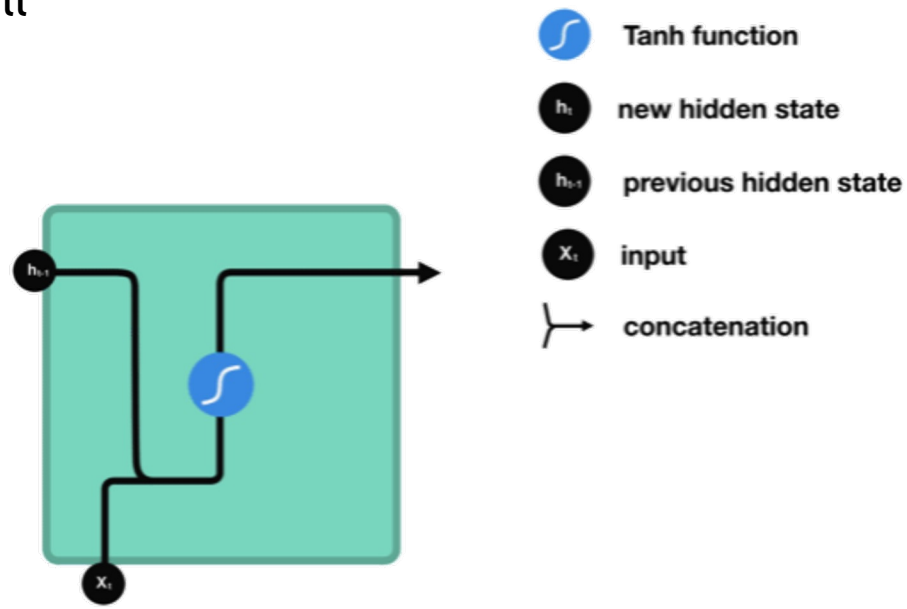
TRANSFORMERS

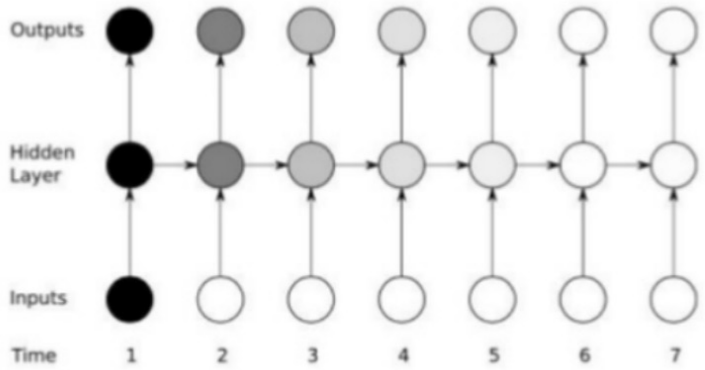
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

05 –LSTM AND GRU

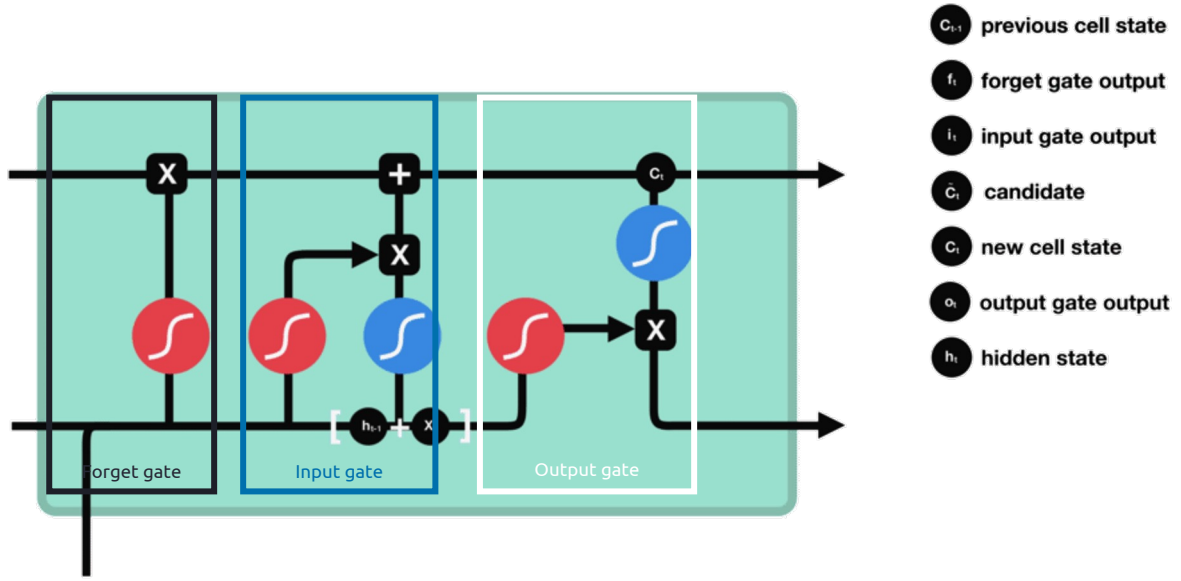
FROM RECURRENT NEURAL NETWORK TO LSTM

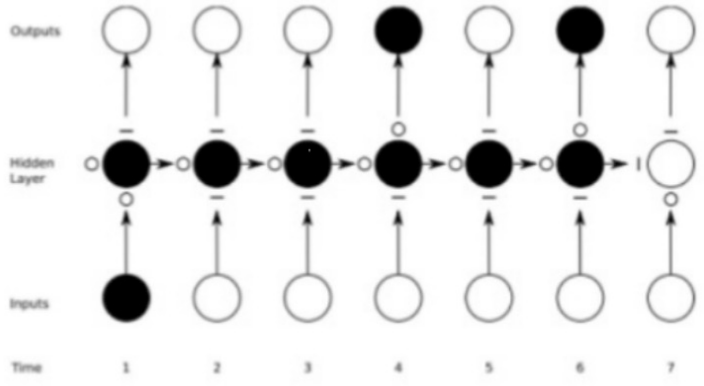
RNN Cell



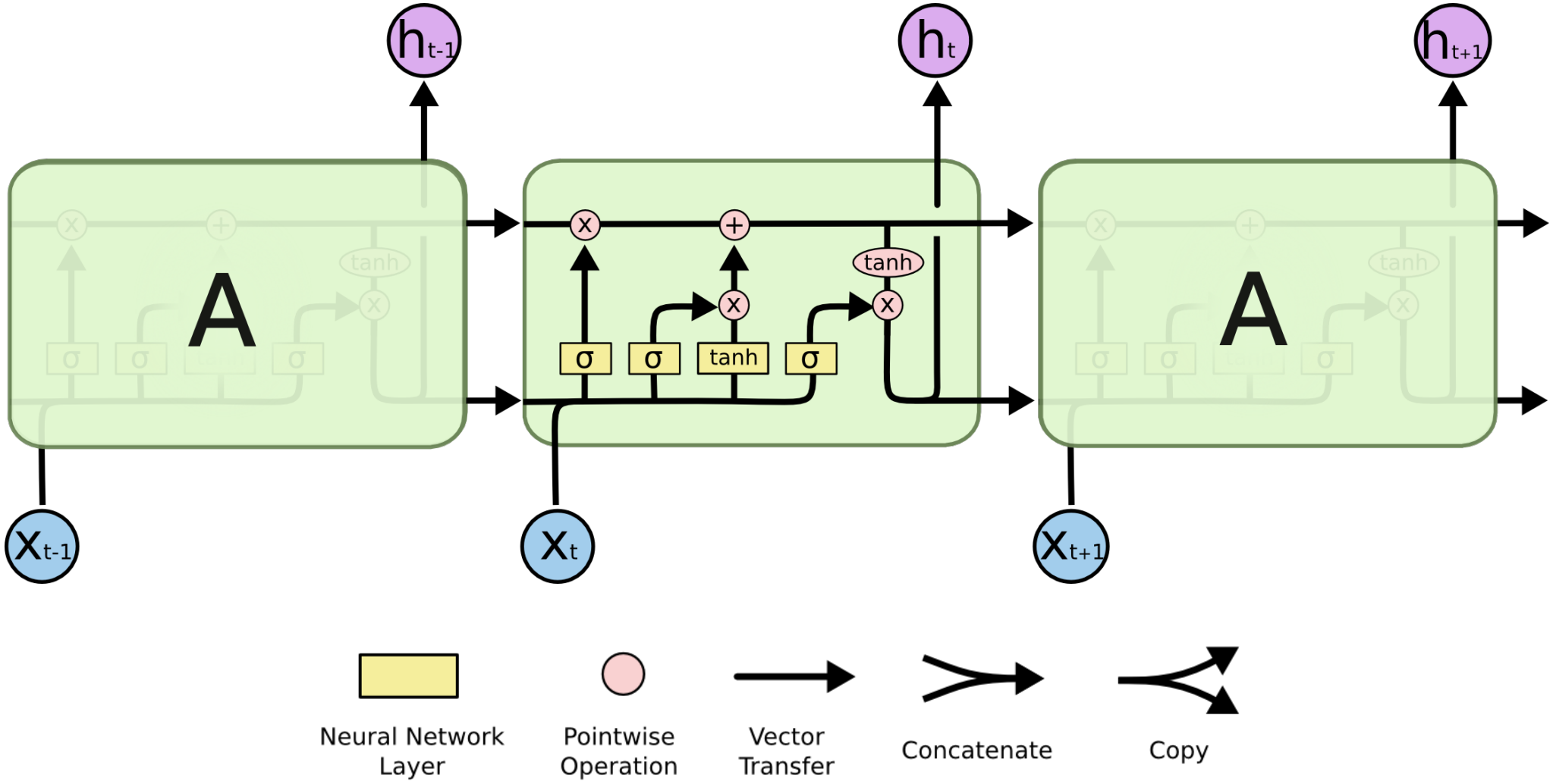


LSTM Cell



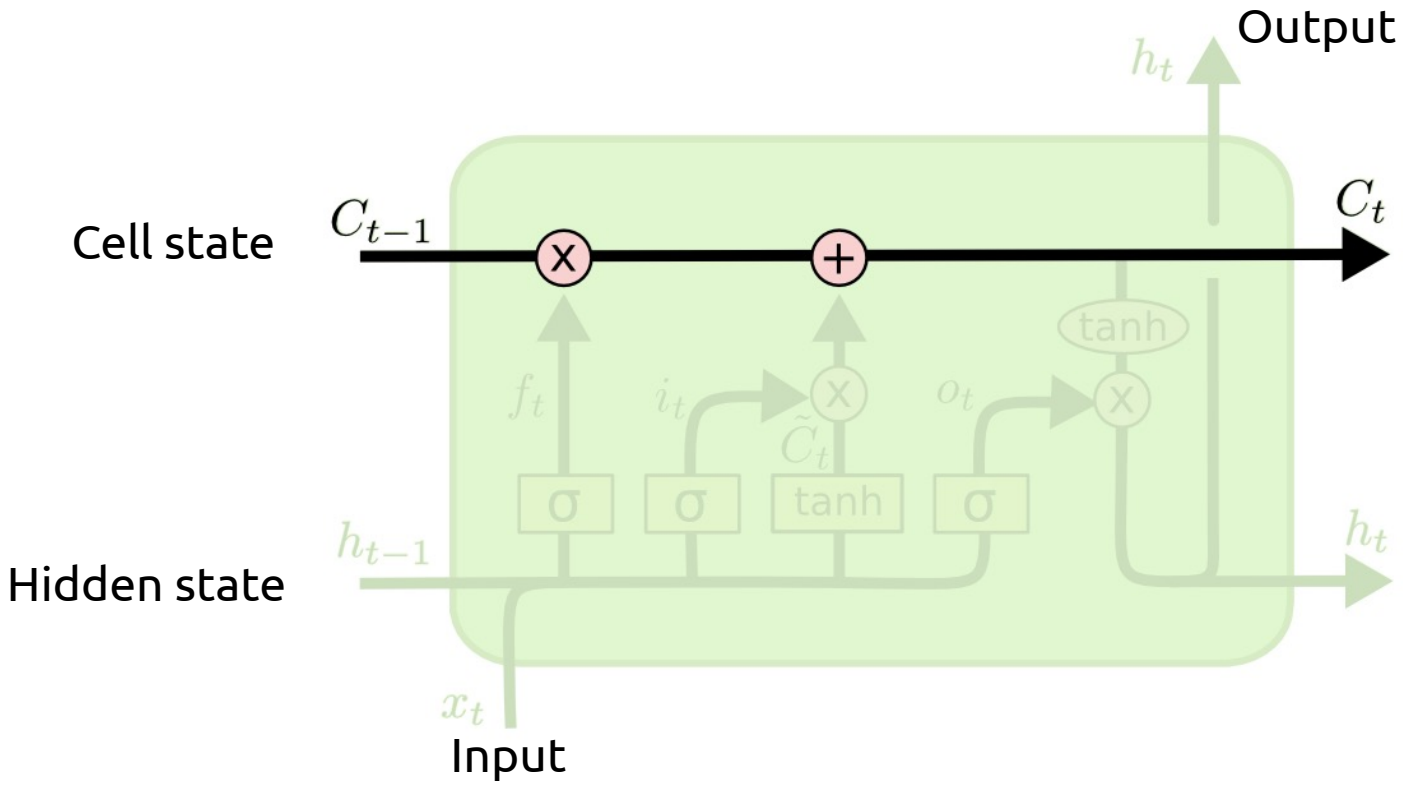


LONG SHORT TERM MEMORY ARCHITECTURE:



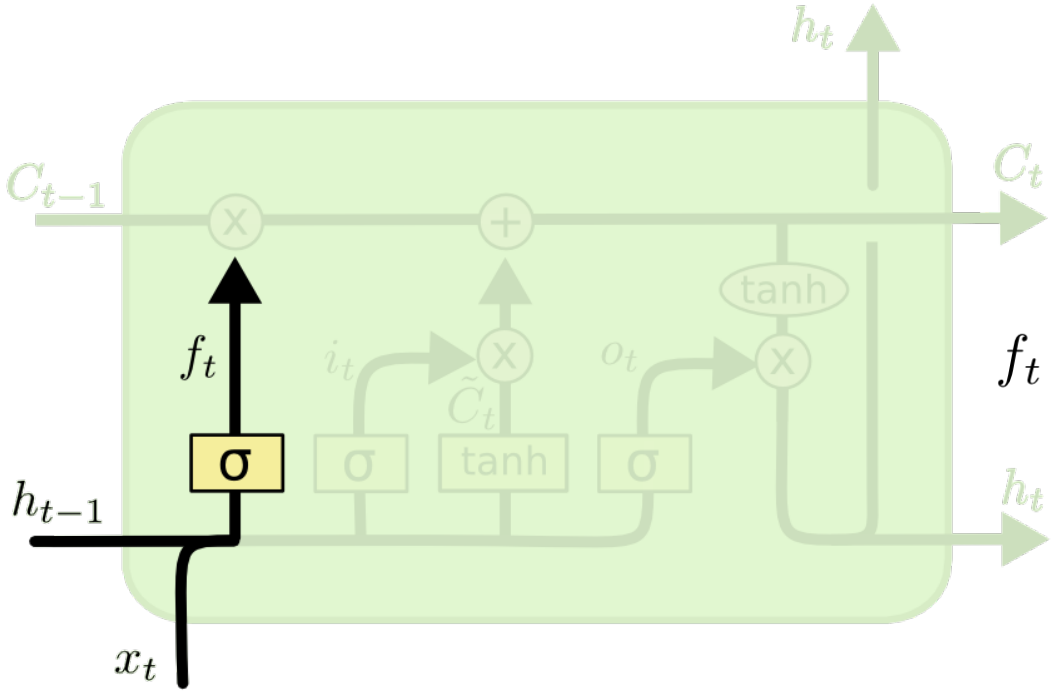
LONG SHORT TERM MEMORY ARCHITECTURE: THE CORE IDEA BEHIND LSTMS

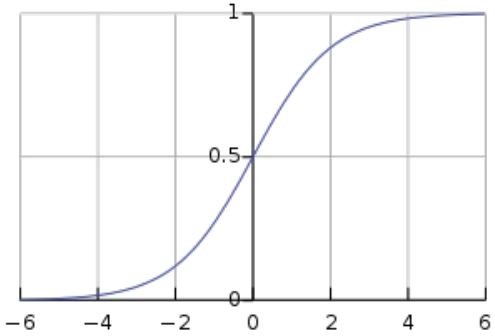
The cell state runs straight down the entire chain, with only some minor linear interactions.
⊕: Add information from previous hidden states to cell state.
⊗: Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication.



LONG SHORT TERM MEMORY ARCHITECTURE: FORGET GATE LAYER

Decide what information we're going to throw away from the cell state
This decision is made by a sigmoid layer called the "forget gate layer."
It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}.



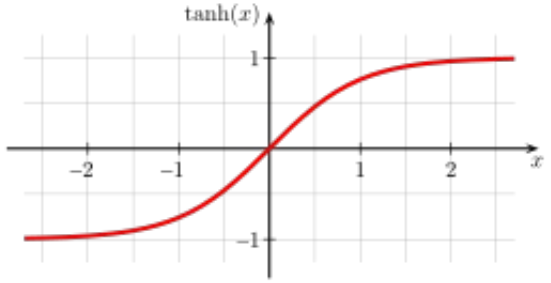


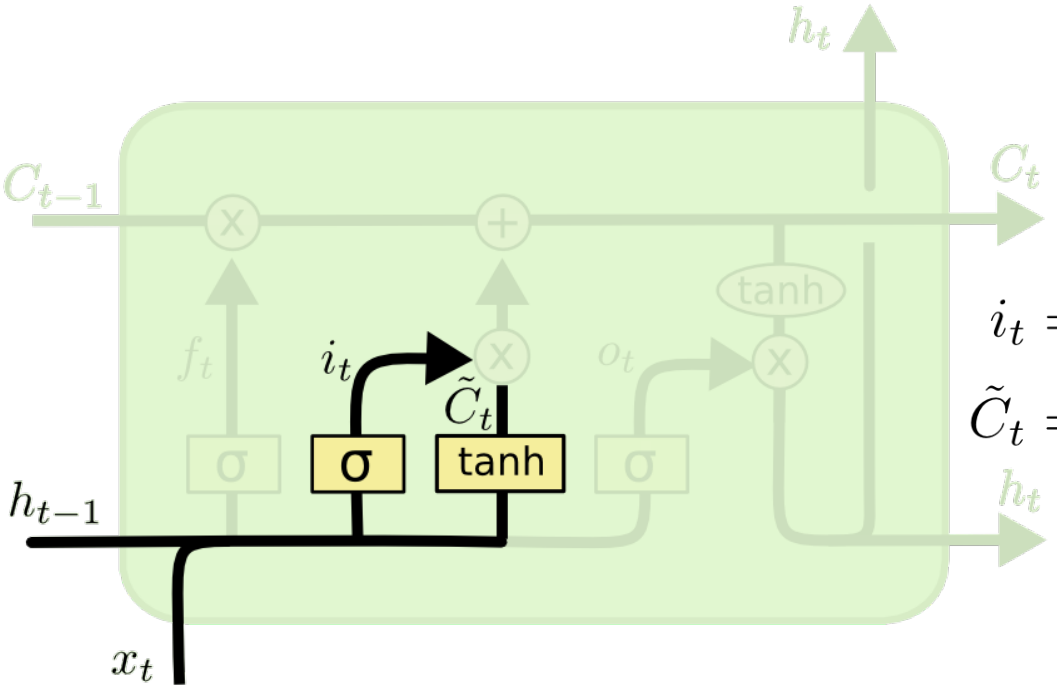
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LONG SHORT TERM MEMORY ARCHITECTURE: INPUT GATE LAYER

Decide what new information we're going to store in the cell state with 2 parts:

- A sigmoid layer called the "input gate layer" decides which values we'll update
- A tanh layer creates a vector of new candidate values $\widetilde{C}_t$, that could be added to the state





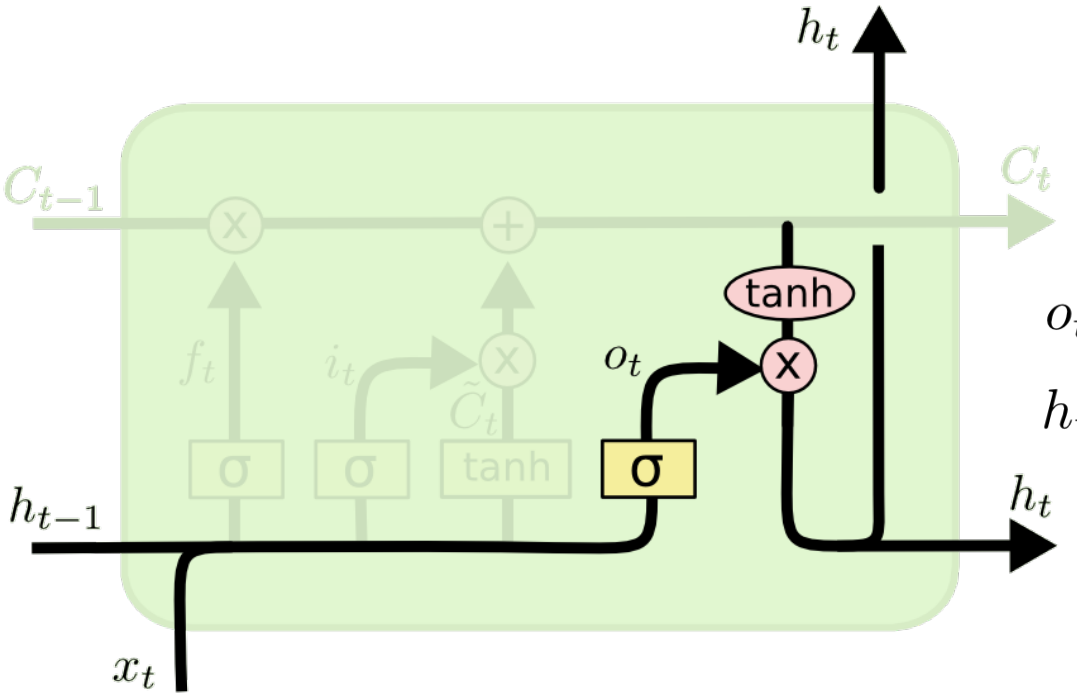
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LONG SHORT TERM MEMORY ARCHITECTURE: OUTPUT GATE LAYER

The output will be based on our cell state, but filtered with the hidden state:

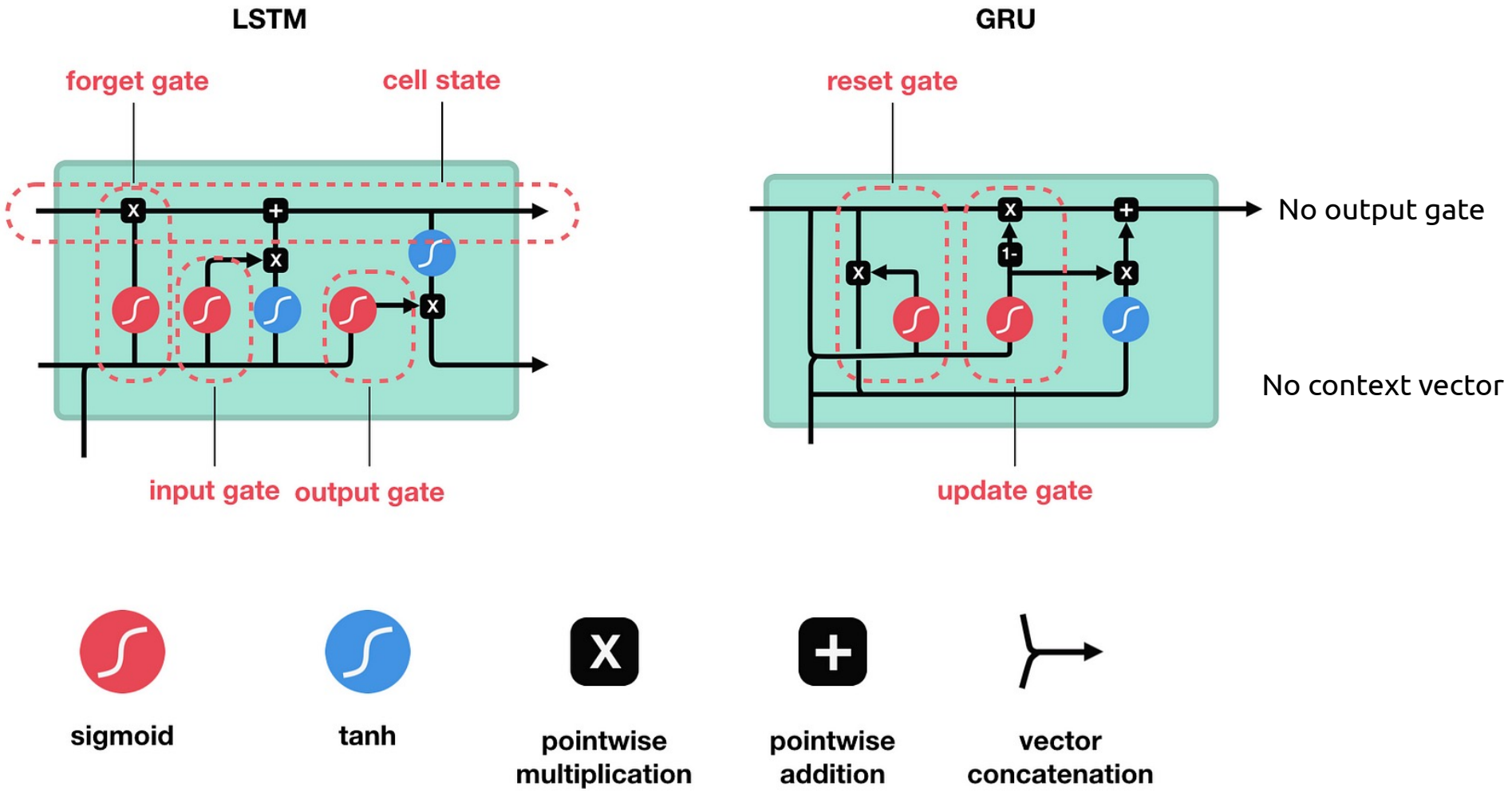
- We run a sigmoid layer which decides what parts of the cell state we're going to output.
- We put the cell state through tanh (to push the values to be between –1 and 1) and multiply it by the output of the sigmoid gate.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

SIMPLIFIED LSTM (WITH FEWER PARAMETERS): GATED RECURRENTS UNIT



PRACTICAL SESSION

https://github.com/CharlesBoydelaTour/TS_XHEC_TP