The state of AI in 2024

What kind of job in AI with a PhD in maths?

Francois Friggit, CIMPA, June 26th, 2024

François Friggit : a quick CV

- **Current position** : Enjoying life since March 2023

- **Experience** : 30 years as “quant” (quantitative analyst) in Paris (2 years), New York (3 years), London (7 years) et Madrid (18 years), including last 5 year on Artificial Intelligence
  Before that, 5 years as engineer specialized in rural development in Burkina Faso, West Africa

- **Studies** : Engineer at Ecole Polytechnique, France, 1985
  Engineer in hydraulics at ENGREF, France, 1987
  Master in International Finance at HEC, France, 1993

The opinions expressed in this presentation only engage the author, and absolutly not any entity with which he is or has been linked, in any possible way

The state of AI in 2024

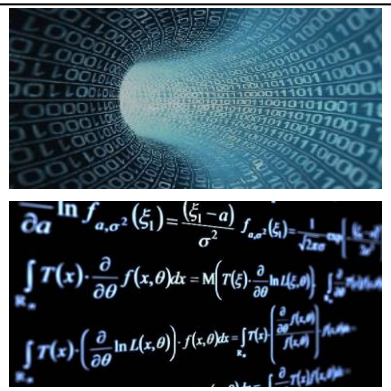
- Definition and history
- AI practical cases for financial markets
- Who is doing what in the AI industry?
- Some advice from an old geezer

What is Artificial Intelligence?

A modern way to analyze data and solve problems

Combining statistics and applied mathematics

- … trying to replicate/improve some human cognitive processes
- One particular branch : Machine Learning. Learn patterns from data.
- Deep Learning = particular case of Machine Learning using Neural Networks technique



Typical problems to solve

- “Classification”: customers segmentation, products classification, medical diagnosis, face recognition, fraud detection, …

- “Prediction”: weather, molecule properties, price adjustments, recommendation engines, automatic translation, question answering, trucks fleet management, …

Historical perspective

for decades | 2000s | Since 2012 | < 2025?

Classic statistics
Digital Analysis 1.0

- Widely used
- Simple, universal concepts
- **Mean, variance, standard deviation, correlation**, PCA (Principal Components Analysis)
- Monte Carlo simulation
- No significant evolution during the past decades

Machine Learning
Digital Analysis 2.0

- **Collection of recipes** or simple models …
- .. to solve a limited collection of **very common problems**
- Kmeans, Random Forest, Gradient Boosting, collaborative filtering, …
- Main idea : human designer choses features combinations …
- … then simple algorithm optimizes the solution
- No more new idea since the 2010s

Deep Learning
Digital Analysis 3.0

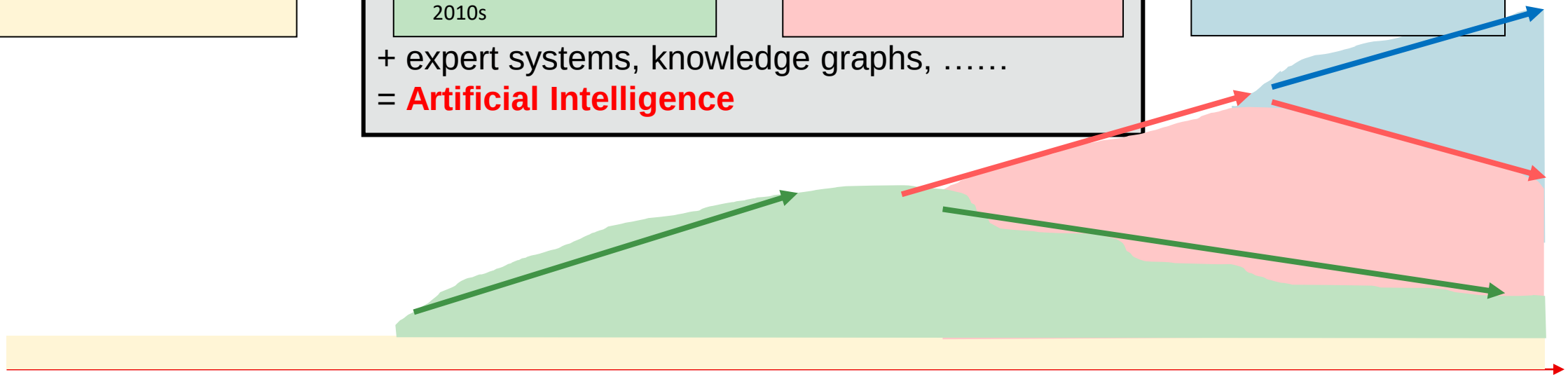
- **Neural Networks**: the algorithm selects optimal combinations of feature combinations
- Convolutional Neural Network, TSTM, BERT, ChatGPT …
- For most ML recipes, a DL solution does it much better.
- Usually requires more data, but not always
- Currently booming, with significant advances each quarter

+ expert systems, knowledge graphs, ……

= **Artificial Intelligence**

???
Digital Analysis 4.0

- Beyond linear combination?
- Beyond activation functions (ReLU, tanh, sigmoid)?
- Beyond gradient descent?
- **Radical new idea?**

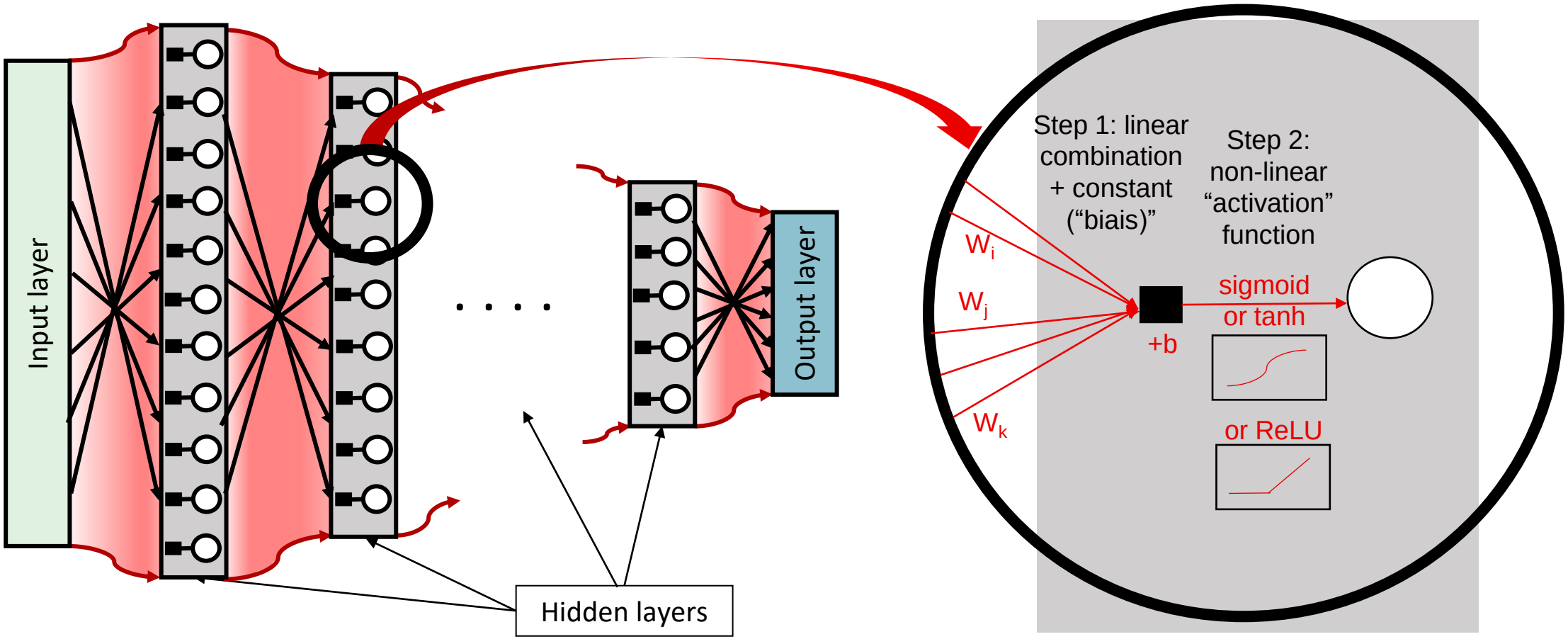


Neural Network (1) - Definition

General structure

- What is it? A Neural Network (NN) is a sequence of layers. Each layer is a set of neurons. Each neuron receives different input signals and transform them into outputs in a specific way: using a function that depends on the neuron nature, using parameters and inputs specific to this neuron.

- What for? To transform input data, layer by layer, into the desired output (i.e. boolean, scalar, vector, …)



Neural Networks (2) - Different types

Most classic types of NN

- Dense layer, or fully connected layer. To use when you have no other idea, or as the last layer in your network
- Convolutional NN. Designed for **image analysis, computer vision**
- RNN – LSTM. Among Recurrent Neural Network, specially designed for **time series**, Long Short term Memory is the most successful one nowadays.

Neural Networks (3) - Inputs: tensors

NN inputs need to be structured

- Examples:
  - IR curves (vectors)
  - volatility matrices, black & white pictures (matrix) 
  - colored pictures(cubes) … 
- General concept: tensor = n-Dimension linear structure of digital data, ideal to **use algebra**
  - 1-Dimension tensor = vector
  - 2-D tensor = matrix
  - 3-D tensor = cube

How to digitalize data?

- Very often, input data are not digital: analogic sound/video signal. Well known digitalization techniques used for decades …
- Special case: words in documents, menus, etc…. Digitalized using Natural Language Processing (NLP) techniques known as **embeddings**

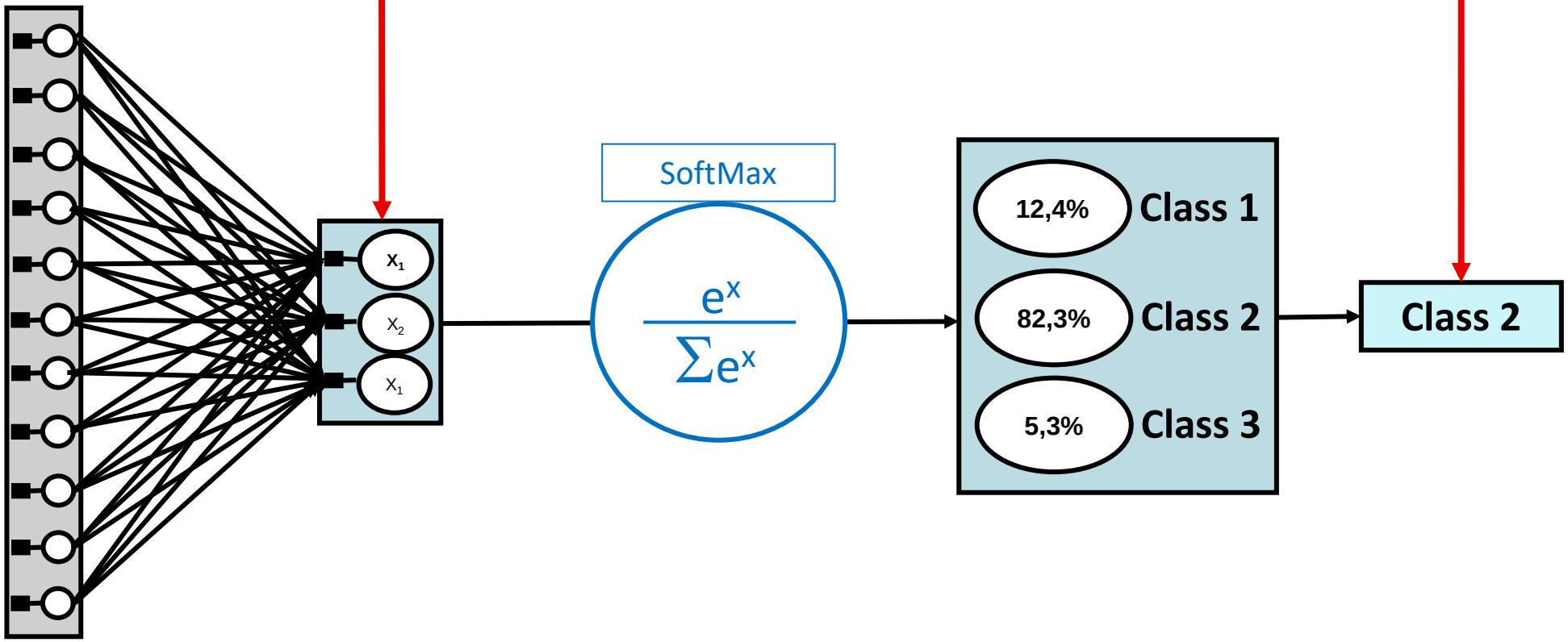
Neural Networks (4) - Output structure depends on purpose

Prediction output

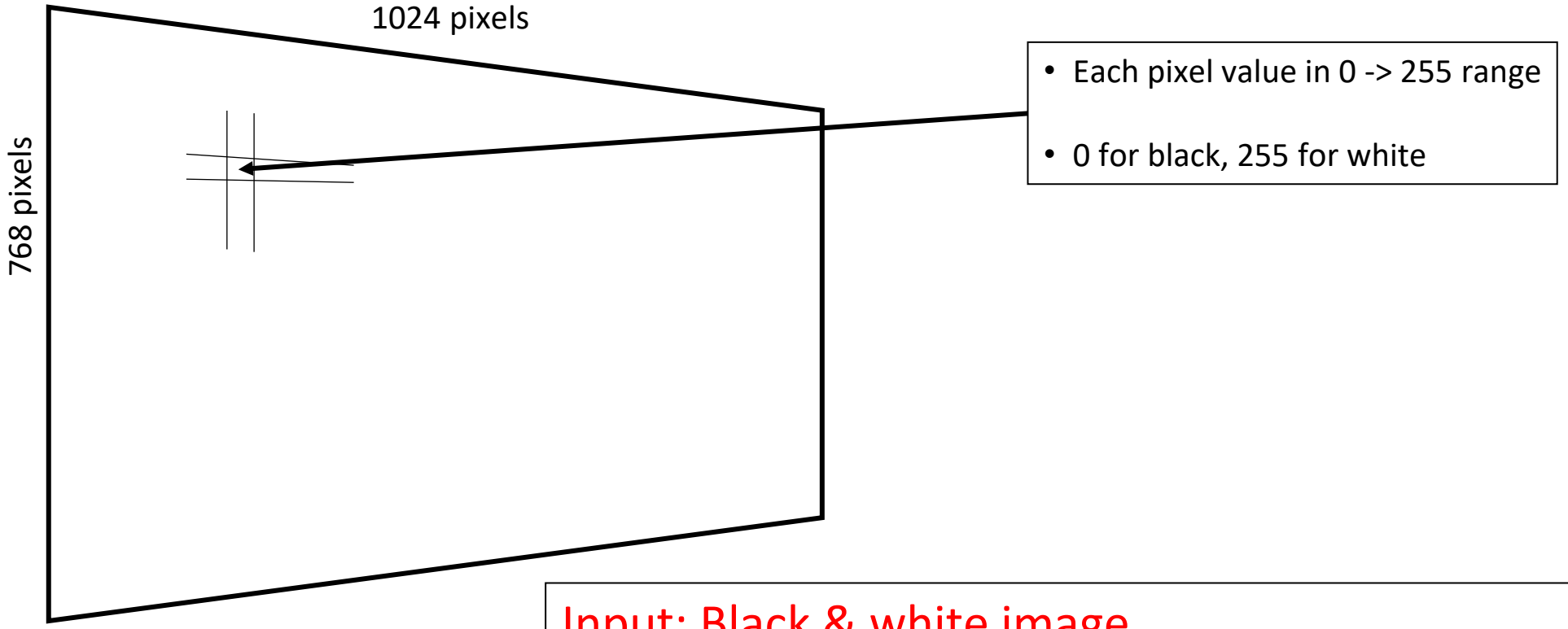
If you want to predict a value, or a vector of values, just create a fully connected layer

Classification output

If you want to predict a class index, apply SoftMax and pick the class index with highest percentage



Convolutional Neural Network (1)

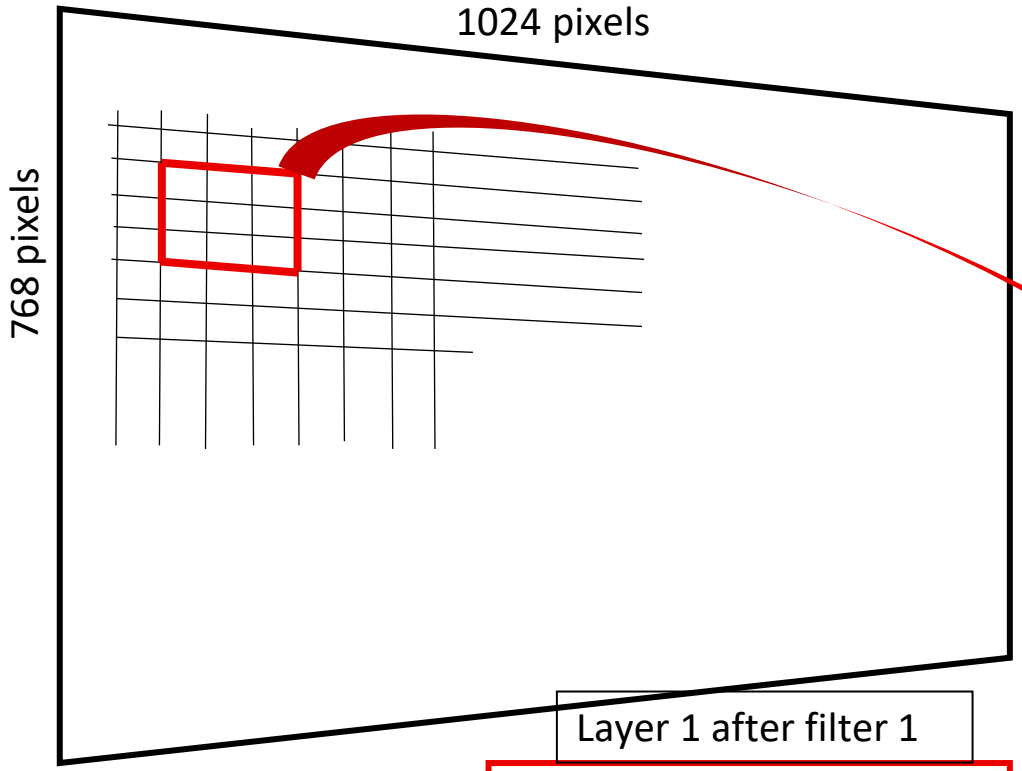


- Each pixel value in 0 -> 255 range
- 0 for black, 255 for white

Input: Black & white image

- Assume 1024 x 768 pixels
- Total number of 786 432 pixels
- With a fully connected first hidden layer with, for example, 20 000 neurons ,
  20 000 x 786 432 = 15 728 640 000 parameters to train!!
- Train with colored 4k video (30 frames/second)?

Convolutional Neural Network (2)



Apply a filter

- Filter: small matrix of coefficient, typically 3x3, 4x5, etc… applied to different parts of picture, gradually shifting left to right and up down.
- All resulting numbers stored in convolutional layer, in a matrix corresponding to this filter

145	148	152
144	147	128
138	121	102

Filter 1

0,0023	0,0061	-0,005
-0,002	0,0137	0,001
0,0061	-0,002	-0,004

X

0,3335	0,9028	-0,760
-0,288	2,0139	0,128
0,8418	-0,242	-0,408

Σ

2.5220

Bias 1

-1,685

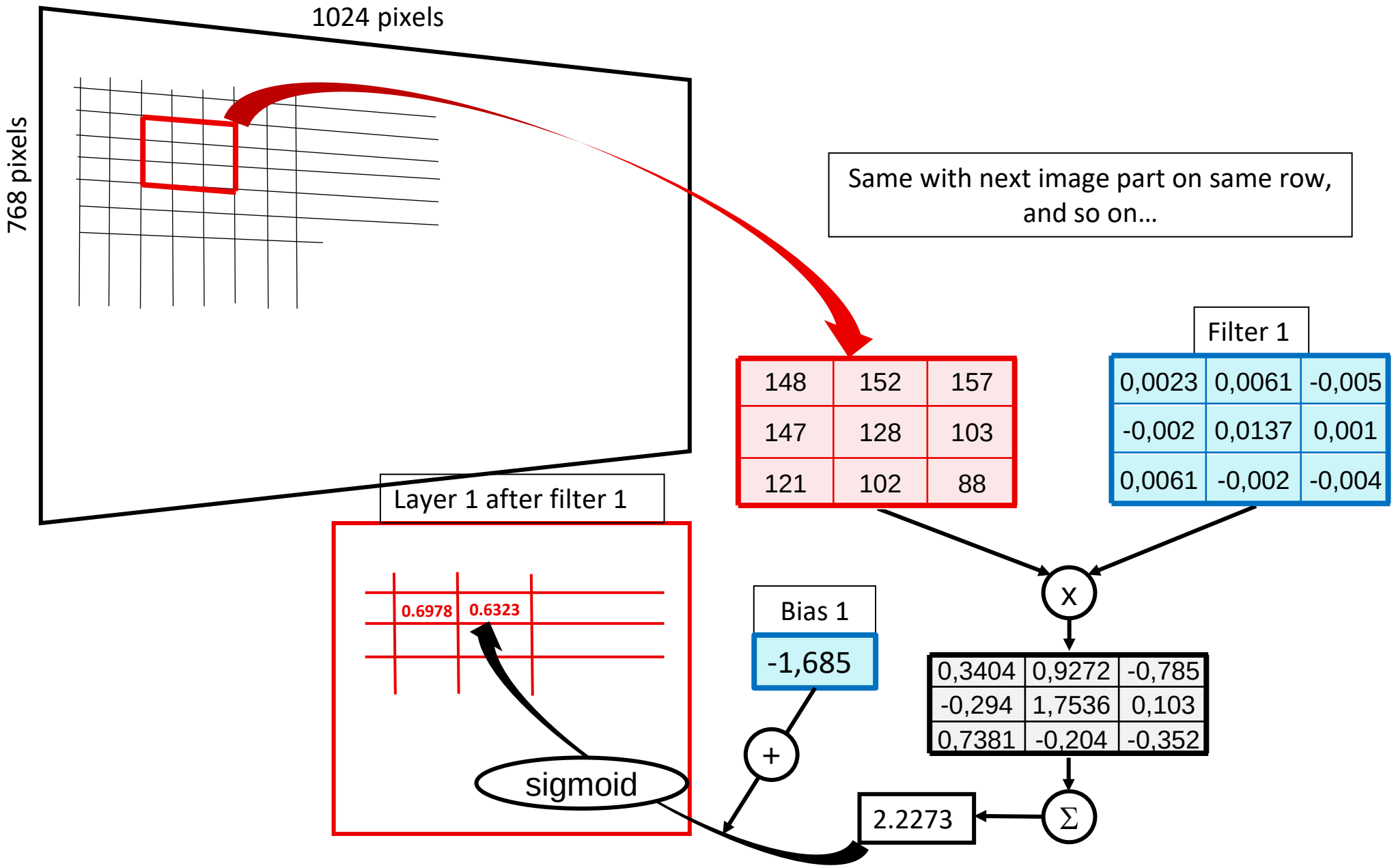
+

sigmoid

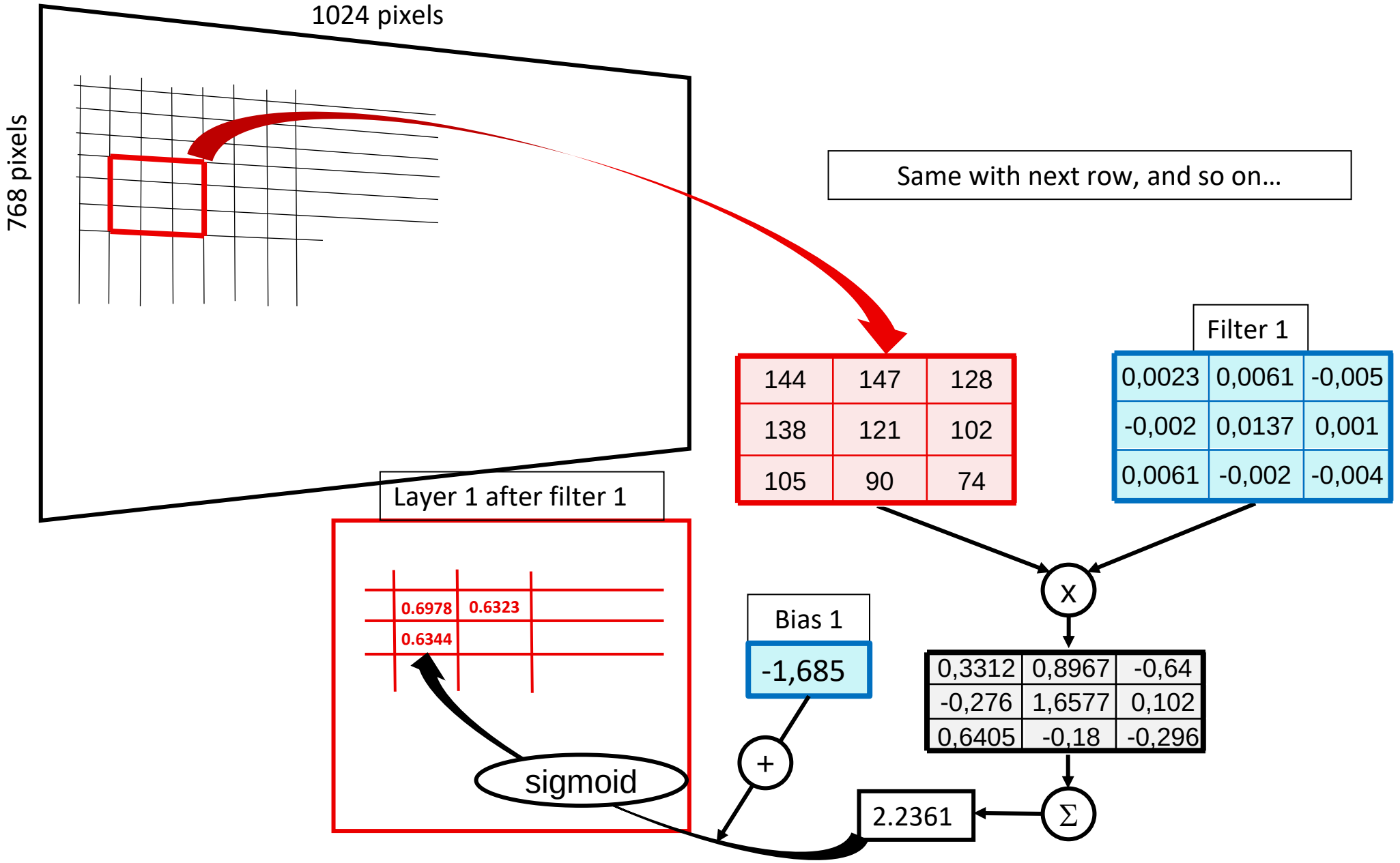
Layer 1 after filter 1

0.6978

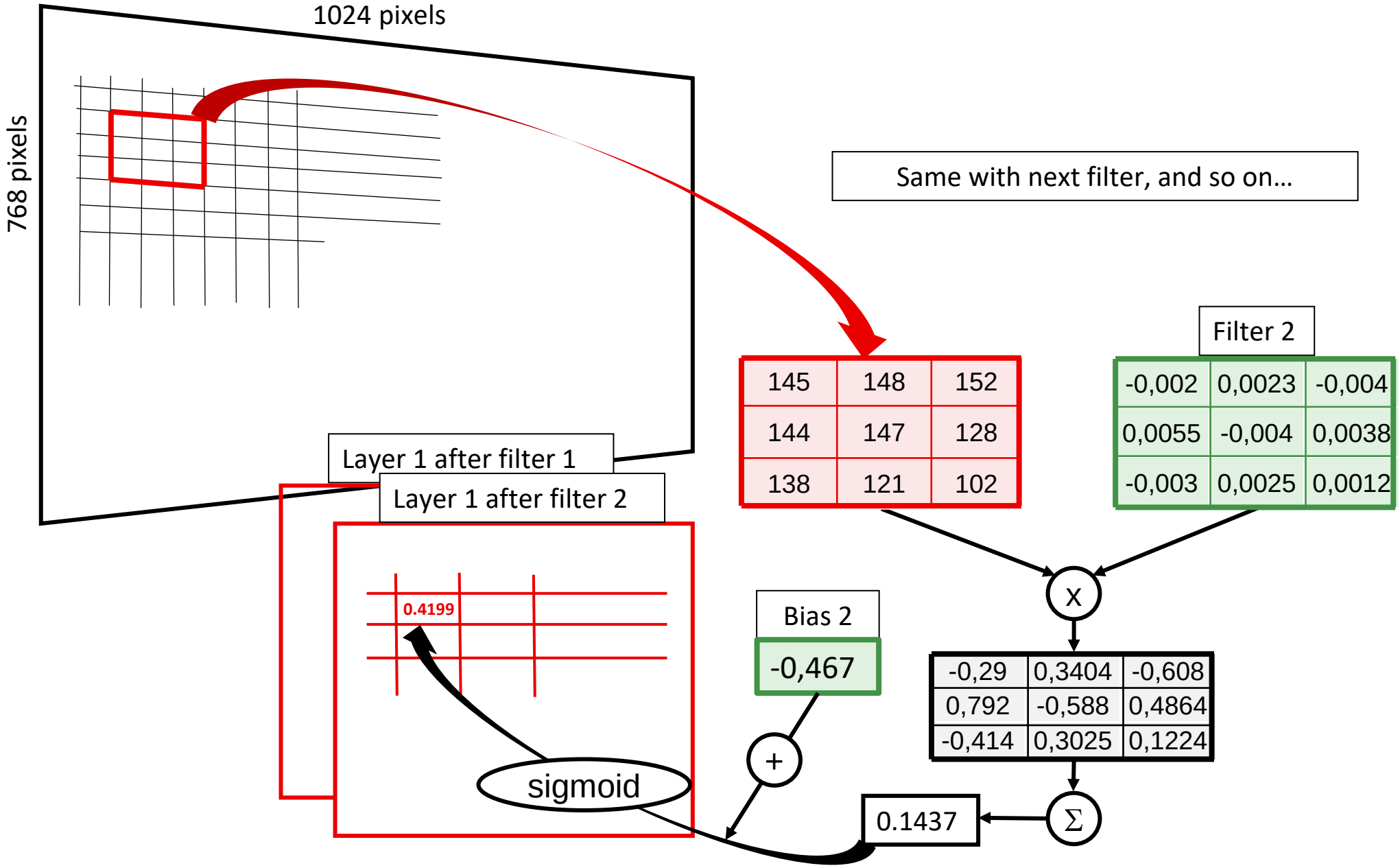
Convolutional Neural Network (3)



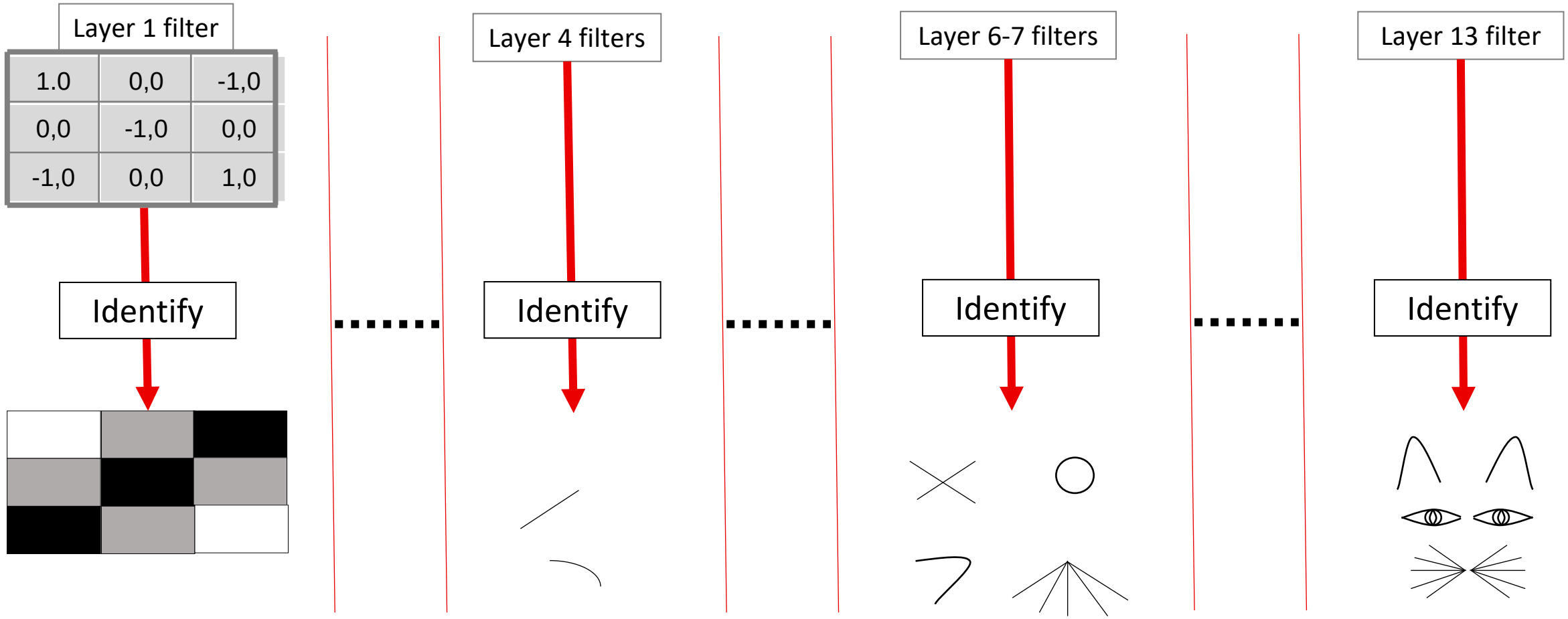
Convolutional Neural Network (4)



Convolutional Neural Network (5)



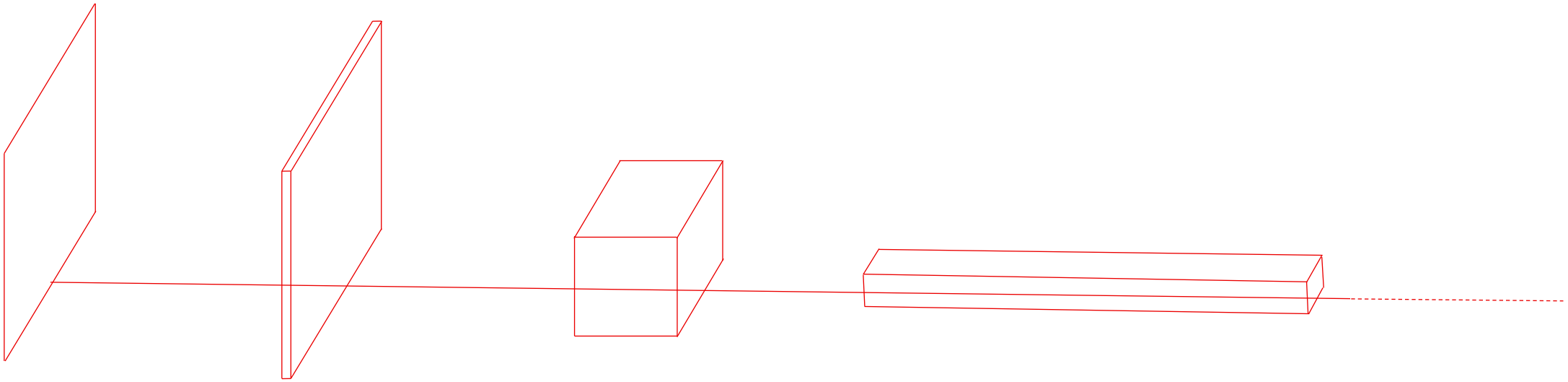
Convolutional Neural Network (6) - What's the idea behind filters?



Convolutional Neural Network (7)

Few parameters per filter, but many filters, many layers (Deep)

- In this case, only 9 parameters per filter! So easy to use for example 12 filters in first layer, then 10 in second layer, etc…
- The deeper the network, the more complex combinations of pixels, to identify sophisticated shapes
- Training decides filter weights, and therefore pixels combinations that better discriminate between images
- General idea: look for specific shapes, whatever the position in the whole picture



Input layer
1024 x 768 x 1

Conv layer 1
12 (3 x 3) filters
1022 x 766 x 12

Conv layer 2
10 (5 x 5) filters
509 x 381 x 120

Conv layer 3
8 (4 x 4) filters
168 x 126 x 960

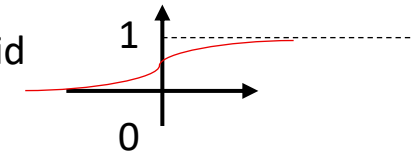
Convolutional Neural Network (8)

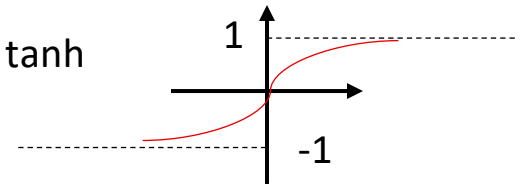
Other aspects

- To reduce number of neurons, use **pooling layers**: group neighboring neurons, taking the max, sometimes the average, …
- After several convolutional layers, use flatten layer, that restructure layer tensor into a vector. Usually followed by fully connected layer
- CNN allow to keep number of parameters (neuron weights) into control. But because of input size, high number of neurons, so heavy computing power still required, particularly with video.

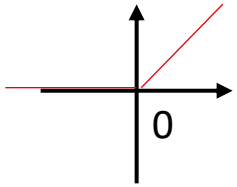
The danger with activation functions

- Without activation functions, NN would only perform linear combination of inputs. No need for several layer, one would be enough
- Initial idea: transform neuron value in a Boolean value (true of false, 0 or 1) in a smooth way

- Sigmoid 

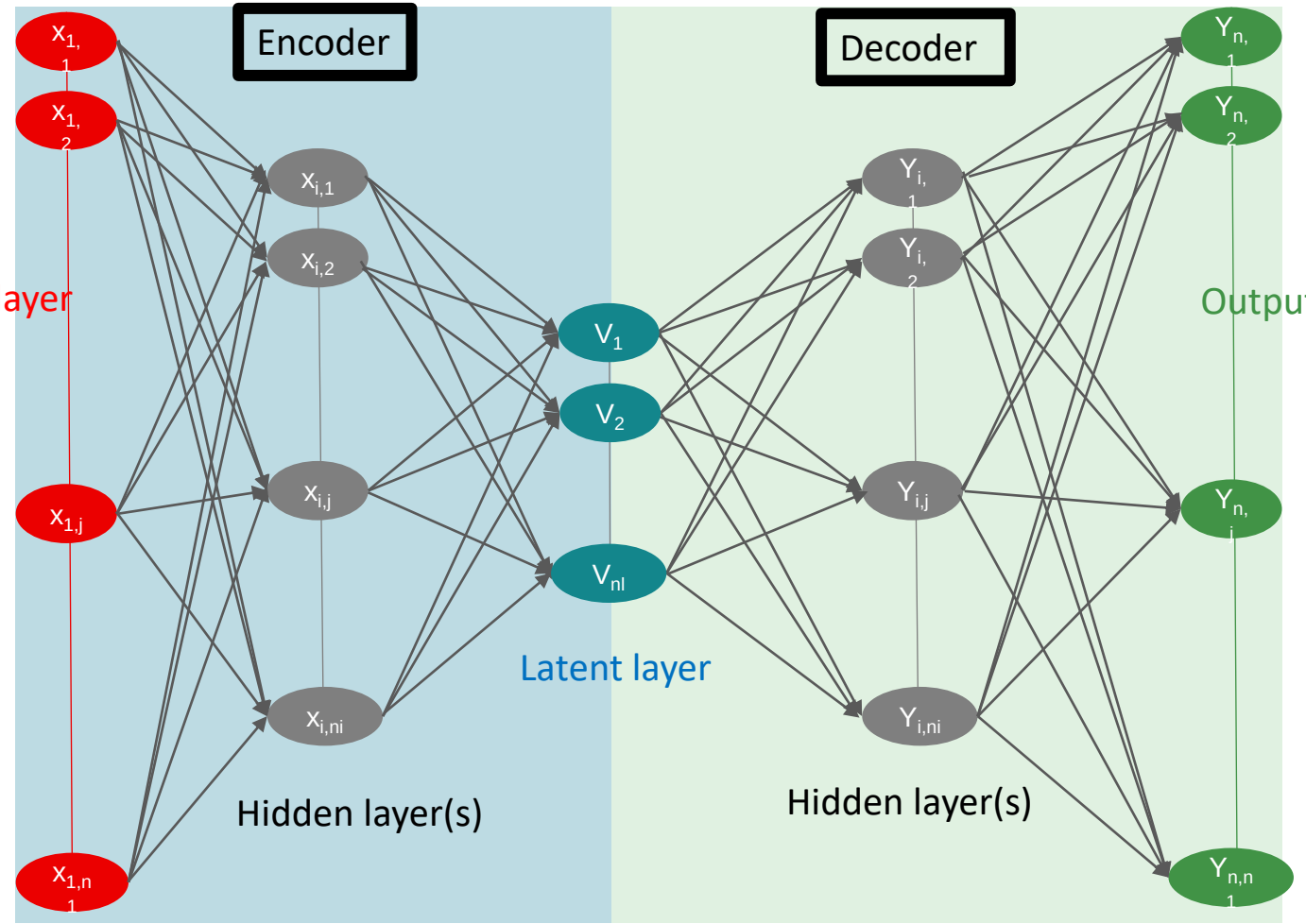
tanh 

- Then, to avoid vanishing gradient, ReLU was introduced

ReLU 

- Problem: when its input is positive, ReLU is identity function (=> linear)
- When their input is around 0, both sigmoid and tanh behave almost as identity

Deep Learning architecture to do better than PCA : Autoencoders



Intuition : project input layer on latent layer, from which the decoder generates output layer as close as possible to the input layer

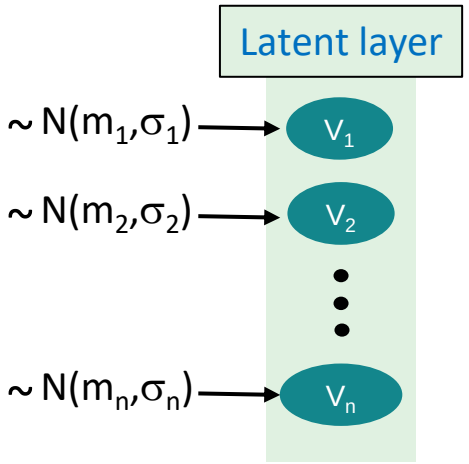
- Input and output layers of same size
- Train network to minimize difference between input and output layers
- Encoder and decoder can be symmetrical, but not necessary
- Layers can be fully connected(=dense), convolutional, LSTM, or of any kind

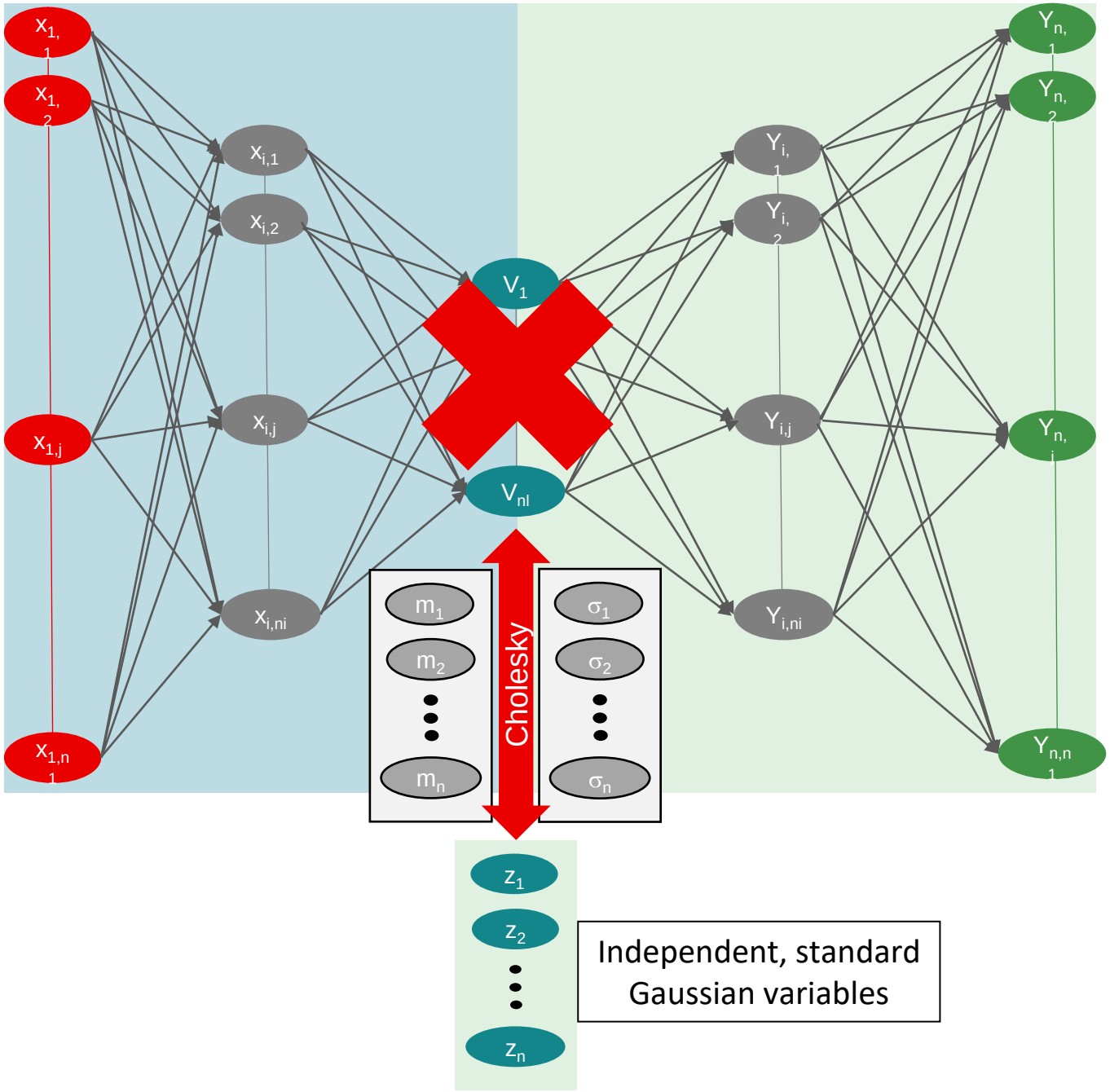
Transform inputs series into series of vectors (“embeddings”)

- Step 1: train the autoencoder (encoder and decoder altogether)
- Step 2: freeze its parameters (weights and biases)
- Step 3: Use encoder to transform inputs into latent layer vector of neuron values

Variational AutoEncoder

- Each V_i from series of latent vectors can be considered as a series of instances of a random variable
- Very often, latent layer variables happened to be Gaussian
  => can be represented by their mean m and standard deviation σ
- In general, when latent layer variables follow a well known probability distribution, you can replace this latent layer with vectors of this probability distribution parameters
- In that case, the AutoEncoder is called **Variational AutoEncoder (VAE)**

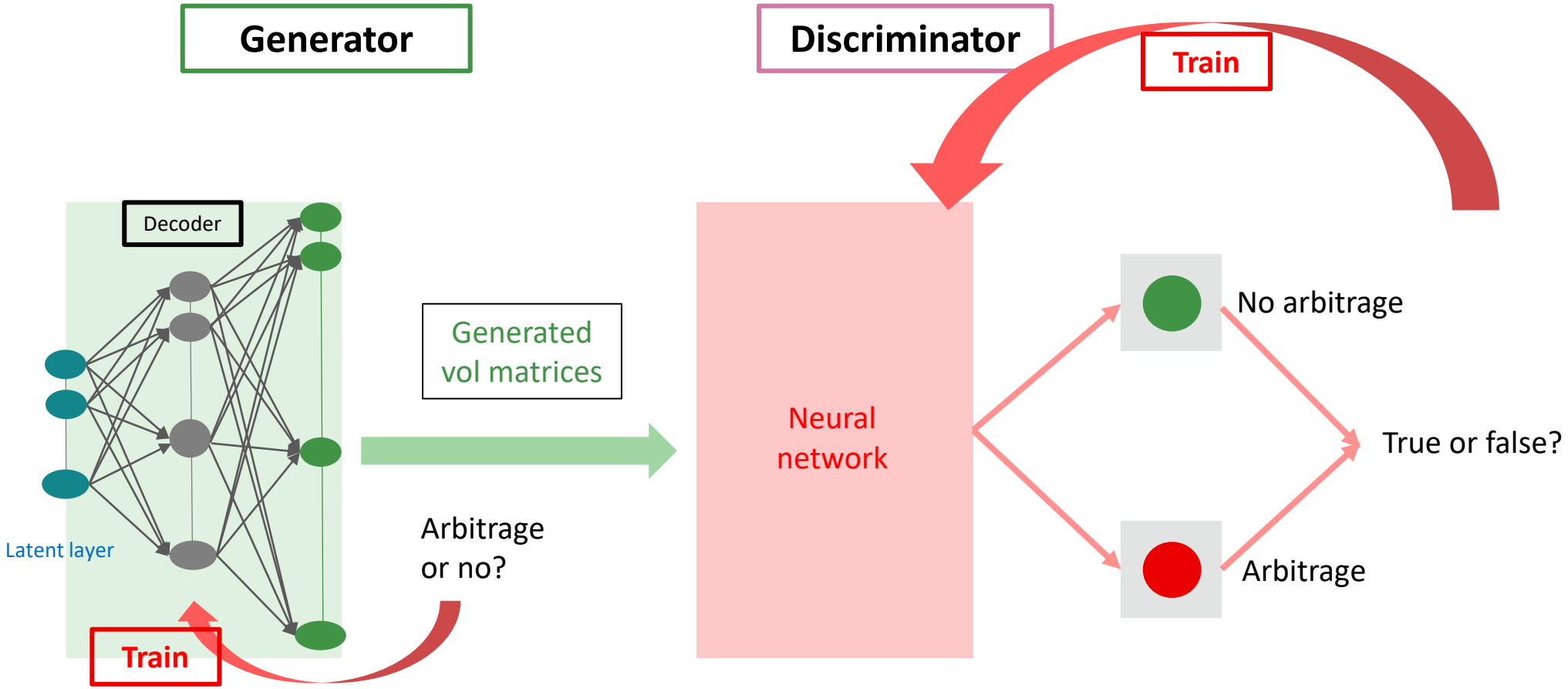




GAN - Generative Adversarial Network

GAN main idea

- In a GAN, Generator tries to fool the Discriminator, and the Discriminator tries not to get fooled
- Both the Generator and the discriminator have to be trained at the same time
- Example: generate non-arbitrable volatility matrices

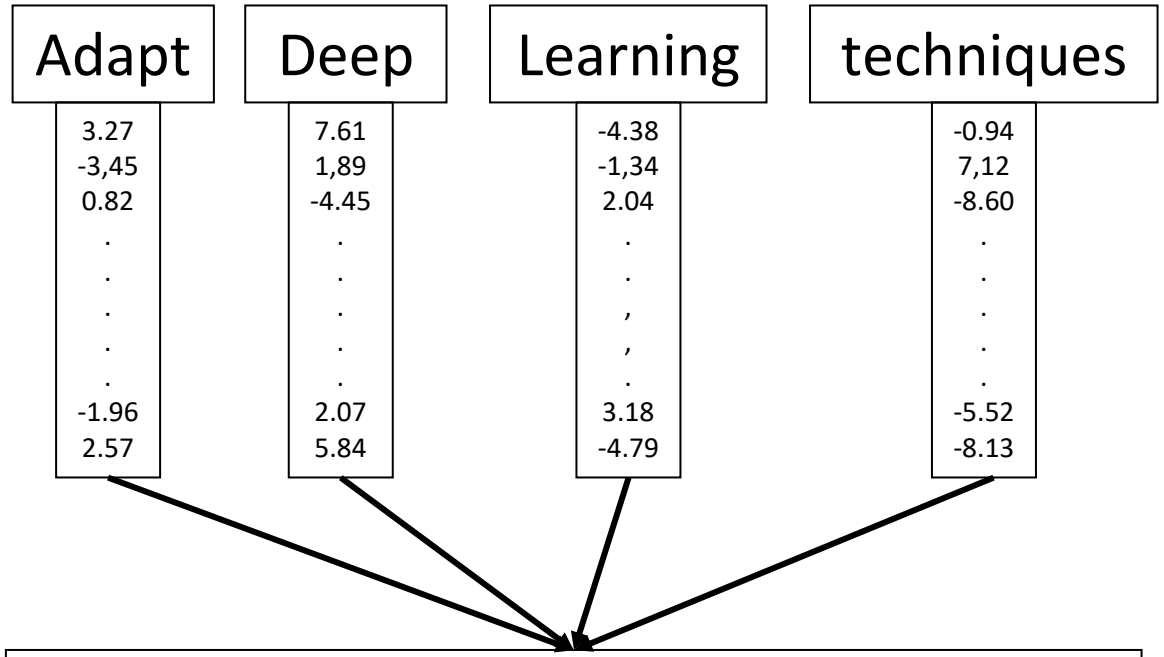


Language pre-treatment for Deep Learning : vectors of embeddings (1)

Vector of embeddings – What is the idea?

- Represent each word by a vector of numbers. Typical vector size : 100 to 500
- Each number role not well known
- In word embeddings vectors information is digital, compact and linear

=> very suitable to apply mathematical techniques like linear algebra

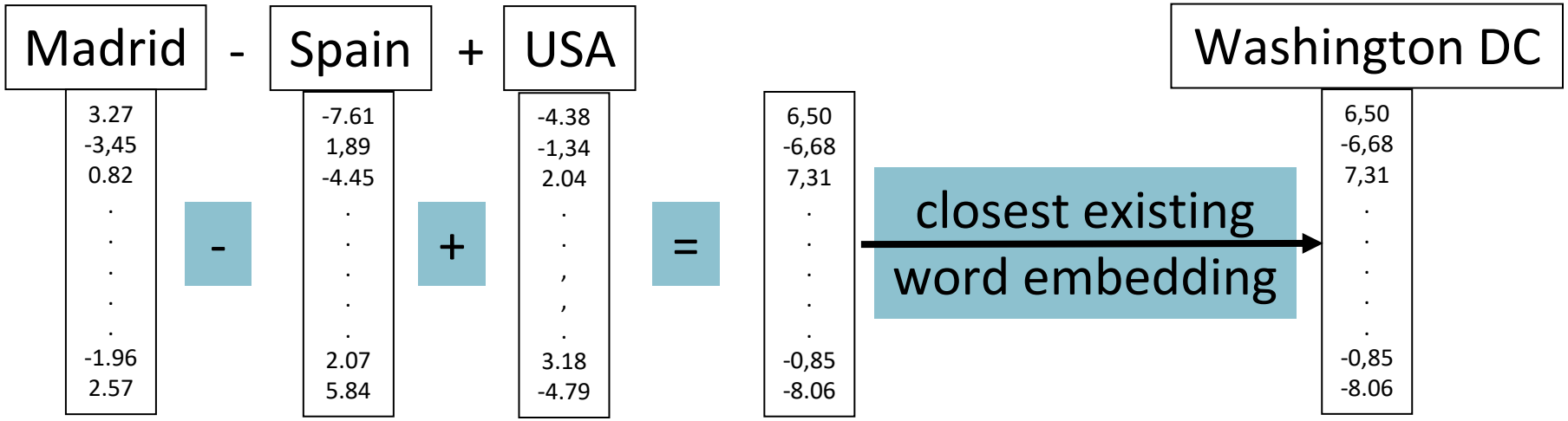


Each vector contains:

- Information about the word itself
- Information about different contexts where the word is used

Language pre-treatment for Deep Learning : vectors of embeddings (2)

Very powerful technique. Example:



Many applications

- Machine translation
- Sentiment analysis
- Next word suggestion
- Missing word suggestion
- Text generation
- …. And many more ….

The state of AI in 2024

- Definition and history
- AI practical cases for financial markets
- Who is doing what in the AI industry?
- Some advice from an old geezer

AI in capital markets : Applications + models

Markets

- Market data simulation
- Portfolio simulation
- Fast pricing
- Cross-markets influence
- Algo trading/hedging
- Markets regime detection, markets surveillance with alerts

Markets + customers

- Intelligent pricing
- Recommendation engines
- Transactions analysis
- Voice command

Customers

- Information extraction from call reports, chats, news
- Customer behavior analysis, sentiment analysis, topic detection, anomaly detection
- Customers and products segmentation
- News analysis for DCM (like Equity Capital Markets)

Models and techniques

- Random Forest, Gradient Boosting
- Convolution Neural Networks
- Recurrent Neural Networks LSTM
- Auto-Encoders, GANs
- Active Learning
- Graph Neural Networks
- Time series analysis
- Reinforcement Learning
- Collaborative Filtering (+Deep)
- NLP – Speech recognition
- NLP – advanced OCR
- NLP – sentiment analysis
- Knowledge graphs
- Clustering/classification

Process control

- Data quality check, data cleansing
- Reconciliation
- Software maintenance & support

Documents

- Information extraction from documents and contracts, CSA/CSD agreements, Term sheets (IBOR)
- Documents/reports generation

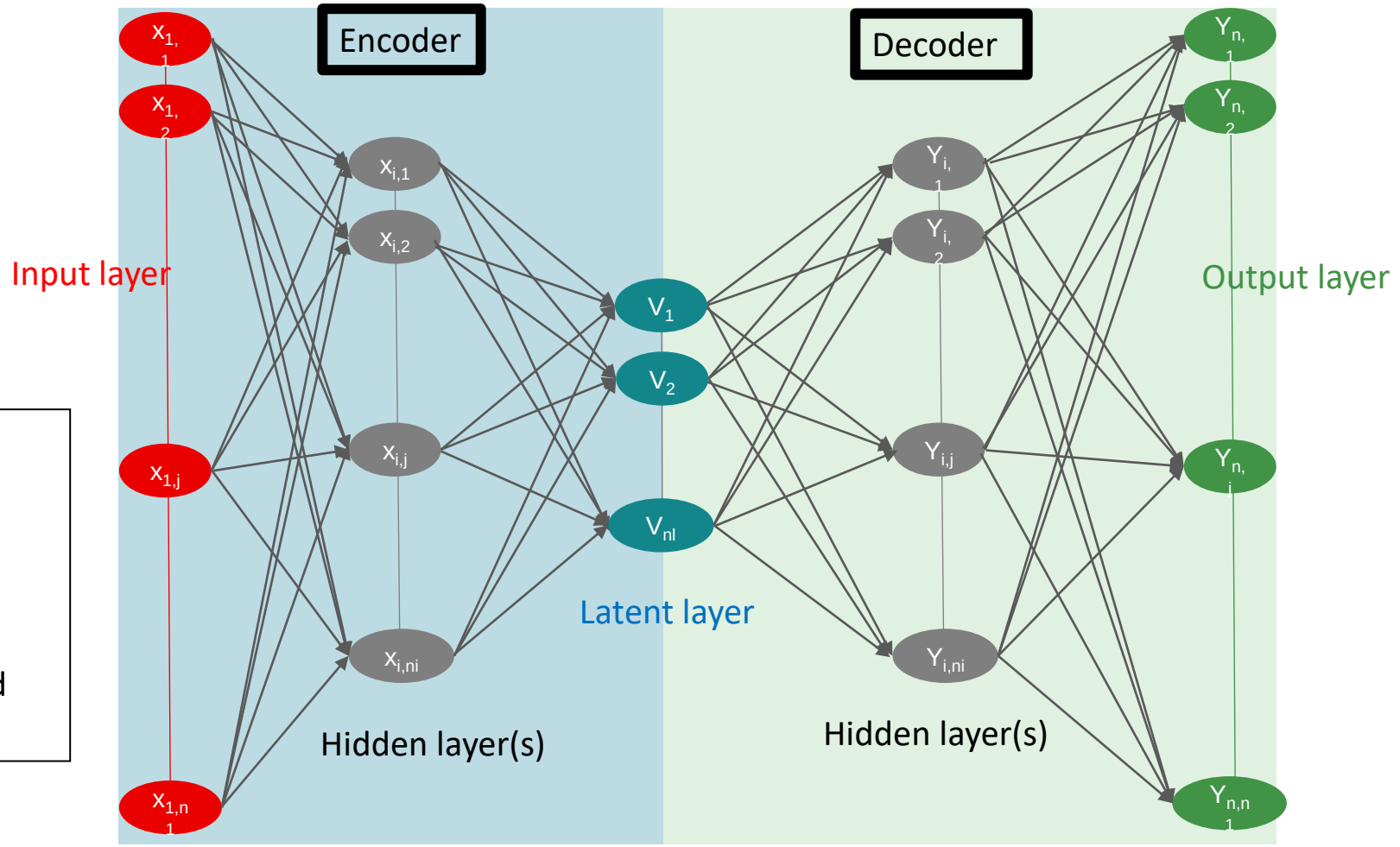
FF expertise:

- Basic
- Decent
- Advanced

Application 1 : Market data historical series

Extract main factors

- Using Variational AutoEncoders technique, an extension of Principal Component Analysis to non-linear complex relations
- Typically reduces swaption vol cubes from 9 000 data to 250 factors, vol surfaces from 500 data to 25 factors, Interest rates curves from 20 data to 2 factors



Applications

- Build scenarios controlling probability distributions (quantiles, …)
- Data quality check, anomaly detection
- Guess missing or wrong historical data
- Imply non-liquid data from more liquid data (US IR curve, main FX pairs, …)

Applis

Applications to transactions data - What are transactions data?

Transactions data

- Many types of transactions: closed sales, “Add to cart”, websites clicks on product page, asked questions, products ratings, …
- Data easily represented the following way

Ope ID	Timestamp	Client ID	Product Ref	Quantity
032 001	2020-04-13-09-34-56	FER_ES	FRYY_202006_S	34 200
032 002	2020-04-13-09-35-04	AIRB_EU	GERZ_202008_M	12 000
032 003	2020-04-13-09-35-47	FER_ES	FRYY_202006_S	85 620
032 004	2020-04-13-09-36-12	BT_UK	GERZ_202008_M	22 900

As opposed to “content” data

- Customers characteristics (= features):, address, age, nationality, incomes, height, weight, …
- Products features: size, unit price, etc… Depends on your activity

Transaction data pre-treatment 1 : generate a client-product matrix

Or user-item matrix

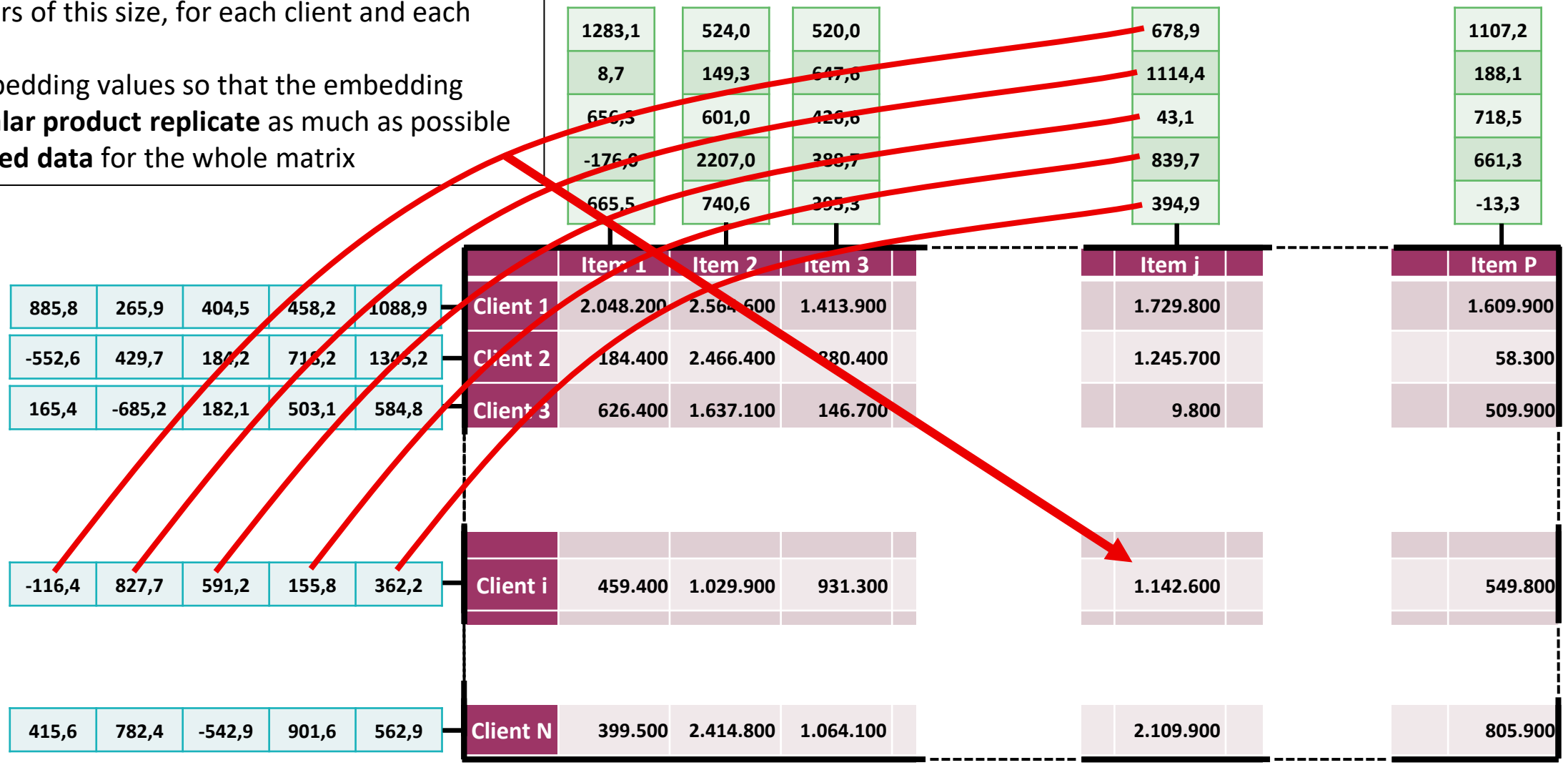
- For a given period, for each pair (client I ; product j), compute the accumulated quantity, based on records presented in previous slide

	Item 1	Item 2	Item 3		Item j		Item P
Client 1	2.048.200	2.564.600	1.413.900		1.729.800		1.609.900
Client 2	184.400	2.466.400	880.400		1.245.700		58.300
Client 3	626.400	1.637.100	146.700		9.800		509.900
Client i	459.400	1.029.900	931.300		1.142.600		549.800
Client N	399.500	2.414.800	1.064.100		2.109.900		805.900

Transaction data pre-treatment 2: calibrate your embedding vectors

Collaborative filtering

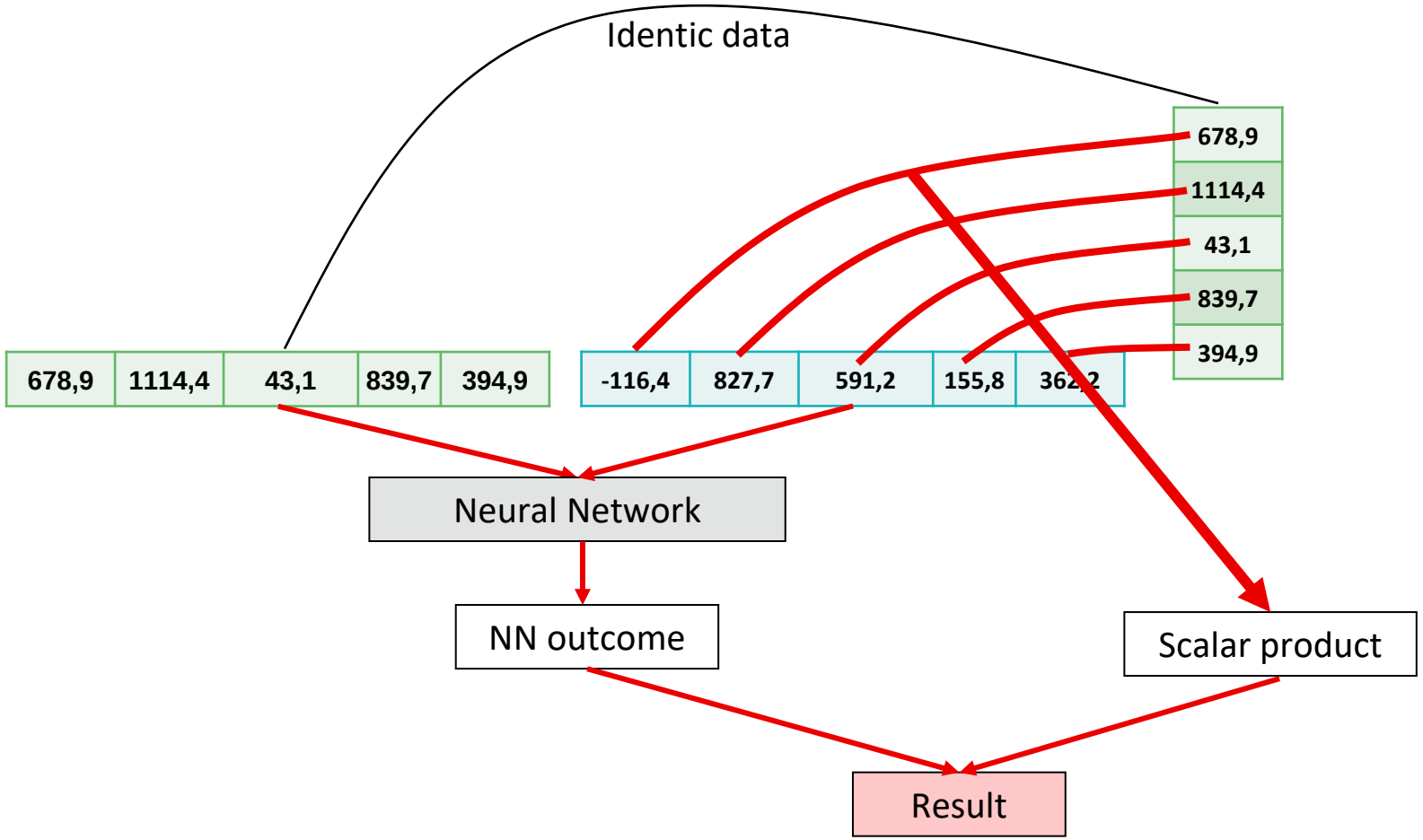
- Choose an **embedding size**, for example 5 or 25, and build vectors of this size, for each client and each product
- Adjust embedding values so that the embedding vectors **scalar product replicate** as much as possible the **observed data** for the whole matrix



Transaction data pre-treatment 2 - alternative: Deep collaborative filtering

Add neural network in parallel

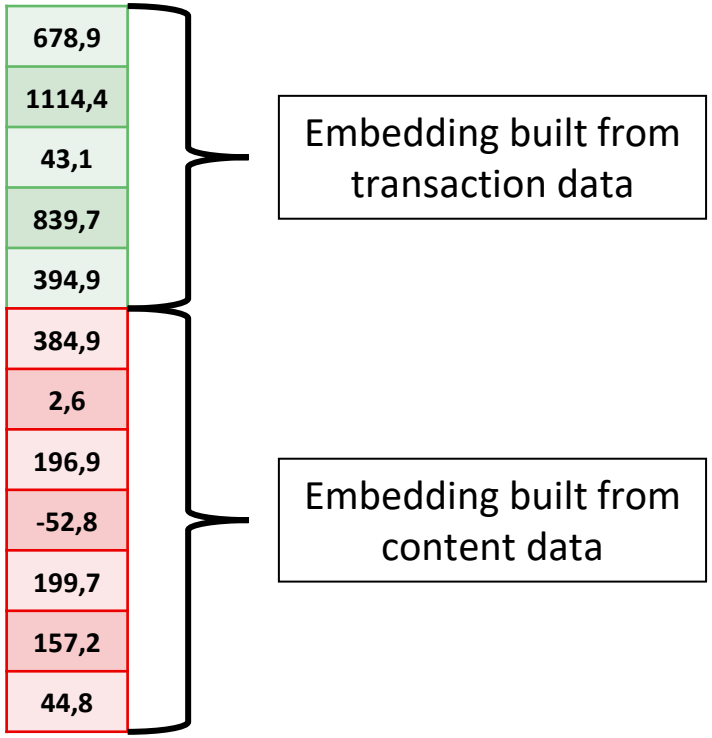
- Everything as before, except vectors scalar product



Transaction data pre-treatment 2 - improvement: embeddings enrichment with content

Combine transaction information with content

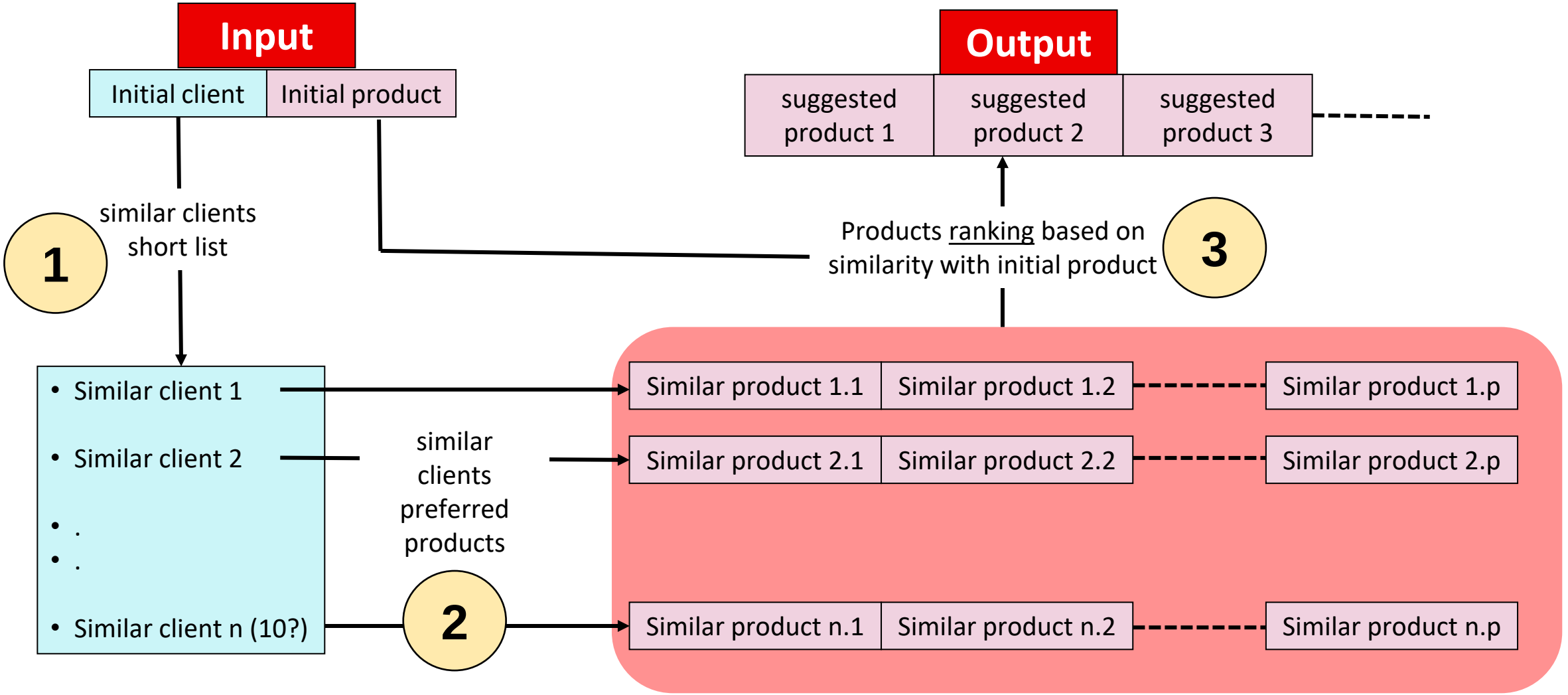
- To enrich the information embedded in vectors
- Simple way: concatenate vectors
- More sophisticated way: take content-based vectors into account when building transaction-based vectors



Application 2 : Recommendation engine

"Amazon-like" selection process

- Context: Initial transaction (closed deal, request for quote, add to basket, click, …) of a client on a product
- Goal: suggest other product(s) that should trigger interest from initial client



Application 3 : Anomaly detection

Identify clients not active enough (or “too” active) in certain products

- Compare model predicted numbers with actual observations
- Pick client-product pairs with higher differences

Observations

	Item 1	Item 2	Item 3			Item j			Item P
Client 1	2.048.200	-	-	1.413.900	-	-	-	-	1.609.900
Client 2	-	-	880.400		2.564.600	-	-	-	-
Client 3	626.400	1.637.100	-	-	-	-	1.245.700	-	-
	-	-	-	-	-	9.800	-	-	-
	-	-	-	146.700	-	-	-	184.400	-
	-	-	-	-	1.729.800	-	-	-	-
	-	-	-	-	-	1.142.600	-	58.300	-
Client i	-	1.029.900	-	-	-	-	-	-	549.800
	-	-	459.400	-	-	-	399.500	-	-
	-	-	-	2.414.800	-	-	-	-	-
	-	-	-	-	-	2.109.900	-	509.900	-
Client N	399.500	-	1.064.100	-	-	-	-	-	805.900

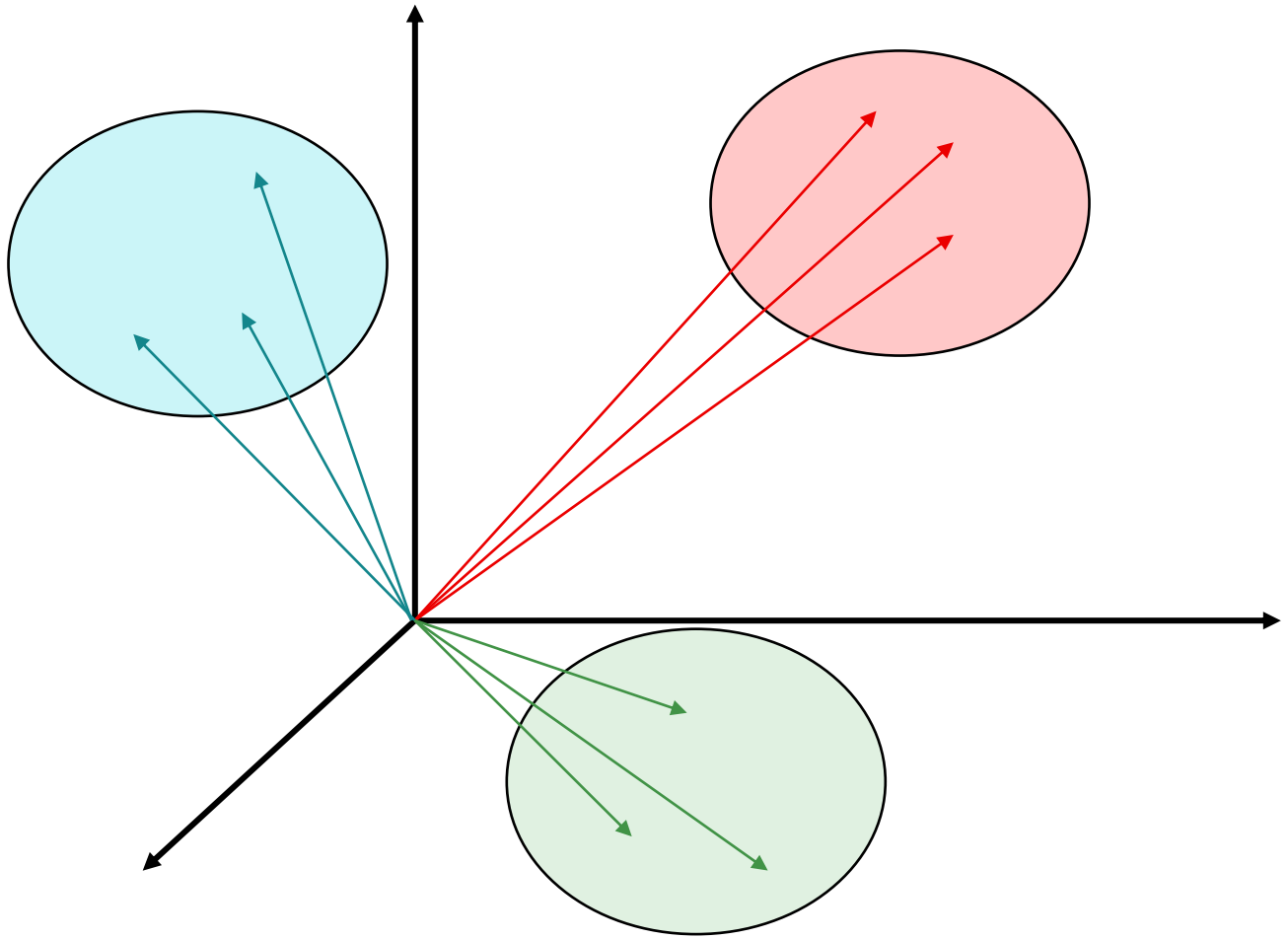
Embedding-based modeled values

	Item 1	Item 2	Item 3			Item j			Item P
Client 1	2.048.200	22.964	120.241	1.413.900	177.001	78.120	121.644	80.747	1.609.900
Client 2	101.198	38.570	880.400	181.307	2.564.600	34.983	93.246	103.740	154.283
Client 3	626.400	1.637.100	205.645	208.884	149.528	159.578	1.245.700	41.449	165.182
	86.931	87.598	80.618	28.488	137.295	9.800	40.794	169.620	58.126
	53.804	36.910	228.302	146.700	4.667	33.682	191.201	184.400	143.489
	26.505	112.799	104.153	133.288	1.729.800	8.688	243.363	180.520	164.518
	130.097	149.848	58.741	92.917	179.361	1.142.600	161.875	58.300	6.302
Client i	46.880	1.029.900	16.932	169.646	9.500	78.339	117.169	59.763	549.800
	197.611	60.292	459.400	14.784	85.692	46.796	399.500	14.702	115.155
	114.306	113.524	140.527	2.414.800	154.693	160.984	8.288	229.516	15.019
	42.905	53.441	53.338	10.087	4.441	2.109.900	51.265	509.900	19.120
Client N	399.500	48.039	1.064.100	41.032	158.078	51.013	100.984	75.578	805.900

Application 4 : Customers and products segmetation

Apply clustering algorithm

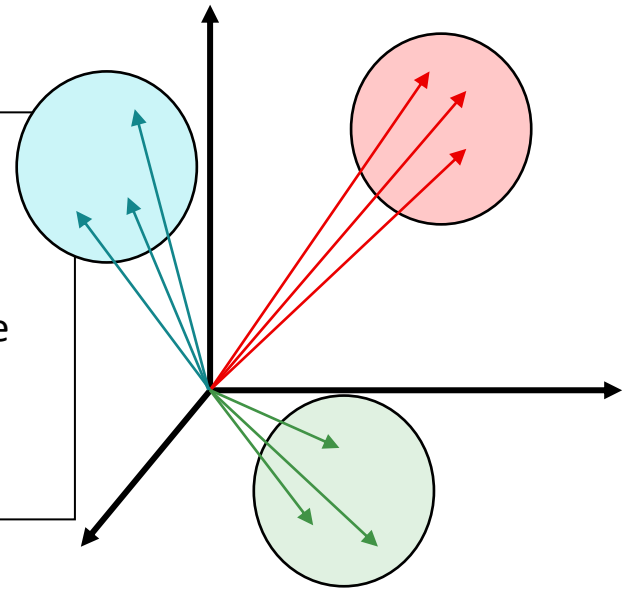
- Using Euclidian distance between embedding vectors
- Group items with small distance between themselves (= similar items) in a common class
- Apply clustering separately to clients on one side, and to products on other side
- In figure below, 3D representation, but in reality, space dimension = embedding vectors size



Application 4 : Customers and products clustering (2)

Several families of clustering algorithms available

- **Centroid-based**: Reduce distance to class centroid. Typical example: k-means
- **Density-based**: Classes defined by homogeneous density. Typical example: DBSCAN
- **Distribution-based**: Fitting probability distributions to classes. Typical example: Gaussian mixture
- **Hierarchical**: Building classes one by one, aggregating ao separating current classes based on distanceTypical example: Agglomerative (bottom-up), Divisive (top-down)
- **Neural gas**: Neural network inspired by self-organizing map



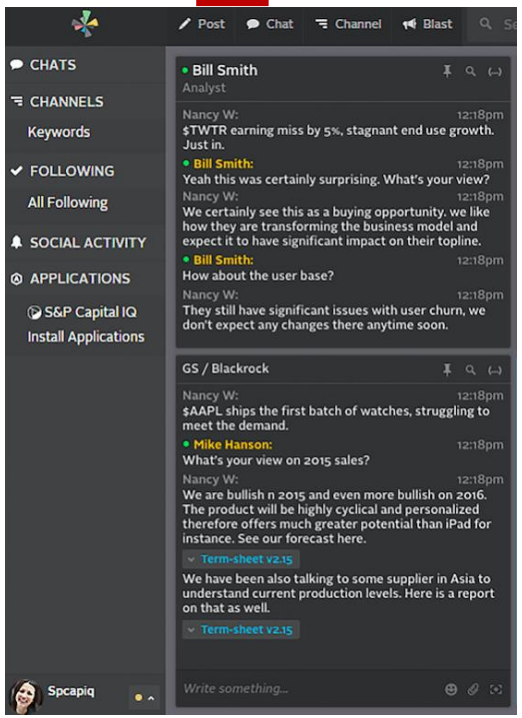
Our original method: combine different algorithms results

- Identify groups of items which stay in the same class with different clustering methods. They are considered as core classes
- Train a supervised classifier on these core classes. It will then give percentage of distribution between classes for unclassified items

Next step

- **Analyze data from a time sequence point of view.**
- **Objectives: prediction, regime/pattern detection**

Application 5 – NLP: Market quotes extraction from chats with brokers



Ope ID	Timestamp	Maturity	Bid	Offer
032 001	2020-04-13-09-34-56	10 y	0,931	0,933
032 002	2020-04-13-09-35-04	5 y	-	0,498
032 003	2020-04-13-09-35-47	7 y	0,692	0,695
032 004	2020-04-13-09-36-12	10 y	0,929	-

Excellent results

- Using a combination of classic NLP (Natural Language Processing) and Machine Learning
- On EUR IRS, up to 99,97% accuracy

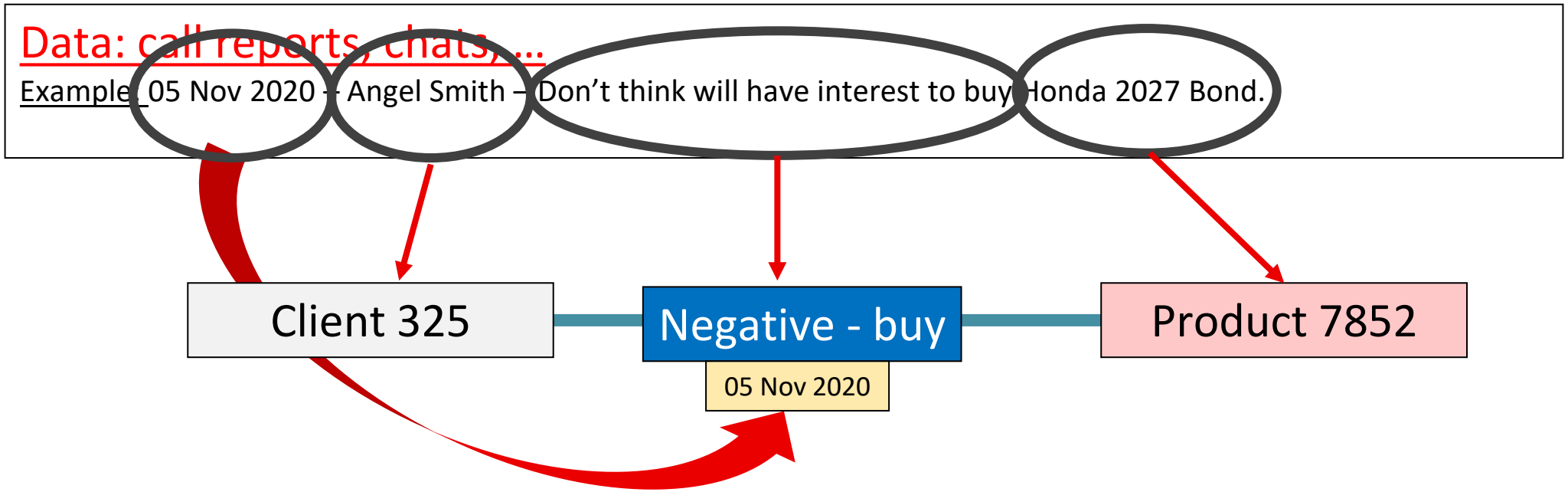
Applis

Application 6 – NLP: Customer feedback exploitation (1)

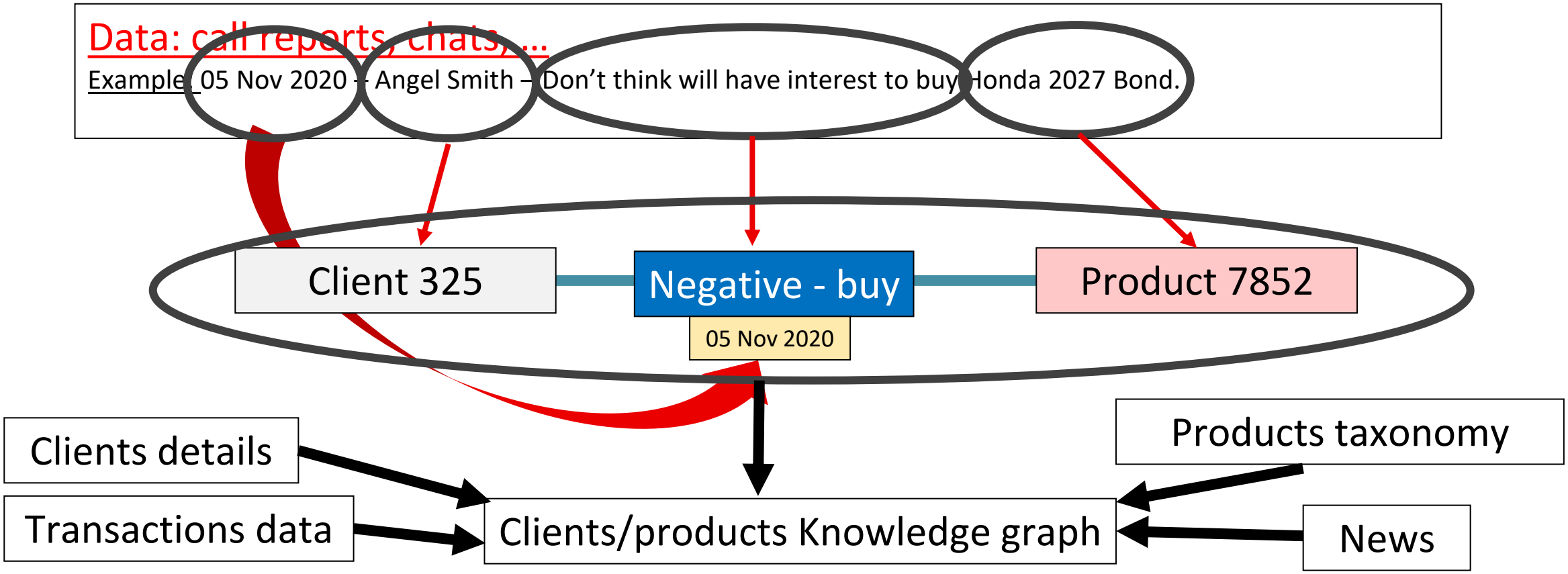
Data: call reports, chats, …

Example: 05 Nov 2020 – Angel Smith – Don’t think will have interest to buy Honda 2027 Bond.

Application 6 – NLP: Customer feedback exploitation (2)



Application 6 – NLP: Customer feedback exploitation (3)



Applications

- Database requests
- Clients segmentation, Products segmentations
- Recommendation engine
- Relations prediction
- …

Applis

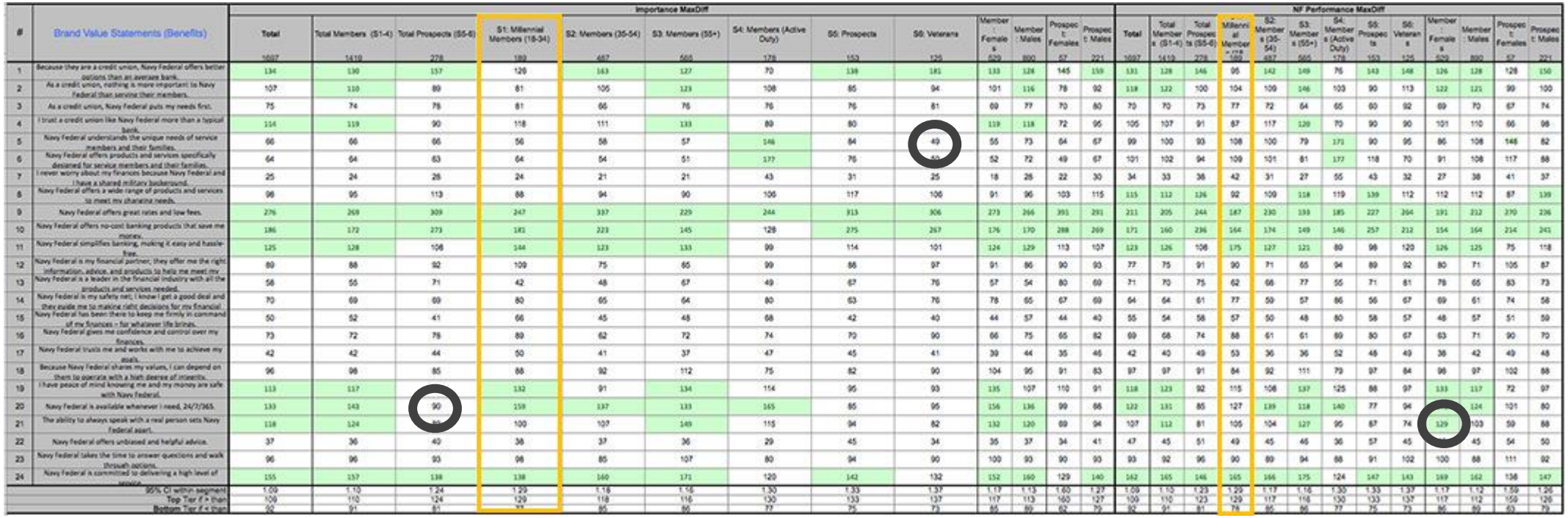
Application 7: Data quality check (1)

#	Brand Value Statements (Benefits)	Importance MaxDiff													NF Performance MaxDiff											
		Total	Total Members (S1-4)	Total Prospects (S5-6)	S1: Millennial Members (18-34)	S2: Members (35-54)	S3: Members (55+)	S4: Members (Active Duty)	S5: Prospects	S6: Veterans	Member : Females	Member : Males	Prospect: Females	Prospect: Males	Total	Total Members (S1-4)	Total Prospects (S5-6)	Millennial Members (18-34)	S2: Members (35-54)	S3: Members (55+)	S4: Members (Active Duty)	S5: Prospects	S6: Veterans	Member : Females	Member : Males	Prospect: Females	Prospect: Males
		1007	1419	278	189	487	565	178	153	125	529	890	57	221	1007	1419	278	189	487	565	178	153	125	529	890	57	221
1	Because they are a credit union, Navy Federal offers better options than an average bank.	134	130	157	126	163	127	70	138	181	133	124	145	159	131	128	146	95	142	149	76	143	148	126	128	128	150
2	As a credit union, nothing is more important to Navy Federal than service their members.	107	110	89	81	105	123	108	85	94	101	116	78	92	118	122	100	104	109	146	103	90	113	122	121	99	100
3	As a credit union, Navy Federal puts my needs first.	75	74	78	81	66	76	76	76	81	69	77	70	80	70	70	73	77	72	64	65	60	92	69	70	67	74
4	I trust a credit union like Navy Federal more than a typical bank.	114	119	90	118	111	133	89	80	103	119	118	72	95	105	107	91	87	117	120	70	90	90	101	110	66	98
5	Navy Federal understands the unique needs of service members and their families.	66	66	66	56	58	57	146	84	49	55	73	64	67	99	100	93	108	100	79	171	90	95	86	108	148	82
6	Navy Federal offers products and services specifically designed for service members and their families.	64	64	63	64	54	51	177	76	50	52	72	49	67	101	102	94	109	101	81	177	118	70	91	108	117	88
7	I never worry about my finances because Navy Federal and I have a shared military background.	25	24	28	24	21	21	43	31	25	18	28	22	30	34	33	38	42	31	27	55	43	32	27	38	41	37
8	Navy Federal offers a wide range of products and services to meet my changing needs.	98	95	113	88	94	90	106	117	106	91	96	103	115	115	112	126	92	109	118	119	139	112	112	112	87	139
9	Navy Federal offers great rates and low fees.	276	269	309	247	337	229	244	313	306	273	266	391	291	211	205	244	187	230	193	185	227	264	191	232	270	236
10	Navy Federal offers no-cost banking products that save me money.	186	172	273	181	223	145	128	275	267	176	170	288	269	171	160	236	164	174	149	146	257	212	154	164	214	241
11	Navy Federal simplifies banking, making it easy and hassle-free.	125	128	106	144	123	133	99	114	101	124	129	113	107	123	126	106	175	127	121	89	98	120	126	125	75	118
12	Navy Federal is my financial partner, they offer me the right information, advice, and products to help me meet my	89	88	92	109	75	65	99	88	97	91	86	90	93	77	75	91	90	71	65	94	89	92	80	71	105	87
13	Navy Federal is a leader in the financial industry with all the products and services needed.	56	55	71	42	48	67	49	67	76	57	54	80	69	71	70	75	62	66	77	55	71	61	78	65	83	73
14	Navy Federal is my safety net; I know I get a good deal and they guide me to making right decisions for my financial	70	69	69	80	65	64	80	63	76	78	65	67	69	64	64	61	77	59	57	86	56	67	69	61	74	58
15	Navy Federal has been there to keep me firmly in command of my finances – for whatever life brings.	50	52	41	66	45	48	68	42	40	44	57	44	40	55	54	58	57	50	48	80	58	57	48	57	51	59
16	Navy Federal gives me confidence and control over my finances.	73	72	78	89	62	72	74	70	90	66	75	65	82	69	68	74	88	61	61	89	80	67	63	71	90	70
17	Navy Federal trusts me and works with me to achieve my goals.	42	42	44	50	41	37	47	45	41	39	44	35	46	42	40	49	53	36	36	52	48	49	38	42	49	48
18	Because Navy Federal shares my values, I can depend on them to operate with a high degree of integrity.	96	98	85	88	92	112	75	82	90	104	95	91	83	97	97	91	84	92	111	79	97	84	98	97	102	88
19	I have peace of mind knowing me and my money are safe with Navy Federal.	113	117	95	132	91	134	154	95	93	135	107	110	91	118	123	92	115	108	137	125	88	97	133	117	72	97
20	Navy Federal is available whenever I need, 24/7/365.	133	143	90	159	137	133	165	85	95	156	136	99	86	122	131	85	127	139	118	140	77	94	141	124	101	80
21	The ability to always speak with a real person sets Navy Federal apart.	118	124	89	100	107	149	115	94	82	132	120	89	94	107	112	81	105	104	127	95	87	74	129	103	59	88
22	Navy Federal offers unbiased and helpful advice.	37	36	40	38	37	36	29	45	34	35	37	34	41	47	45	51	49	45	46	36	57	45	46	45	54	50
23	Navy Federal takes the time to answer questions and walk through options.	96	96	93	98	85	107	80	94	90	100	93	90	93	93	92	96	90	89	94	88	91	102	100	88	111	92
24	Navy Federal is committed to delivering a high level of service.	155	157	138	138	160	171	120	142	132	152	160	129	140	162	165	146	165	166	175	124	147	143	169	162	138	147
	95% CI within segment	1.09	1.10	1.24	1.29	1.18	1.16	1.30	1.33	1.37	1.17	1.13	1.60	1.27	1.09	1.10	1.23	1.29	1.17	1.16	1.30	1.33	1.37	1.17	1.12	1.59	1.26
	Top Tier if > than	109	110	124	129	118	116	130	133	137	117	113	160	127	109	110	123	129	117	116	130	133	137	117	112	159	126
	Bottom Tier if < than	92	91	81	**	85	86	72	75	73	85	89	62	79	92	91	81	78	85	86	77	75	73	86	89	63	79

Application 7: Data quality check (2)

Version 1 : replicate teams results

- M/O or risk teams currently working on checking data quality, identifying wrong data
- Train Machine Learning algorithms to replicate their results
- Already in production for daily CVA/DVA deal filters



#	Brand Value Statements (Benefits)	Importance MaxDiff													NF Performance MaxDiff										
		Total	Total Members (S1-4)	Total Prospects (S5-6)	S1: Millennial Members (18-34)	S2: Members (35-54)	S3: Members (55+)	S4: Members (Active Duty)	S5: Prospects	S6: Veterans	Member : Females	Member : Males	Prospect: Females	Prospect: Males	Total	Total Members (S1-4)	Total Prospects (S5-6)	Millennial Members (18-34)	S2: Members (35-54)	S3: Members (55+)	S4: Members (Active Duty)	S5: Prospects	S6: Veterans	Member : Females	Member : Males	Prospect: Females	Prospect: Males
		1007	1419	278	189	487	565	178	153	125	529	800	57	221	1007	1419	278	189	487	565	178	153	125	529	800	57	221
1	Because they are a credit union, Navy Federal offers better options than an average bank.	134	130	157	126	163	127	70	138	181	133	124	145	159	131	128	146	95	142	149	76	143	148	126	128	128	150
2	As a credit union, nothing is more important to Navy Federal than service their members.	107	110	89	81	105	123	108	85	94	101	116	78	92	118	122	100	104	109	146	103	90	113	122	121	99	100
3	As a credit union, Navy Federal puts my needs first.	75	74	78	81	66	76	76	76	81	69	77	70	80	70	70	73	77	72	64	65	60	92	69	70	67	74
4	I trust a credit union like Navy Federal more than a typical bank.	114	119	90	118	111	133	89	80		119	118	72	95	105	107	91	87	117	120	70	90	90	101	110	66	98
5	Navy Federal understands the unique needs of service members and their families.	66	66	66	56	58	57	146	84	49	55	73	64	67	99	100	93	108	100	79	171	90	95	86	108	148	82
6	Navy Federal offers products and services specifically designed for service members and their families.	64	64	63	64	54	51	177	76	49	52	72	49	67	101	102	94	109	101	81	177	118	70	91	108	117	88
7	I never worry about my finances because Navy Federal and I have a shared military background.	25	24	28	24	21	21	43	31	25	18	28	22	30	34	33	38	42	31	27	55	43	32	27	38	41	37
8	Navy Federal offers a wide range of products and services to meet my changing needs.	98	95	113	88	94	90	106	117	106	91	96	103	115	115	112	126	92	109	118	119	139	112	112	112	87	139
9	Navy Federal offers great rates and low fees.	276	269	309	247	337	229	244	313	306	273	266	391	291	211	205	244	187	230	193	185	227	264	191	232	270	236
10	Navy Federal offers no-cost banking products that save me money.	186	172	273	181	223	145	128	275	267	176	170	288	269	171	160	236	164	174	149	146	257	212	154	164	214	241
11	Navy Federal simplifies banking, making it easy and hassle-free.	125	128	106	144	123	133	99	114	101	124	129	113	107	123	126	106	175	127	121	89	98	120	126	125	75	118
12	Navy Federal is my financial partner, they offer me the right information, advice, and products to help me meet my	89	88	92	109	75	65	99	88	97	91	86	90	93	77	75	91	90	71	65	94	89	92	80	71	105	87
13	Navy Federal is a leader in the financial industry with all the products and services needed.	56	55	71	42	48	67	49	67	76	57	54	80	69	71	70	75	62	66	77	55	71	61	78	65	83	73
14	Navy Federal is my safety net; I know I get a good deal and they guide me to making right decisions for my financial	70	69	69	80	65	64	80	63	76	78	65	67	69	64	64	61	77	59	57	86	56	67	69	61	74	58
15	Navy Federal has been there to keep me firmly in command of my finances - for whatever life brings.	50	52	41	66	45	48	68	42	40	44	57	44	40	55	54	58	57	50	48	80	58	57	48	57	51	59
16	Navy Federal gives me confidence and control over my finances.	73	72	78	89	62	72	74	70	90	66	75	65	82	69	68	74	88	61	61	89	80	67	63	71	90	70
17	Navy Federal trusts me and works with me to achieve my goals.	42	42	44	50	41	37	47	45	41	39	44	35	46	42	40	49	53	36	36	52	48	49	38	42	49	48
18	Because Navy Federal shares my values, I can depend on them to operate with a high degree of integrity.	96	98	85	88	92	112	75	82	90	104	95	91	83	97	97	91	84	92	111	79	97	84	98	97	102	88
19	I have peace of mind knowing me and my money are safe with Navy Federal.	113	117		132	91	134	154	95	93	135	107	110	91	118	123	92	115	108	137	125	88	97	133	117	72	97
20	Navy Federal is available whenever I need, 24/7/365.	133	143	90	159	137	133	165	85	95	156	136	99	86	122	131	85	127	139	118	140	77	94	129	124	101	80
21	The ability to always speak with a real person sets Navy Federal apart.	118	124	96	100	107	149	115	94	82	132	120	89	94	107	112	81	105	104	127	96	87	74	103	59	88	
22	Navy Federal offers unbiased and helpful advice.	37	36	40	38	37	36	29	45	34	35	37	34	41	47	45	51	49	45	46	36	57	45	45	54	50	
23	Navy Federal takes the time to answer questions and walk through options.	96	96	93	98	85	107	80	94	90	100	93	90	93	93	92	96	90	89	94	88	91	102	100	88	111	92
24	Navy Federal is committed to delivering a high level of service.	155	157	138	138	160	171	120	142	132	152	160	129	140	162	165	146	165	166	175	124	147	143	169	162	138	147
	95% CI within segment	1.09	1.10	1.24	1.29	1.18	1.16	1.30	1.33	1.37	1.17	1.13	1.60	1.27	1.09	1.10	1.23	1.29	1.17	1.16	1.30	1.33	1.37	1.17	1.12	1.59	1.26
	Top Tier if > than	109	110	124	129	118	116	130	133	137	117	113	160	127	109	110	123	129	117	116	130	133	137	117	112	159	126
	Bottom Tier if < than	92	91	81	**	85	86	77	75	73	85	89	62	79	92	91	81	78	85	86	77	75	73	86	89	63	79

Version 2:

- Typical “anomaly detection” problem
- Algorithm learns directly from data the mainstream data structure, mainstream behavior
- => identifies strange data, strange behavior.

Applis

Application 8: Heavy pricer replication

Problem

- Some pricers are heavy in computation time
- Make portfolio behavior analysis time-consuming. Real-time analysis is untractable
- Example: CVA Monte-Carlo pricing, FX tarkos, Bermudan swaptions, …

Solution

- Train a Neural Network to replicate the heavy pricer
- Training is resource- and time-consuming, but execution is fast
- Can be applied to greeks as well, considering each Greek as another pricer to replicate

The state of AI in 2024

- Definition and history
- AI practical cases for financial markets
- Who is doing what in the AI industry?
- Some advice from an old geezer

AI long term history : A series of winters and hype (1)

First hype – 1950s/60s : NLP translation

- 1949 : optimistic report on Machine Translation. IBM and MIT very pushy. Demo in 1954 on few Russian documents. US government and CIA interested. Media exaggerated the potential impact and researcher underestimated the difficulties
- 1966 : US National Research Council report : Machine Translation is too expensive and inefficient. Funds removed.
- 1969 : Book on the limits of single-layer Perceptron.

First winter – 1970s

- 1969: DARPA reduced AI funding then stopped few years later.
- 1973: UK report saying AI computation was untractable.
- Nevertheless, somerResearchers kept on working on AI.

Second hype – 1980s : expert systems

- Early 1980s : various expert systems using LISP language, in-house developed in many companies.
- 1981 : Japanese Fifth Generation Computer Program launched
- 1983 : DARPA Launched a similar project
- 1987 : DARPA dramatically cut AI funding, due to disappointing results
- 1987 : Sun workstations overpowered specialized LISP machines, soon followed by desktop computers
- Early 1990s : expert systems proved difficult to update
- 1992 : Fifth Generation Program stopped due to lack of tangible results.

Second winter – 1990s + early 2000s

- 2005 + 2006 + 2007 : Newspaper articles explain that researchers avoid using the words “AI”. Researchers started using “cognitive systems”, “machine learning”, etc… to get funded

AI long term history : A series of winters and hype (2)

"Classic Machine Learning" models development – 2000s

- No hype any more, but a series of simple techniques developed to successfully solve a limited, well-defined set of problems : Support-Vector Machine, Random Forest, collaborative filtering, etc …
- Early 2010s : no progress made any more. Neural Networks take the lead.

Neural Network for Image recognition hype – 2010-1017

- 2006-2009 : Stanford Professor Fei Fei Li and team, in collaboration with other universities, build ImageNet, a large hand-annotated images database, and make it public
- 2010-2017: ImageNet challenge ILSRVC allows image recognition top-5 error rate to dramatically drop from 28% to < 3%, less that human error rate
- 2012 : AlexNet, 6-layer NN, won with 16% error rate. Second candidate got 26%! From now on, only NN submitted

More examples

- Self-driving cars
- Big data
- Chatbots
- **GAN**
- **…….**

What Impact will have GenAI?

Where to work?

Companies typology

- Magnificent Seven – Ex GAFAM. You may end up being just a pawn in the game. But very valuable reference for your future carrer.
- Other techs using AI. Mainly startups, but some few unicorns. Probably the most interesting companies to work in, but apply carefully your criteria.
- E-commerce, advertising industry. Very data driven. Also generally very interesting to work in.
- Large banks, insurance, financial companies. Innovation is opposite to their culture.
- Consulting. Very similar culture to their main clients', large financial and industrial companies.
- Industrial companies. All depends on team leader. Usually difficult to get recognized. Managers often feels they need AI but don't understand why and how to use it.
- Government research. More freedom, but often difficult to get enough data and computing power.

Your criteria

- Growth industry with constantly evolving technologies and businesses. You have to **make bets**.
- The **leaders**. Go were both the investor and the strategist understand what they are doing, i.e. share a **realistic vision** on the company, with the right **business model**, and building a **collaborative working environment** to promote creativity and innovation.
- Open-source/freeware can be a competitor. Check that the company added-value is worth paying for it. Added-value aiming at **increasing user productivity** rather than replacing her/him is more accurate.
- Make sure the environment helps you to **constantly learn and produce**. This way your market value improves.

Personal strategy to attract employers' attention (1)

Make use of your maths abilities

- **Barriers to entry are very low**. Any person with very light maths understanding can use, even manage an AI application. Although less people, many still can program an AI application.
- **Produce personal, original programs, algorithms, models**. It's one of the best way to learn, and gives you some credibility.
- Always be able to **go deep in the algorithm "belly"**, challenging common practice, trying to understand why it does not work as expected and improving it. Most of AI programmers download a github code, adapt to ther Python or Tensorflow version and try it. If it does not work as expected, they just drop it and look for another one on github.

Get specialized in some Deep Learning building blocks

- Convolutional Neural Networks, Transformers, …
- Reinforcement Learning?, …

Get specialized in some Deep Learning weaknesses

- **Data quality check** is boring to most of the people, but it is crucial for model training.
- **Semi-supervised or weak supervision** Learning, **Transfer Learning**. Data labeling is expensive.
- **Training process** is very inefficient.
    - **Gradient descent** algorithm is very basic, at the end of the day.
    - Cold start is a classic problem
    - Local minima are a serious issue, as we try to minimize a one-dimension error function

Personal strategy to attract employers' attention (2)

Communicate

- **Publish your AI production**. On github, leaving your code as open source, but also on X/Twitter, LinkedIn, even TowardsDataScience. Be also active in AI frameworks users' forums and in Stack Overflow.
- **Compete in Kaggle challenges**.
- Go to **AI conferences and meetups**. Even though most of the time is wasted, it helps you understand who's who and build your network.

Scientific watch is an absolute necessity

- In AI with such a rapid growth, diploma matters far less than knowledge and above all know-how
- Excellent courses from best universities available for free on Youtube. But you miss exercises and professors feedback on your work.
- Plenty of new material available on Youtube and internet in general, literally everyday.

The state of AI in 2024

- Definition and history
- AI practical cases for financial markets
- Who is doing what in the AI industry?
- Some advice from an old geezer

Users initially enthusiastic, then quickly frustrated

Let's assume that no model = 50% good results

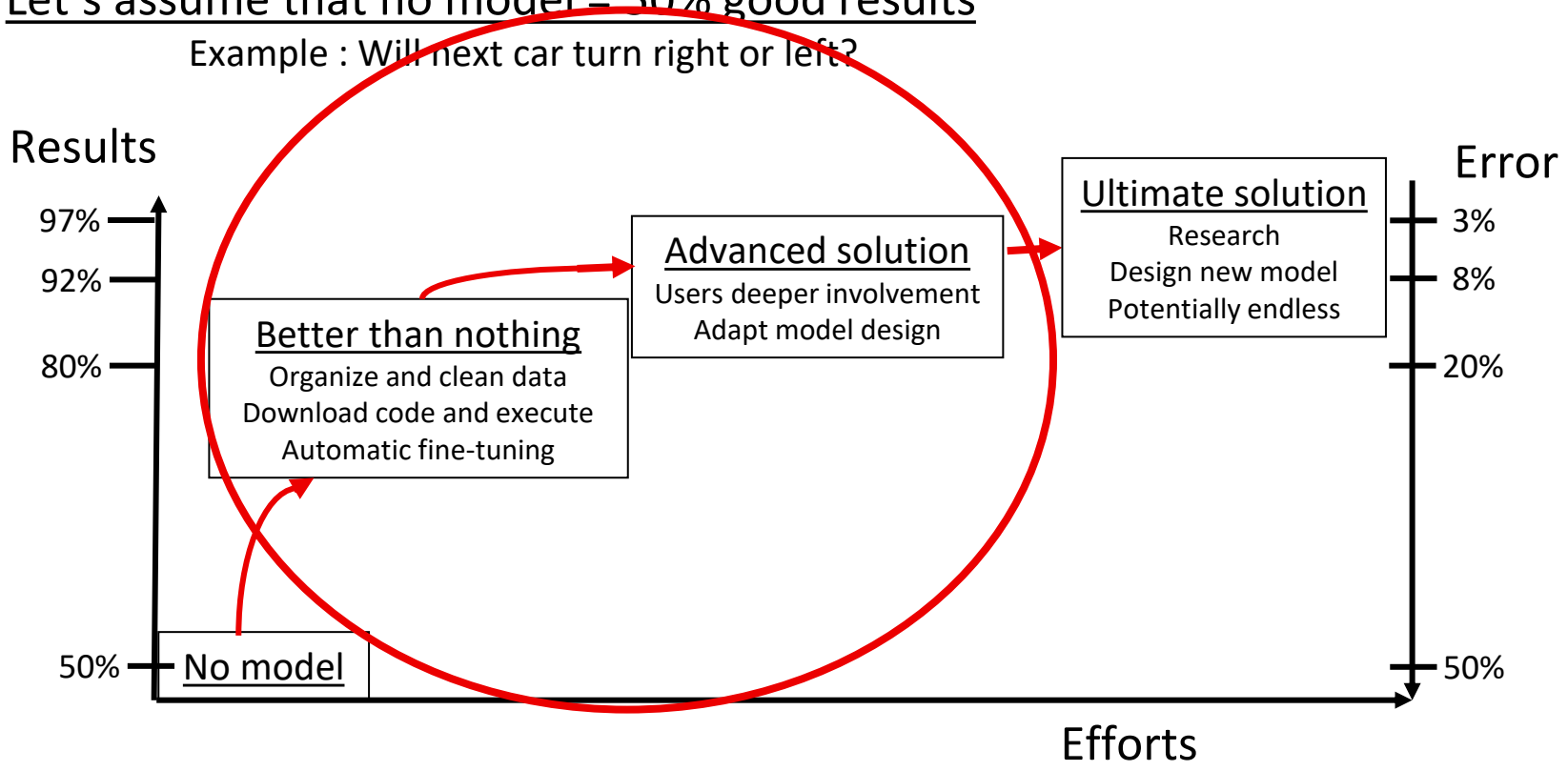
Example : Will next car turn right or left?

- **Initial result seems extraordinary**. 80 % accuracy!! You are the king.
- After few weeks, the **user focuses on the 20 % failure** rate. Pressure is on you.
  - => always try to improve the model
  - => always interact with user to better understand how she/he uses it

Find the optimal strategy

Let's assume that no model = 50% good results

Example : Will next car turn right or left?



Key to success

- Opensource frameworks (ScikitLearn, Tensorflow, Keras, PyTorch,…) allow you to program a first "better-than-nothing" program that works
- For each problem, correctly position the desired solution on this figure

Mainly depends on users' tolerance to error

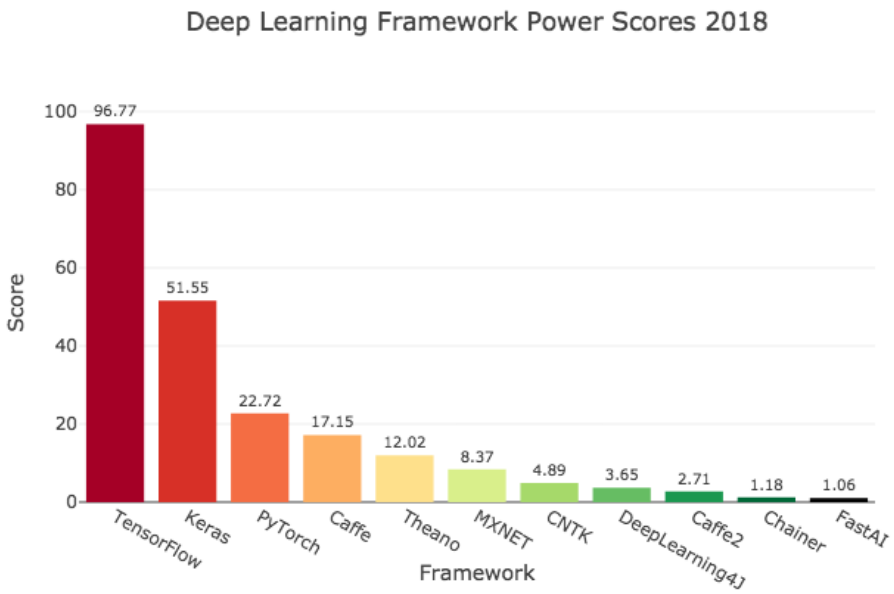
Carefully choose your framework(s)

Why use a ML/DL framework?

- A collection of high- and medium-level functions to easily program ML and DL models
- Allows to program a NN that gives a result in 20-40 lines. Later, weeks of work needed to improve your NN.
- Above all, computation adaptation to your GPU/TPU/hardware come for free ….
- … together with automatic differentiation, absolutely needed for gradient descent training

History + current situation/panorama

- Started with R, then evolve into Python (standard)
- ML framework: leader = **ScikitLearn**. Integrated in DL frameworks
- DL frameworks initially developed by top universities in early 2010s, then reshaped and improved by GAFAs
- **Tensorflow** (Google) is leader. Version 2.0 released in Q1 2019
- **PyTorch** (Facebook) more flexible, used in research and universities. Caffe2 (Facebook) more adapted for production
- **Keras** (Google) on top of Tensorflow, simpler and more intuitive. Ad.hoc version now included in Tensorflow 2.0
- MxNet (Amazon), CNTK (Microsoft), Gluon (API on top of MxNet and CNTK), Paddle (Baidu)
- Natural Language Processing (NLP) specific framworks: **NLTK** is "older", still more complete, **SpaCy** is more recent, **Hugging Face**, …
- NLP pre-trained models, not frameworks : **BERT** (Google), **GPT-3** (OpenAI), …
- Opensource, and many contributors



https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a
https://skymind.ai/wiki/comparison-frameworks-dl4j-tensorflow-pytorch

Build your own AI setup

Hardware

- Own hardware, or cloud?
- Nvidia graphic card minimum for Deep Learning, even TPU if you can afford
- Google Colab to start with

Data

- If own data is enough, you are lucky
- First check if generate your own additional data is feasible
- If no data or not enough, use classic dataset for research purpose. Many available online (GitHub, Kaggle, universities, …) or in frameworks

Working remotely is tricky

- 28 year experience in designing, developing, releasing and maintaining models with cross-border teams
- Build a team spirit is difficult without physical presence.
- Way more difficult to integrate new team members, or generate new ideas and start new projects.
- More possible for dull, repetitive tasks. But AI is particularly efficient for these tasks. Outsourcing opportunities should be shrinking and outdated. Pure accounting savings initially, but soon economic loss
- Probable optimal scheme : mixed presential/remote periods, with frequent gathering
- But a mixed presential/remote team is very dangerous. Bad talking is too easy at the coffee machine.
- When things get wrong, first out

Effort balance to develop models

Stochastic model in market finance

- 10-20% on theory and design. **Key factor**
- 50-60 % programming
- 30% interaction with users

AI models

- 40% design (theory = 0%)
- 20 % programming
- 40% interaction with users. **Key factor**

Facts on models

- **Life is unfair**. Some easy developments generate spectacular results, and some painful research and development efforts lead to confidential results, difficult to sell.
- **No warranty on new models**. Innovation requires a lot of trial and error. For management the risk is to spend money on a team of lazy clowns, or even if the team is bright, to eventually get a disappointing result.

Conclusion

Never be scared or too impressed

- AI Gurus constantly make (plenty of) mistakes
- Hierarchy often does not understand
- Humble, measured, really experienced experts are more interesting to listen too
- **Managing expectations** is key**.**

You are facing a unique opportunity in world history

- Shortage of mathematicians.
- Mathematical models are at the chore center of the most invested-in economic sector