Data science pour la prévision de séries temporelles

Sylvain BARTHELEMY

Plan du cours

- Machine learning et séries temporelles
- Deep learning temporel LSTM, GRU et TCN
- Réseaux de neurones traditionnels et CNN
- Librairie Prophet
- Autres méthodes: TimeGPT, TimesFM, Chronos, MUIRAI

Series Temporelles et Machine Learning

Séries temporelles et machine learning

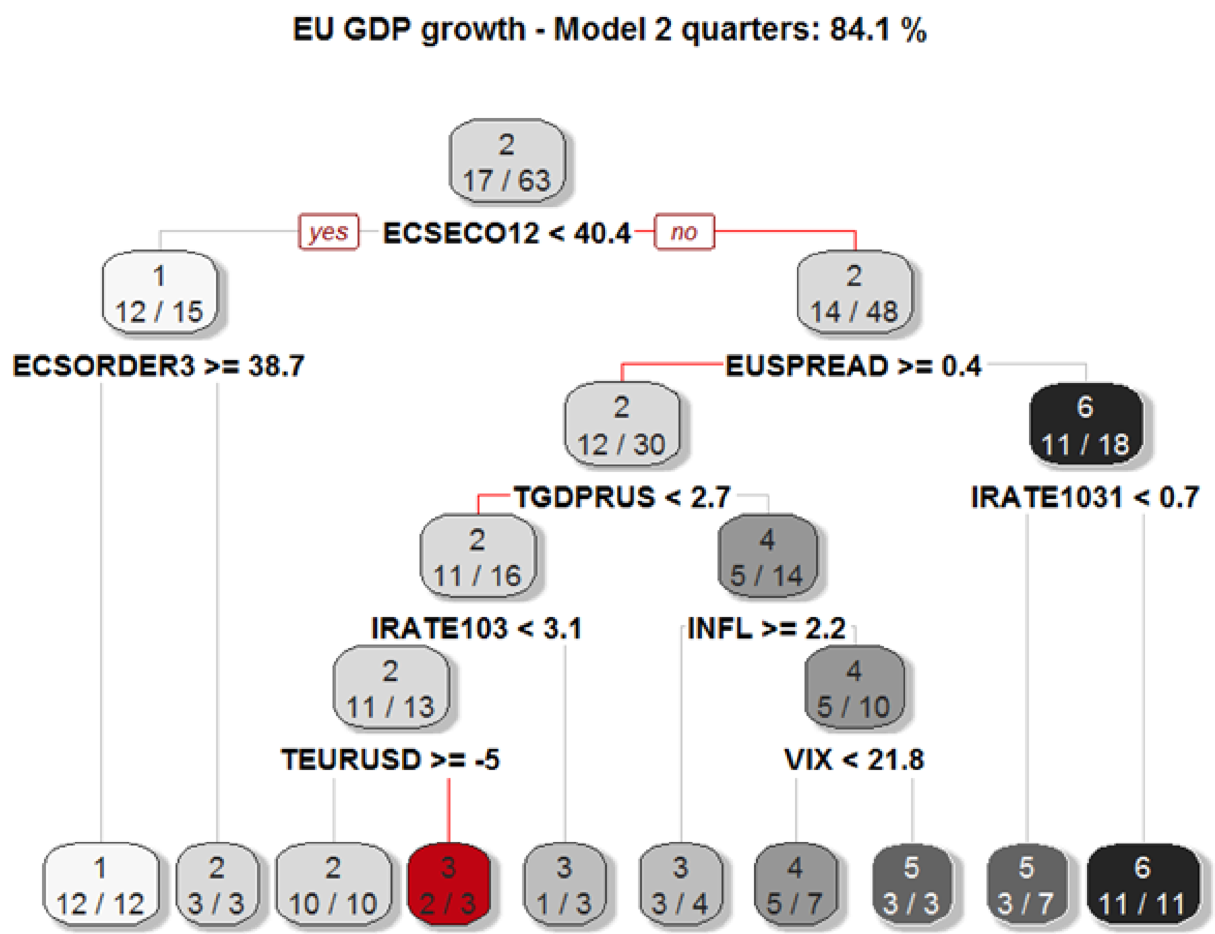
- **Calibrage d’un modèle de machine learning sur des séries temporelles**
  - Manipulation de séries temporelles en Python.
  - Attention au train/test => pas de sklearn traditionnel
  - Création de variables artificielles « décalées »
  - Calibrage d’un RandomForest « basique »
- **Gestion des saisonnalités et des tendances**
  - Quel est le problème ?
  - Différenciation simple
  - Gestion via l’ajout de variables dummy (ex: le mois)
  - Gestion via une décomposition des séries temporelles

Séries temporelles et machine learning

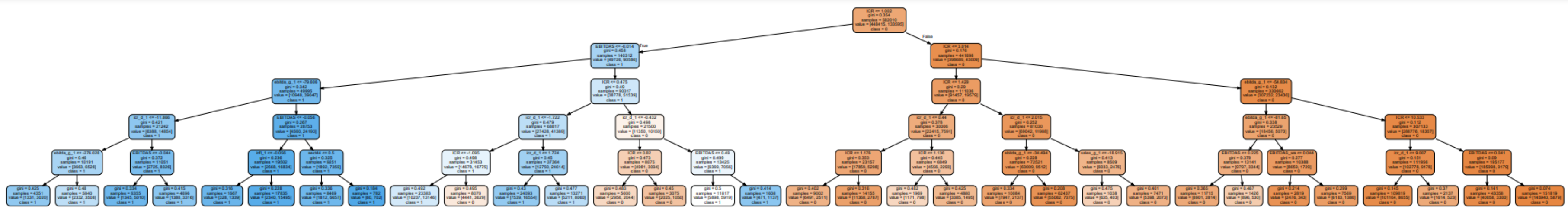
- **Prévision numériques vs classes**
  - Transformation des variables en classes
  - Rappels sur la matrice de confusion
  - Rappels sur la ROC Curve et la difficulté du choix du seuil
  - Rappels sur les classes minoritaires (SMOTE)
- **Travail sur le Bitcoin en journalier**
  - Travailler un modèle à un jour
  - Travailler un modèle à 1 mois
  - Travailler un modèle à 6 mois

Analyse de conjoncture

CART Applied to Real GDP Growth Forecast



Exemple d'arbres

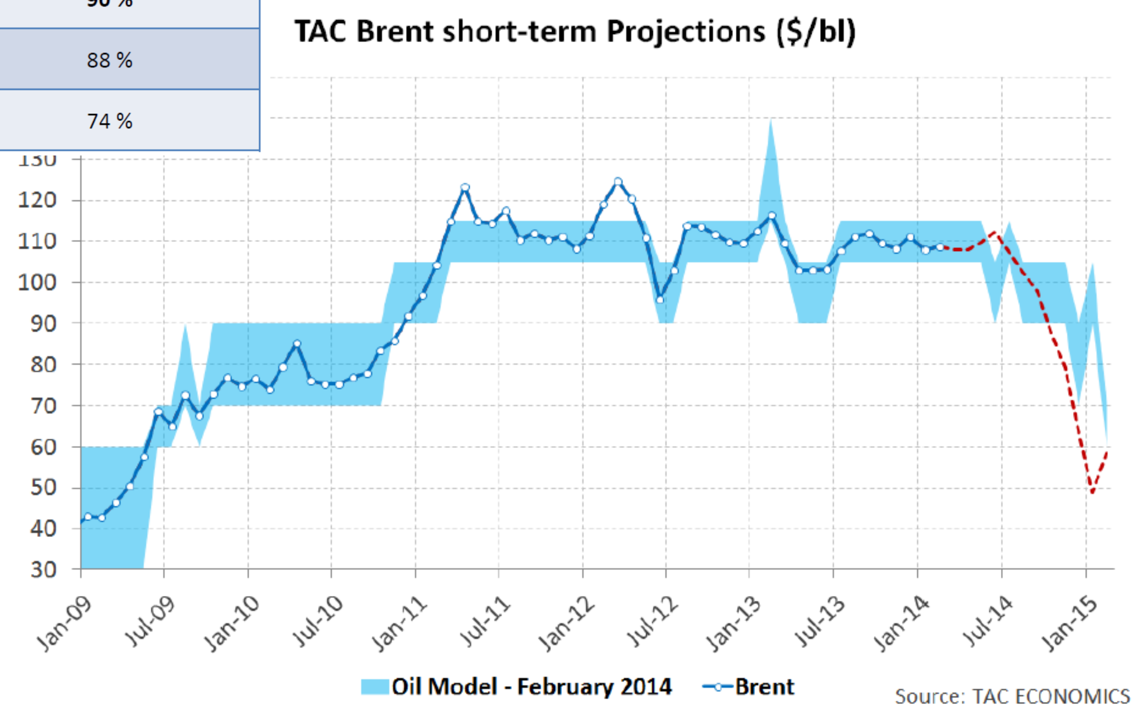


Exemple de performances d'évaluation du risque de paiement sur 350,000 PME avec moins de 5 variables

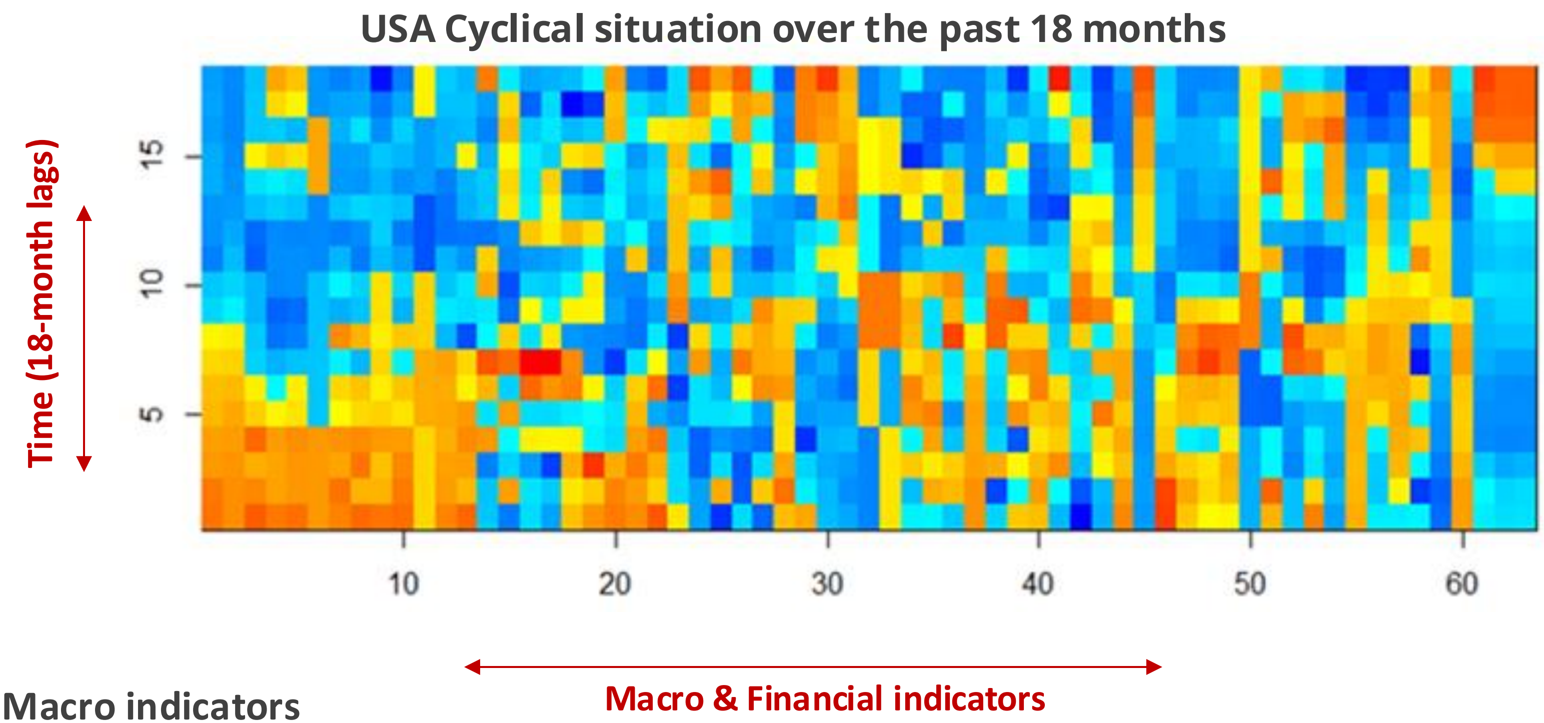
Horizon	CART Simplifié	RF Simplifié	CART Standard	RF Standard
1 an	88%	89%	86%	87%
2 ans	83%	84%	81%	82%

Machine Learning on Oil Price Forecasts

	Average accuracy (training dataset)	Average accuracy (testing, dataset 1)
Naive Bayes	96 %	88 %
Tree Bagging	99.9 %	**90 %**
Gradient Boosted Machine	100 %	**90 %**
Supervised SOM	86 %	75 %
Neural Network multilayer perceptron	82 %	74 %
Random Forest	100 %	**90 %**
Support Vector Machine	96 %	88 %
k-nearest neighbors	84 %	74 %



Heatmaps d'indicateurs macroéconomiques

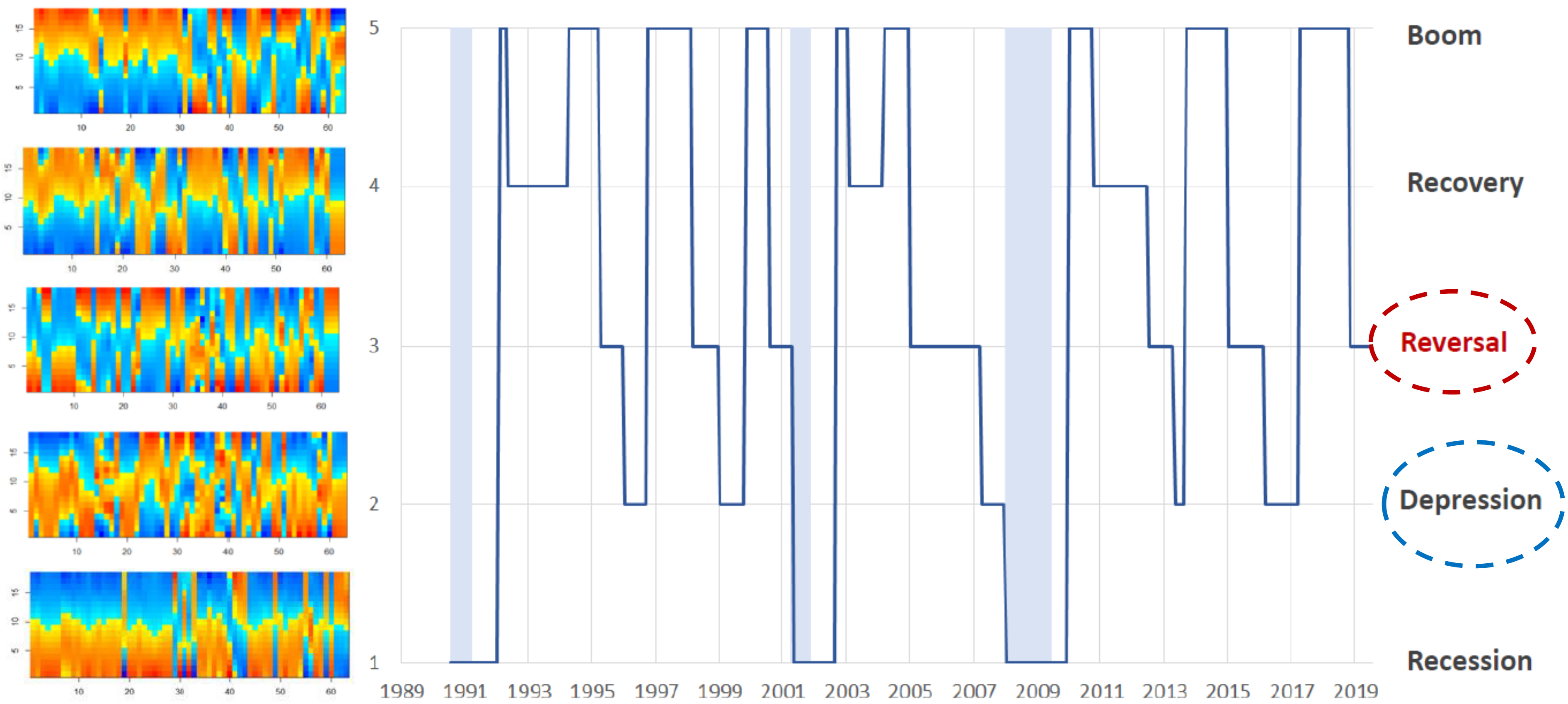


Macro indicators

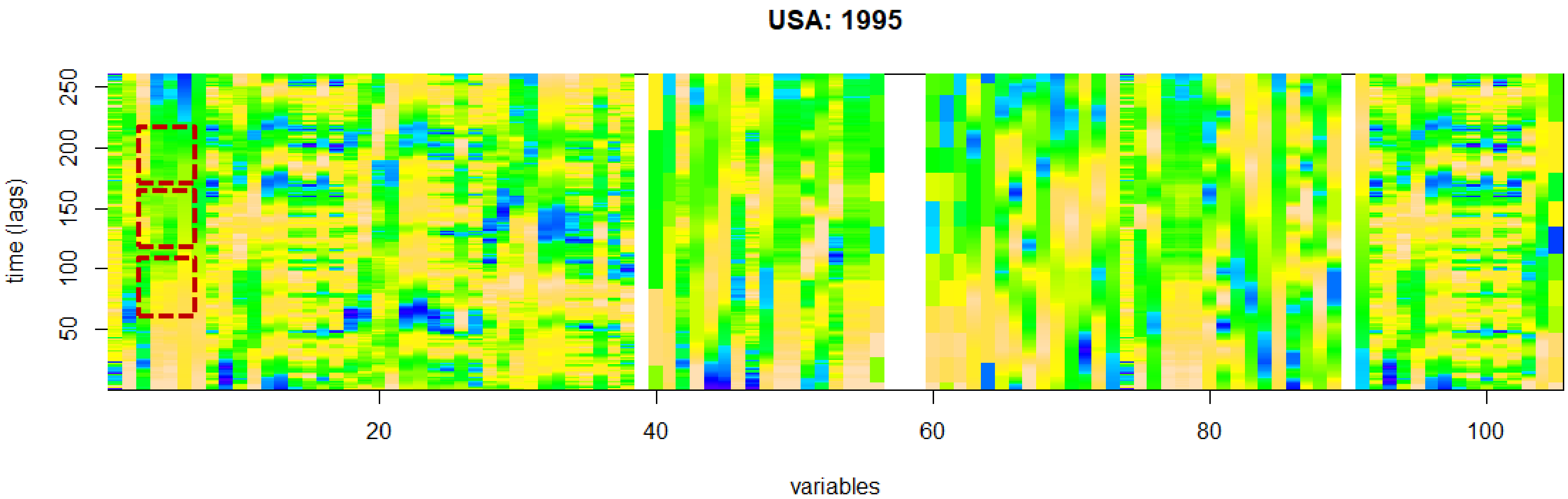
- Leading indicators
- Manufacturing sector (production, survey, capacity
- Household situation (income, confidence, consumption, labor market)
- Housing market (permits, price)
- Prices
- Financial market (rates, stocks, financial conditions, exchange rate)

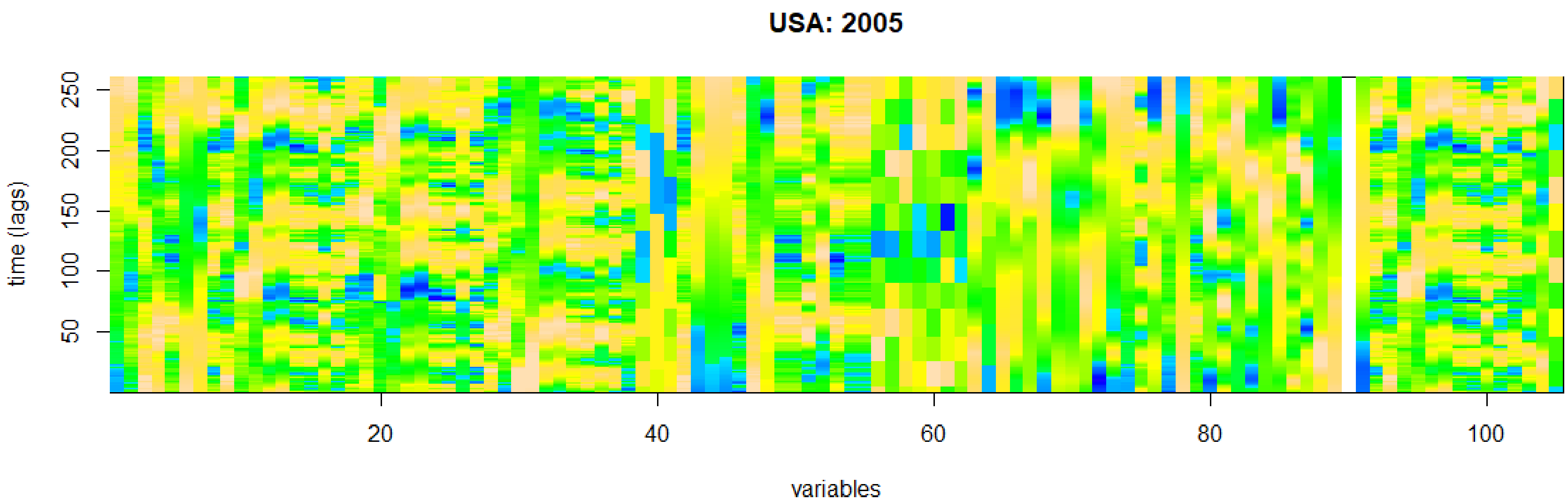
Séquences de prototypes d'images et prévisions

Historical changes in "archetypal" US Cyclical Heatmap

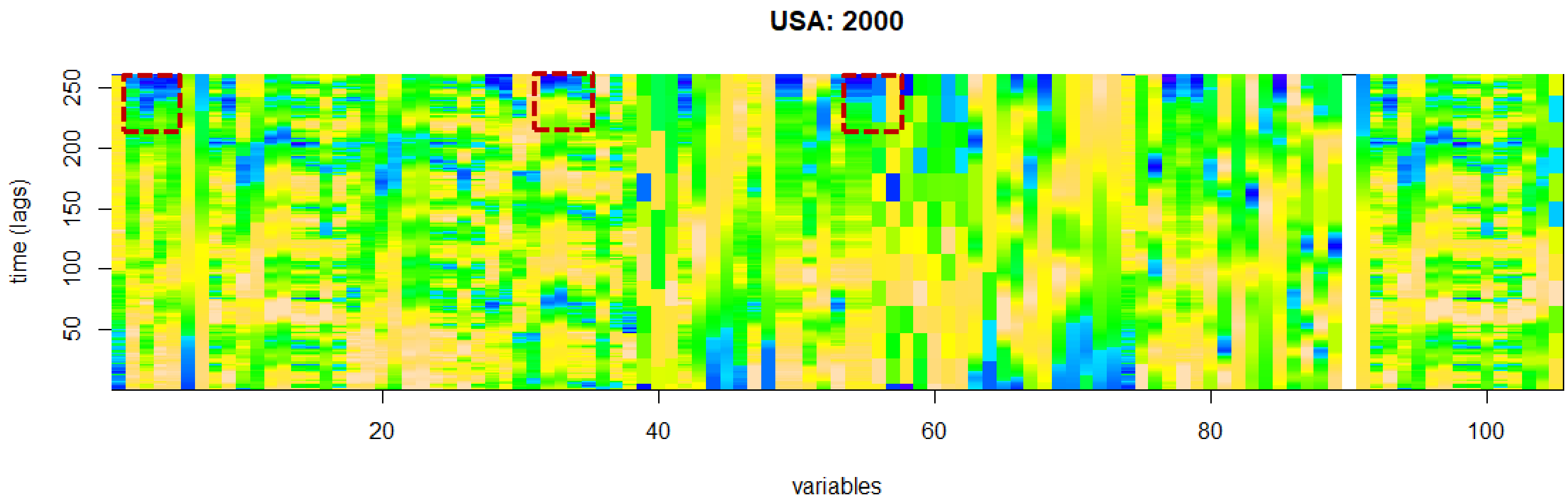


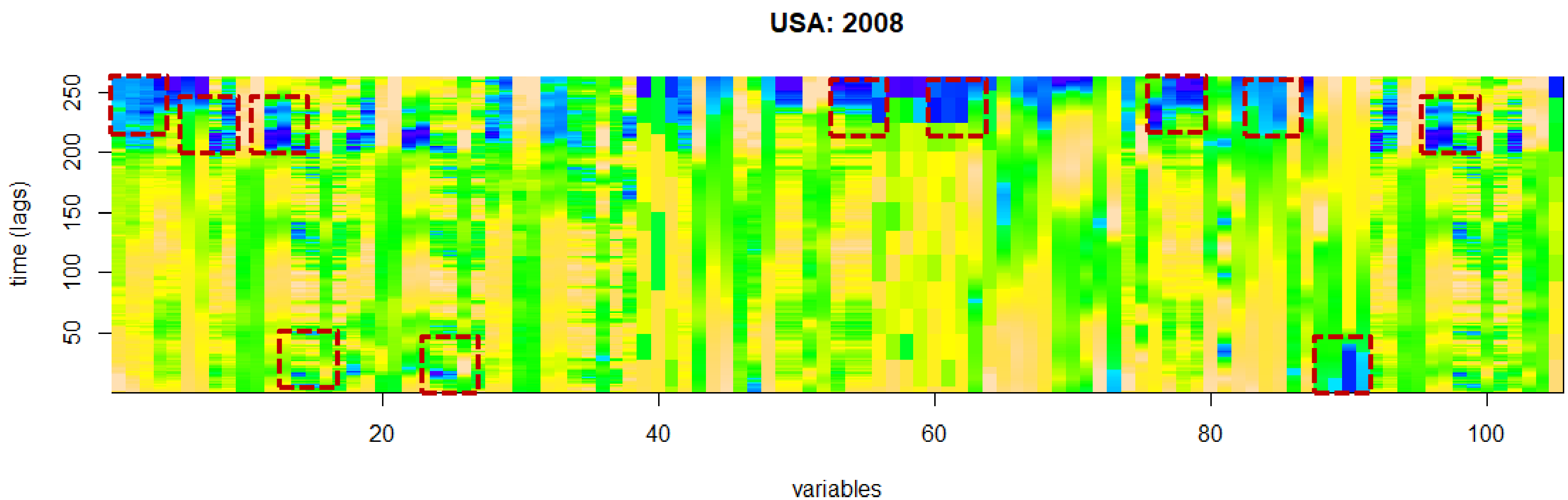
Macrofinancial « Patterns » and QMA



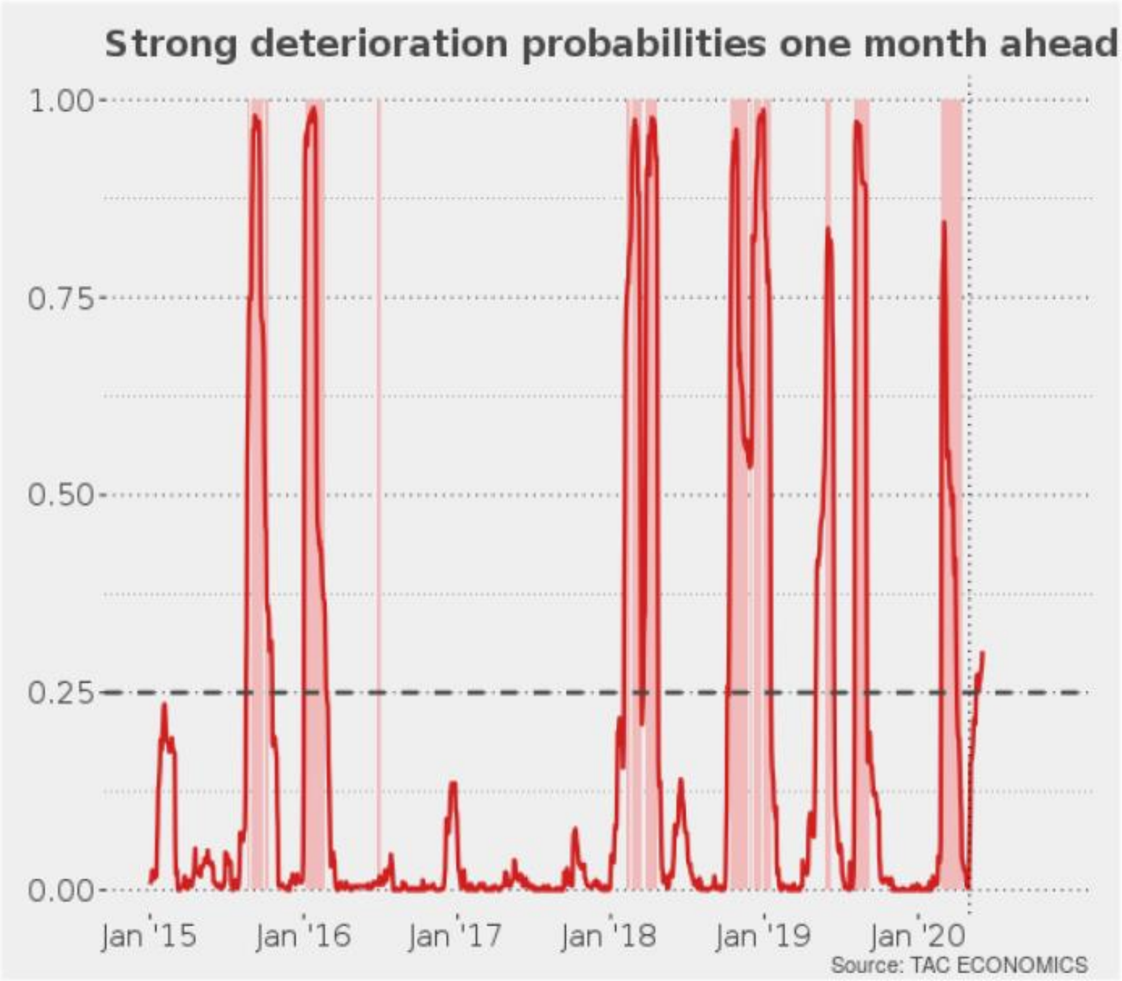


Macrofinancial « Patterns » and QMA



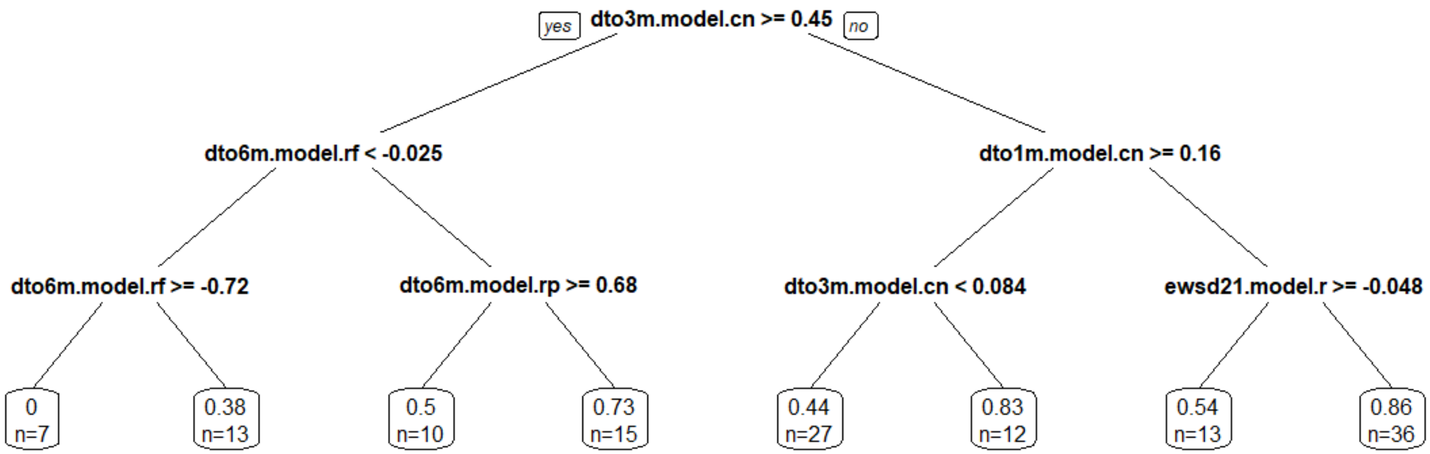


Performances et signaux du QMA sur le S&P500

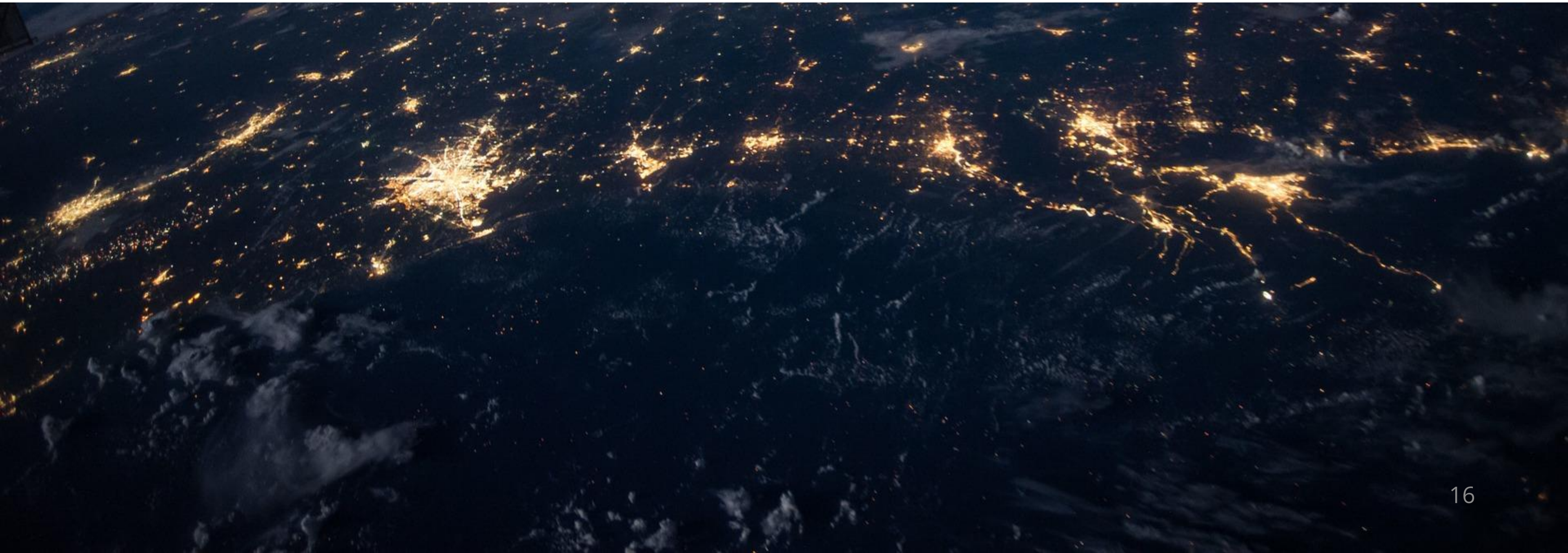


Méta-stratégies et logiques de combinaisons de signaux

Exemple sur le S&P 500



Early Warning Systems on Economic & Financial Crises



Economic & financial crises since 2017

CURRENCIES
Argentina -20%
Turkey -15%
Brazil -10% Russia -8%
In 2017:
Egypt -44%
Uzbekistan -43%
Congo-Kinshasa -31%

ACTIVITY
Venezuela (-13.2%),
Congo-Brazzaville (-4.6%),
Kuwait (-2.5%)

China
6% vs 14% in 2007

INFLATION
Argentina >30%,
Nigeria 17%, Iran 10%,
Turkey 15%, Ukrain 14%
Congo-Kinshasa 42%
Libyia 28%, Egypt 24%
Venezuela >1000%

Forecasting economic and financial crises

- Krugman (1979), Obstfeld (1994), Cantor and Packer (1996), Eichengreen et al. (1996), Frankel and Rose (1996), Goldstein (1996), Goldstein and Turner (1996), Kaminsky and Reinhart (1999), Komulainen and Lukkarila (2003) , …
- But forecasting economic and financial crises a few years in advance remains a very difficult exercise, particularly on emerging markets.

Kaminsky, Lizondo and Reinhart (1998) "Leading indicators of currency crises" (1/2)

Table 1. *Performance of Indicators Under the "Signals" Approach*

	Number of crises for which there are data (1)	Percentage of crises called[a] (2)	Good signals as percentage of possible good signals (3)	Bad signals as percentage of possible bad signals (4)	Noise/ signal (adjusted)[b] (5)	P(crisis/ signal)[c] (6)	P(crisis/ signal) −P(crisis)[d] (7)
In terms of the matrix in the text			A/(A+C)	B/(B+D)	[B/(B+D)]/ [A/(A+C)]	A/(A+B)	
Real exchange rate	72	57	25	5	0.19	67	39
Banking crises	26	37	19	6	0.34	46	17
Exports	72	85	17	7	0.42	49	20
Stock prices	53	64	17	8	0.47	49	18
M2/international reserves	70	80	21	10	0.48	46	17
Output	57	77	16	8	0.52	49	16
"Excess" M1 balances	66	61	16	8	0.52	43	15
International reserves	72	75	22	12	0.55	41	13
M2 multiplier	70	73	20	12	0.61	40	11
Domestic credit/GDP	62	56	14	9	0.62	39	11
Real interest rate	44	89	15	11	0.77	34	6
Terms of trade	58	79	19	15	0.77	36	6
Real interest differential	42	86	11	11	0.99	29	0
Imports	71	54	9	11	1.16	26	−3
Bank deposits	69	49	16	19	1.20	25	−4
Lending rate/deposit rate	33	67	13	22	1.69	18	−9

[a] Percentage of crises in which the indicator issued at least one signal in the previous 24 months, out of the total number of crises for which data are available.

[b] Ratio of false signals (measured as a proportion of months in which false signals could have been issued) to good signals (measured as a proportion of months in which good signals could have been issued).

[c] Percentage of the signals issued by the indicator that were followed by at least one crisis within the subsequent 24 months.

[d] P(crisis) is the unconditional probability of a crisis, (A+C)/(A+B+C+D) in terms of the matrix in the text. This probability ranges from 27 percent to 33 percent depending on the indicator. The unconditional probability varies across indicators because not all of them have observations for the same period.

Kaminsky, Lizondo and Reinhart (1998) "Leading indicators of currency crises" (2/2)

Table 2. *Average Lead Time*

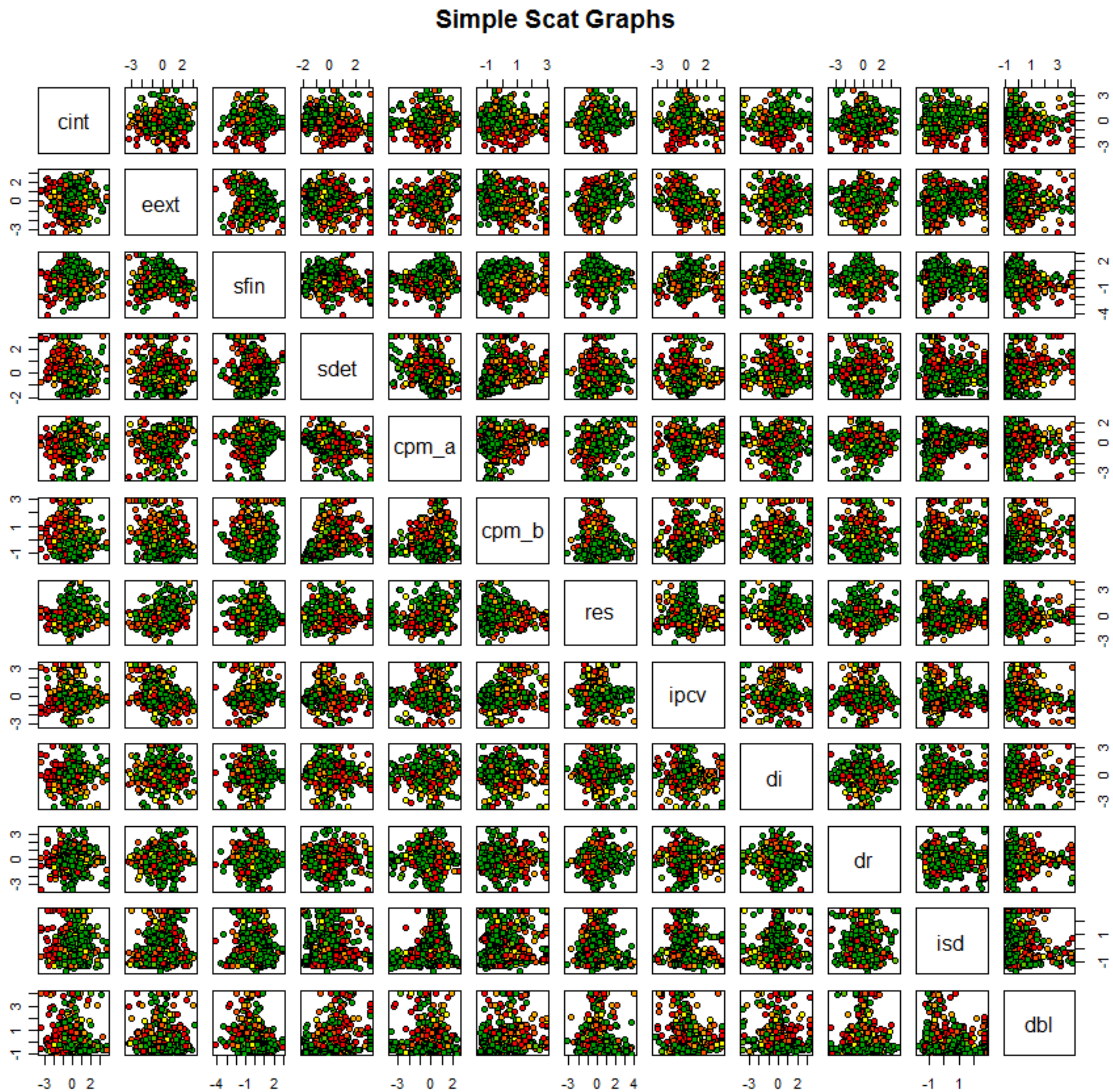
Indicator	Number of months in advance of the crisis when first signal occurs
Banking crisis	19
Real exchange rate	17
Real interest rate	17
Imports	16
M2 multiplier	16
Output	16
Bank deposits	15
"Excess" M1 balances	15
Exports	15
Terms of trade	15
International reserves	15
Stock prices	14
Real interest differential	14
M2/international reserves	13
Lending rate/deposit rate	13
Domestic credit/GDP	12

TAC ECONOMICS/RiskMonitor Fundamental Balances

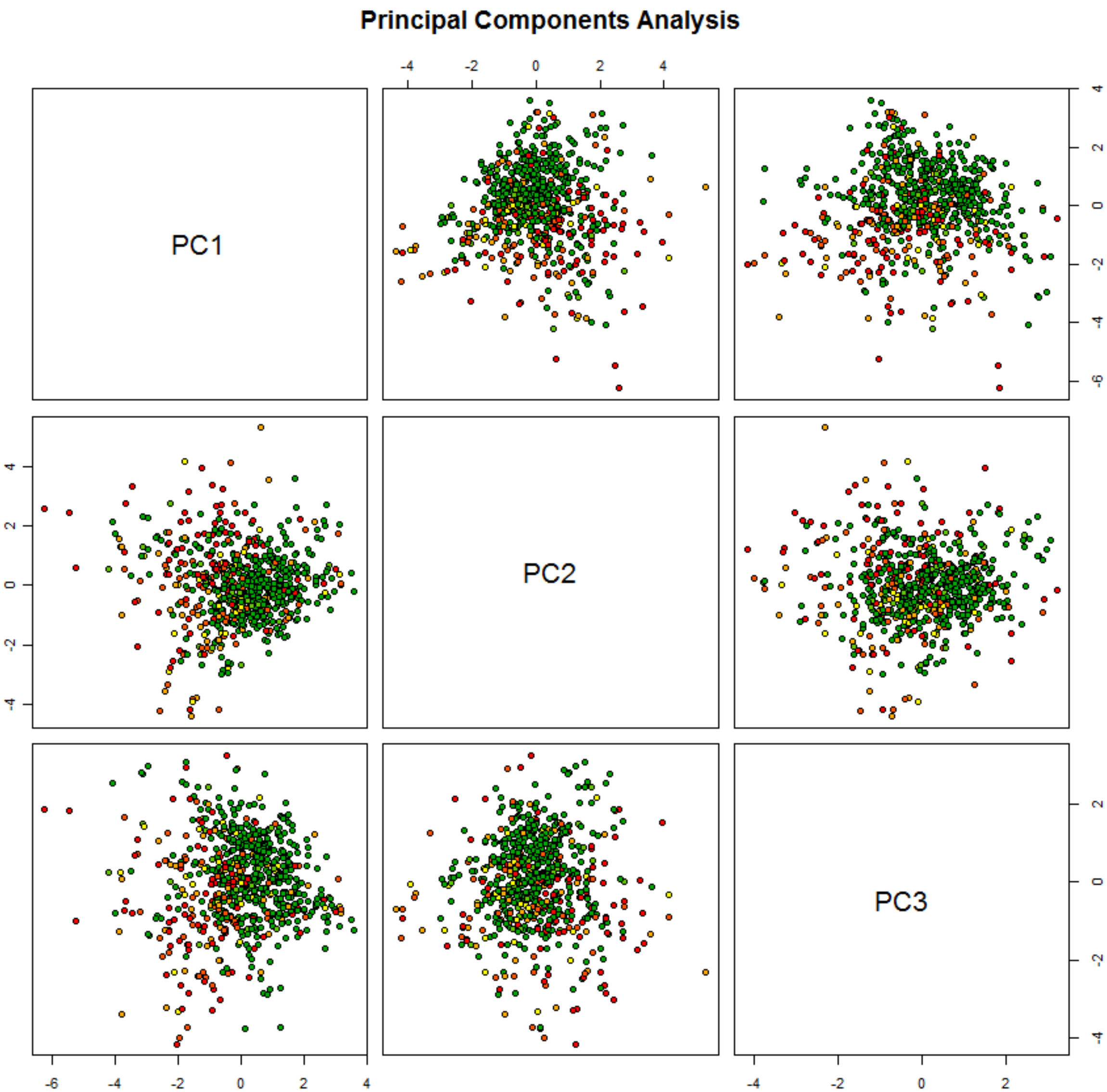
Macroeconomic indicators of the RiskMonitor fundamental balances

Indicator	Periodicity	Description
Economic growth	Annual	GDP growth
External balance	Annual	External balance sustainability
Financing stability	Annual	Stability of FDI inflows
Debt service	Annual	External financing
Forex liquidity	Quarterly	Foreign currency situation
Maximum potential service	Quarterly	Short-term foreign currency liabilities
Forex reserves quality	Quarterly	Dynamics in forex reserves
Exch. rate competitiveness	Quarterly	International competitiveness of exchange rate
Monetary pressure	Quarterly	Quality of monetary policy
Real economic pressure	Quarterly	Momentum of domestic activity
Domestic leverage	Quarterly	Activity and banks' health
Foreign financing	Quarterly	Dependence on foreign financing

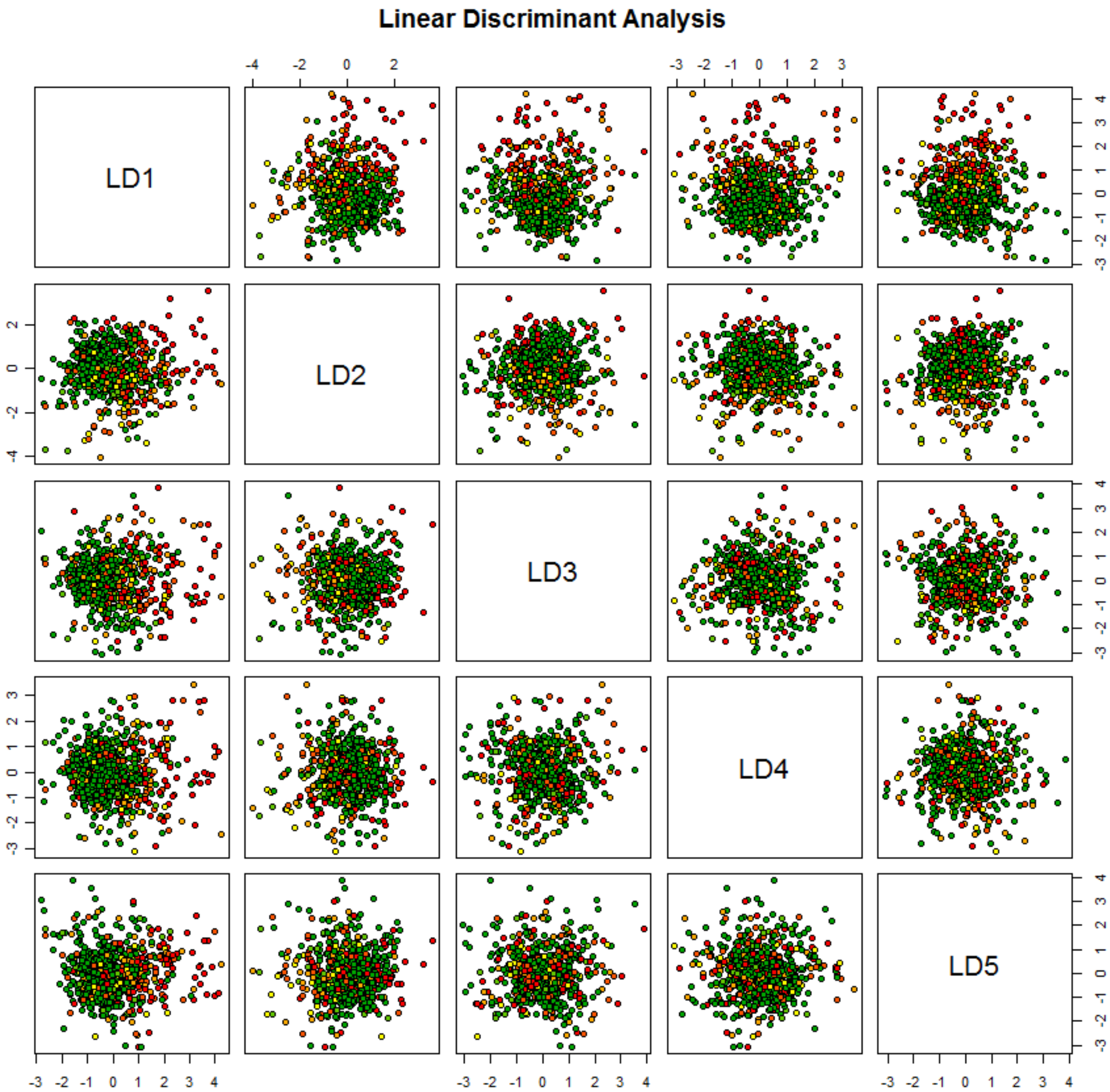
Classification of crises (in red) and stable periods (in green)



Classification of crises (in red) and stable periods (in green)

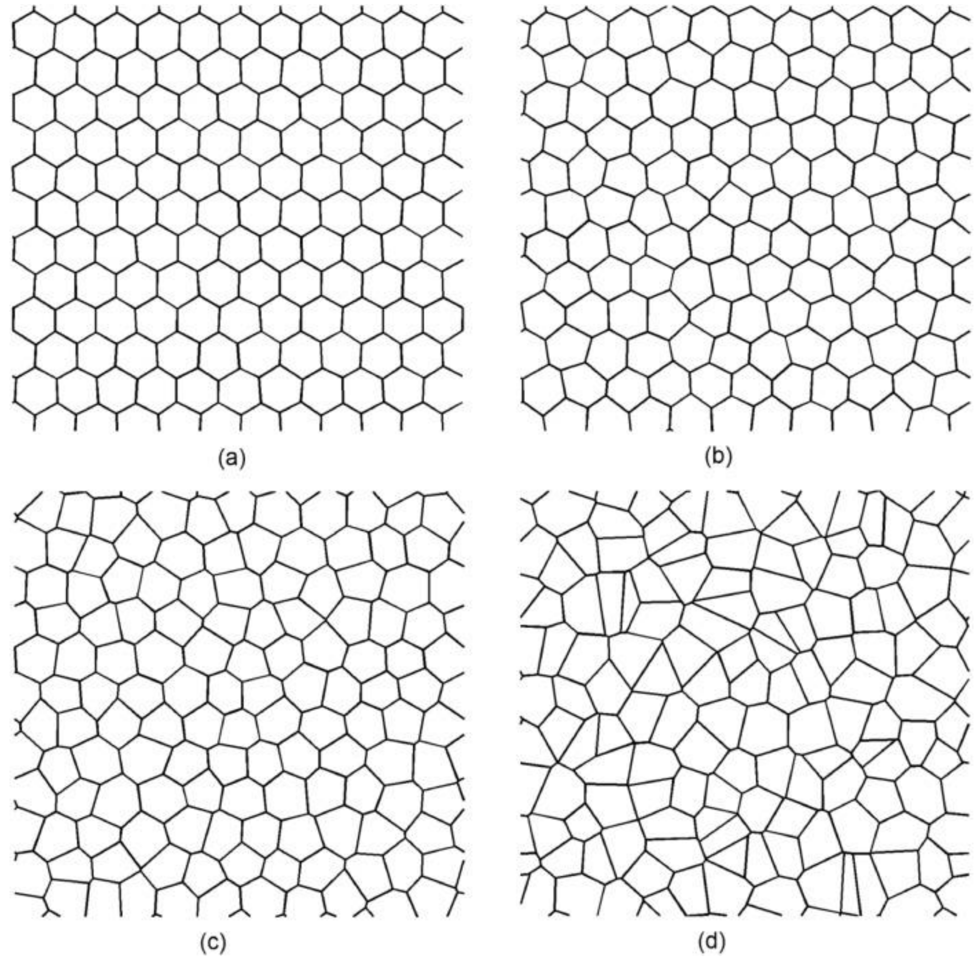


Classification of crises (in red) and stable periods (in green)



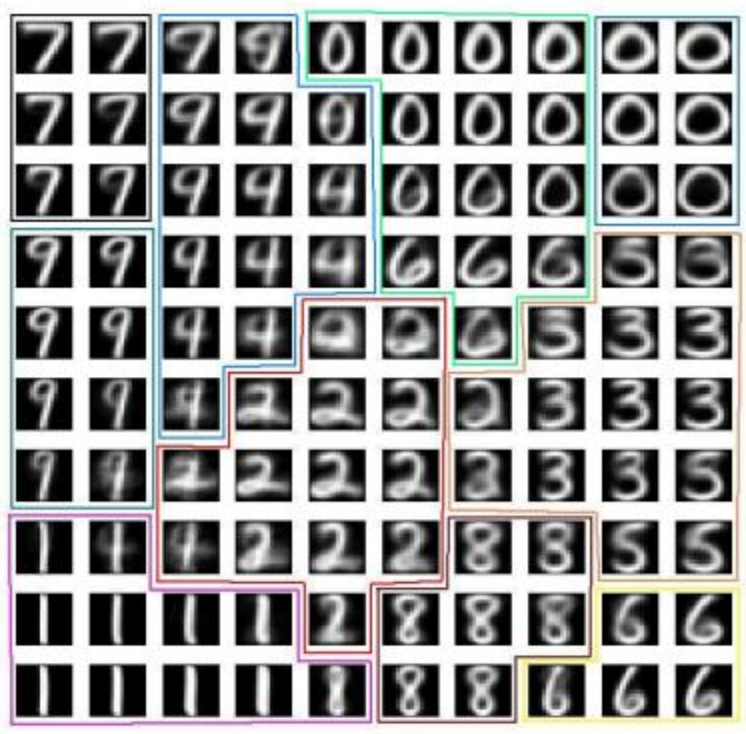
Self Organizing Maps… et Tessellations de Voronoi

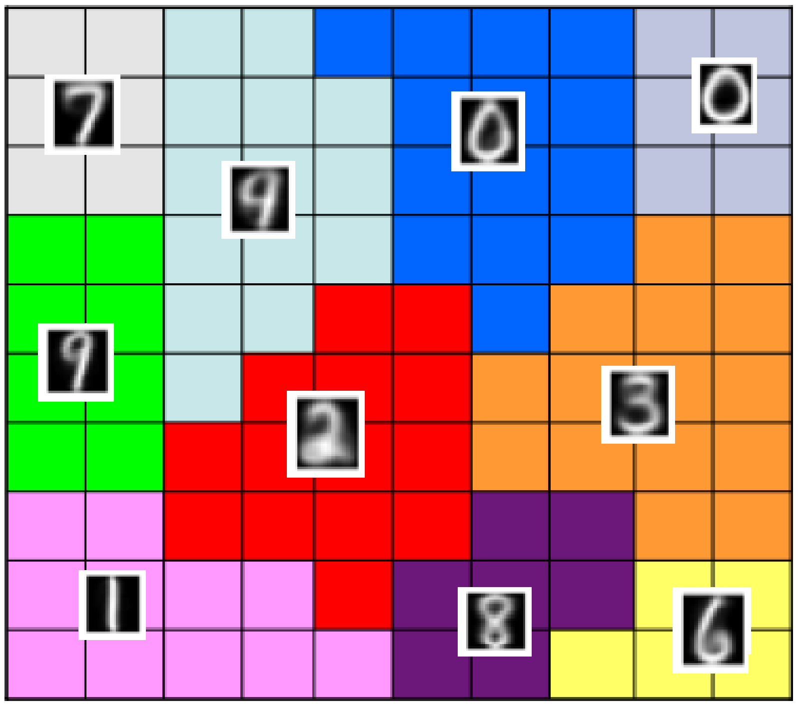
Cas particulier d'un réseau de neurons "non supervise", adapté à l'analyse de données… et à l'identification d'analogies entre individus, entre périodes, entre pays, entre secteurs, …



Source
"Evaluating the performance of microstructure generation algorithms for 2-d foam-like representative volume elements"
https://www.sciencedirect.com/science/article/pii/S016766361630028X

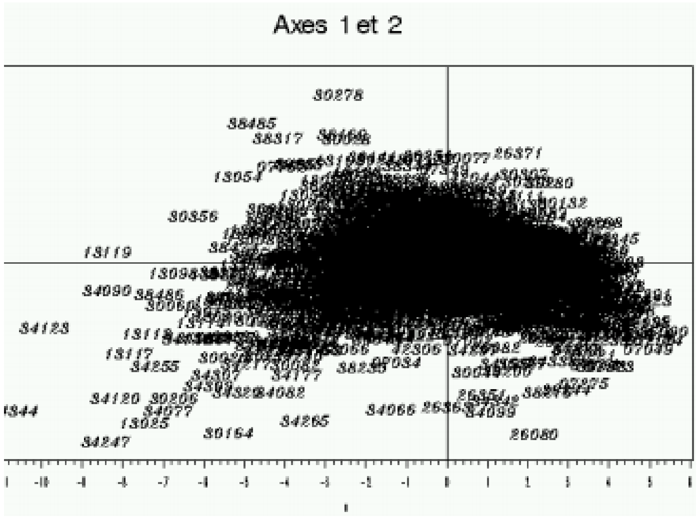
Self Organizing Maps… et Tessellations de Voronoi

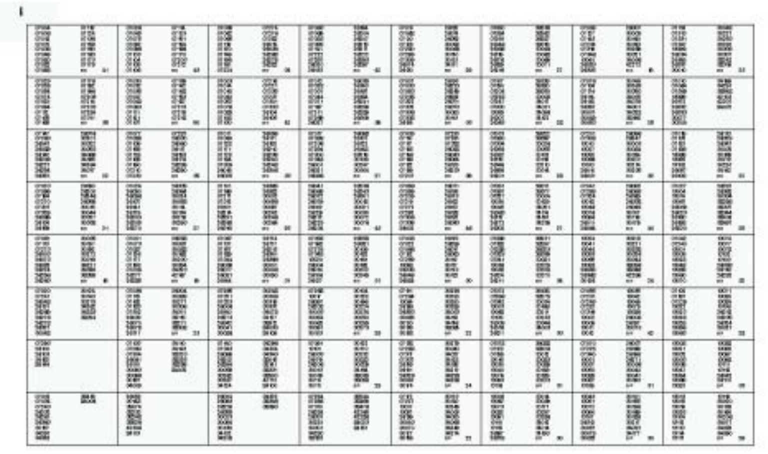




Source
Data Analysis using Self-Organizing Maps
Marie Cottrell and Patrick Letrémy
https://samos.univ-paris1.fr/IMG/pdf_Porvoo_Kohonen_Data_Analysis_V3-2.pdf

Self Organizing Maps… et Tessellations de Voronoi





Source

Data Analysis using Self-Organizing Maps

Marie Cottrell and Patrick Letrémy

https://samos.univ-paris1.fr/IMG/pdf_Porvoo_Kohonen_Data_Analysis_V3-2.pdf

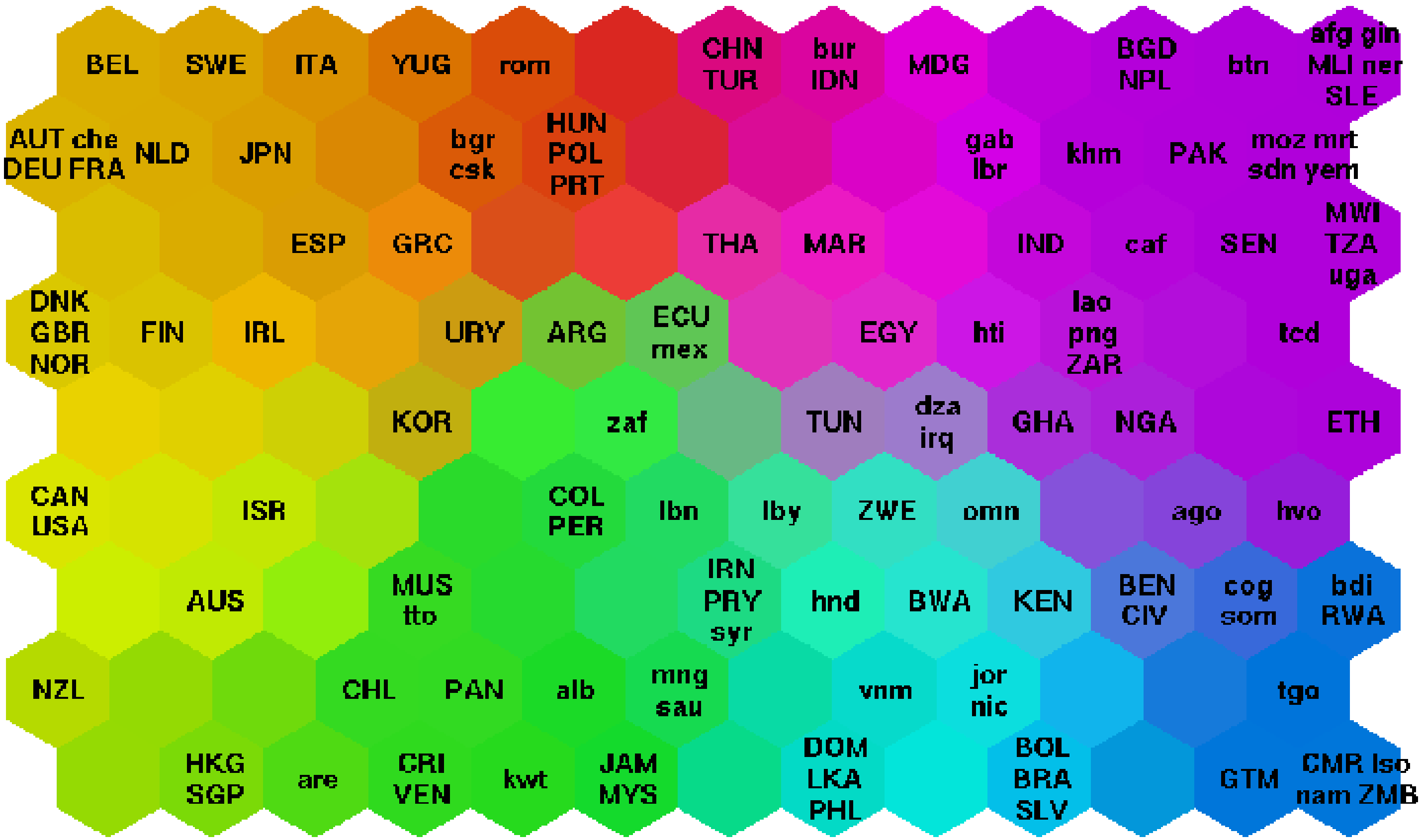
World Poverty Map using Self Organizing Maps
An example on World Bank data, by Samuel Kaski

- 39 World Bank country indicators describing various quality-of-life factors, such as state of health, nutrition, educational services, etc, were used.
- The complex joint effect of these factors can be visualized by organizing the countries using the self-organizing map.
- The map consists of a regular grid of processing units, "neurons". A model of some multidimensional observation, eventually a vector consisting of features, is associated with each unit.
- The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At the same time the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other.

Source: http://www.cis.hut.fi/research/som-research/worldmap.html

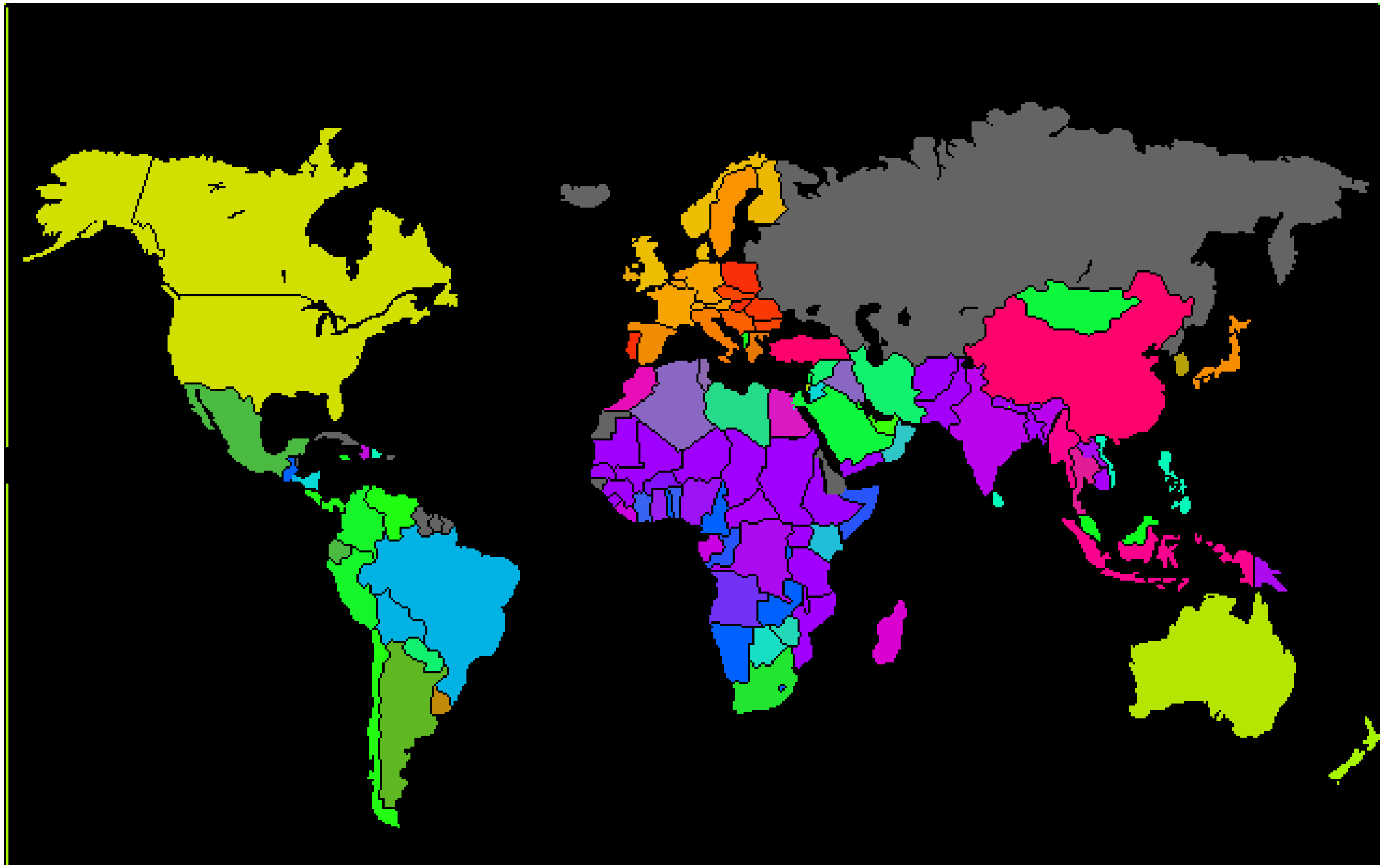
World Poverty Map using Self Organizing Maps
An example on World Bank data, by Samuel Kaski

each country is automatically assigned a color describing its poverty type in relation to other countries



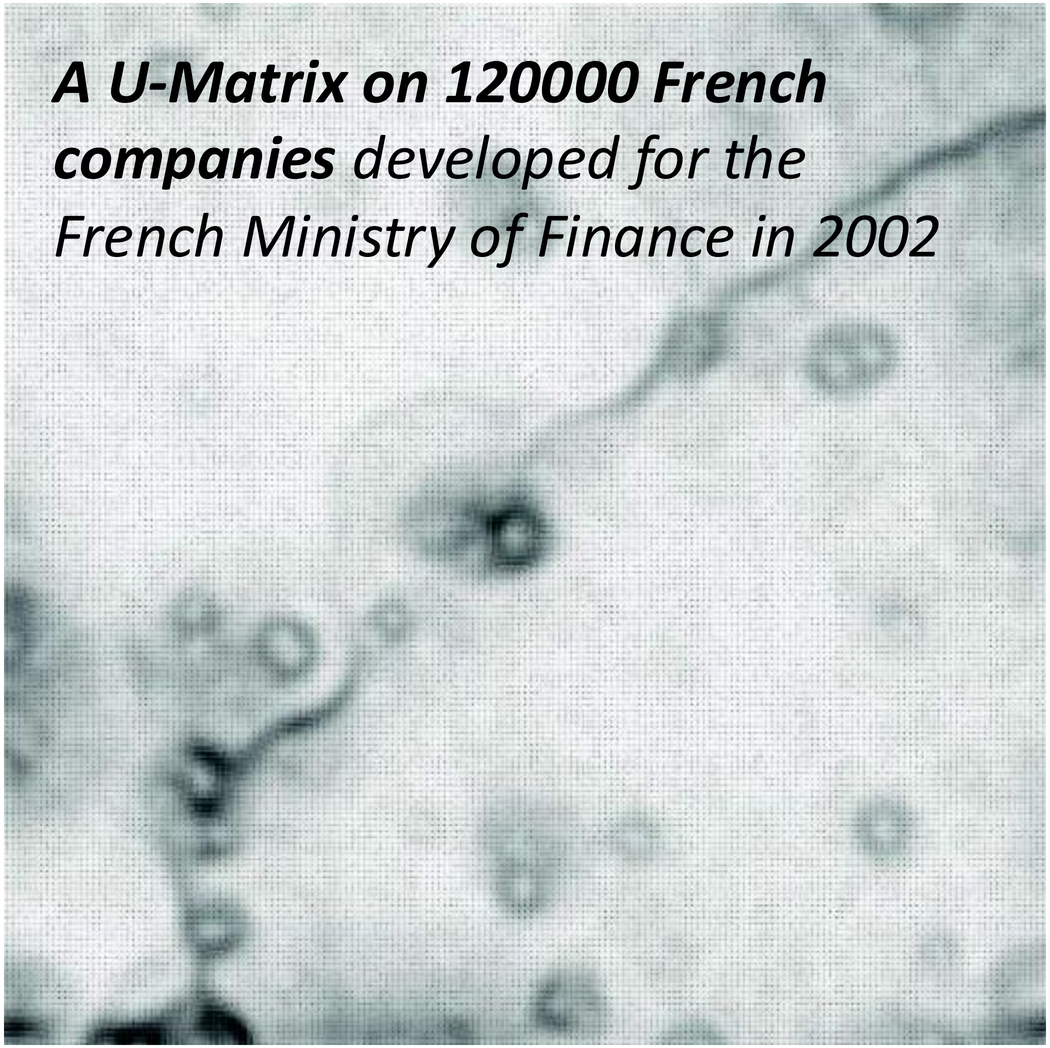
Source: http://www.cis.hut.fi/research/som-research/worldmap.html

World Poverty Map using Self Organizing Maps
An example on World Bank data, by Samuel Kaski

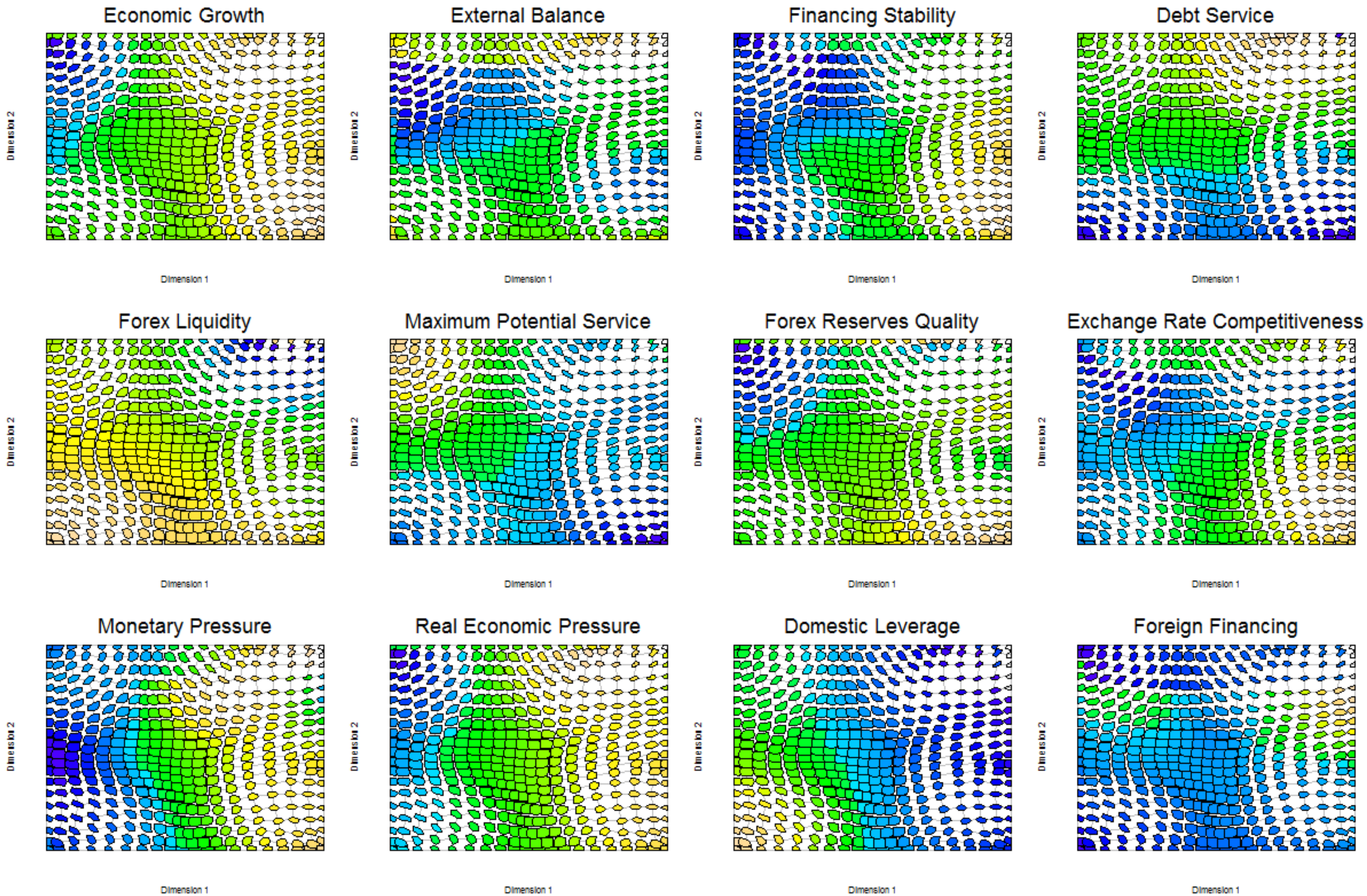


Source: http://www.cis.hut.fi/research/som-research/worldmap.html

Typologie of French Corporates

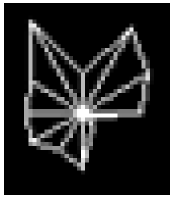
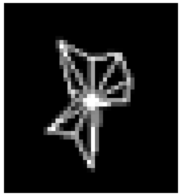
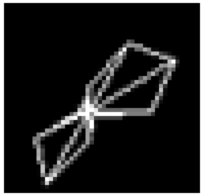
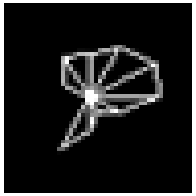


Non-linear patterns revealed by a self organizing map



Web charts on macroeconomic indicators

Countries, indicators, years… and bufferflies

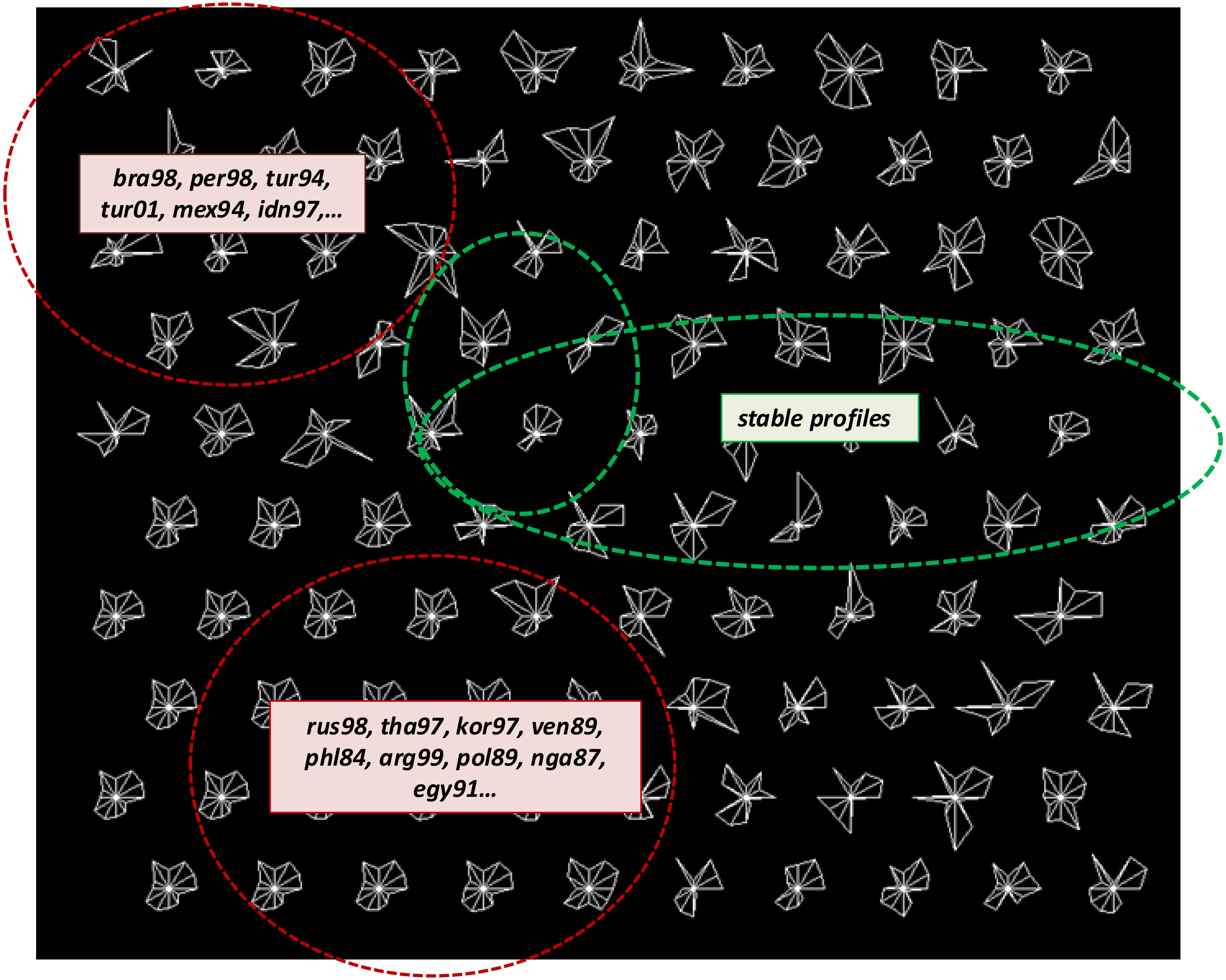
   

on more than 100 countries since the 80s

100 countries x 40 years x 4 quarters per year

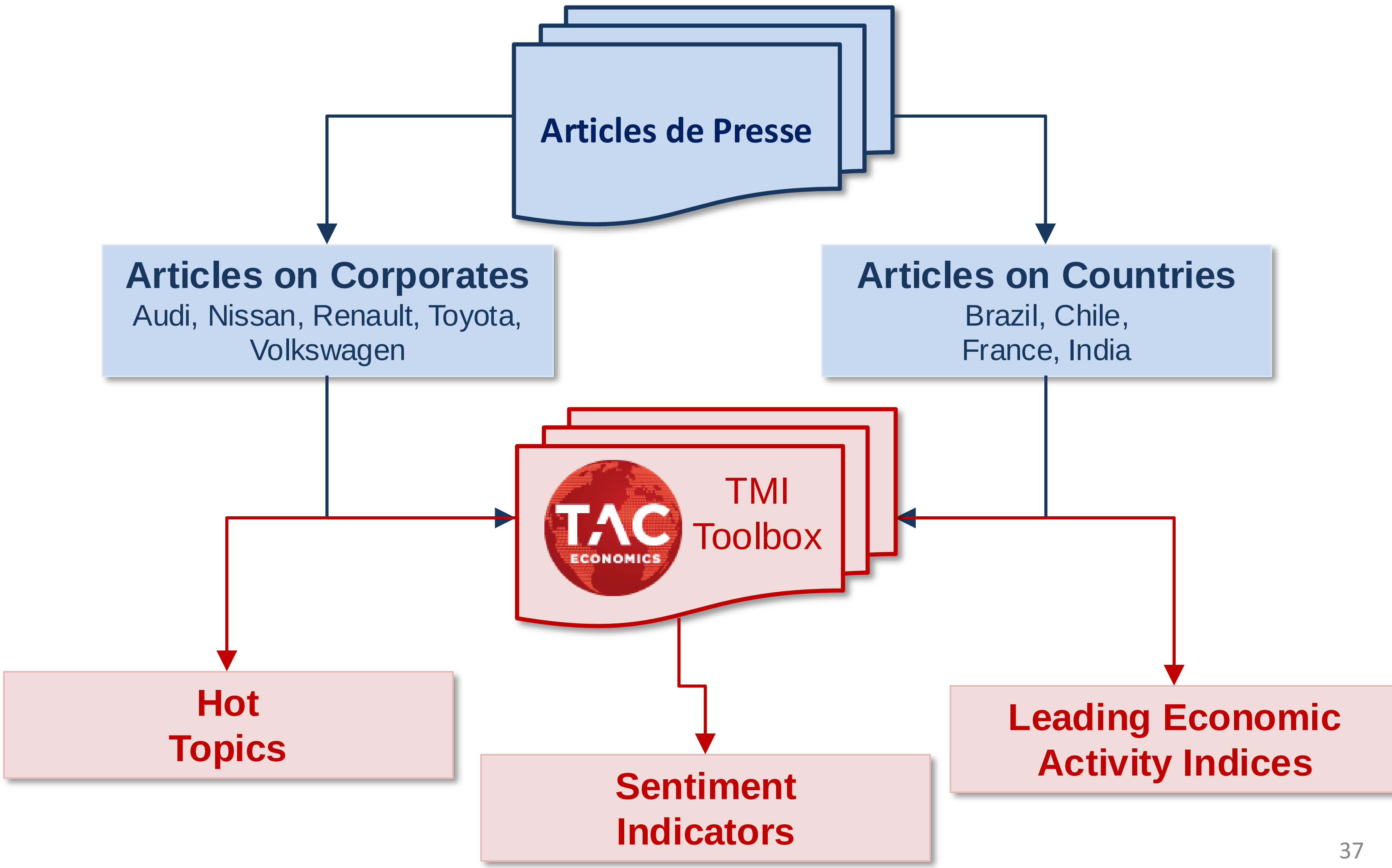
16,000 country web charts to analyse!

Similarities in macroeconmic patterns and neural networks



Text Mining, Sentiment, Topics, …

Objectives


Articles de Presse
Articles on Corporates
Audi, Nissan, Renault, Toyota, Volkswagen
Articles on Countries
Brazil, Chile, France, India
TAC
ECONOMICS
TMI Toolbox
Hot Topics
Sentiment Indicators
Leading Economic Activity Indices

The notion of Term Frequency-Inverse Document Frequency

	obama	biden	trump	inflation	paris	eurozone	emploi
document 1	0	0	5	0	7	0	0
document 2	0	0	0	0	5	0	0
document 3	10	0	0	0	5	0	0
document 4	6	3	0	0	1	0	0
document 5	0	2	0	0	2	0	0
document 6	0	0	5	0	0	2	2
document 7	0	5	0	0	0	3	0
document 8	0	0	2	0	0	20	0
document 9	0	0	4	0	5	0	0
document 10	0	2	3	4	0	0	0
document 11	0	2	0	20	0	10	0
document 12	1	0	2	0	1	0	0
document 13	0	2	0	0	20	0	0
document 14	0	0	20	0	0	0	0
document 15	0	0	0	2	0	5	0
document 16	20	0	0	20	0	0	0

Term Frequency Matrix (TF)

% of occurrence of each word in each doc

	obama	biden	trump	inflation	paris	eurozone	emploi
document 1	0%	0%	42%	0%	58%	0%	0%
document 2	0%	0%	0%	0%	100%	0%	0%
document 3	67%	0%	0%	0%	33%	0%	0%
document 4	60%	30%	0%	0%	10%	0%	0%
document 5	0%	50%	0%	0%	50%	0%	0%
document 6	0%	0%	56%	0%	0%	22%	22%
document 7	0%	63%	0%	0%	0%	38%	0%
document 8	0%	0%	9%	0%	0%	91%	0%
document 9	0%	0%	44%	0%	56%	0%	0%
document 10	0%	22%	33%	44%	0%	0%	0%
document 11	0%	6%	0%	63%	0%	31%	0%
document 12	25%	0%	50%	0%	25%	0%	0%
document 13	0%	9%	0%	0%	91%	0%	0%
document 14	0%	0%	100%	0%	0%	0%	0%
document 15	0%	0%	0%	29%	0%	71%	0%
document 16	50%	0%	0%	50%	0%	0%	0%

Inverse Document Frequency Matrix (IDF)

$$IDF(t) = log\left(\frac{ND}{df(d)}\right)\ ou\ log\left(1 + \frac{ND}{df(d)}\right)$$

ND: number of documents and df(d) : number of documents width the word « d »

If a word is rarely mentioned in documents, it is considered as « more important » and its weight is therefore high

The notion of TF-IDF

$$w(t,d) = TF(t,d) * log\left(1 + \frac{ND}{f(t)}\right)$$

We use these algorithms to optimise websites. The underlying idea is not to use frequent terms, but to try to write « unique web pages », to increase their score.

The notion of TF-IDF

Example

The word « recession » is only mentioned twice in a document of 100 words. Its TF is therefore 2%.

If my dataset consists of 10,000,000 of documents, and only three of them mention the word « recession ». Then, the IDF is log(10,000,000/3) = 6.5

The TF-IDF is then 2% * 6.5 = 13%

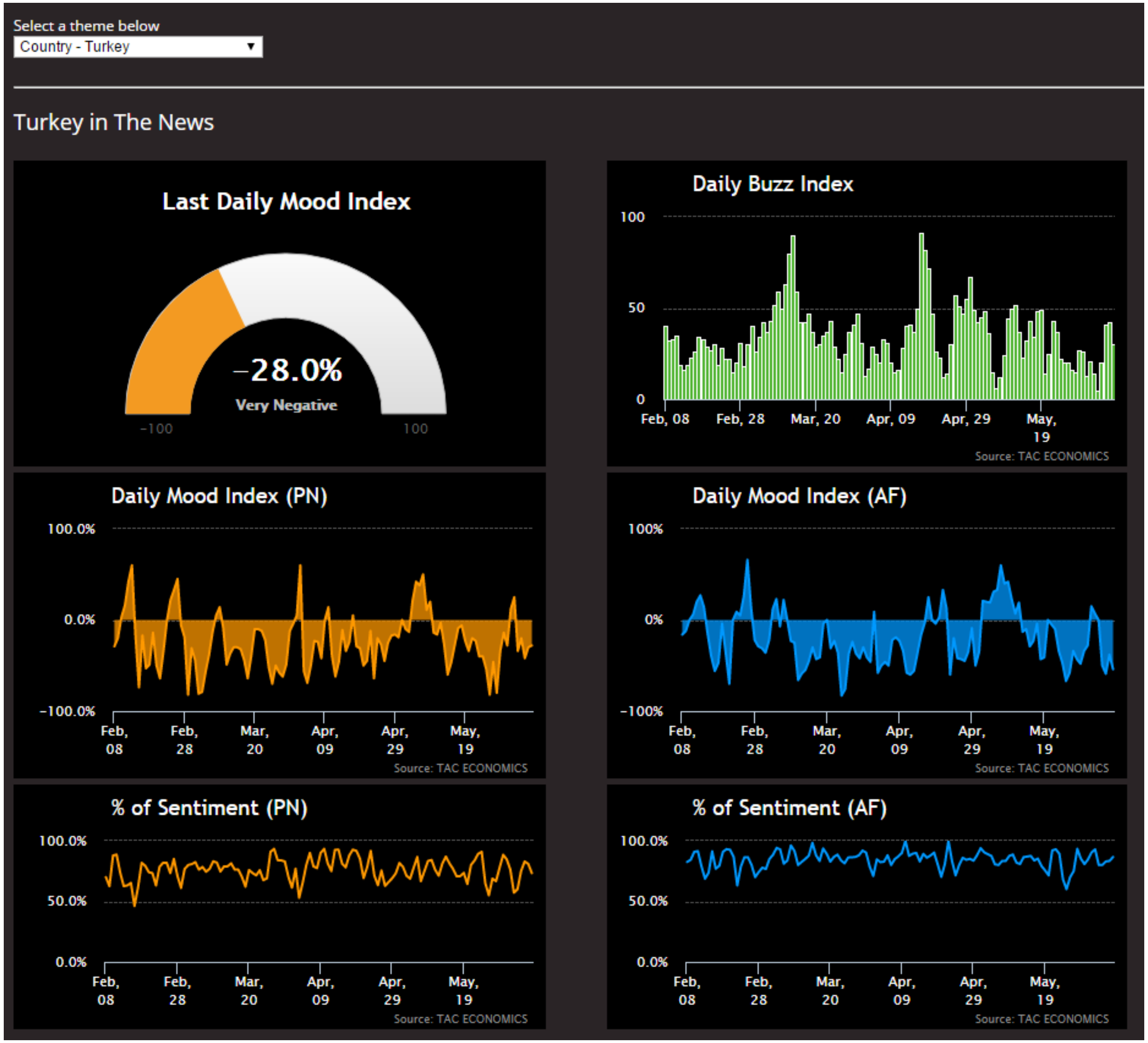
The database of News & Tweets

- A list of 1000 websites, newspapers, information providers and blogs, in different languages.
- Possibility to extent the dataset to larger databases, such as Lexis Nexis or Factiva.
- Every 3 hours our web-crawlers store the information published on these websites on internal databases (approx. 20,000 per day).
- A different set of crawlers extract and store tweets on a pre-defined list of keywords or corporates. However, limitations imposed by Tweeter (GNIP, Datasift, Topsy,…).

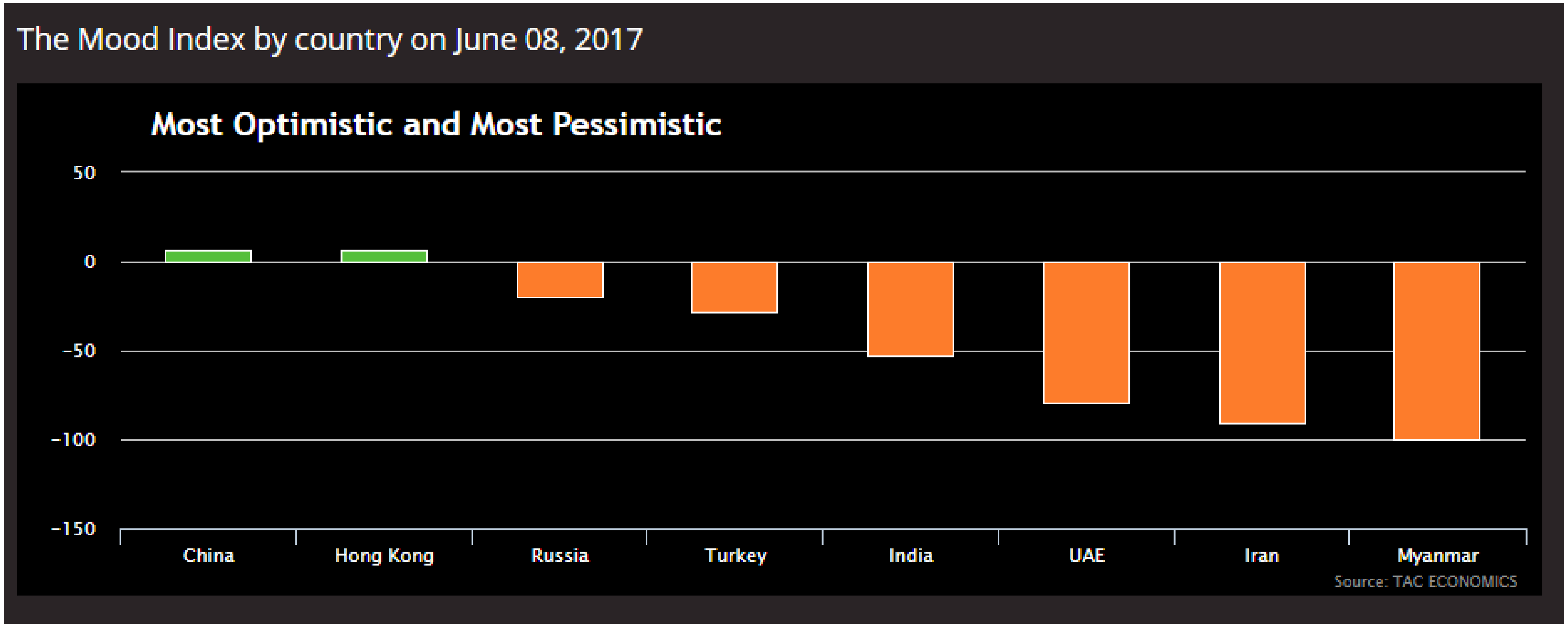
Extraction of Hot Topics in the News

- An internal algorithm to extract most important articles and topics, on a daily, weekly and monthly basis.
- The list of key topics is available on a large sample of countries and some multinational companies.
- The list can be modified to include any theme or keyword.
- Topics automatically published on our portal in real-time.

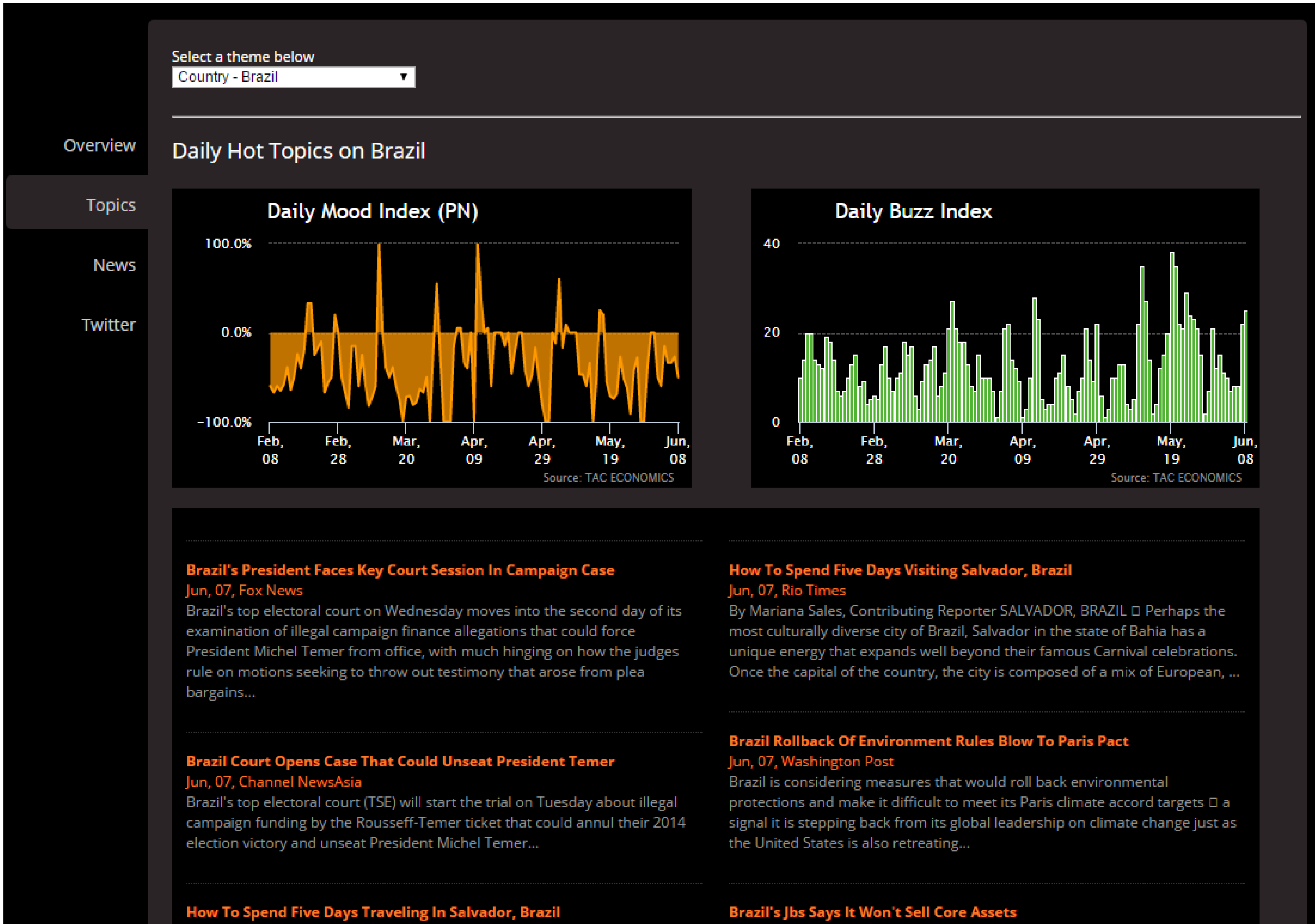
Sentiment analysis world news



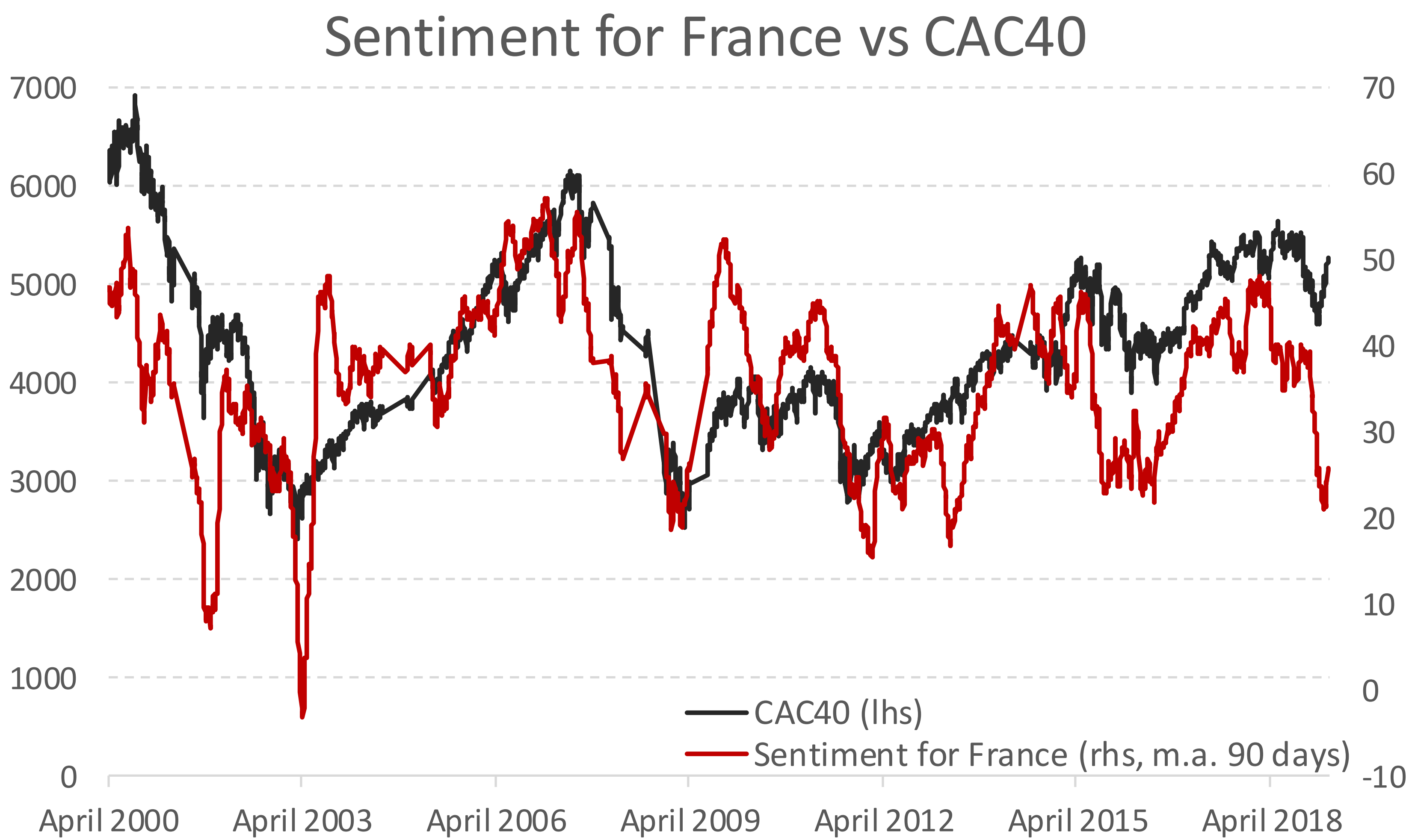
TAC Mood Index by Country



Hot Topics of the Day

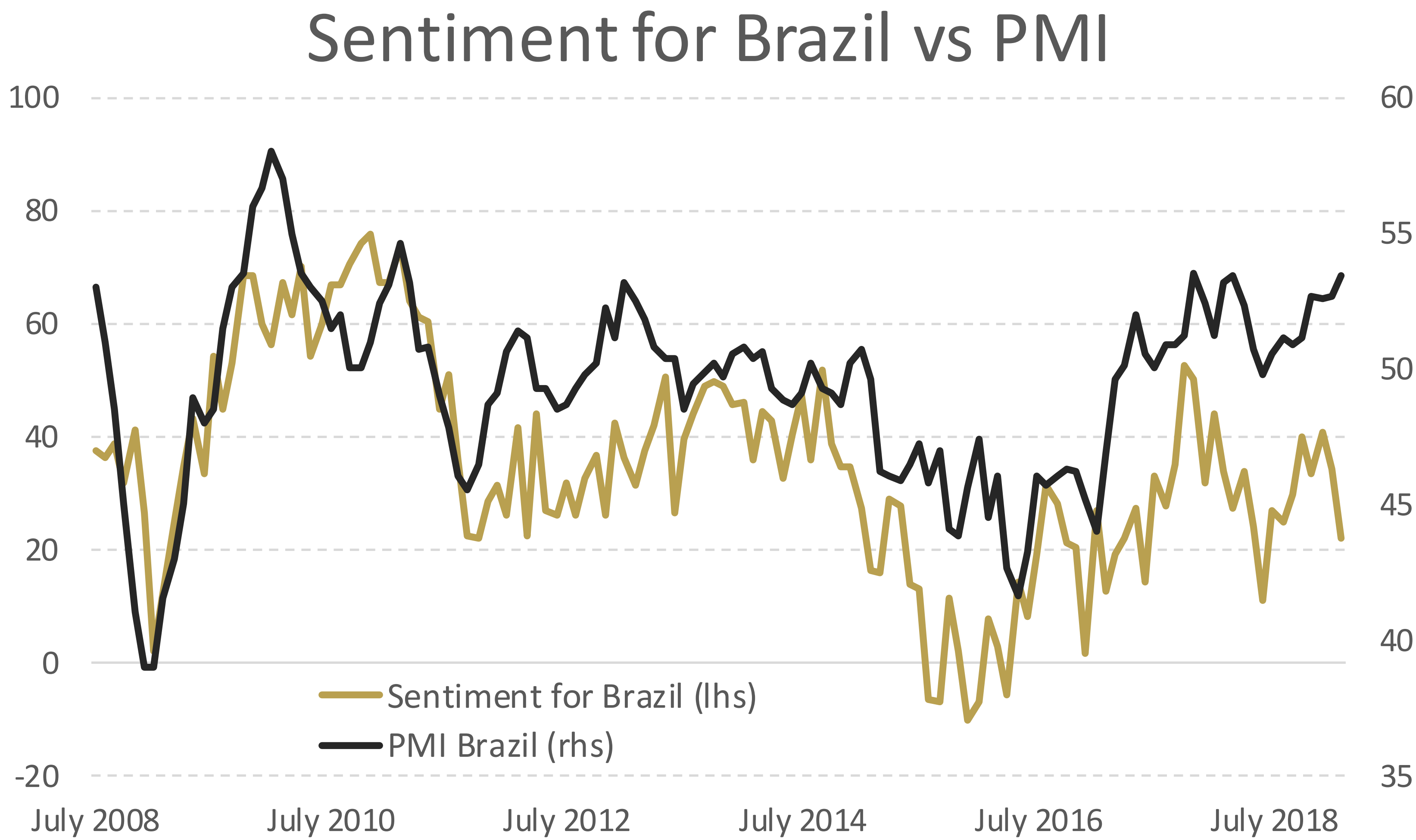


Sentiment Indicators and Financial Markets



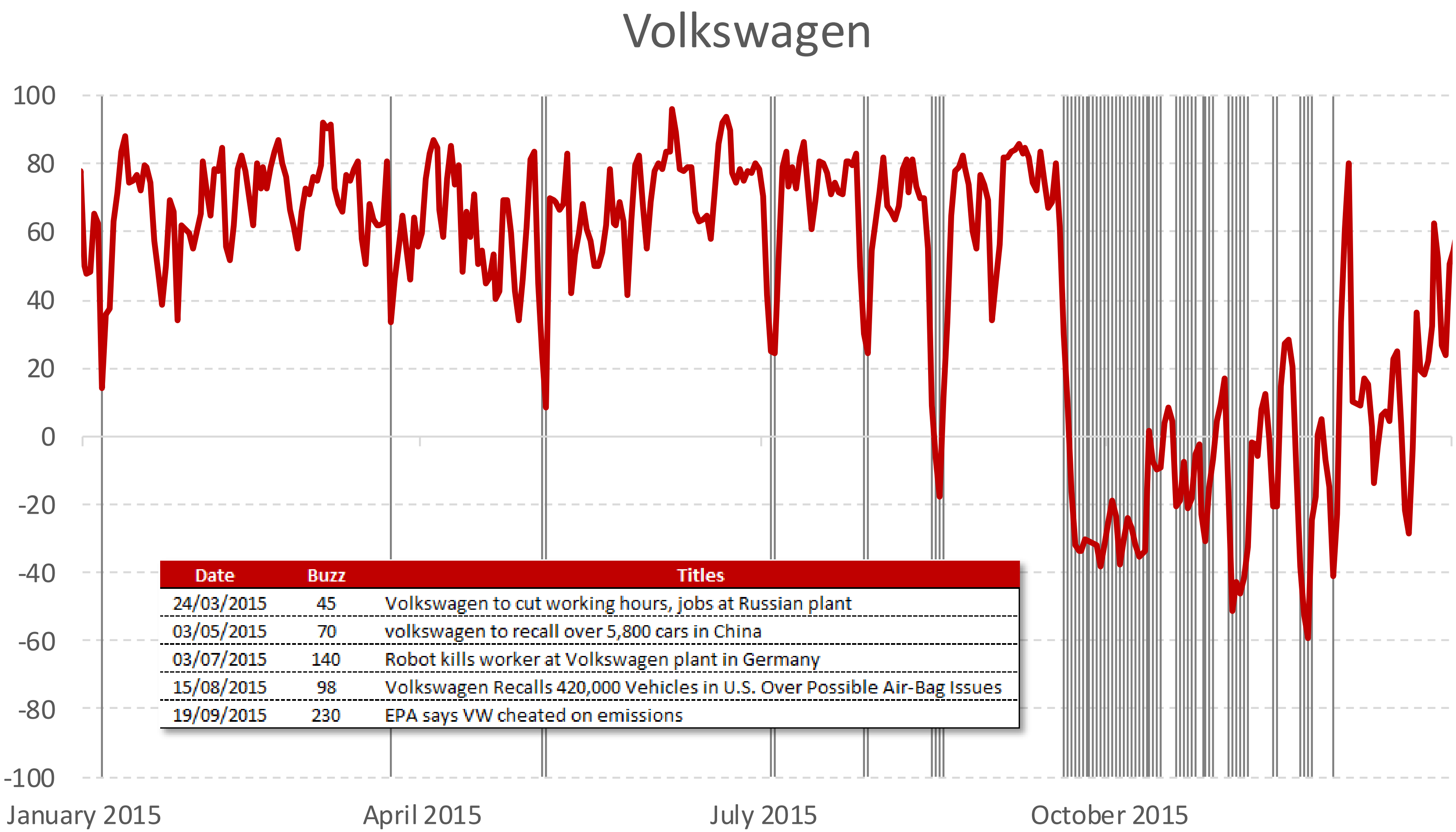
Source: TAC ECONOMICS

Brazil



Source: TAC ECONOMICS

Reputational Risk: Volkswagen & the Diesel Gate



Date	Buzz	Titles
24/03/2015	45	Volkswagen to cut working hours, jobs at Russian plant
03/05/2015	70	volkswagen to recall over 5,800 cars in China
03/07/2015	140	Robot kills worker at Volkswagen plant in Germany
15/08/2015	98	Volkswagen Recalls 420,000 Vehicles in U.S. Over Possible Air-Bag Issues
19/09/2015	230	EPA says VW cheated on emissions

Source: TAC ECONOMICS

Deep Learning Applied to SP500 Forecasting

Neural networks in finance?

- 90s: neural networks were viewed as a substitute to expert systems.
- NN applied to stock prices forecasting: Kimoto et Yoda (1993), on the Tokyo Stock Index with… 5 input variables!
- NN also applied to gold price and S&P500 by Grudnitski & Obsburn (1993) or Quang Do (1995).
- NN applied to exchange rate forecasting mid-90, by Rawani (1993), Azoff (1994) and Avouyi Dovi (1995).
- Today, a very large number of machine learning models applied in the field of finance… but not much deep learning and convolutional networks at this stage.

Largest hedge funds & quantitative finance

	AUM in $bn in 2017	Quantitative
Bridgewater Associates	122.2	No
AQR Capital Management	69.6	Yes
JPMorgan AM	45.0	(Yes)
Renaissance Technologies	42.0	Yes
Two Sigma	38.9	Yes
De Shaw & Co	34.7	Yes
Man Group	33.9	Yes
Millennium Management	33.9	Yes
Och-Ziff Capital Management	33.5	(Yes)
Winton Group	32.0	(Yes)

Deep learning in finance ?

Stephen Roberts et de Sid Ghoshal (University of Oxford)

« Thresholded ConvNet Ensembles: Neural Networks for Technical Forecasting » (July 2018)

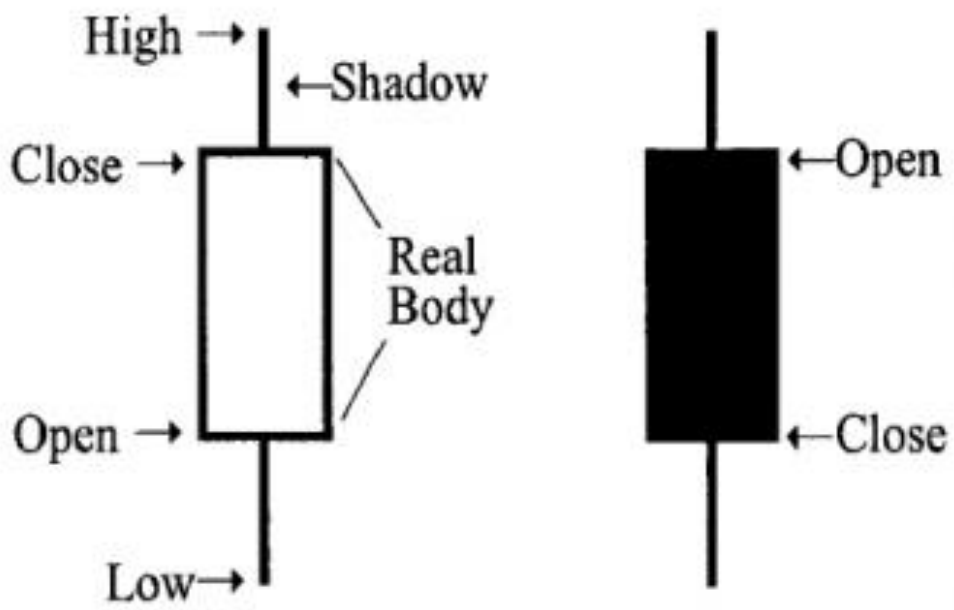


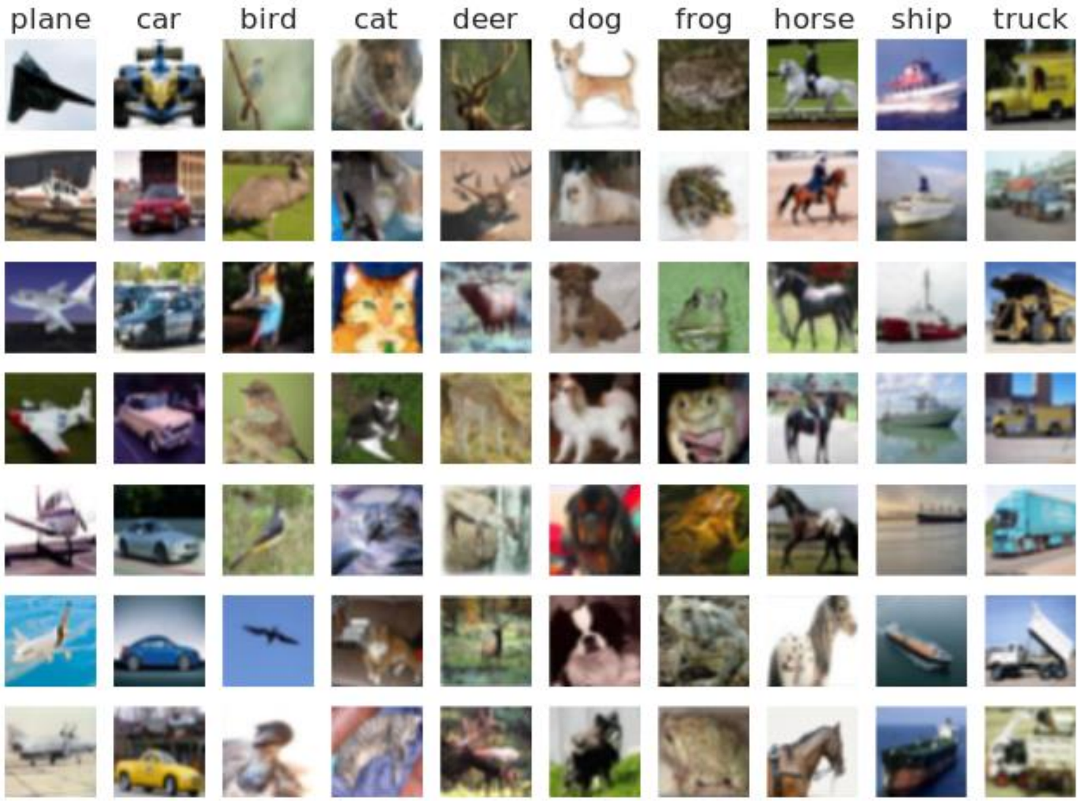
Figure 1: Candlestick representation of financial time series data.

Technival analysis and machine learning

Source: https://www.abcbourse.com/apprendre/11_heikin_ashi_et_hollow_candle.html

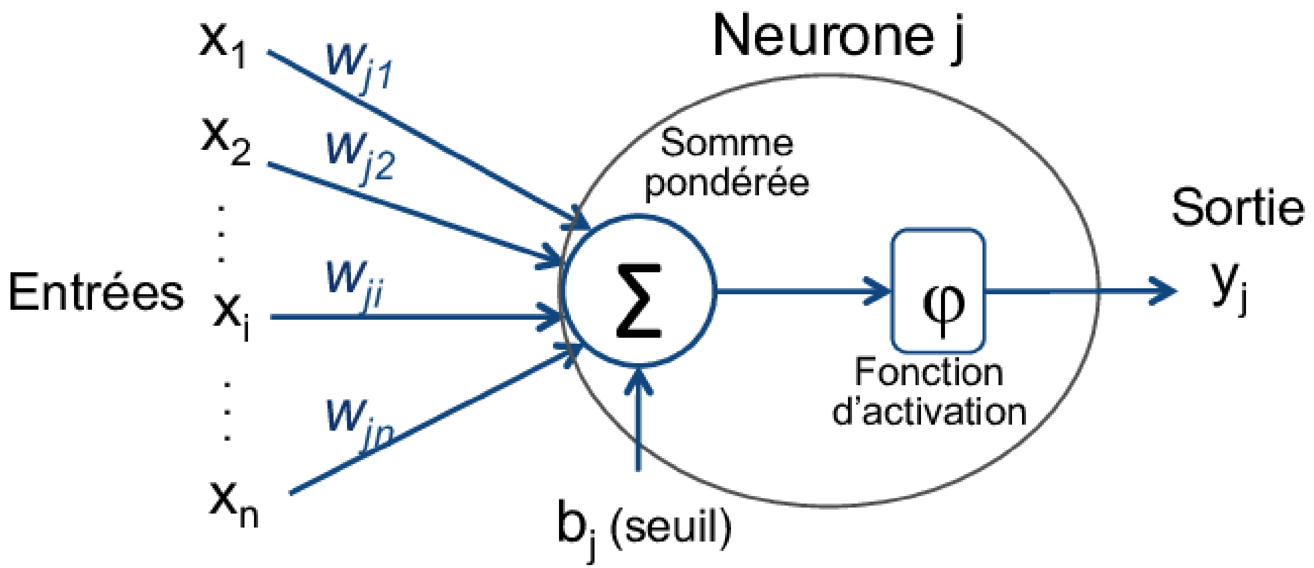
Computer vision & artificial intelligence

easy for humans, but more difficult for computers

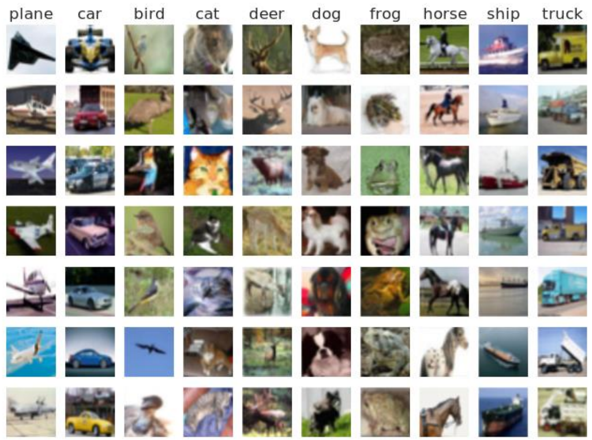


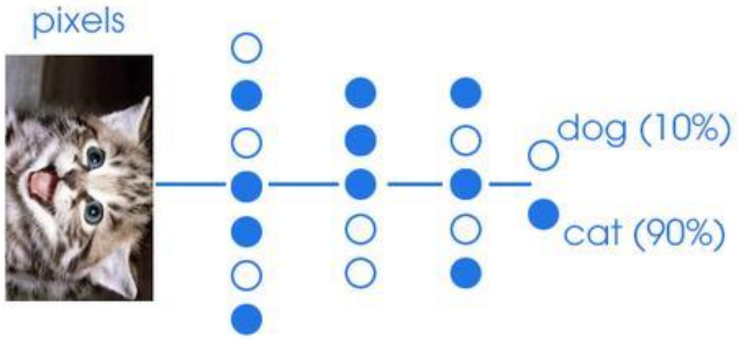
Source: dominodatalab

Structure élémentaire du réseaux de neurones



Standard machine learning techniques



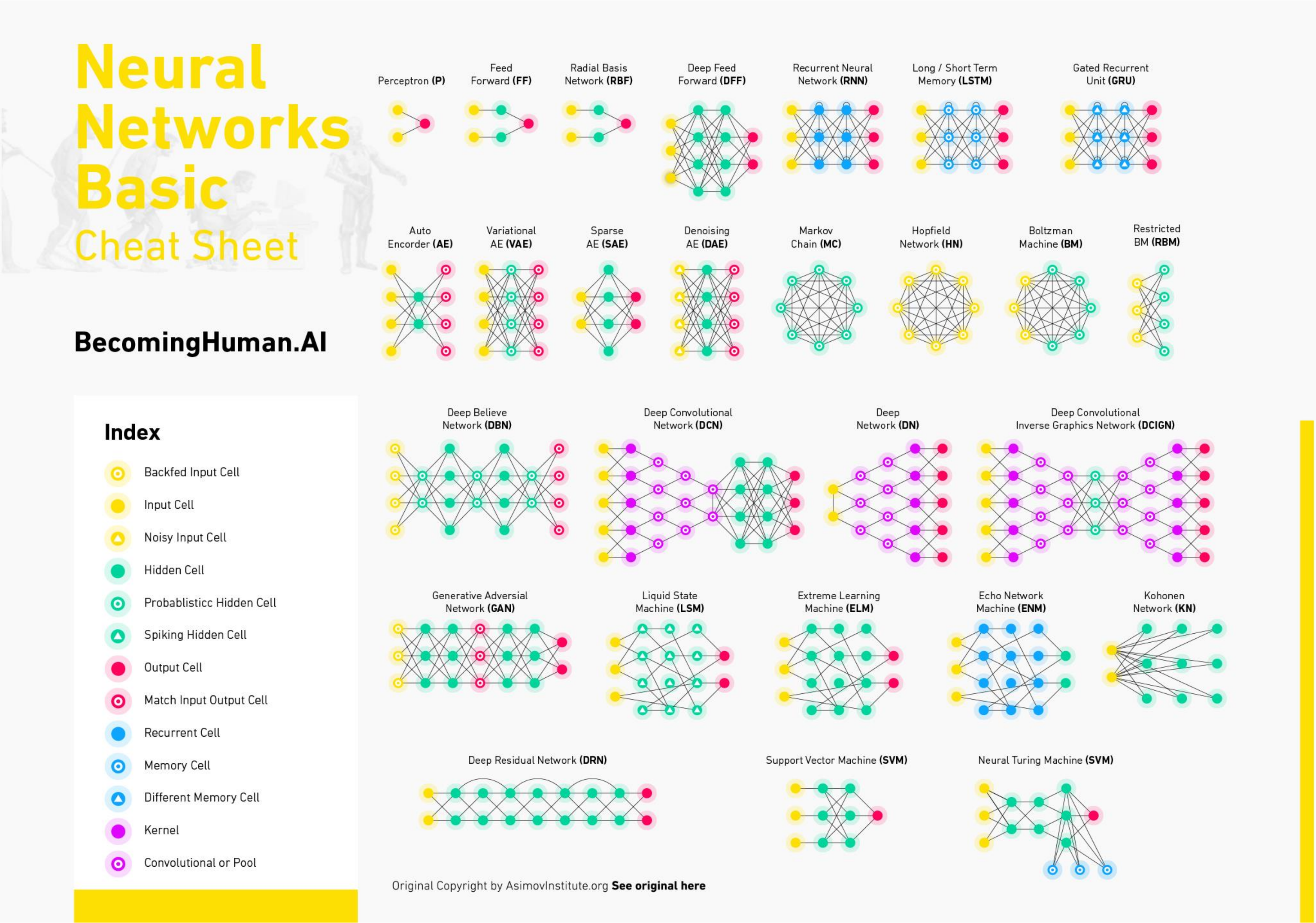


The input of these neural networks are the pixels,
and the values are the colors

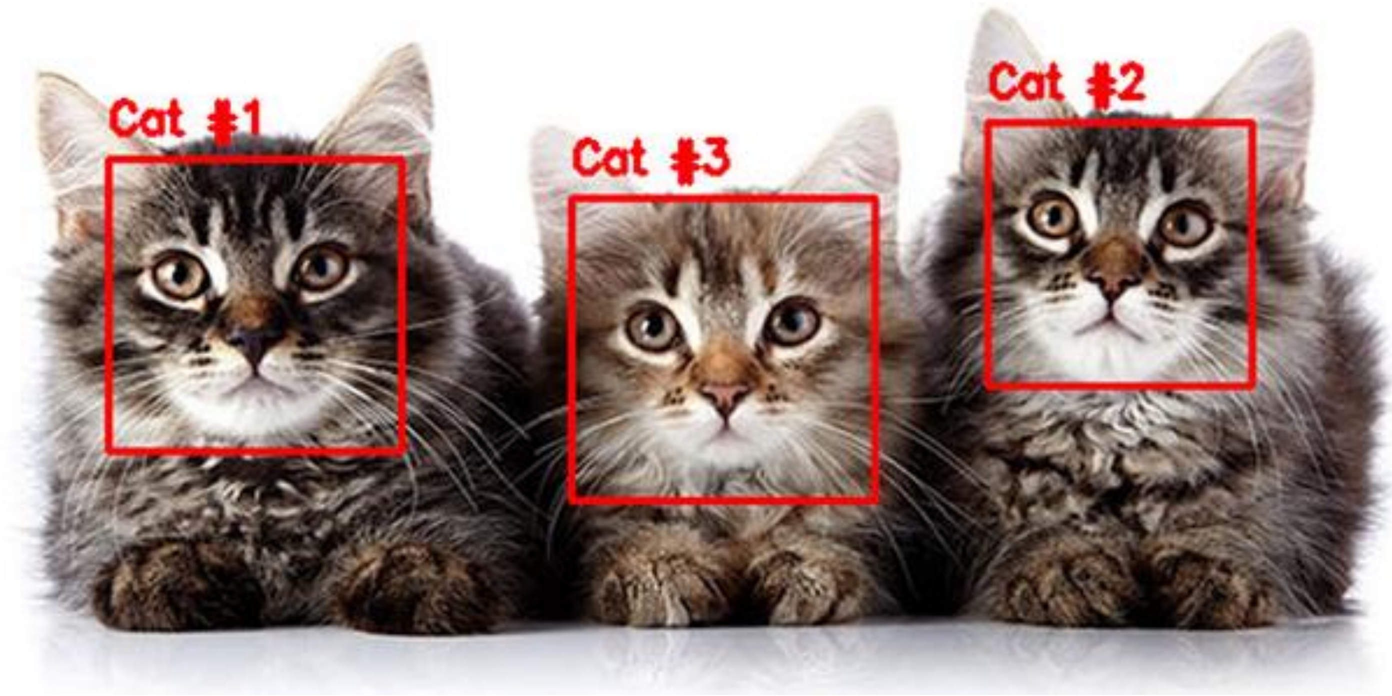
The example of dogs and muffins

Sometimes it is also difficult for humans!

L'écosystème du « deep learning »

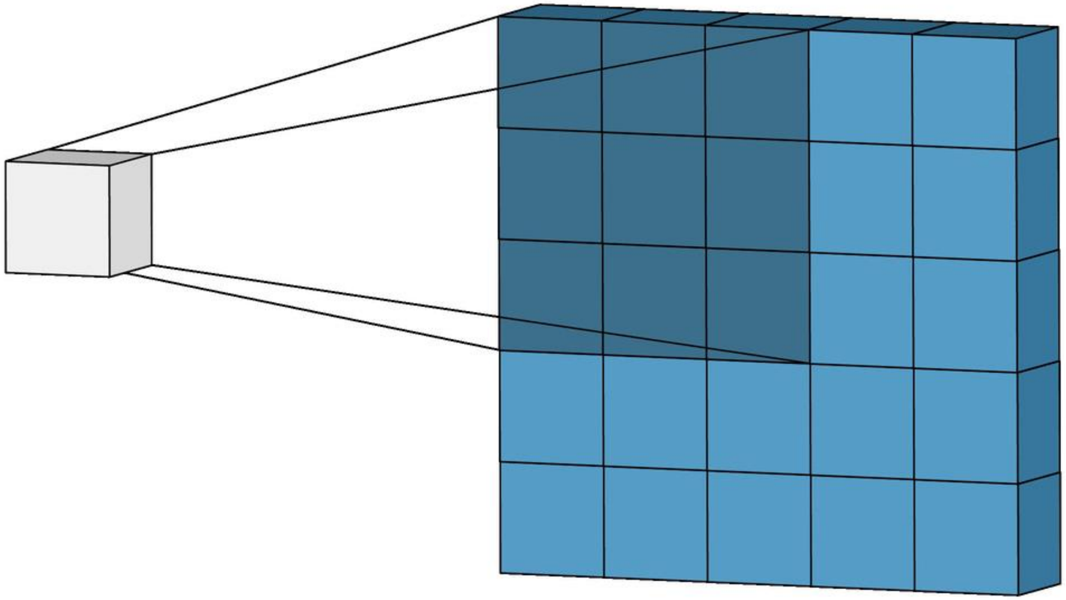


Deep Learning and Convolution?



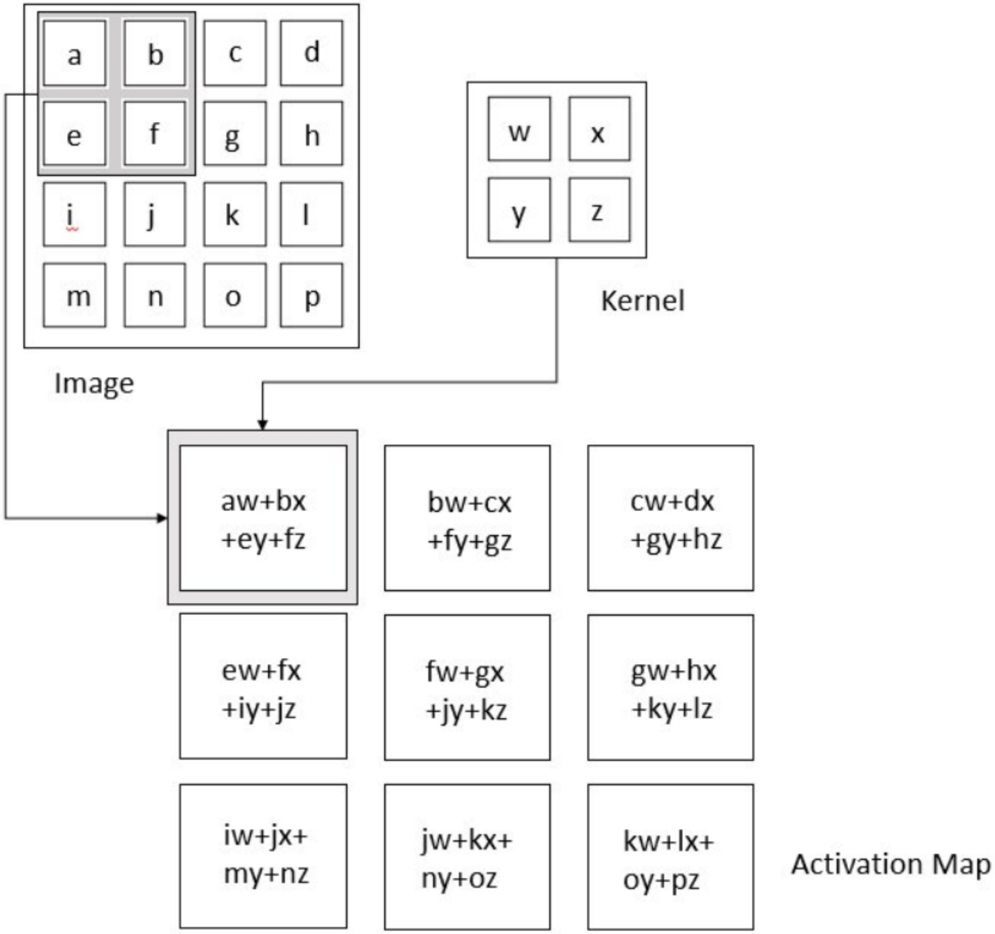
The input of these neural networks are the pixels,
and the values are the colors…
… but the convolution goes by itself beyond the pixels!

How convolution works?



Source:
https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

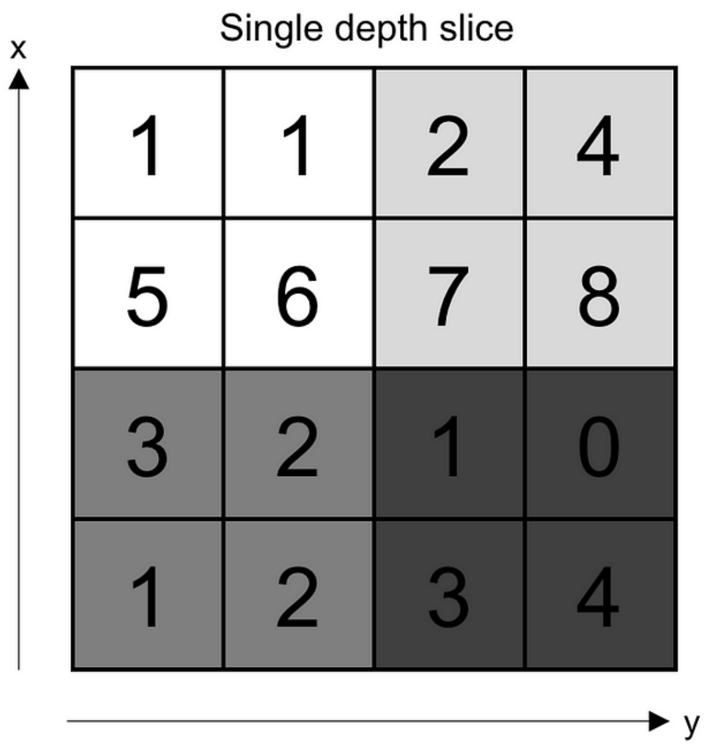
Convolution



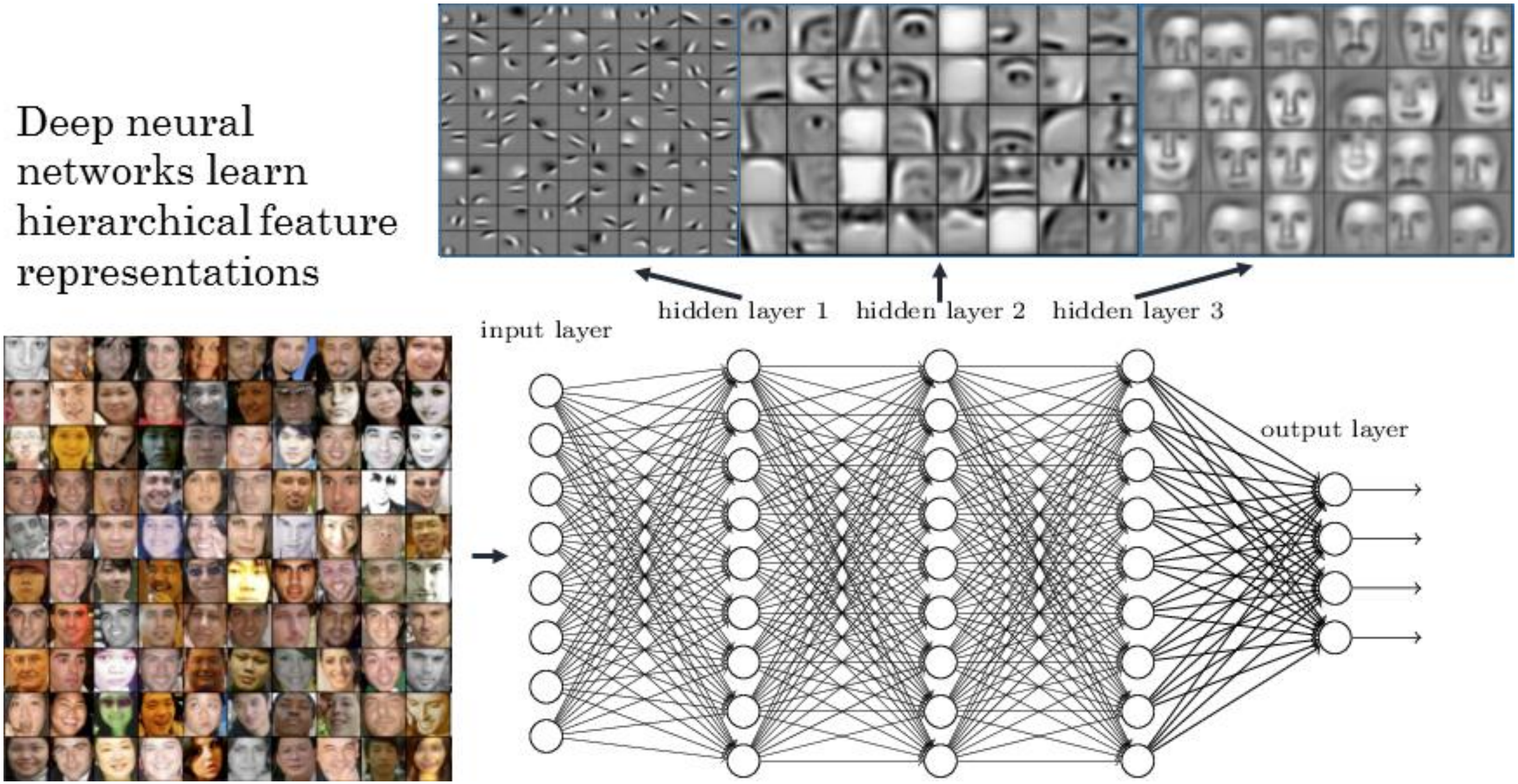
Source:
Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Pooling

Objectif de réduire la quantité d’information à conserver au voisinage d’un point.



How convolution works?



Source:
https://www.strong.io/blog/deep-neural-networks-go-to-the-movies

Technical analysis and SP500

PATTERN	γ	$\mu(bp)$	$\sigma(bp)$
UNCONDITIONAL		4.26	229.40
HAMMER	5.13	-15.80	223.04
INVERTED HAMMER	5.00	+13.75	211.62
HANGING MAN	3.71	-14.92	222.06
SHOOTING STAR	4.78	+12.01	232.42
DRAGONFLY DOJI	14.73	-25.81	219.99
GRAVESTONE DOJI	12.93	+22.41	223.57
BULLISH SPINNING TOP	2.64	-0.72	214.70
BEARISH SPINNING TOP	1.67	+0.94	213.30
BULLISH ENGULFING	1.61	-0.28	236.50
BEARISH ENGULFING	4.16	+5.75	238.47
BULLISH HARAMI	1.07	+5.5	222.17
BEARISH HARAMI	1.51	-0.96	219.43
PIERCING LINE	2.29	+0.78	241.42
CLOUD COVER	1.06	-0.75	218.24
TWEEZER BOTTOM	1.76	+4.19	223.23
TWEEZER TOP	1.22	+2.97	221.93
ABANDONED BABY-	3.29	-4.04	232.45
ABANDONED BABY+	1.27	+2.94	232.28
EVENING STAR	2.89	-0.27	231.76
MORNING STAR	1.80	+2.59	231.89
THREE BLACK CROWS	6.85	+13.09	229.40
THREE WHITE SOLDIERS	6.30	-11.77	203.26
THREE INSIDE DOWN	1.63	+2.72	233.12
THREE INSIDE UP	2.50	+0.13	220.75

*Source: **Stephen Roberts et de Sid Ghoshal** (University of Oxford): « Thresholded ConvNet Ensembles: Neural Networks for Technical Forecasting »*

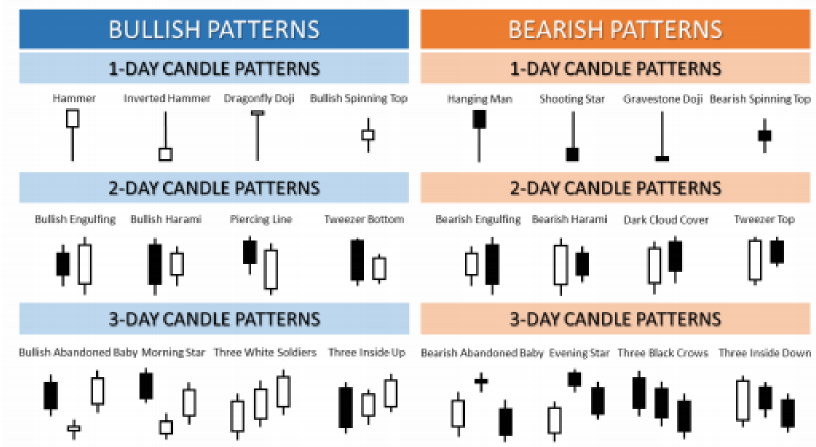


Figure 2: For each timescale (1-day, 2-day and 3-day), we specify 8 chartist patterns and the future direction they predict ('bullish' for positive returns, 'bearish' for negative returns).

Deep learning applied to SP500

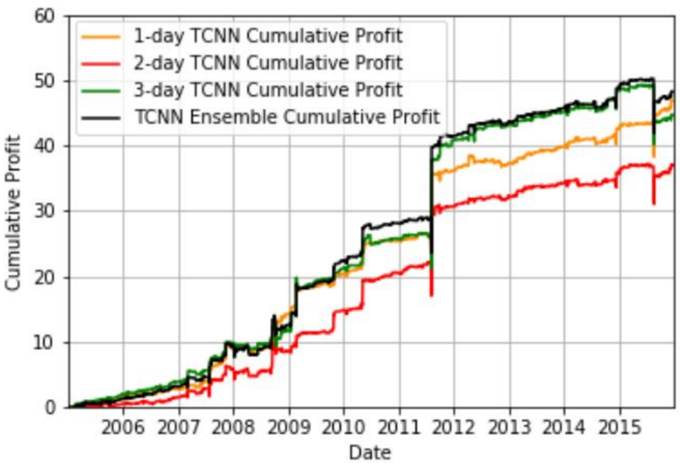


Figure 8: Cumulative profit (as a multiple of starting wealth, per Table 6) generated by the various TCNN models between Jan-2005 and Dec-2015, in the absence of friction costs. The models are steadily profitable, with occasional spikes related to recognising major events. Drawdowns are infrequent and of limited scale.

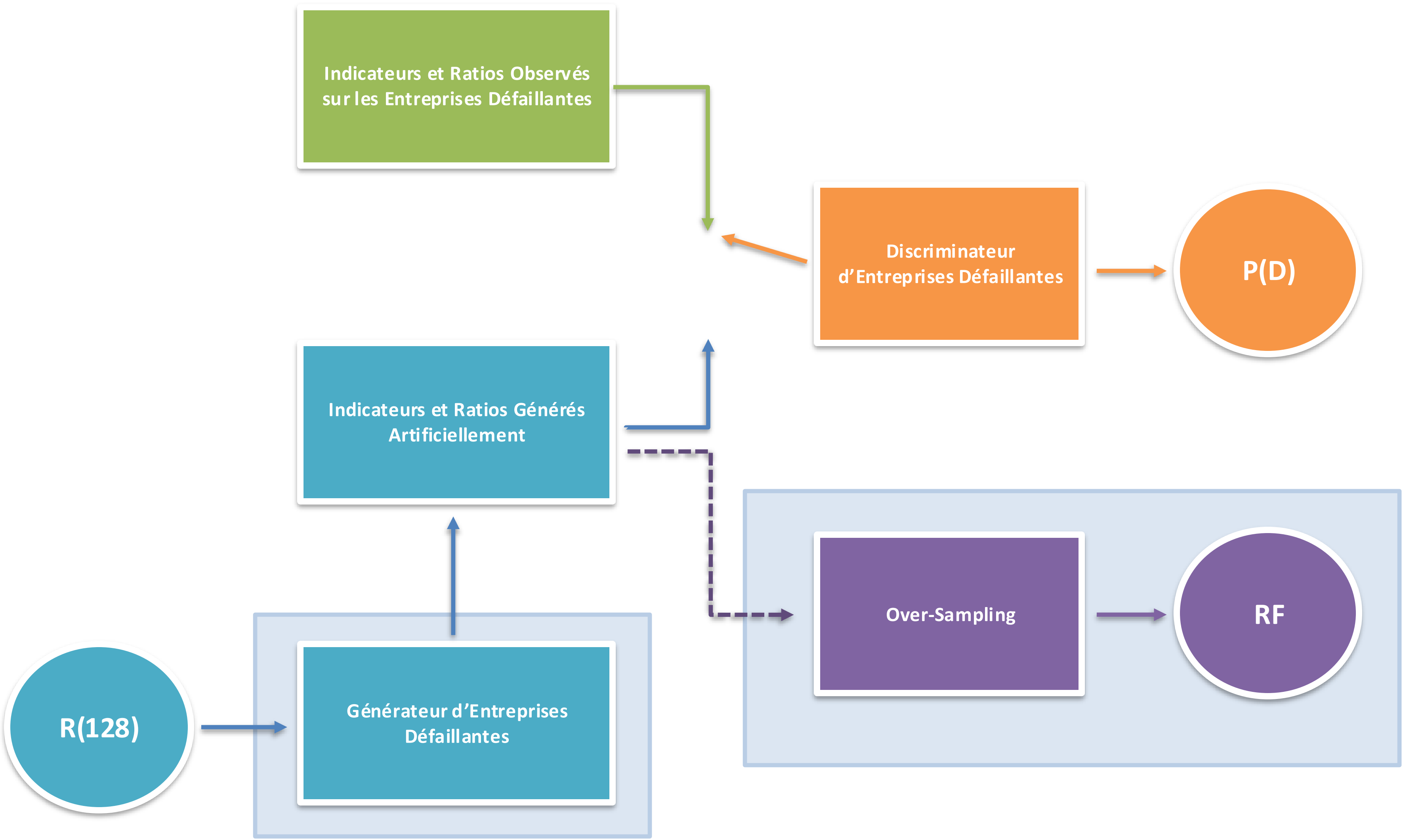
Table 8: Benchmark performance across a range of supervised learning models trained on S&P500 technical data for Jan 1994 - Dec 1994 and tested on Jan 2005 - Dec 2015.

Model	Accuracy (%)	AUC (%)	Z	Significance
MLP	50.6	51.1	23.766	> 0.9999
Technical NN	49.8	49.9	−1.878	–
1-day CNN	51.3	51.8	36.546	> 0.9999
2-day CNN	51.2	51.5	31.291	> 0.9999
3-day CNN	51.2	51.5	31.423	> 0.9999
CNN Ensemble	51.2	51.7	35.628	> 0.9999
1-day TCNN	56.7	57.2	14.533	> 0.9999
2-day TCNN	56.3	56.5	13.017	> 0.9999
3-day TCNN	55.9	56.2	12.493	> 0.9999
TCNN Ensemble	57.5	57.5	15.301	> 0.9999
RNN-LSTM	50.8	51.0	19.616	> 0.9999
RNN-GRU	50.9	51.2	24.880	> 0.9999
1-NN	50.0	50.1	1.087	0.8614
10-NN	49.9	49.8	-3.317	–
100-NN	49.7	49.6	-7.651	–
Linear SVM	49.9	49.8	-0.962	–
RBF SVM	49.9	49.8	-2.416	–
10-RF	50.0	50.0	0.256	0.6009
50-RF	49.8	49.7	-5.986	–
100-RF	49.8	49.6	-7.628	–

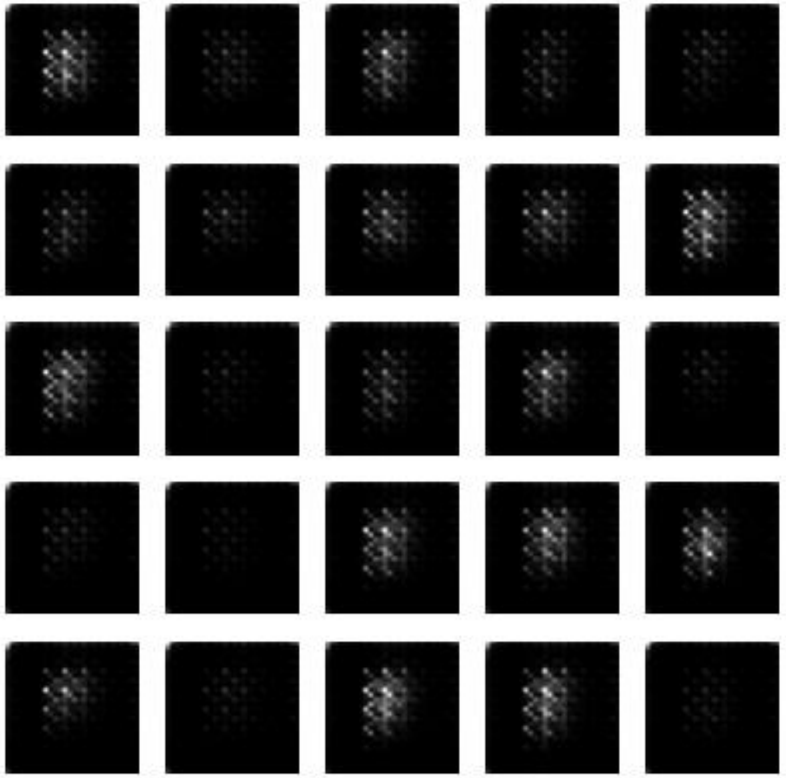
*Source: **Stephen Roberts et de Sid Ghoshal** (University of Oxford): « Thresholded ConvNet Ensembles: Neural Networks for Technical Forecasting »*

Echantillons asymétriques et apprentissage autonome

Exemple d'application de modèles génératifs antagonistes à la défaillance d'entreprises

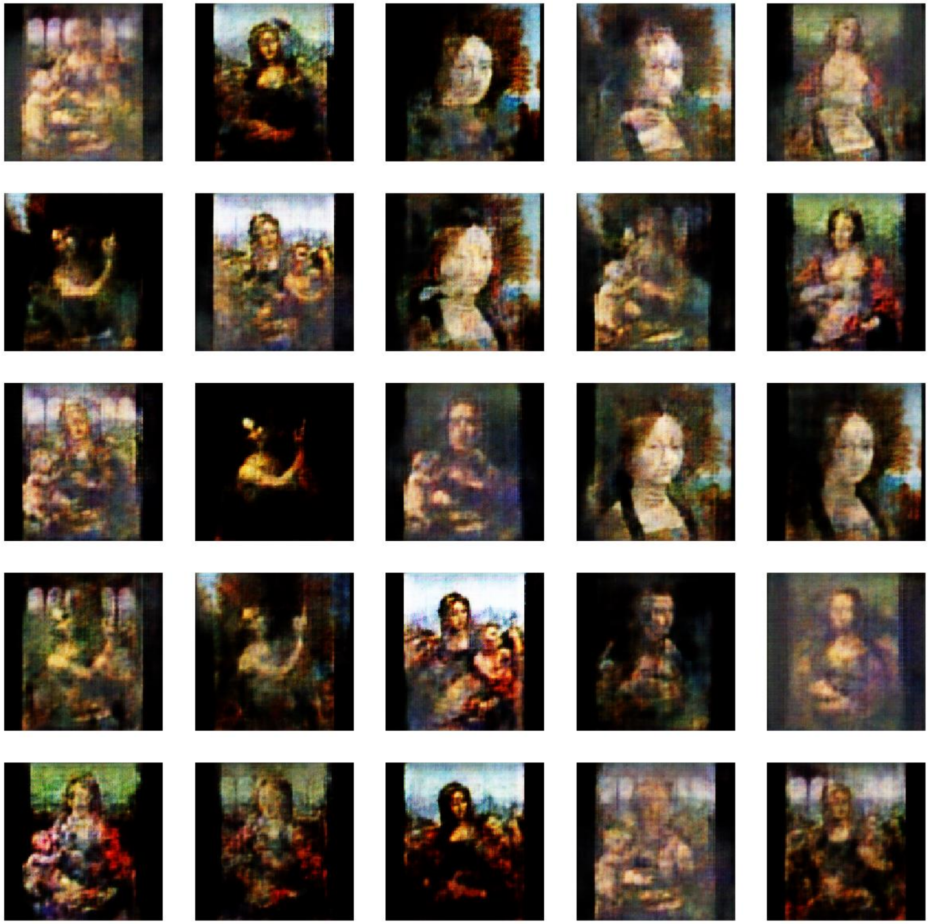


Exemple de Modèle Génératifs Antagoniste (GAN)



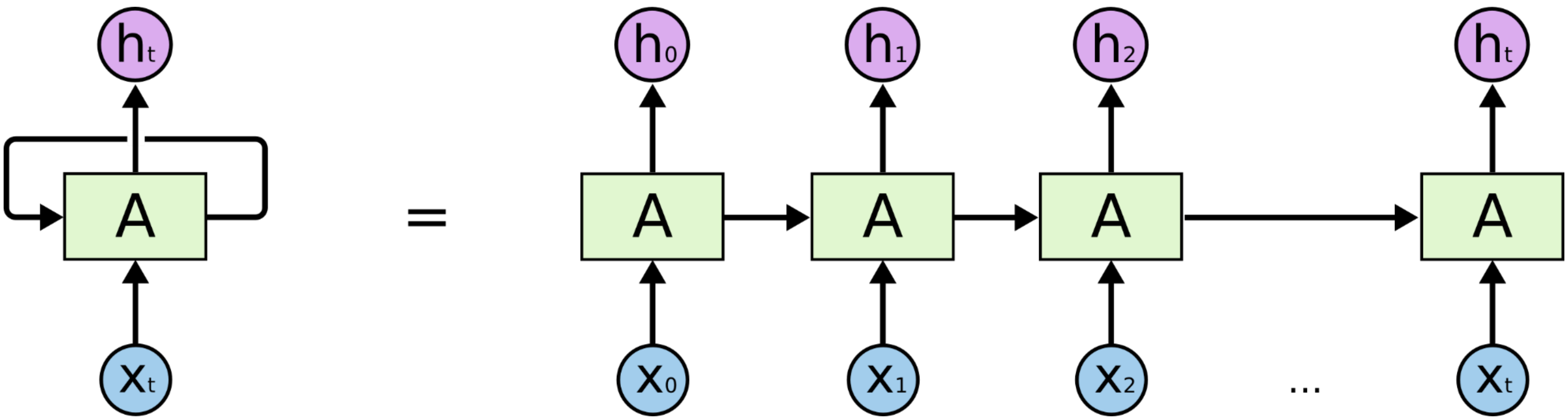
Exemple de Modèle Génératifs Antagoniste (GAN)

Exemple de Modèle Génératifs Antagoniste (GAN)



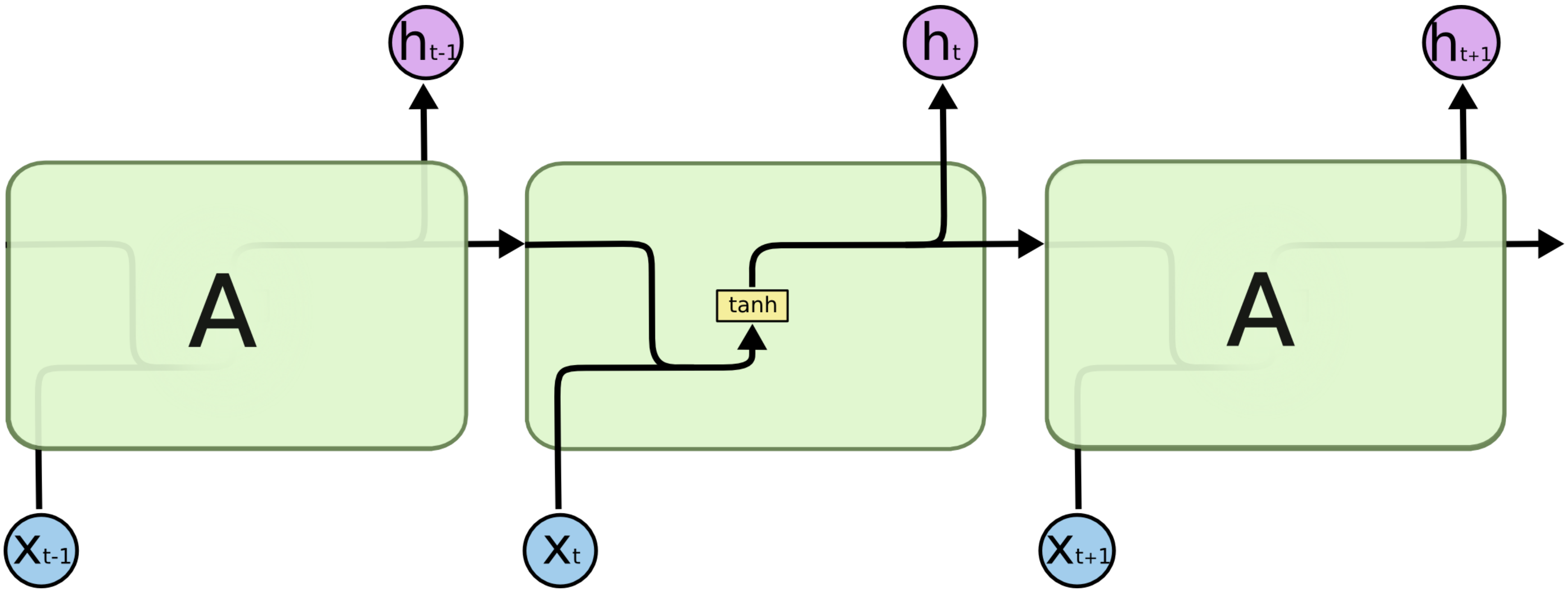
Introduction aux LSTM

Réseaux Récurrents



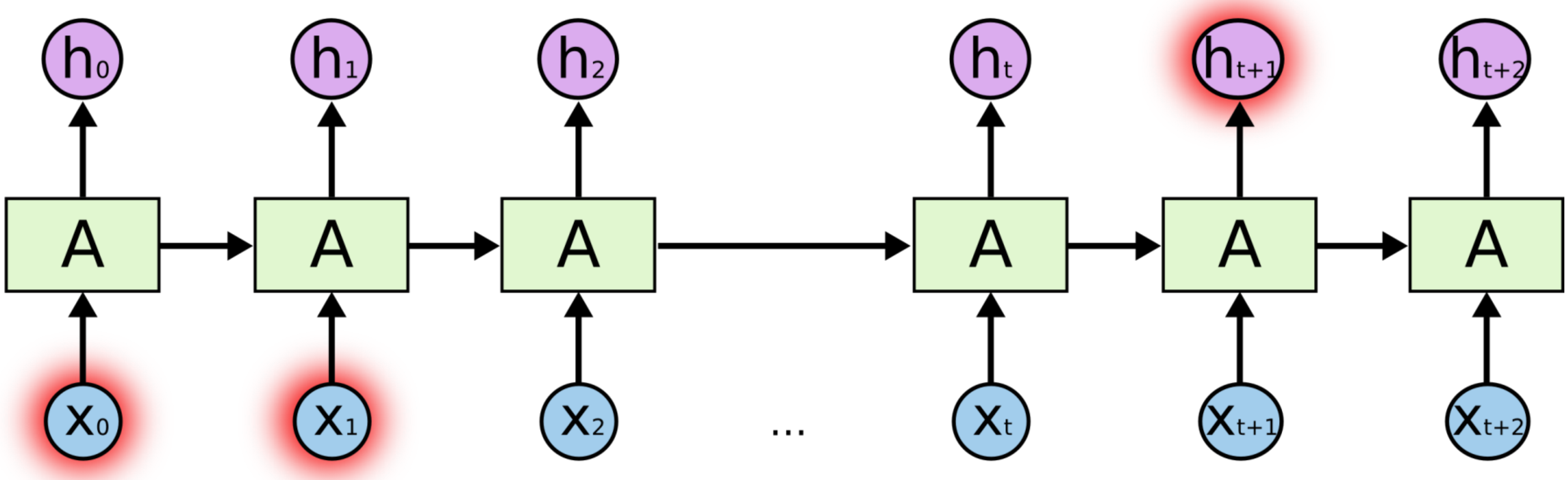
Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

La structure du RNN



Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Le « long term dependency problem »

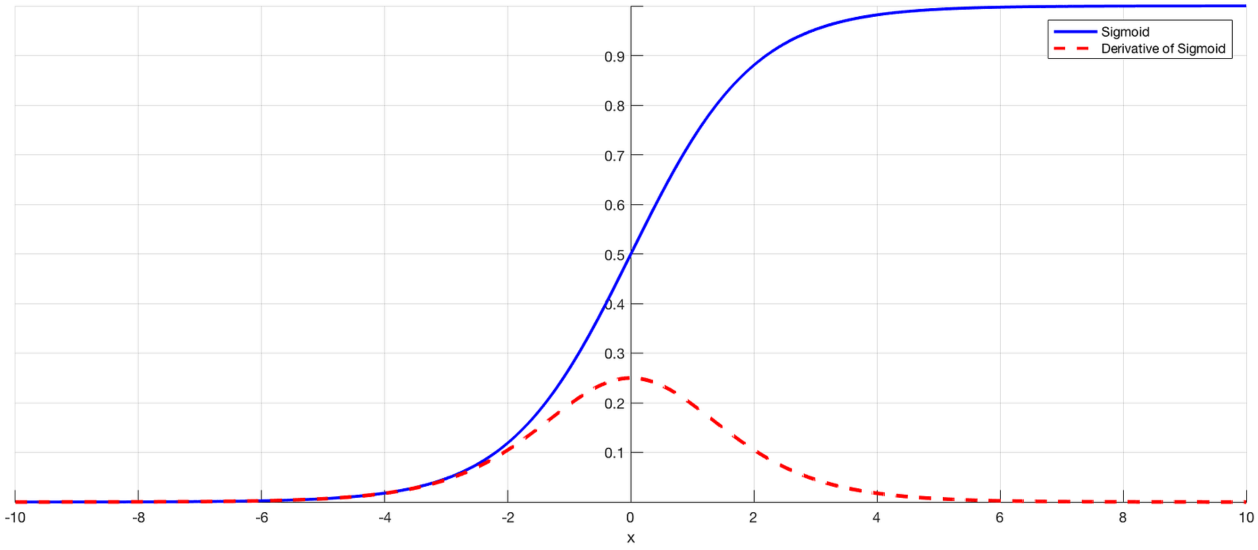


Sources:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Hochreiter (1991), Bengio & al. (1994)

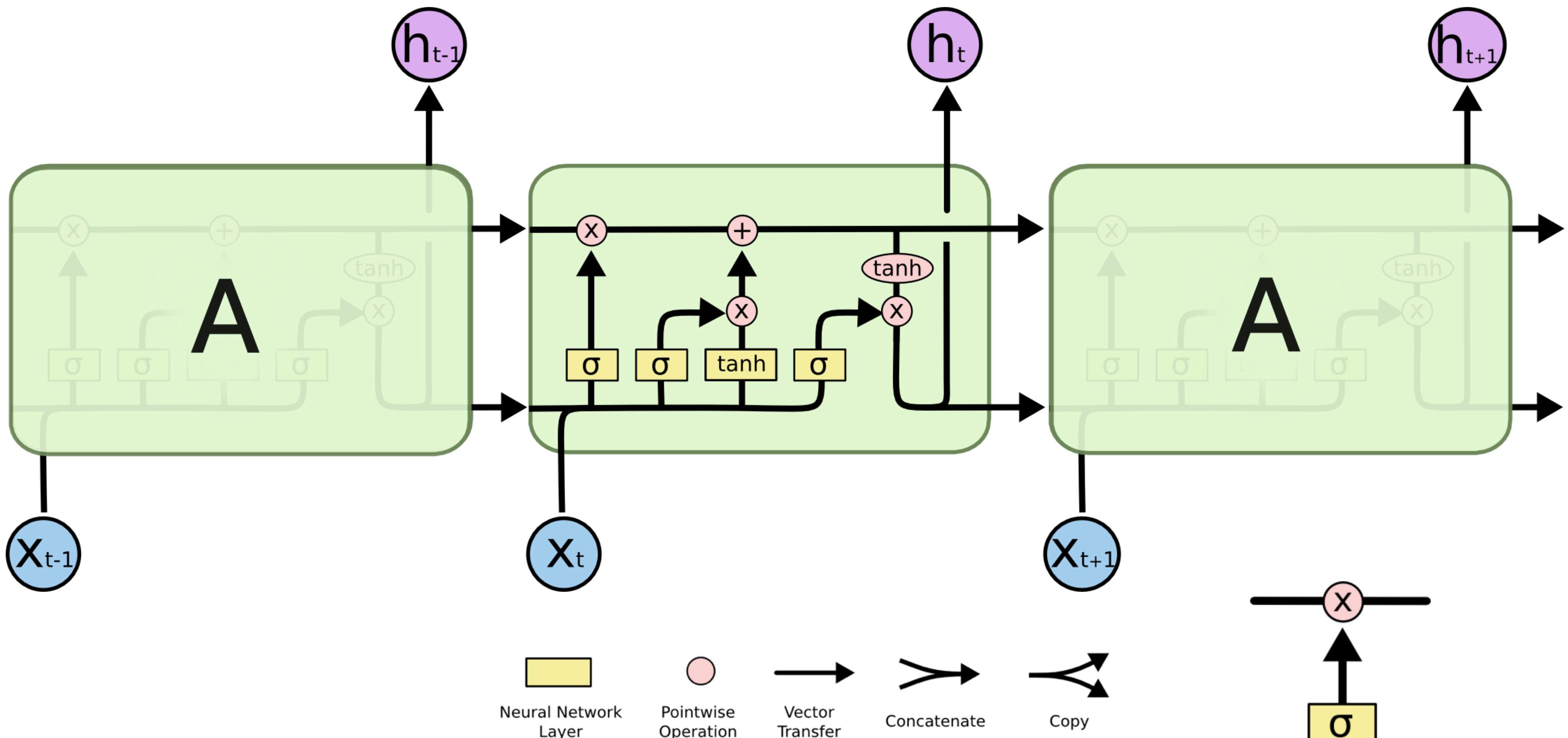
Problème du Vanishing Gradient

- Lorsque n couches cachées utilisent une fonction d'activation sigmoïde, n petites dérivées sont multipliées ensemble. En conséquence, le gradient diminue exponentiellement lorsque nous le propageons vers les couches initiales, ce qui rend le réseau de neurones difficile à entrainer
- Une fonction d'activation de type ReLU (max(0,k)) peut accélérer la convergence jusqu'à un facteur de 5 ou 6, d'où leur popularité ces dernières années.



La structure du LSTM

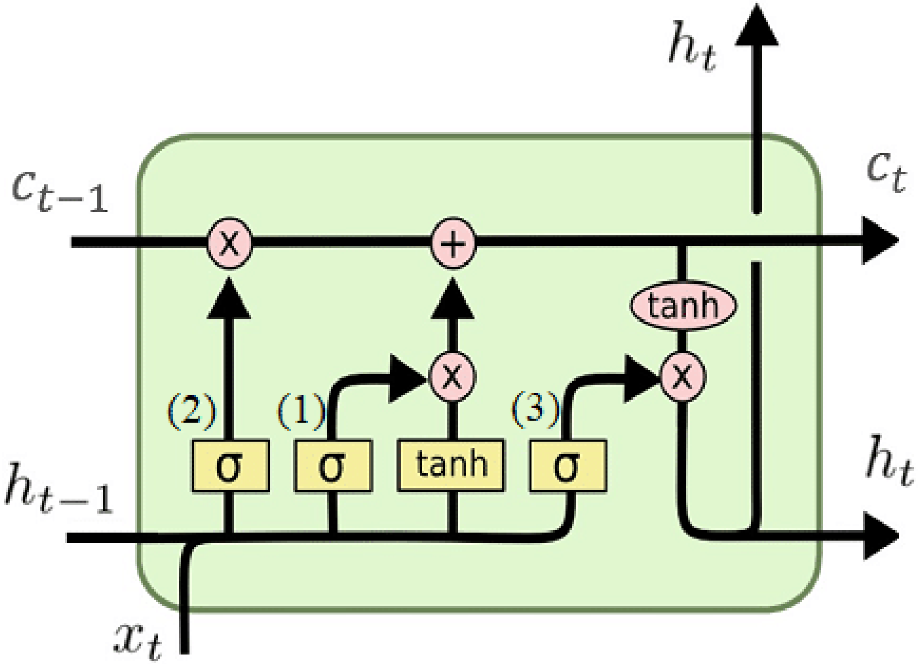
Hochreiter & Schmidhuber (1997)



Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

La structure du LSTM

- **C(t) est le « cell state »** ou la mémoire à long terme du RNN en t.
- **H(t) le hidden-state** ou mémoire courte du RNN en t.
- **(2) la porte d'oubli (forget gate layer)** permet de sélectionner l'information qui sera oubliée de celle qui sera conservée dans le « cell state »
- **(1) la « porte d'entrée »** permet de sélectionner l'information qui sera stockée dans le « cell state » en combinant la « input gate layer » (sigmoid de sélection) à un tanh (l'information elle-même).
- (3) est la sortie du RNN, et permet d'obtenir un nouveau « hidden state », en combinant le nouvel « cell state »



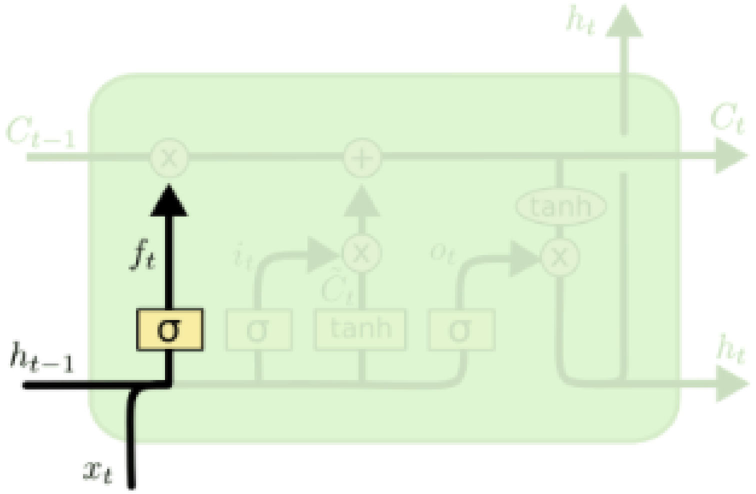
LSTM
(Long-Short Term Memory)

Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Zoom sur la porte d'oubli

Quelle information supprime-t-on du « cell state » ?

Une sigmoïde nous permet de faire la sélection



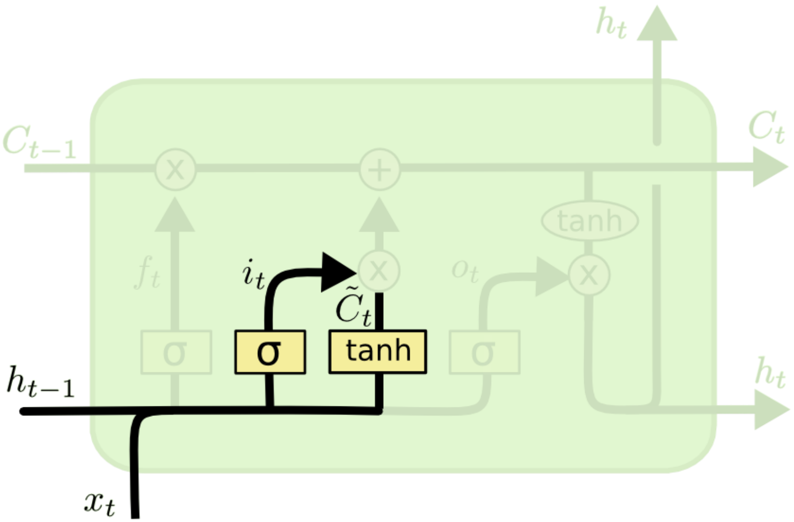
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Zoom sur la porte d'entrée

Quelle information ajoutons-nous au « cell state » ?

« i » détermine les valeurs à ajuster et une « C » via une tanh les valeurs

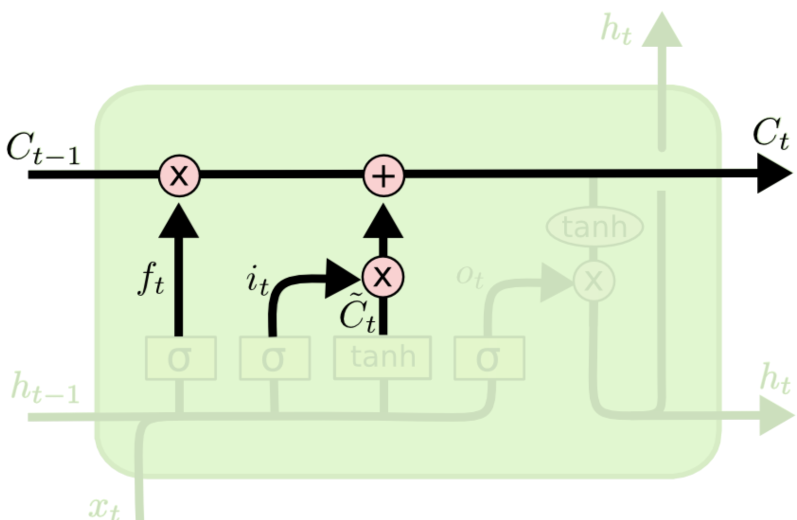


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oubli + Entrée



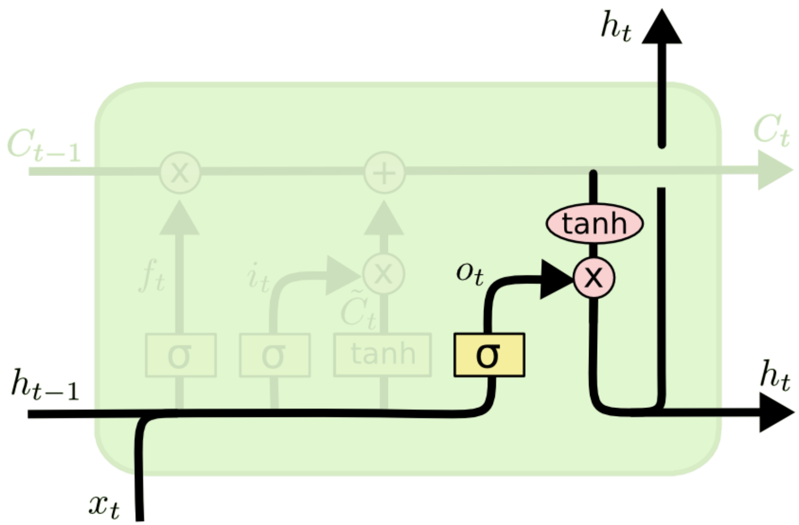
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Zoom sur l'output

Quelle information avons-nous en sortie ?

Une version filtrée du « cell state » via une sigmoid

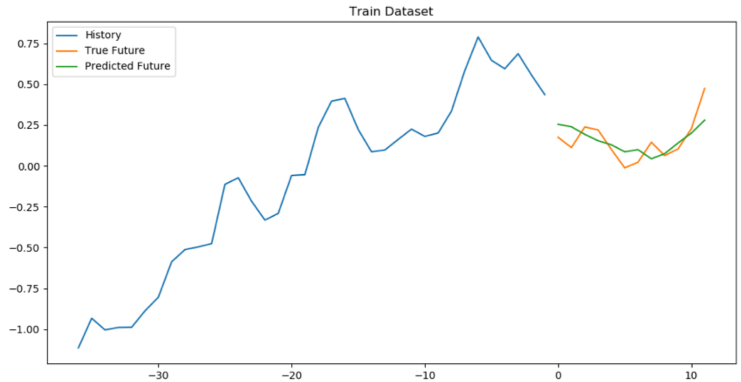


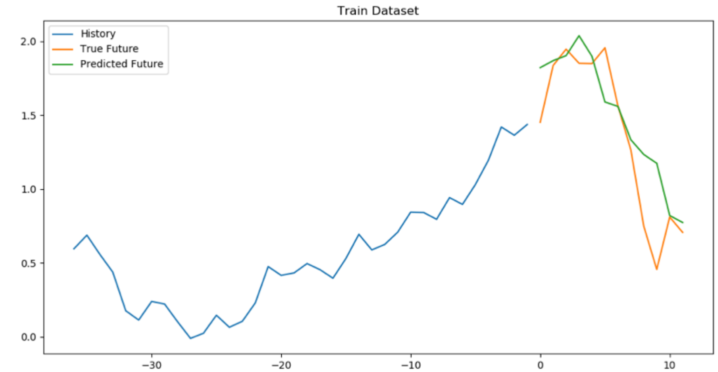
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

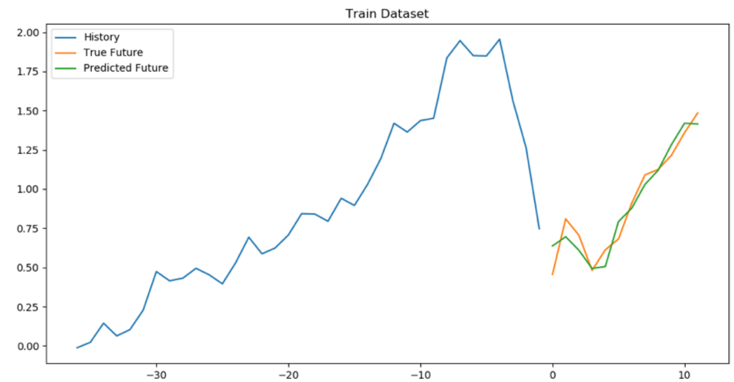
$$h_t = o_t * \tanh\left(C_t\right)$$

Source:
Understanding LSTM Networks. Aug. 2015
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM applied to EUR/USD







Exemple d'un LSTM avec TensorFlow

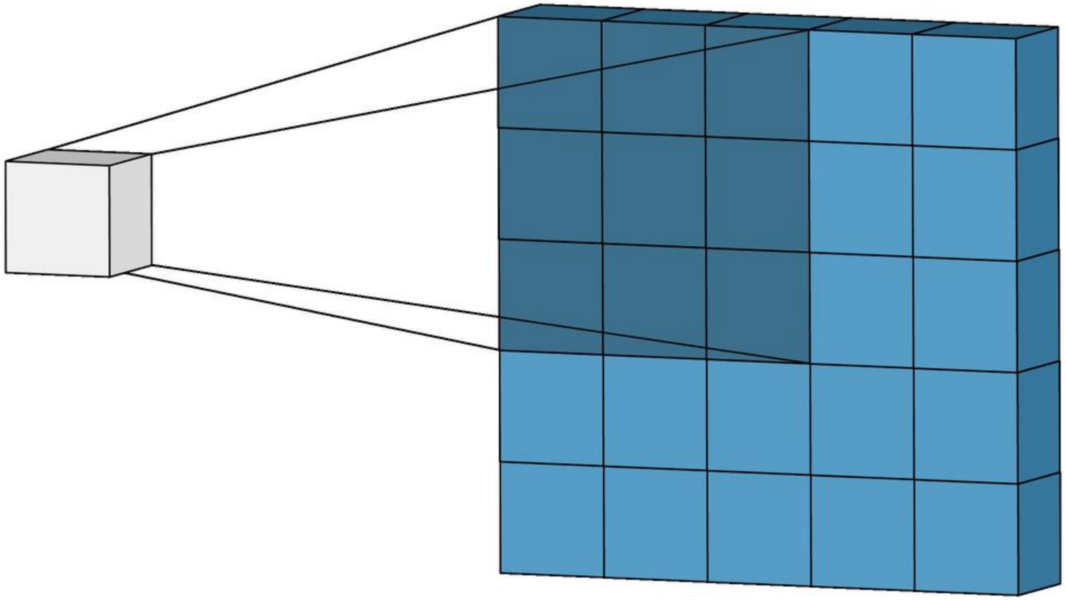
```python
import tensorflow as tf

# Modèle LSTM sur données journalière, prenant un historique de 20 jours
model = tf.keras.Sequential([
    tf.keras.layers.LSTM(64, input_shape=(20, 1)),
    tf.keras.layers.Dense(1)
])

# Compilation du modèle
model.compile(loss='mse', optimizer='adam', metrics=['accuracy'])

# Prepare data
X_train = ... # input data with shape (batch_size, 20, 1)
y_train = ... # target data with shape (batch_size, 1)

X_test = ... # test data with shape (batch_size, 20, 1)
y_test = ... # target test data with shape (batch_size, 1)


# train du modèle sur 20 epochs sans validation
model.fit(X_train, y_train, epochs=20)

# train du modèle sur 20 epochs avec validation
model.fit(X_train, y_train, epochs=20, validation_data=(X_test, y_test))
```

Exercice

- En démarrant avec l'exemple fourni sur github (https://github.com/sylbarth/series-temporelles), essayer de programmer un premier LSTM permettant de prévoir une fonction cosinus.
- Question 1 : les prévisions sont-elles fiables ?
- Question 2 : essayer de prévoir sur plus de 1-step ?
- Question 3 : comment se modèle se compare à un ARIMA ?
- Question 4 : estimer un CART et un Random Forest sur la même cible, simplement avec les « variables contemporaines ».
- Question 5 : ajouter des retards au Random Forest et comparer les performances.
- Question 6: ajouter du bruit dans vos données, et comparez les performances.

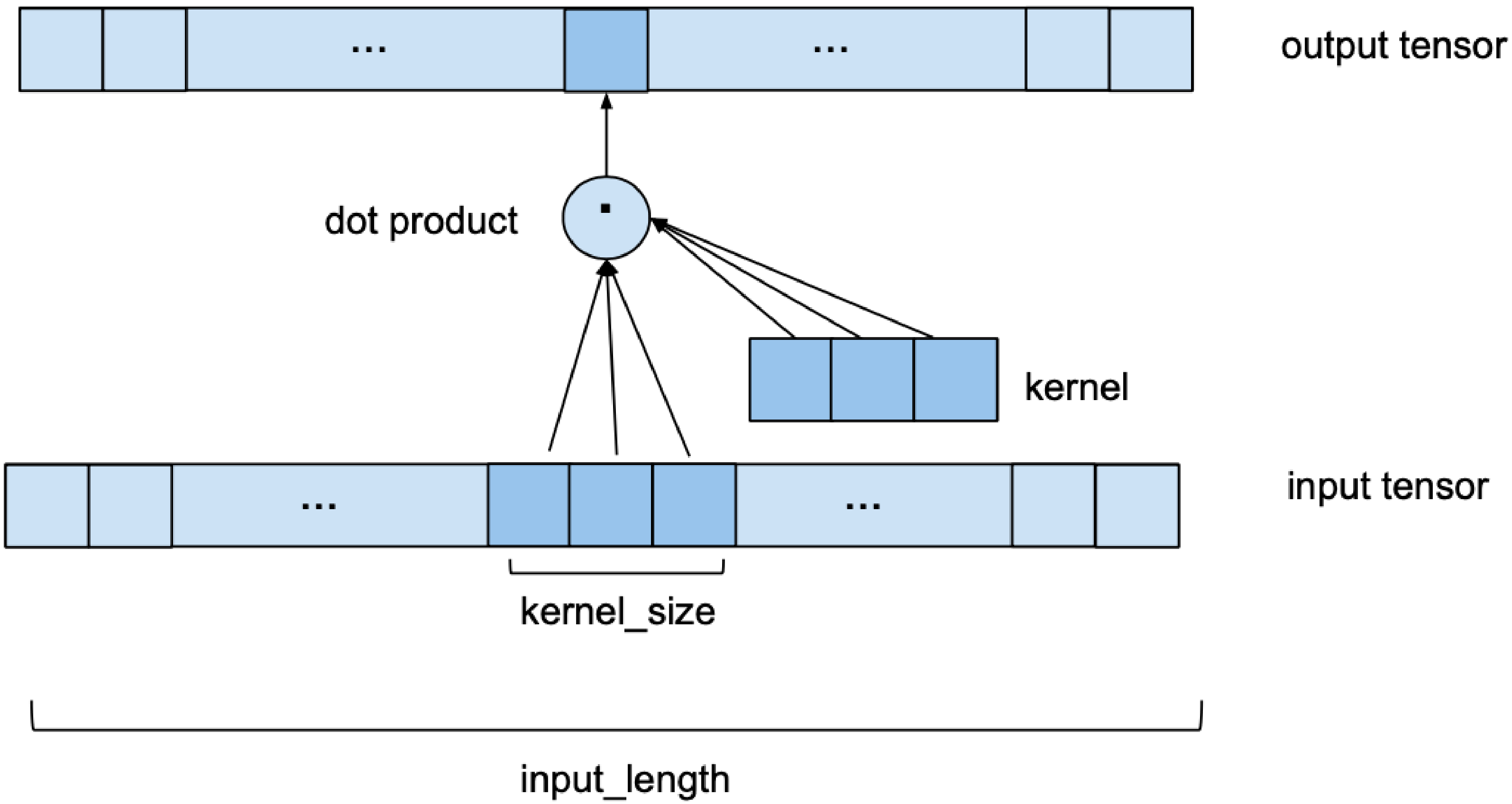
Introduction aux TCNN

Convolution



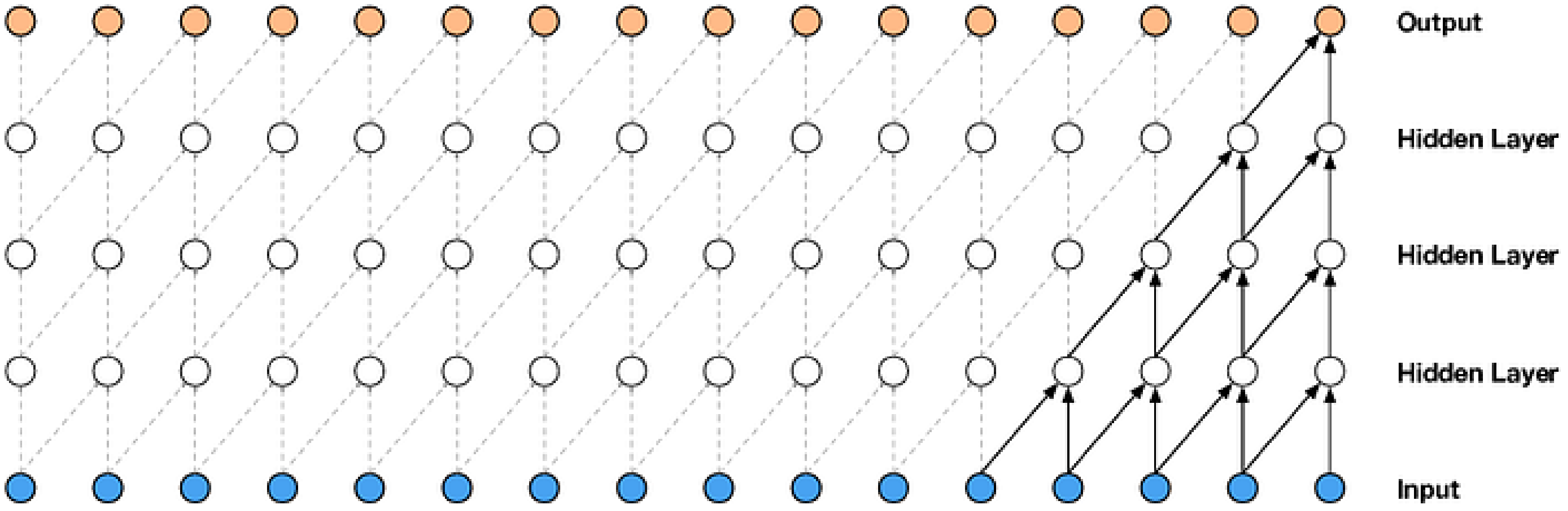
Source:
https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

1D Convolutional Network

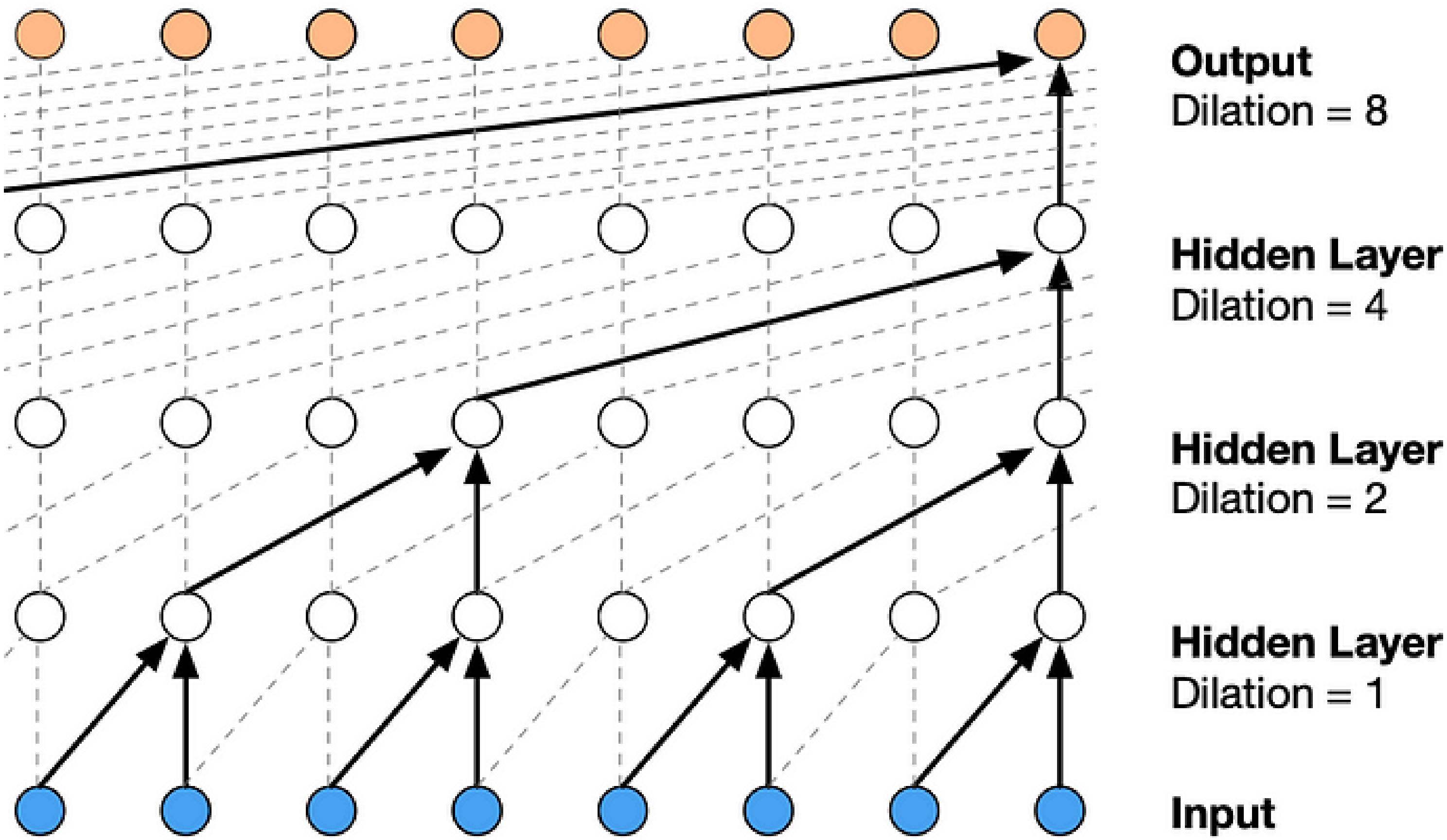


Source:
https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/

Approche convolutionnelle traditionnelle



Approche avec « dilatation »



Formule

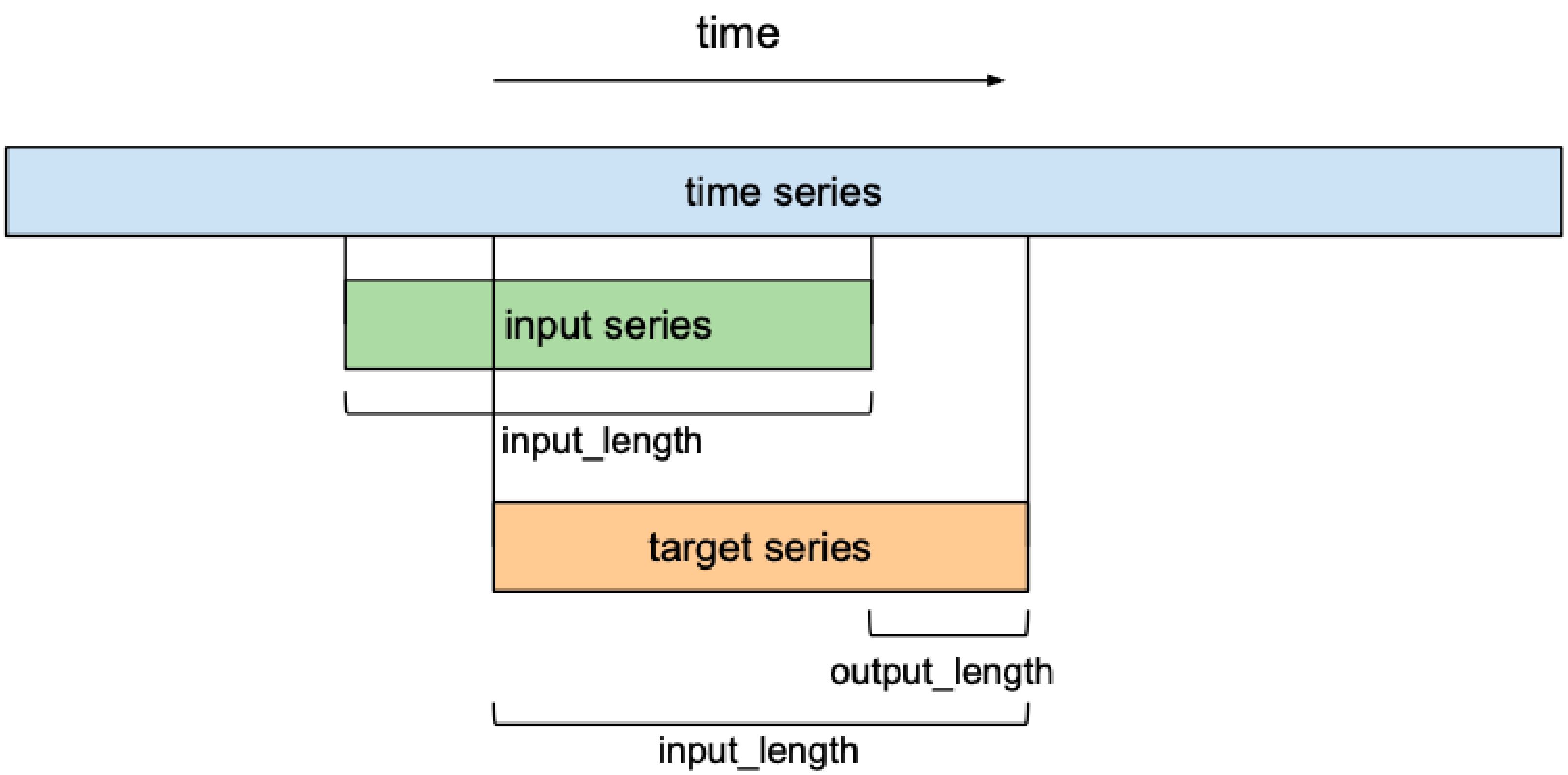
$$x_l^t = g\left(\sum_{k=0}^{K-1} w_{lk} \cdot x_{l-1}^{t-(k\times d)} + b_l\right)$$

Where:

- *xt_l* is the output of the neuron at position *t* in the *l-th* layer.
- *K* is the kernel's with, determining the number of past time windows considered.
- *wlk* is the weight for the *k-th* position in the kernel used to give importance to past data.
- *d* is the dilation factor, or the space between inputs allowing the network to integrate historical information.
- *bl* is the bias term.
- *g* is **ReLU** defined as *g(x) =max(0,x)*.

https://medium.com/call-for-atlas/temporal-convolutional-neural-network-with-conditioning-for-broad-market-signals-9f0b0426b2b9

Forecasting



Source:
https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/

Synthèse

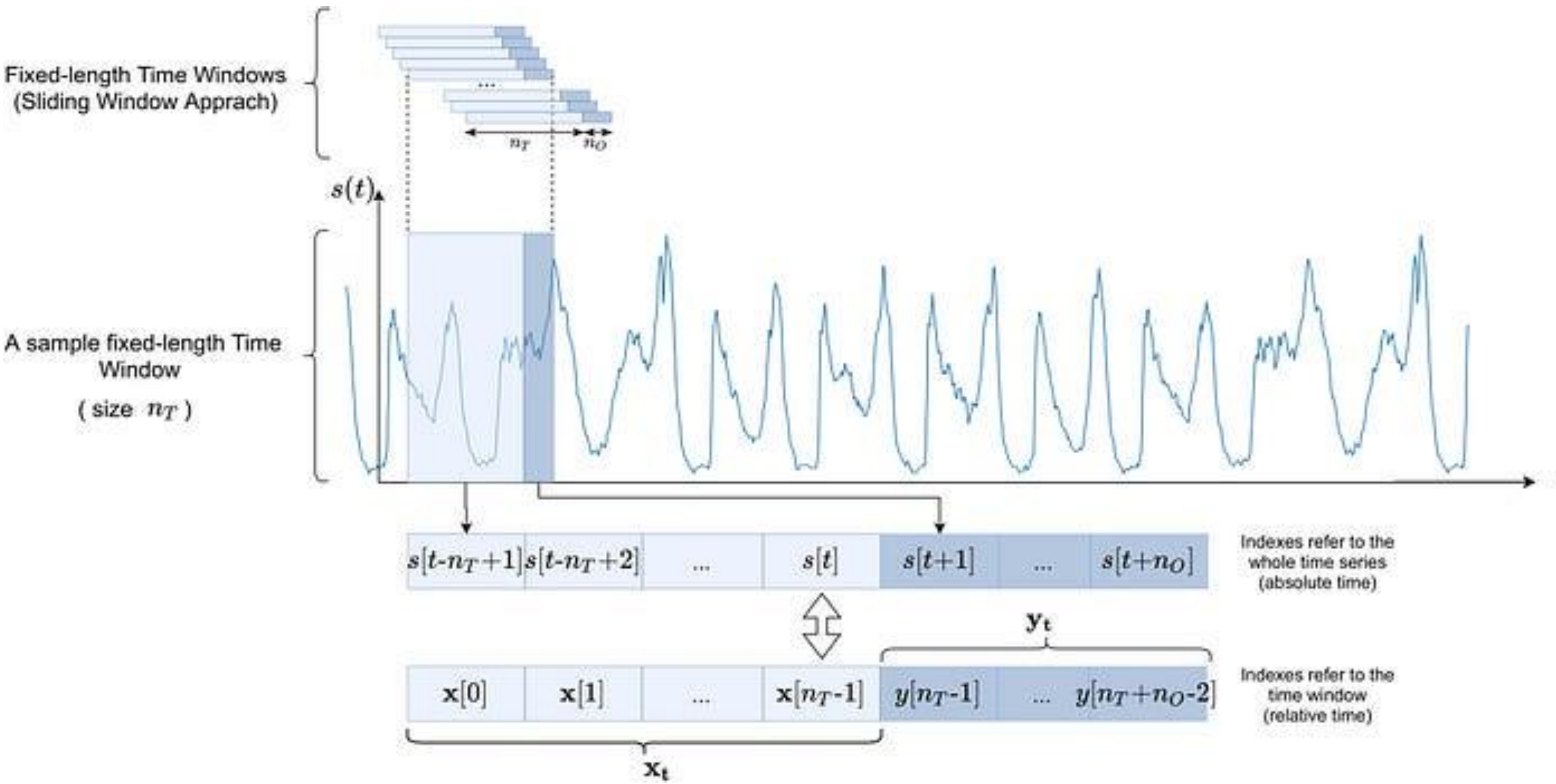
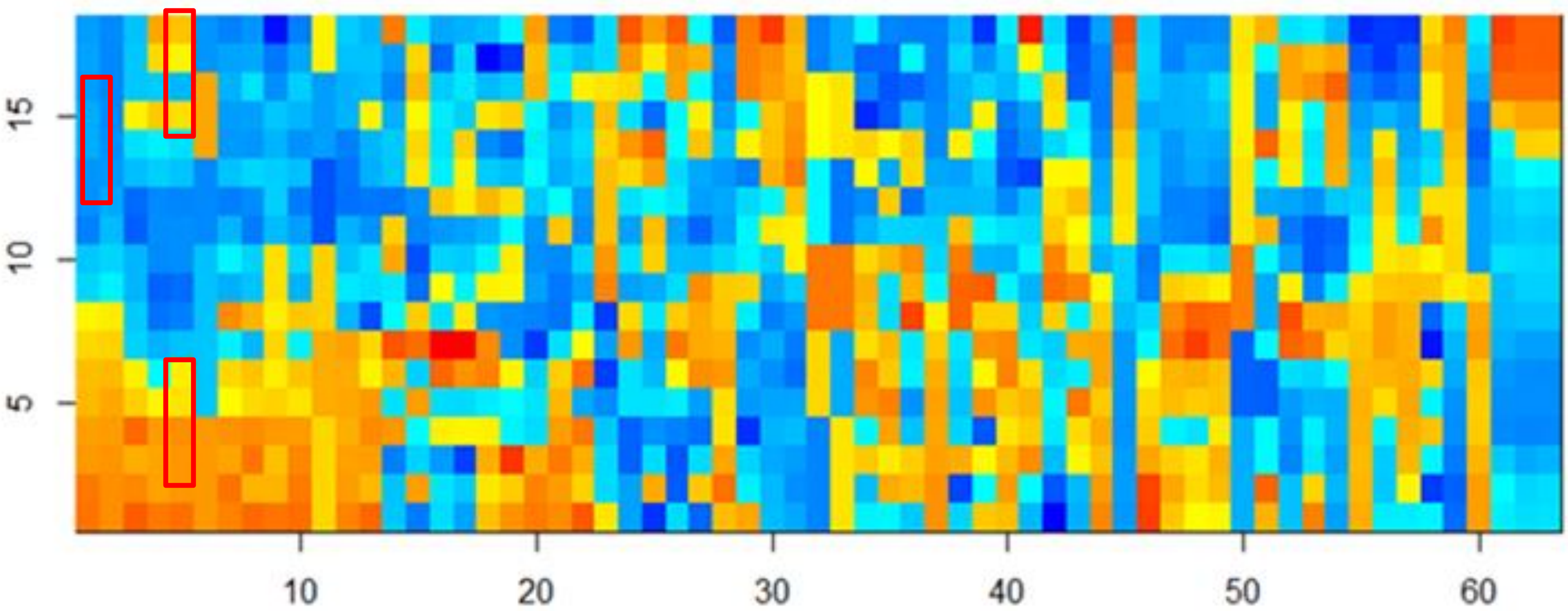


Image from *Gasparin et al.*

Logique 1D



Technical analysis and SP500

Pattern	γ	$\mu(bp)$	$\sigma(bp)$
Unconditional		4.26	229.40
Hammer	5.13	-15.80	223.04
Inverted Hammer	5.00	+13.75	211.62
Hanging Man	3.71	-14.92	222.06
Shooting Star	4.78	+12.01	232.42
Dragonfly Doji	14.73	-25.81	219.99
Gravestone Doji	12.93	+22.41	223.57
Bullish Spinning Top	2.64	-0.72	214.70
Bearish Spinning Top	1.67	+0.94	213.30
Bullish Engulfing	1.61	-0.28	236.50
Bearish Engulfing	4.16	+5.75	238.47
Bullish Harami	1.07	+5.5	222.17
Bearish Harami	1.51	-0.96	219.43
Piercing Line	2.29	+0.78	241.42
Cloud Cover	1.06	-0.75	218.24
Tweezer Bottom	1.76	+4.19	223.23
Tweezer Top	1.22	+2.97	221.93
Abandoned Baby-	3.29	-4.04	232.45
Abandoned Baby+	1.27	+2.94	232.28
Evening Star	2.89	-0.27	231.76
Morning Star	1.80	+2.59	231.89
Three Black Crows	6.85	+13.09	229.40
Three White Soldiers	6.30	-11.77	203.26
Three Inside Down	1.63	+2.72	233.12
Three Inside Up	2.50	+0.13	220.75

*Source: **Stephen Roberts et de Sid Ghoshal** (University of Oxford): « Thresholded ConvNet Ensembles: Neural Networks for Technical Forecasting »*

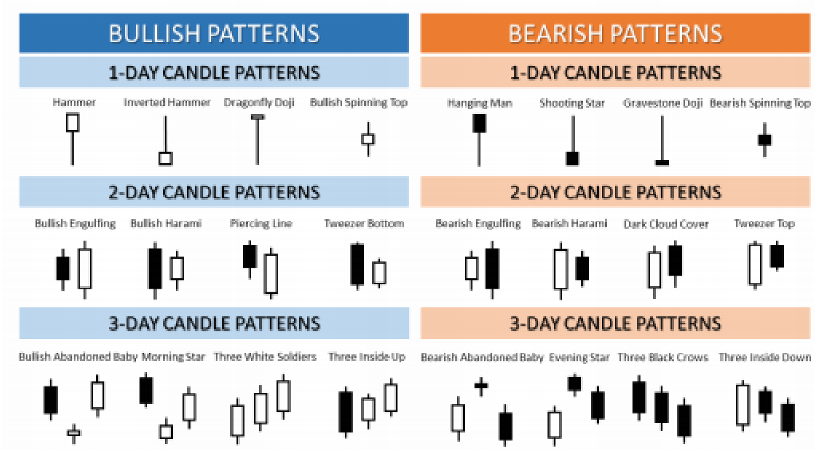


Figure 2: For each timescale (1-day, 2-day and 3-day), we specify 8 chartist patterns and the future direction they predict ('bullish' for positive returns, 'bearish' for negative returns).

Deep learning applied to SP500

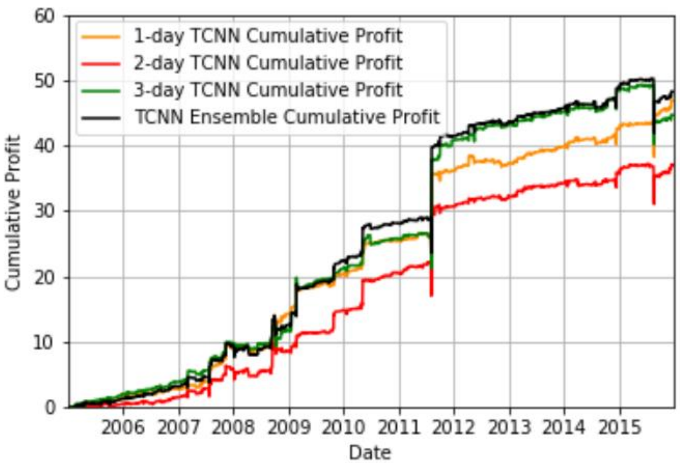


Figure 8: Cumulative profit (as a multiple of starting wealth, per Table 6) generated by the various TCNN models between Jan-2005 and Dec-2015, in the absence of friction costs. The models are steadily profitable, with occasional spikes related to recognising major events. Drawdowns are infrequent and of limited scale.

Table 8: Benchmark performance across a range of supervised learning models trained on S&P500 technical data for Jan 1994 - Dec 1994 and tested on Jan 2005 - Dec 2015.

Model	Accuracy (%)	AUC (%)	Z	Significance
MLP	50.6	51.1	23.766	> 0.9999
Technical NN	49.8	49.9	−1.878	–
1-day CNN	51.3	51.8	36.546	> 0.9999
2-day CNN	51.2	51.5	31.291	> 0.9999
3-day CNN	51.2	51.5	31.423	> 0.9999
CNN Ensemble	51.2	51.7	35.628	> 0.9999
1-day TCNN	56.7	57.2	14.533	> 0.9999
2-day TCNN	56.3	56.5	13.017	> 0.9999
3-day TCNN	55.9	56.2	12.493	> 0.9999
TCNN Ensemble	57.5	57.5	15.301	> 0.9999
RNN-LSTM	50.8	51.0	19.616	> 0.9999
RNN-GRU	50.9	51.2	24.880	> 0.9999
1-NN	50.0	50.1	1.087	0.8614
10-NN	49.9	49.8	-3.317	–
100-NN	49.7	49.6	-7.651	–
Linear SVM	49.9	49.8	-0.962	–
RBF SVM	49.9	49.8	-2.416	–
10-RF	50.0	50.0	0.256	0.6009
50-RF	49.8	49.7	-5.986	–
100-RF	49.8	49.6	-7.628	–

*Source: **Stephen Roberts et de Sid Ghoshal** (University of Oxford): « Thresholded ConvNet Ensembles: Neural Networks for Technical Forecasting »*

Biblio

- **TCNN: a Tensor Convolutional Neuro-Network for big data anomaly detection**
  https://berthuang.com/courses/opt18/projects/tcnn.pdf

- **How a convolutional network with some simple adaptations can become a powerful tool for sequence modeling and forecasting.**
  https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/

- **An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling**
  https://arxiv.org/pdf/1803.01271.pdf

Exemples d’applications

- **Keras TCN**
  https://github.com/philipperemy/keras-tcn

- **Temporal Convolutional Network using Keras-TCN**
  https://www.kaggle.com/code/carlmcbrideellis/temporal-convolutional-network-using-keras-tcn

- **Tensorflow TCN**
  https://github.com/Baichenjia/Tensorflow-TCN

  Explications sur: Implementing Temporal Convolutional Networks
  https://medium.com/the-artificial-impostor/notes-understanding-tensorflow-part-3-7f6633fcc7c7

- **Exemple en finance**
  https://medium.com/call-for-atlas/temporal-convolutional-neural-network-with-conditioning-for-broad-market-signals-9f0b0426b2b9

Mise en Pratique

Exercice 1

- **Télécharger le Weather CSV File:**
  https://corgis-edu.github.io/corgis/csv/weather/

- **Prévisions précipitation (RF/LSTM/TCN)**
  Réaliser une prévision à une semaine avec RF et LSTM.
  Tenter de prévoir à une puis deux semaines.
  Utiliser le code github pour ajoute un modèle TCN

- **Prévisions températures (RF/LSTM/TCN)**
  Réaliser une prévision à une semaine.
  Tenter de prévoir à une puis deux semaines.

- **Prévisions sur toutes les stations:**
  Estimer les modèles sur toutes les stations.
  Comparer les performances.

Exercice 2

- **Data**
  Travail au choix sur EUR/USD ou BTC

- **Modélisation**
  Réaliser une prévision à un mois puis trois mois avec plusieurs modèles.
  Comparer les performances

- **Améliorations possibles**
  Proposer et tester des idées pour améliorer les performances du modèle.

Sylvain BARTHELEMY

sylvain.barthelemy@gwenlake.com

www.gwenlake.com