TRAITEMENT DE LA PAROLE

RECONNAISSANCE DU LOCUTEUR

Le Mans Université

Anthony Larcher - Le Mans Université - 2019

RÉSEAUX DE NEURONES EN RECONNAISSANCE DU LOCUTEUR

- Prémices en 2012
- Premiers résultats positifs en 2014
- Deuxième résultat en 2016
- État actuel:
  - systèmes basés sur le paradigme des i-vecteurs
  - Premiers systèmes entièrement neuronaux (end-to-end)

PLAN DU COURS

- Apparition de réseaux de neurone dans le paradigme i-vecteurs
- X-vecteurs
- Système end-to-end
- Fonctions de coût discriminantes

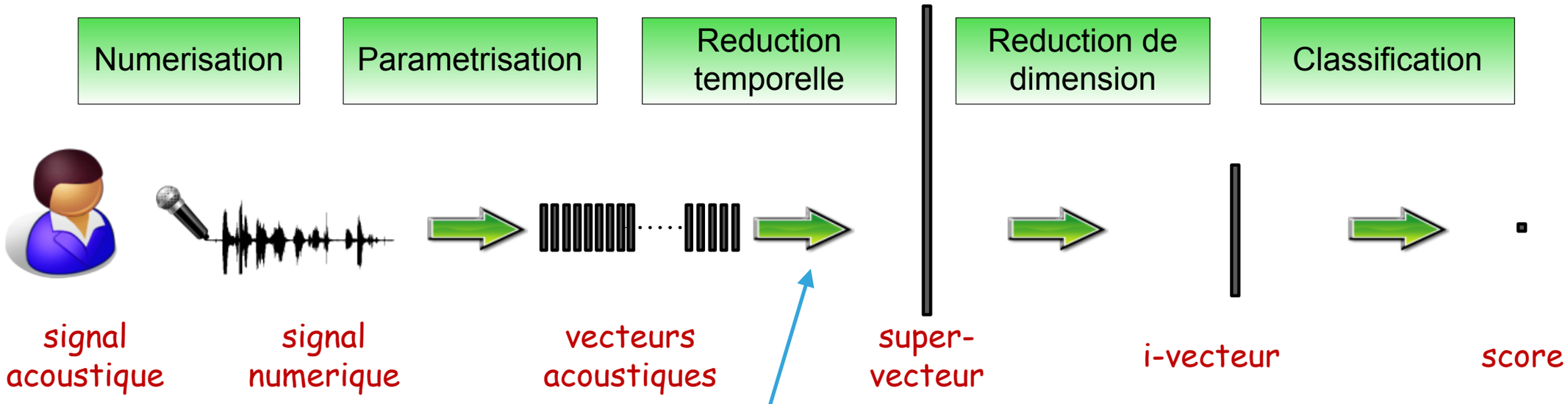
TRAITEMENT DE LA PAROLE

RAPPEL SUR LES I-VECTEURS

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

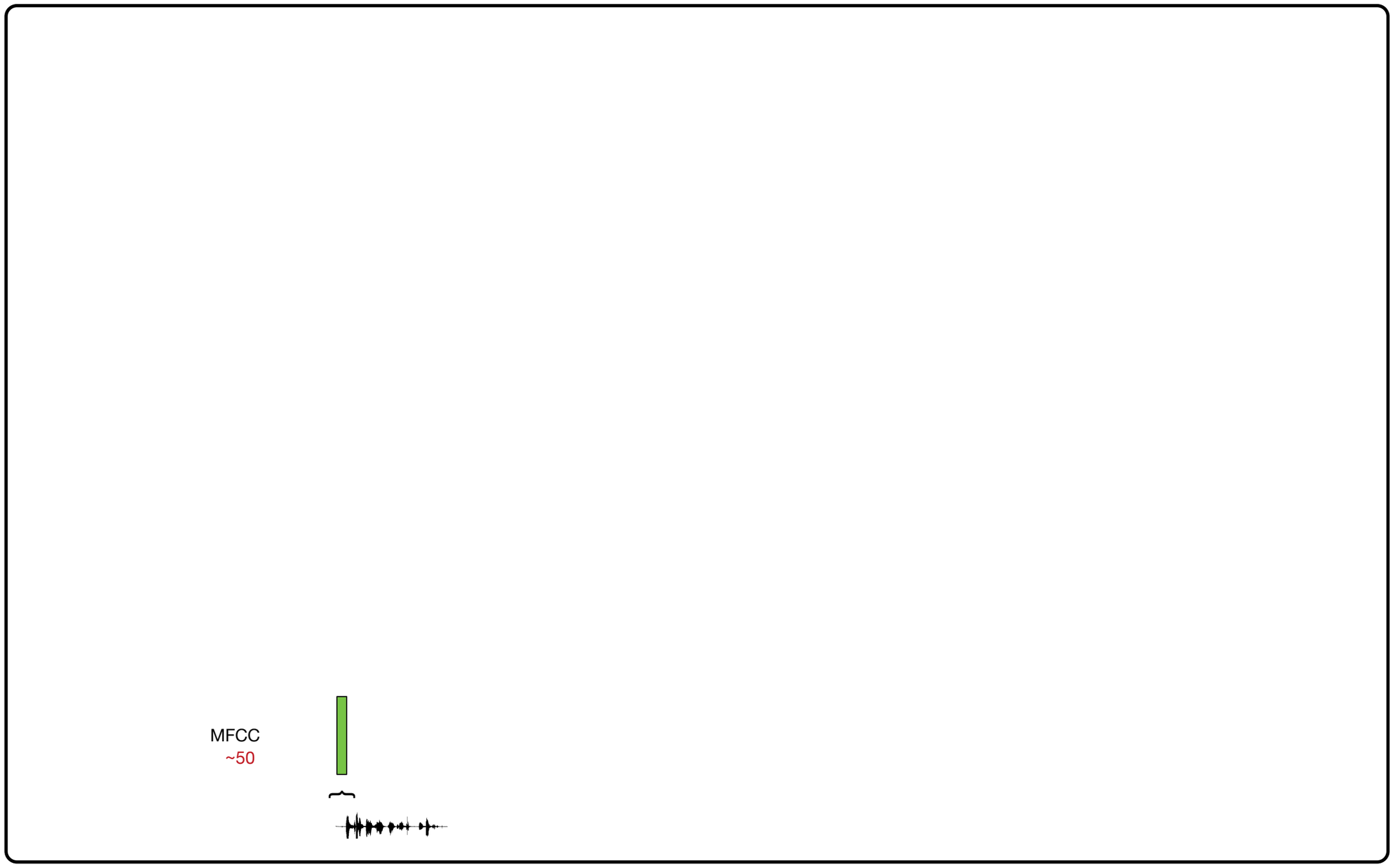
RAPPEL SUR LES I-VECTEURS



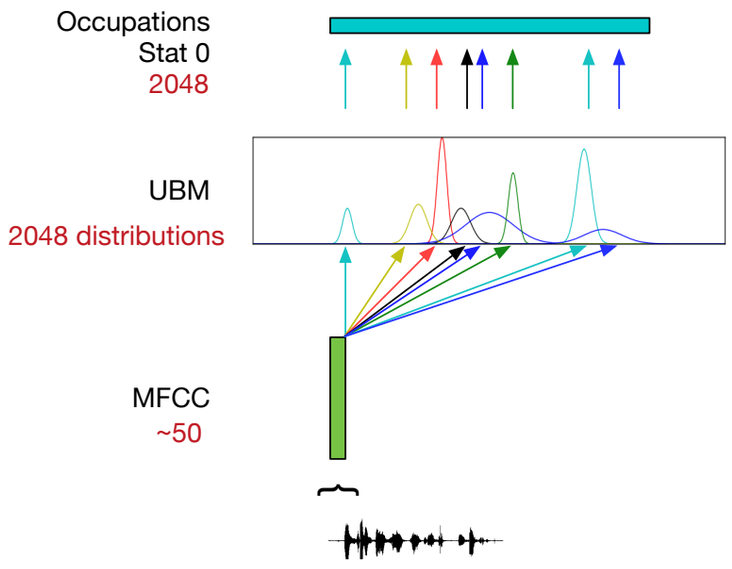
Mixture de Gaussiennes (GMM)
Statistiques d'ordre 1 et 2 dans un Factor Analysis

RAPPEL SUR LES I-VECTEURS

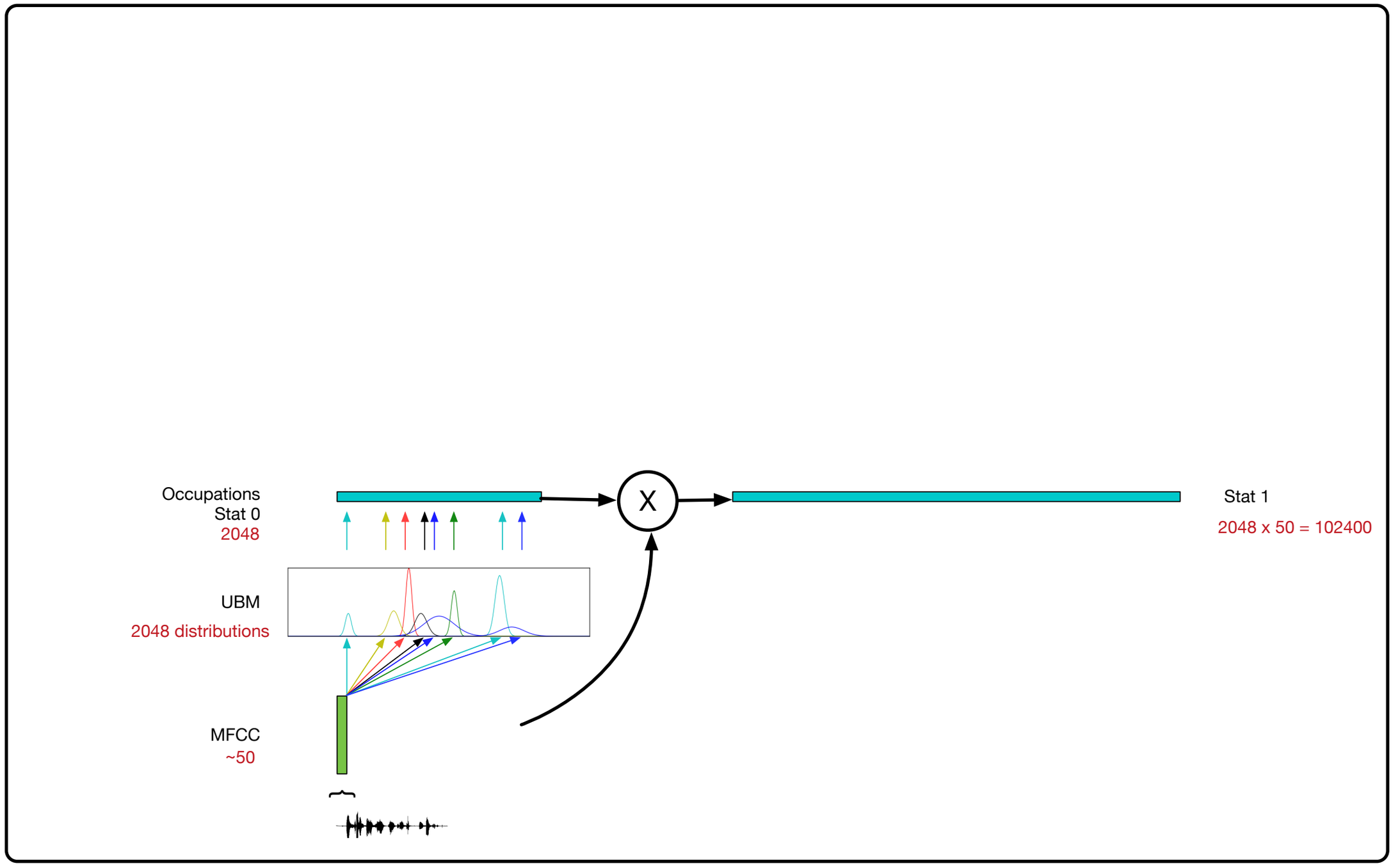
- état de l'art 2014


MFCC
~50

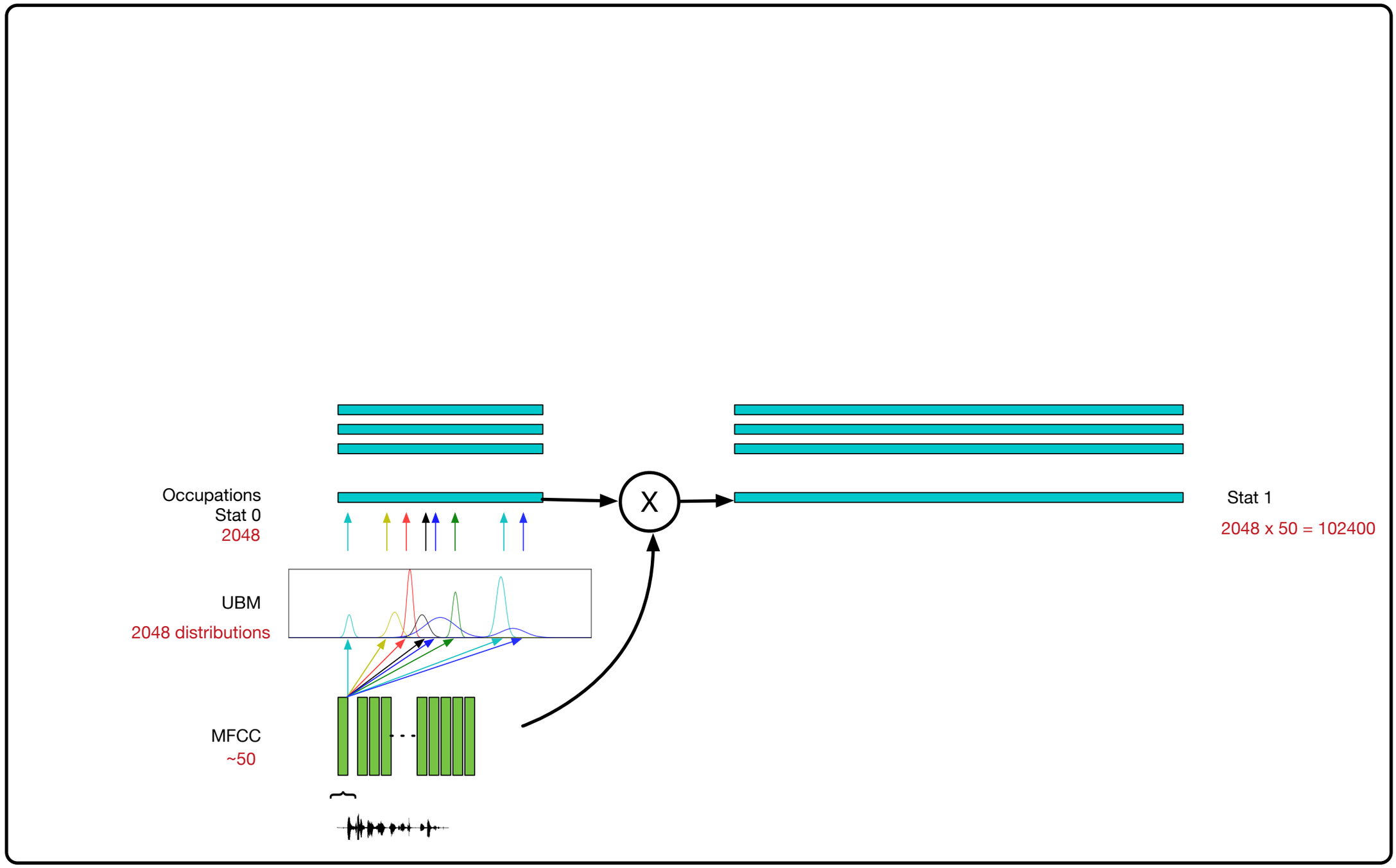
TRAITEMENT DE LA PAROLE

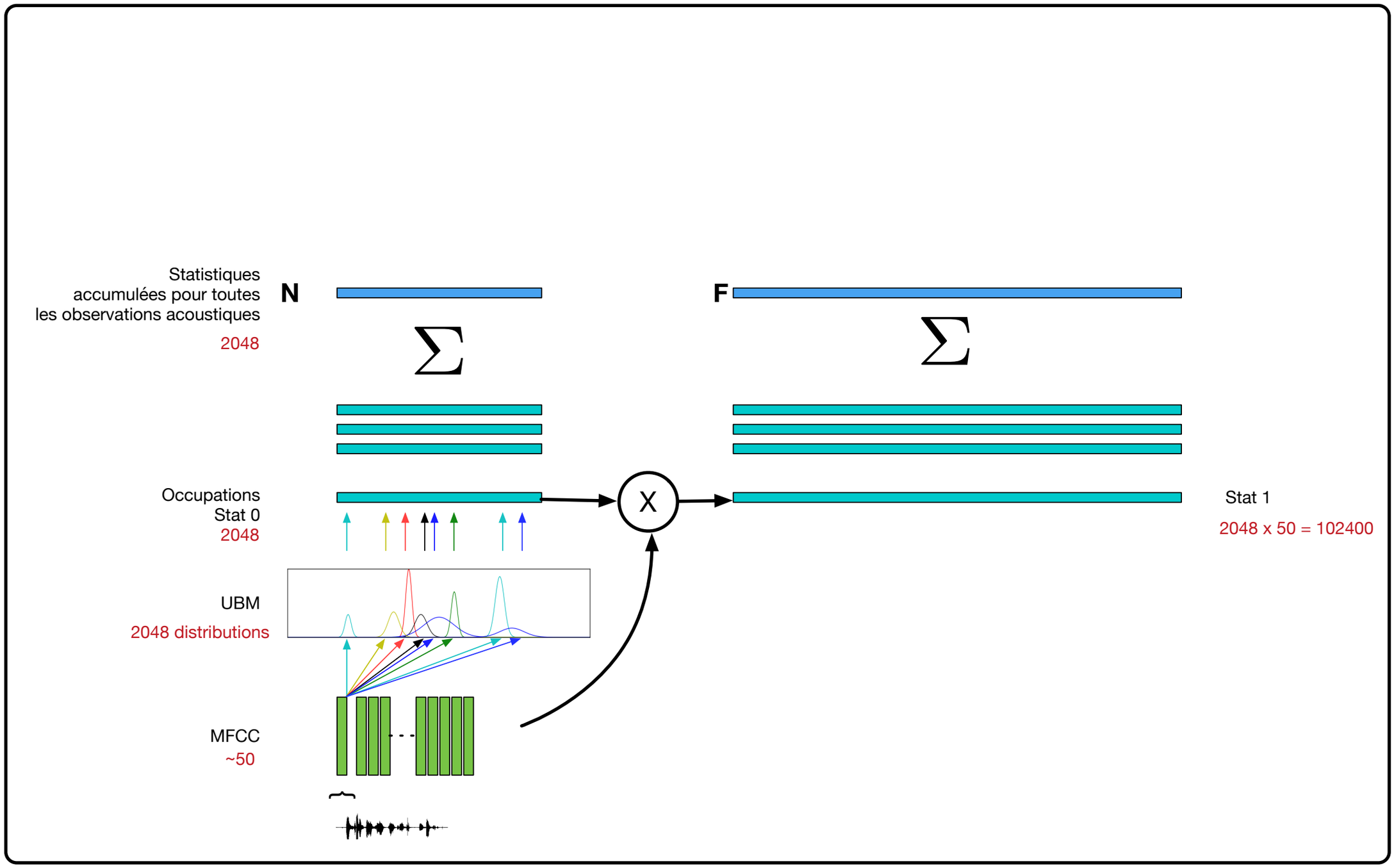


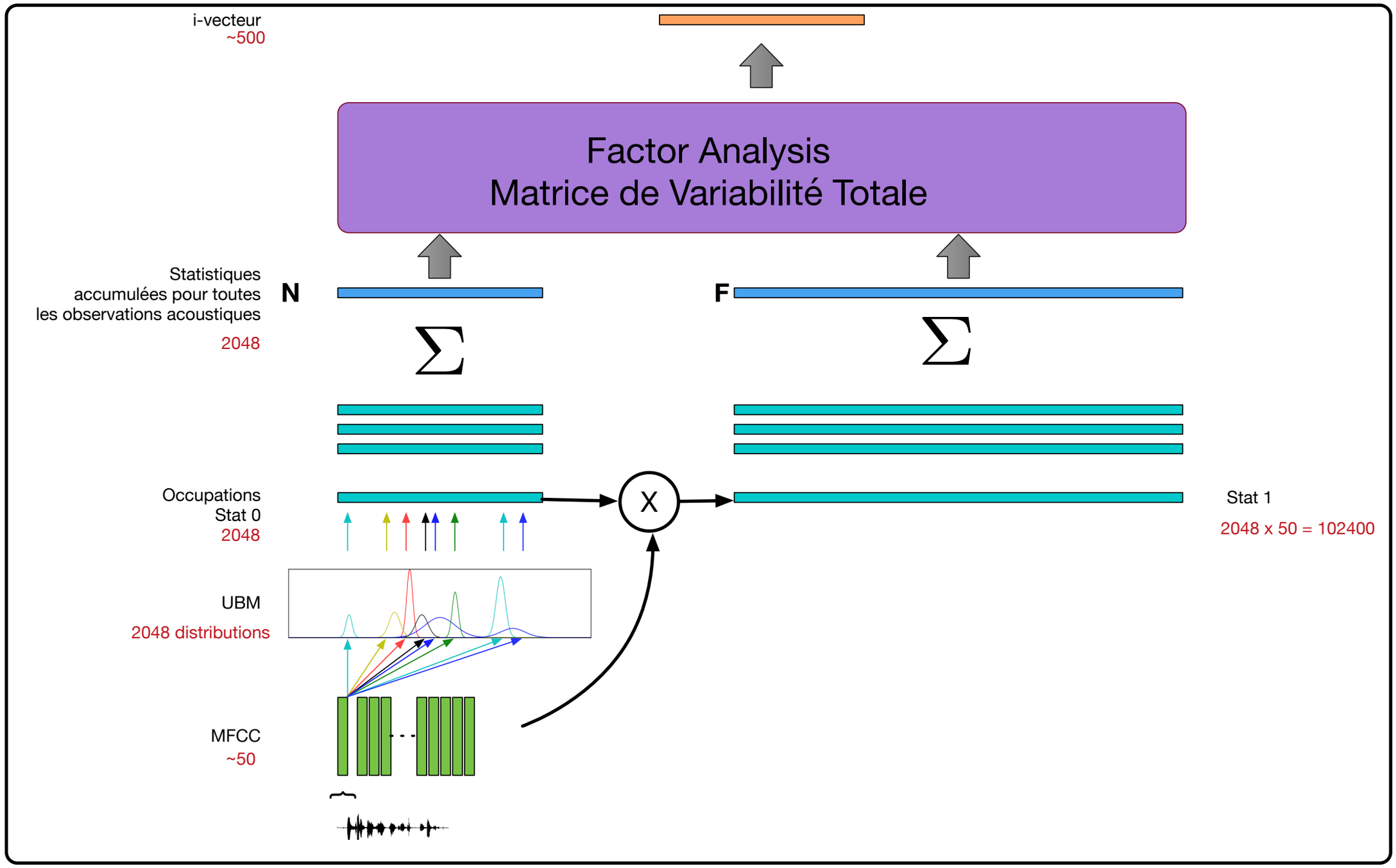
TRAITEMENT DE LA PAROLE

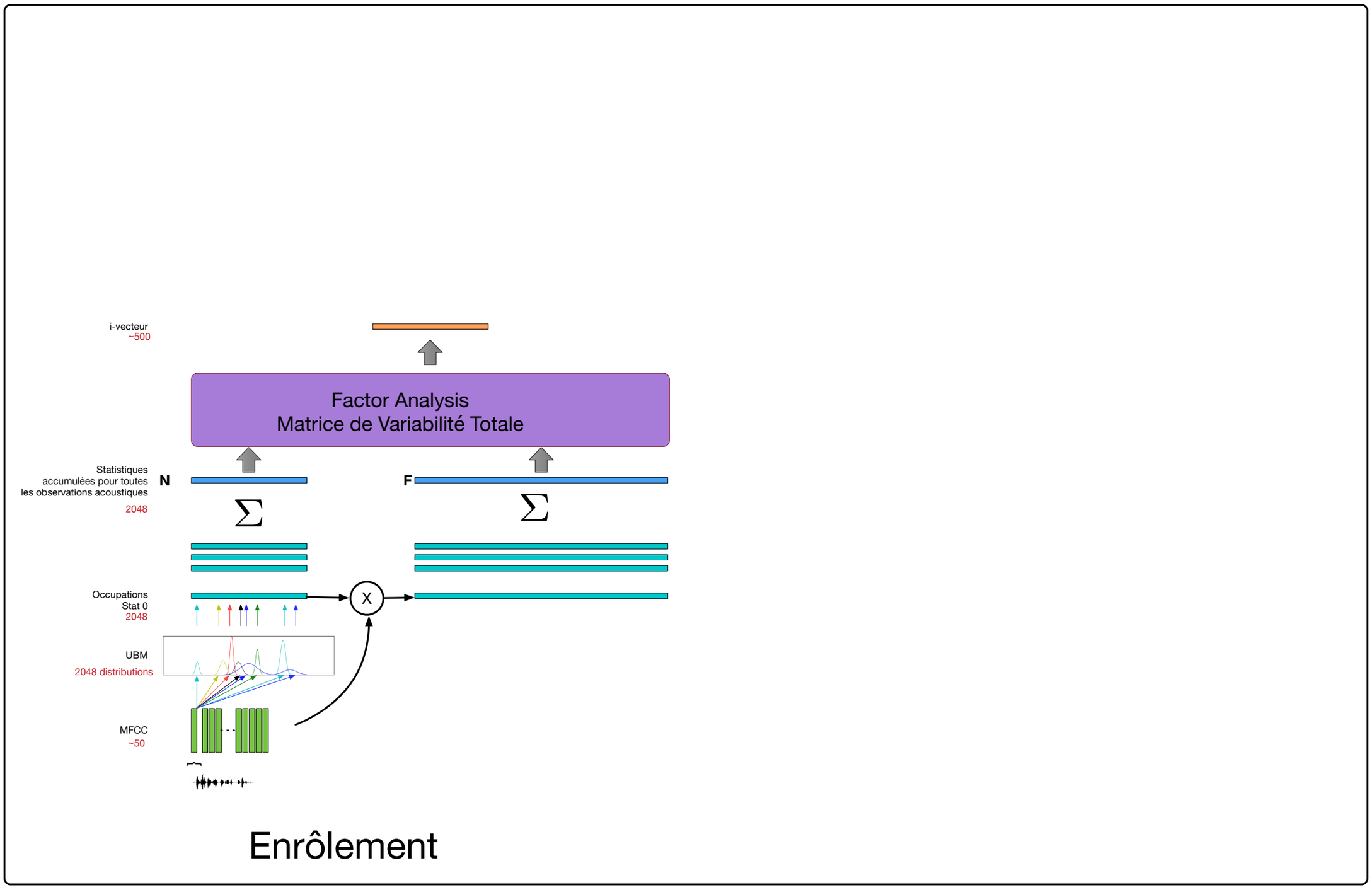


TRAITEMENT DE LA PAROLE

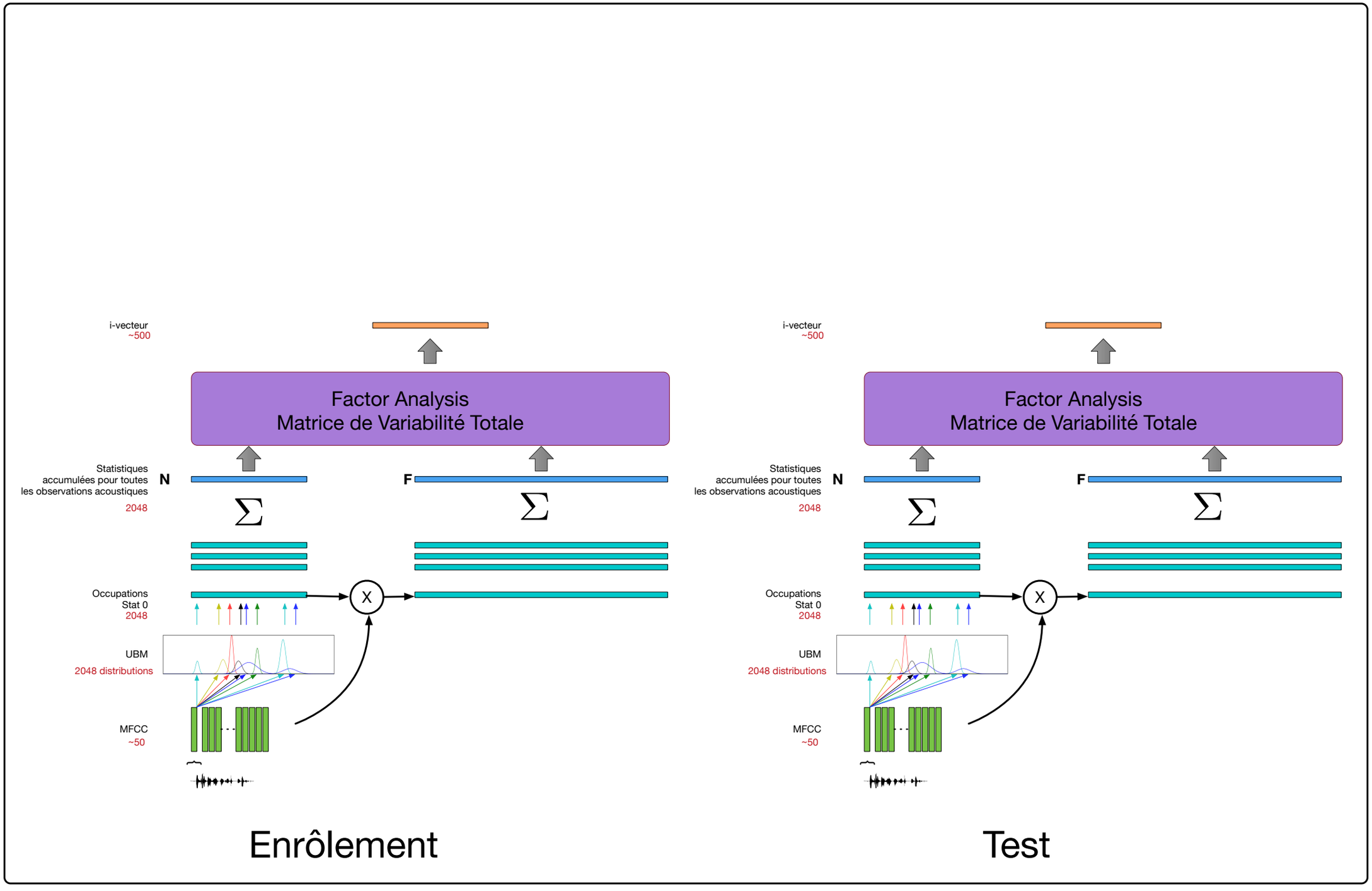



Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Occupations
Stat 0
2048
UBM
2048 distributions
MFCC
~50
X
Stat 1
2048 x 50 = 102400

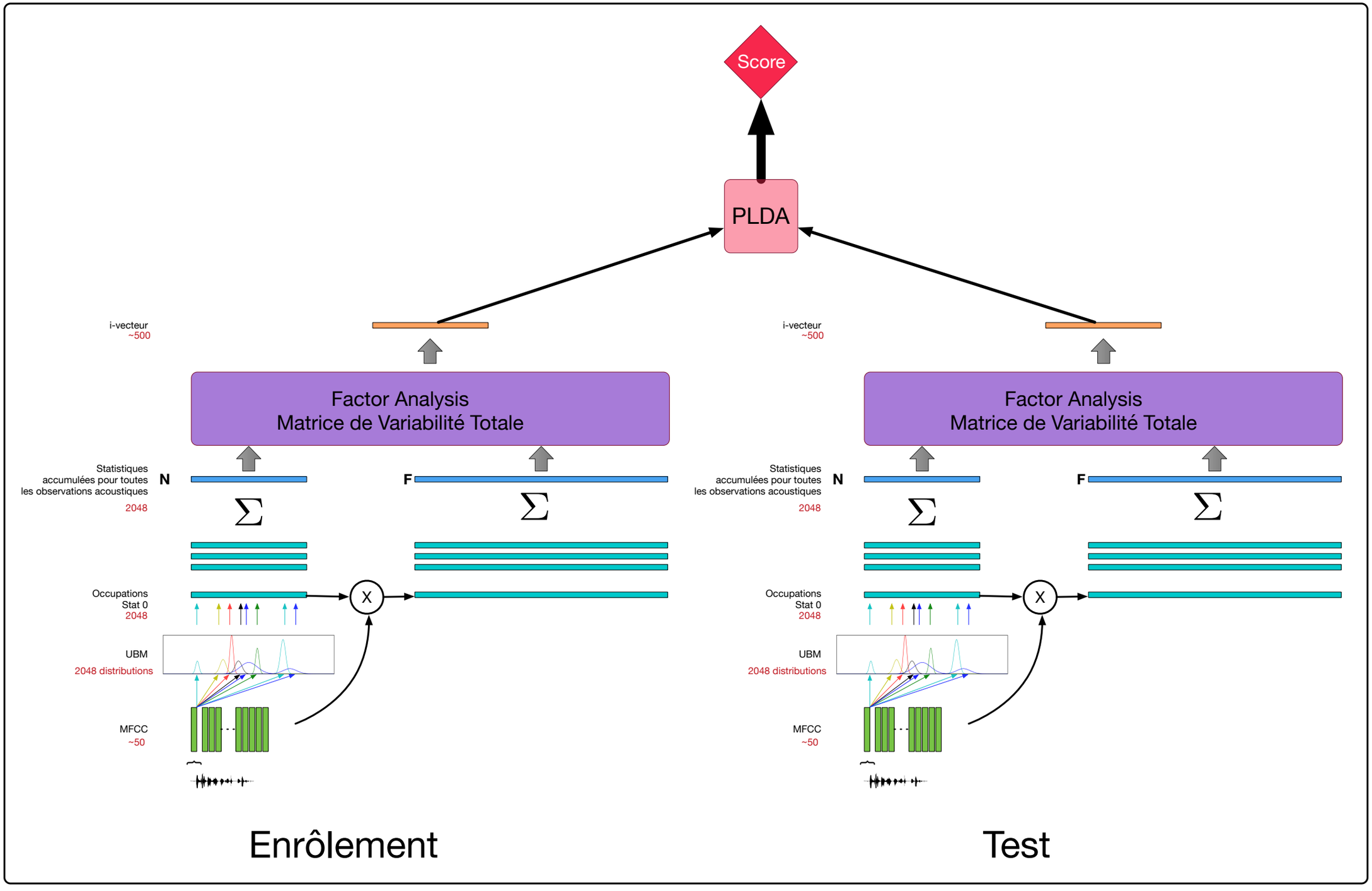

i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
X
Stat 1
2048 x 50 = 102400
UBM
2048 distributions
MFCC
~50


i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
X
UBM
2048 distributions
MFCC
~50

Enrôlement


i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
UBM
2048 distributions
MFCC
~50
X
Enrôlement
i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
UBM
2048 distributions
MFCC
~50
X
Test

TRAITEMENT DE LA PAROLE



TRAITEMENT DE LA PAROLE

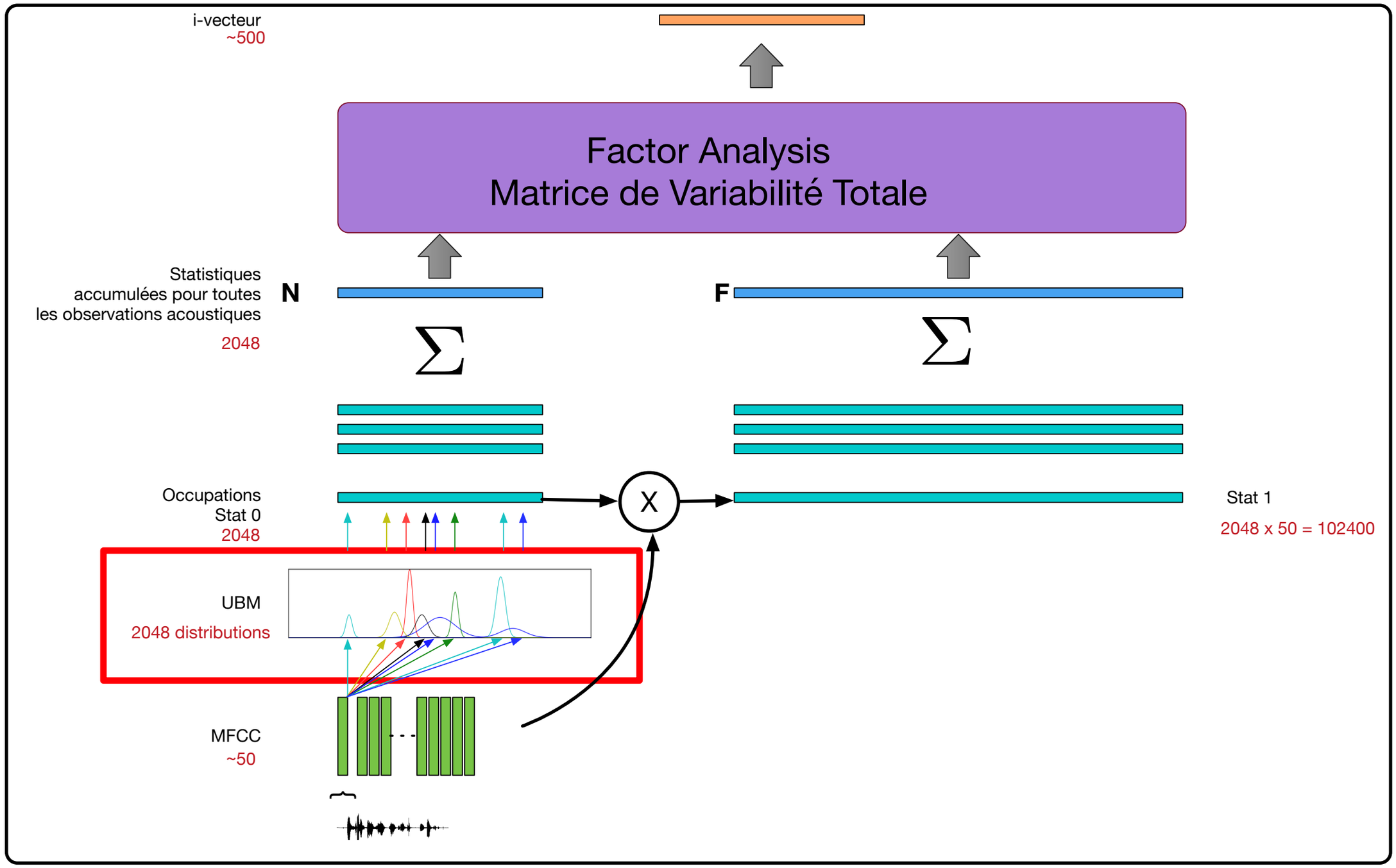
1 DNN POUR REMPLACER LE UBM

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

RÉSEAU DE NEURONES PROFOND POUR REMPLACER LE UBM

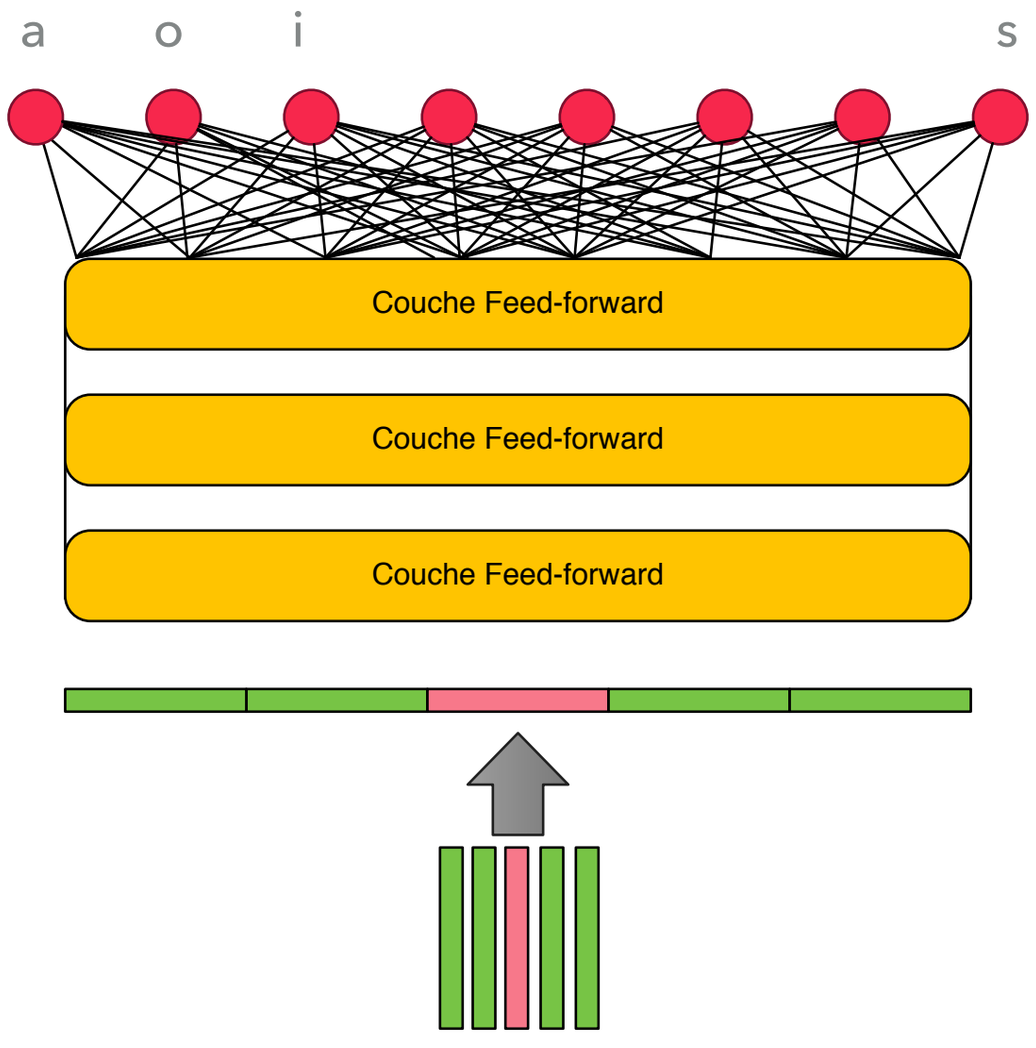
- le modèle du monde (UBM) sert à classifier les vecteurs acoustiques dans l'espace
- un DNN serait plus précis (approche discriminante)
- utiliser une approche phonétique: système venant de la transcription de parole

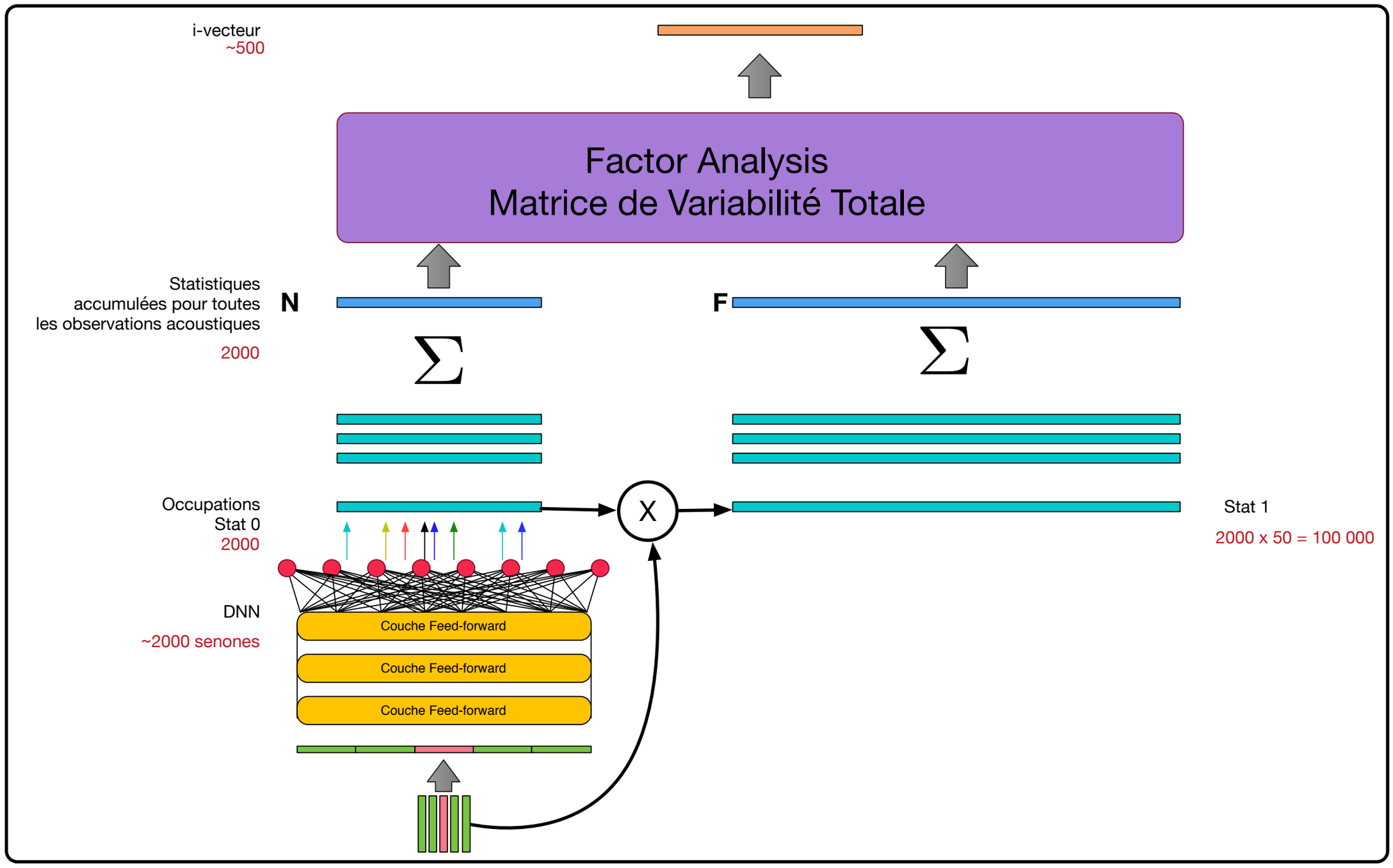

i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
X
Stat 1
2048 x 50 = 102400
UBM
2048 distributions
MFCC
~50

DNN POUR LA TRANSCRIPTION DE PAROLE

1 classe par phonème (ou phonème en contexte)

Chaque vecteur acoustique + son contexte est classifié




i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2000
N
F
Occupations
Stat 0
2000
DNN
~2000 senones
Couche Feed-forward
Couche Feed-forward
Couche Feed-forward
X
Stat 1
2000 x 50 = 100 000

RÉSEAU DE NEURONES PROFOND POUR REMPLACER LE UBM

UBM-DNN: –30 % de taux d'erreur par rapport à un système i-vecteur utilisant une mixture de Gaussiennes comme modèle du monde

TRAITEMENT DE LA PAROLE

1 DNN POUR EXTRAIRE DES PARAMÈTRES

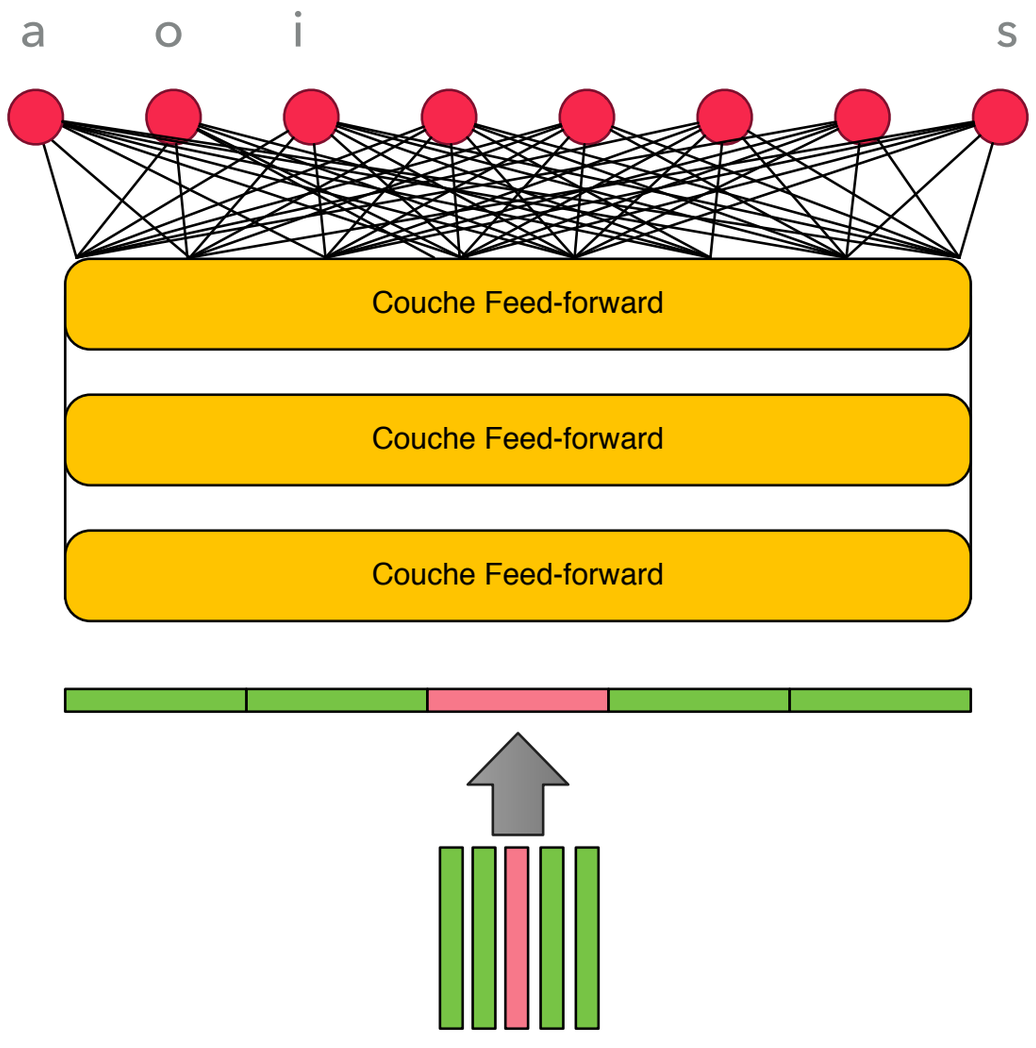
Le Mans Université

Anthony Larcher - Le Mans Université - 2017

DNN POUR LA TRANSCRIPTION DE PAROLE

1 classe par phonème (ou phonème en contexte)

Chaque vecteur acoustique + son contexte est classifié

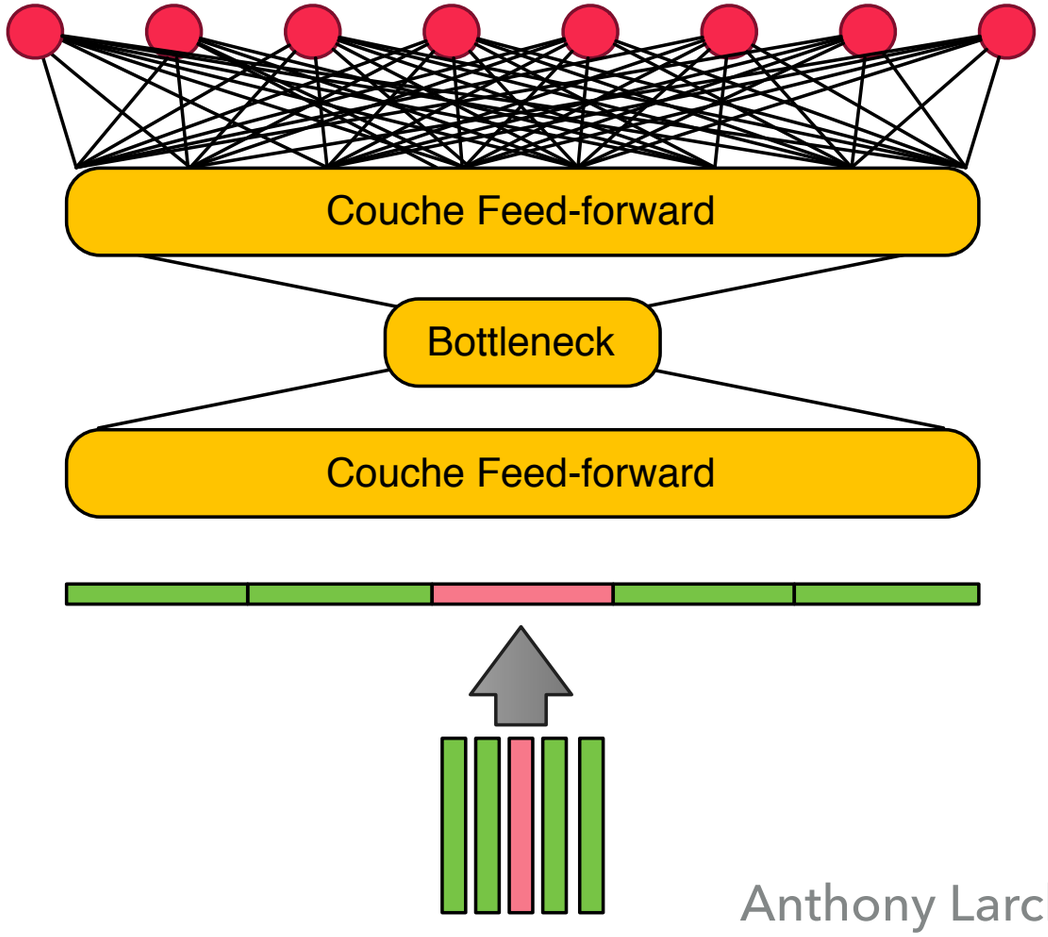


DNN POUR LA TRANSCRIPTION DE PAROLE

1 classe par phonème (ou phonème en contexte)

Chaque vecteur acoustique + son contexte est classifié

On intègre une couche « bottleneck »

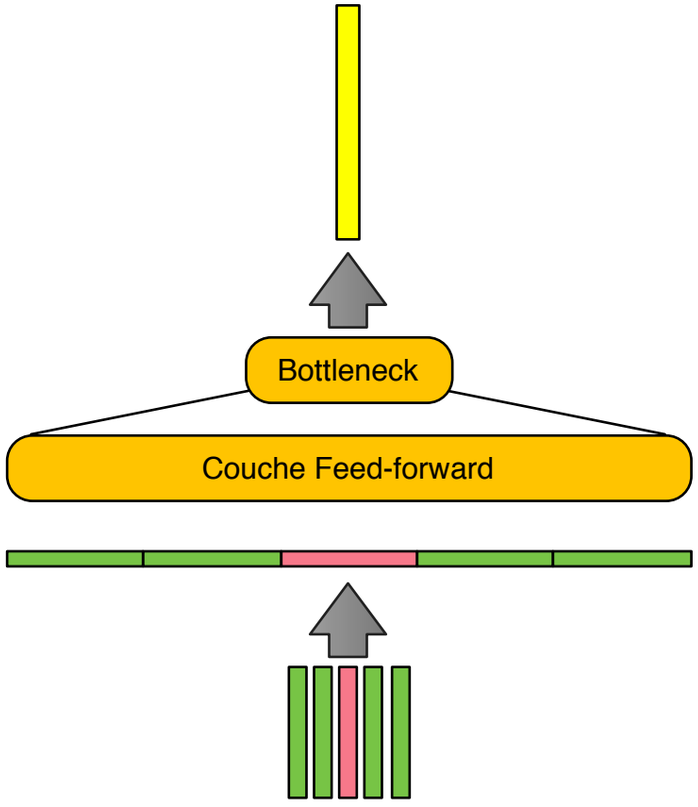


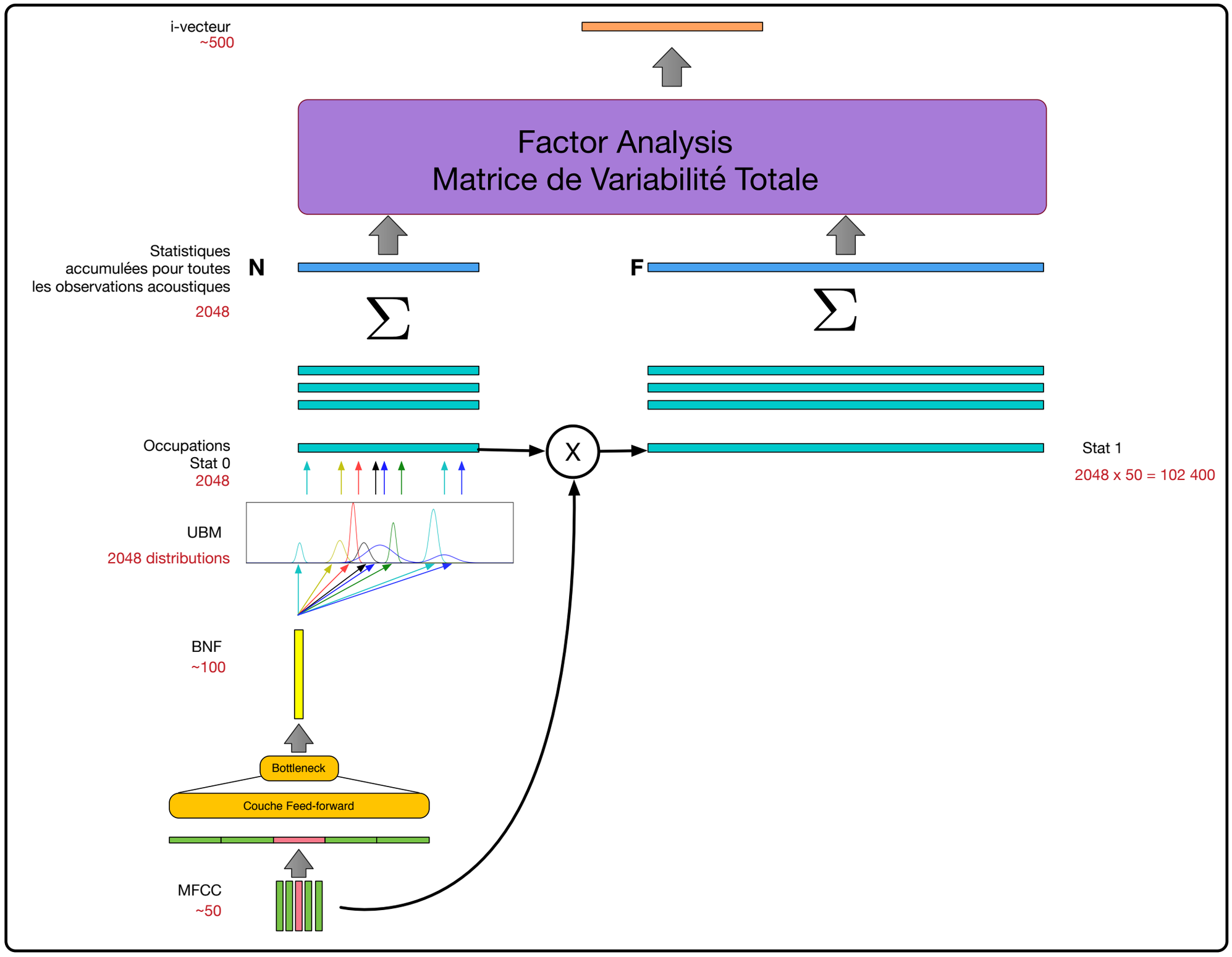
40 < dimension < 150

DNN POUR EXTRAIRE DES « BOTTLENECK FEATURES »

DNN appris pour classifier les phonèmes

Sensé supprimer la variabilité locuteur




i-vecteur
~500
Factor Analysis
Matrice de Variabilité Totale
Statistiques
accumulées pour toutes
les observations acoustiques
2048
N
F
Σ
Σ
Occupations
Stat 0
2048
X
Stat 1
2048 x 50 = 102 400
UBM
2048 distributions
BNF
~100
Bottleneck
Couche Feed-forward
MFCC
~50

RÉSEAU DE NEURONES PROFOND POUR EXTRAIRE DES FEATURES

UBM-DNN: –30 % de taux d'erreur par rapport à un système i-vecteur utilisant une mixture de Gaussiennes comme modèle du monde

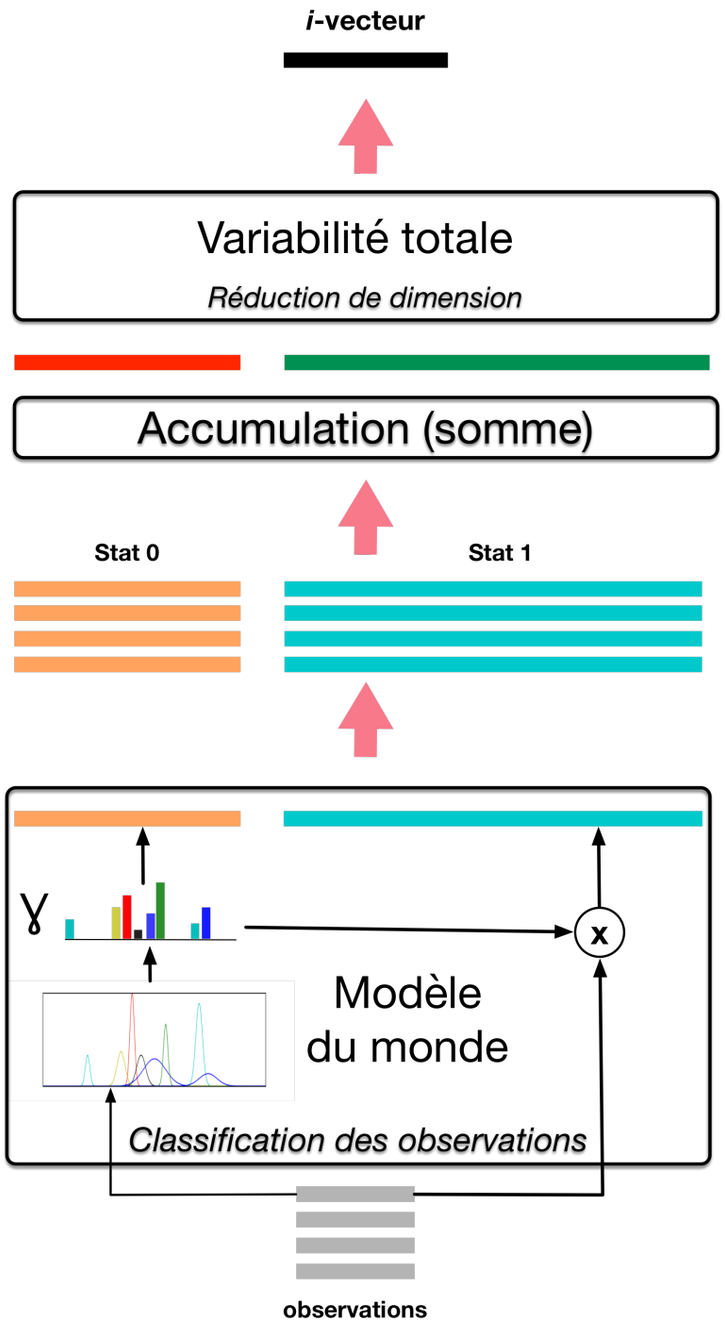
TRAITEMENT DE LA PAROLE

X-VECTEURS

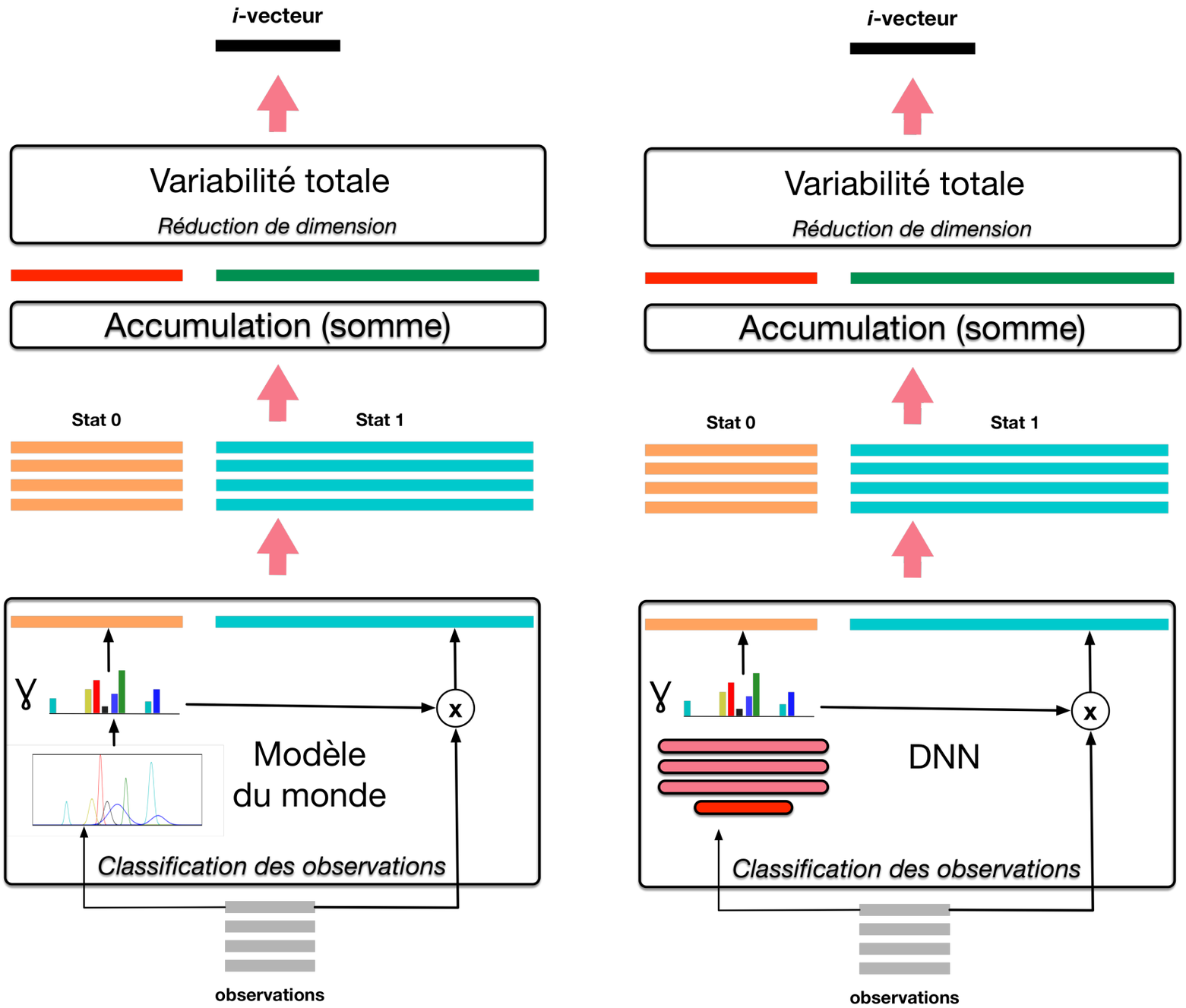
Le Mans Université

Anthony Larcher - Le Mans Université - 2017

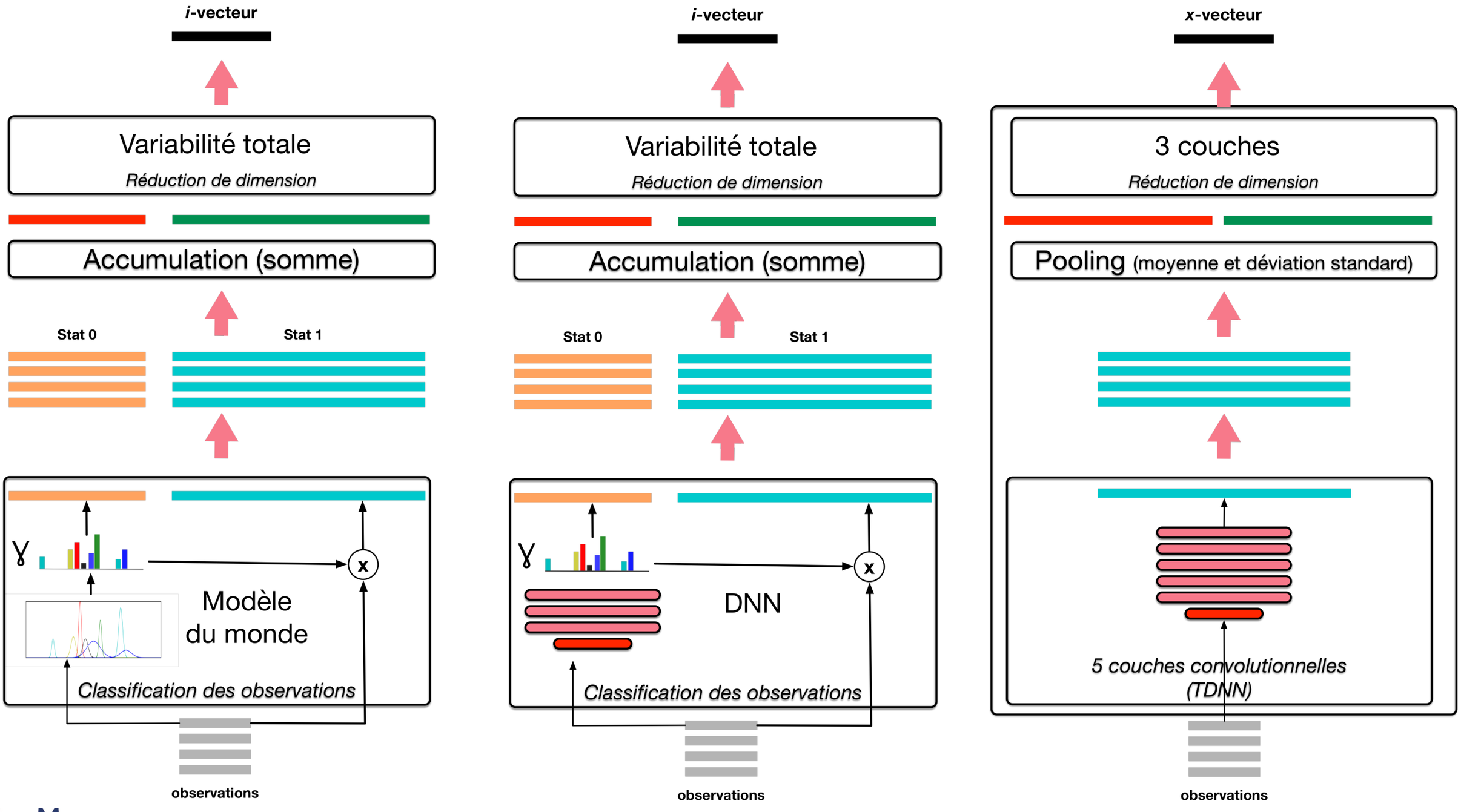
X-VECTEURS: LE RÉSEAU



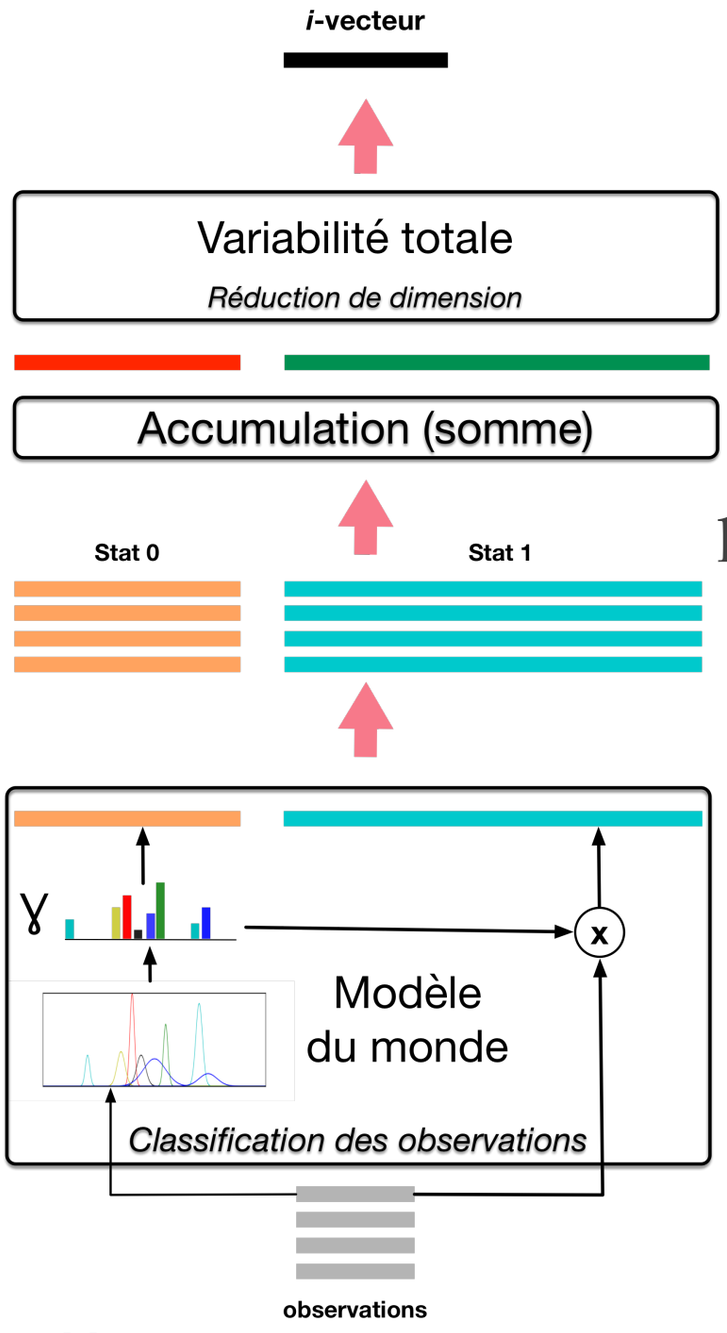
X-VECTEURS: LE RÉSEAU



X-VECTEURS: LE RÉSEAU

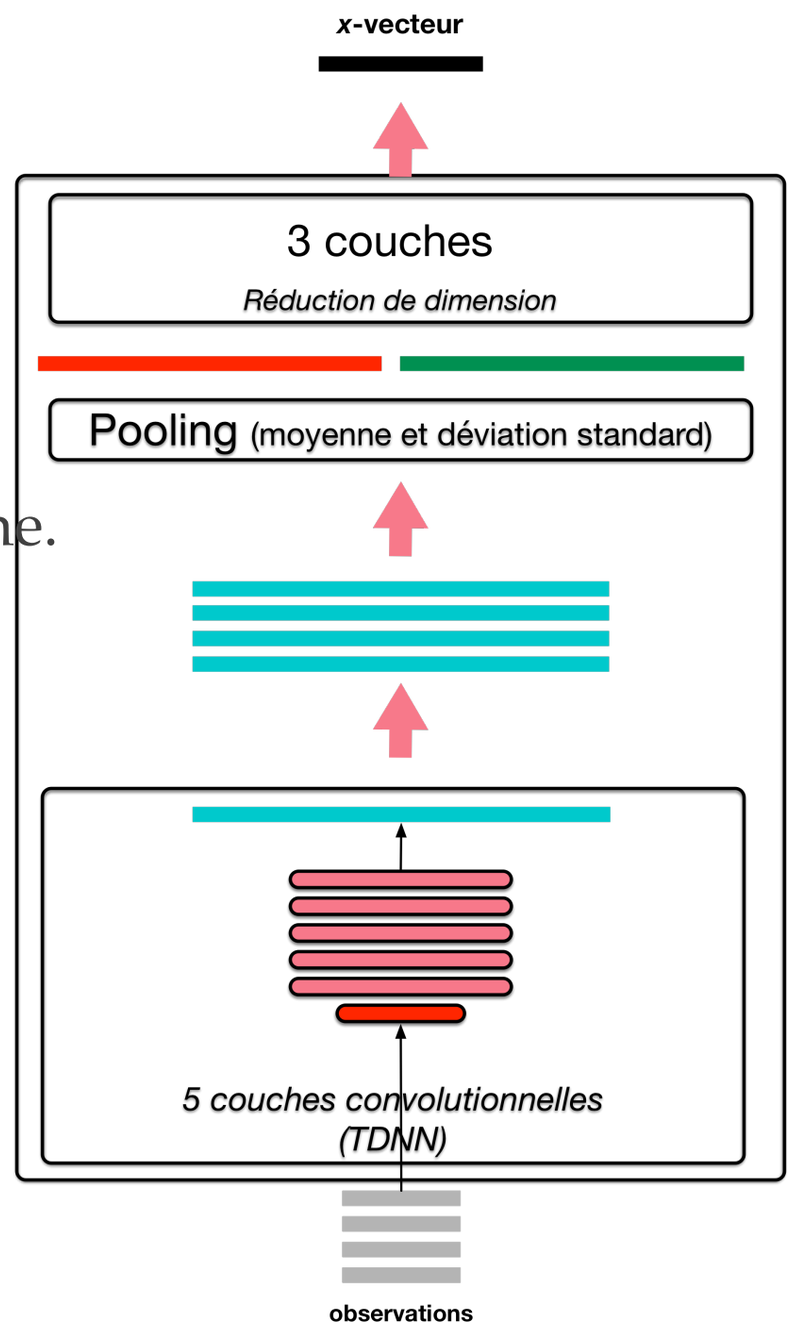


X-VECTEURS: LE RÉSEAU

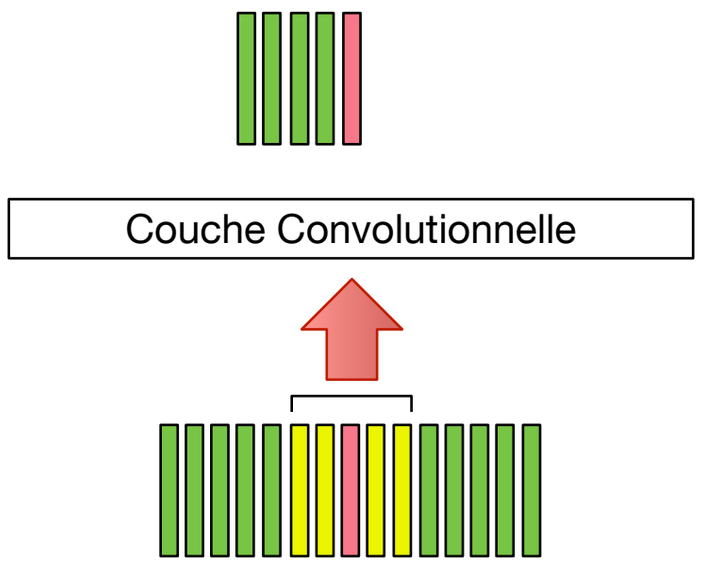


Même structure
mais
possible changement de paradigme.

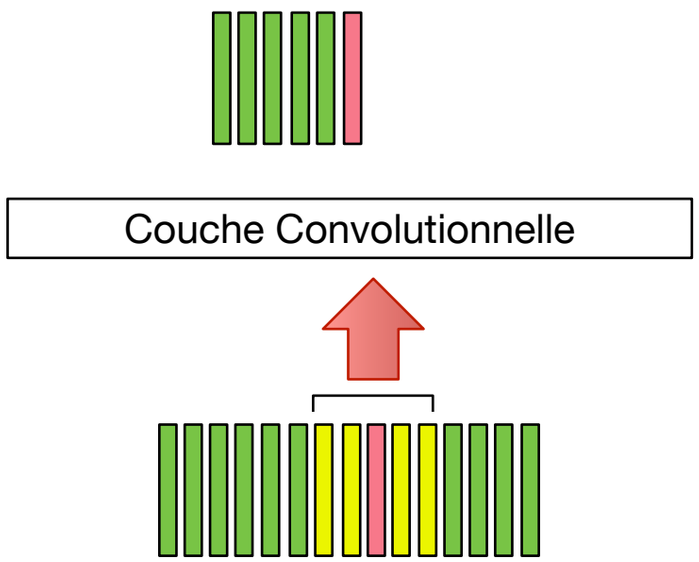
Les statistiques d'ordre 0 ne sont plus explicites!



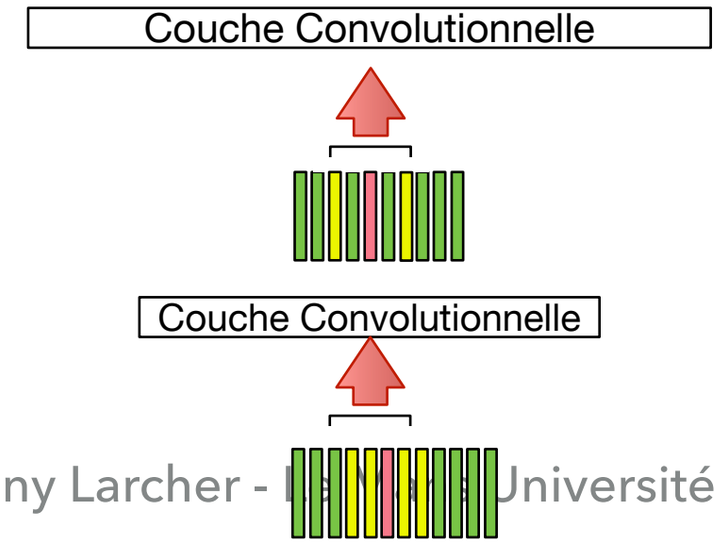
X-VECTEURS: LE RÉSEAU



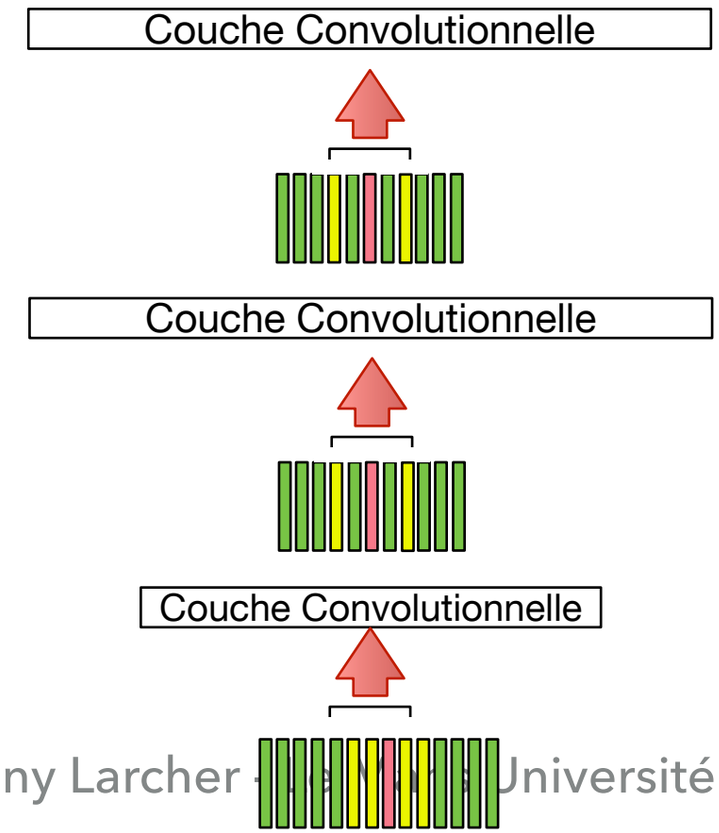
X-VECTEURS: LE RÉSEAU



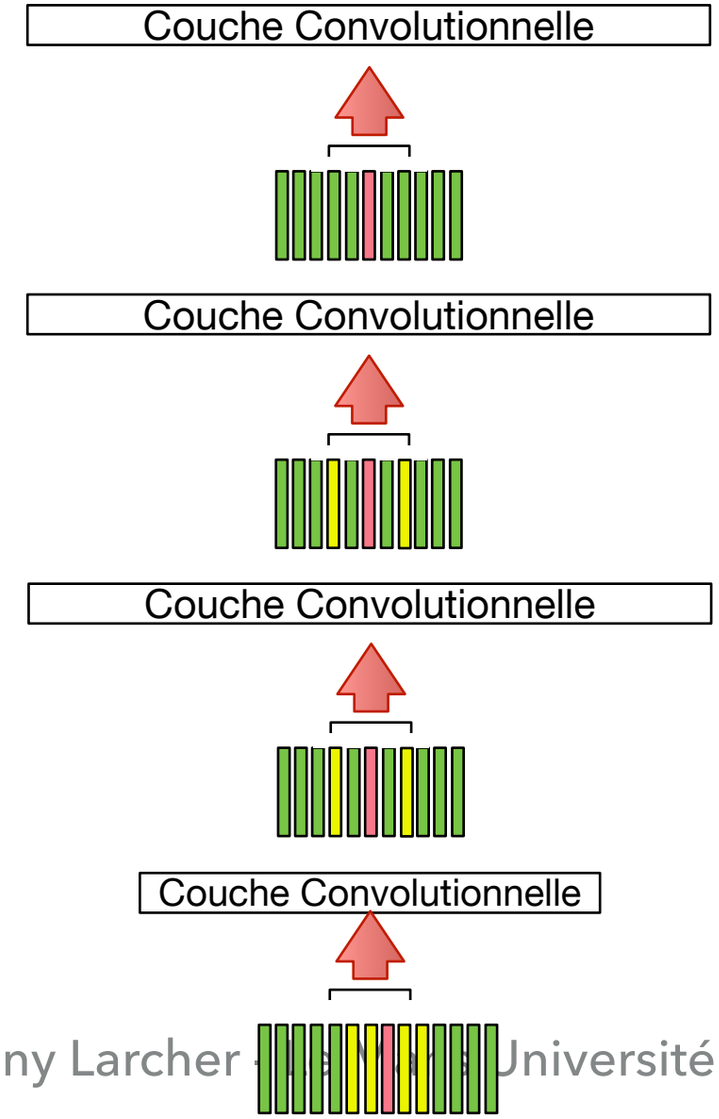
X-VECTEURS: LE RÉSEAU



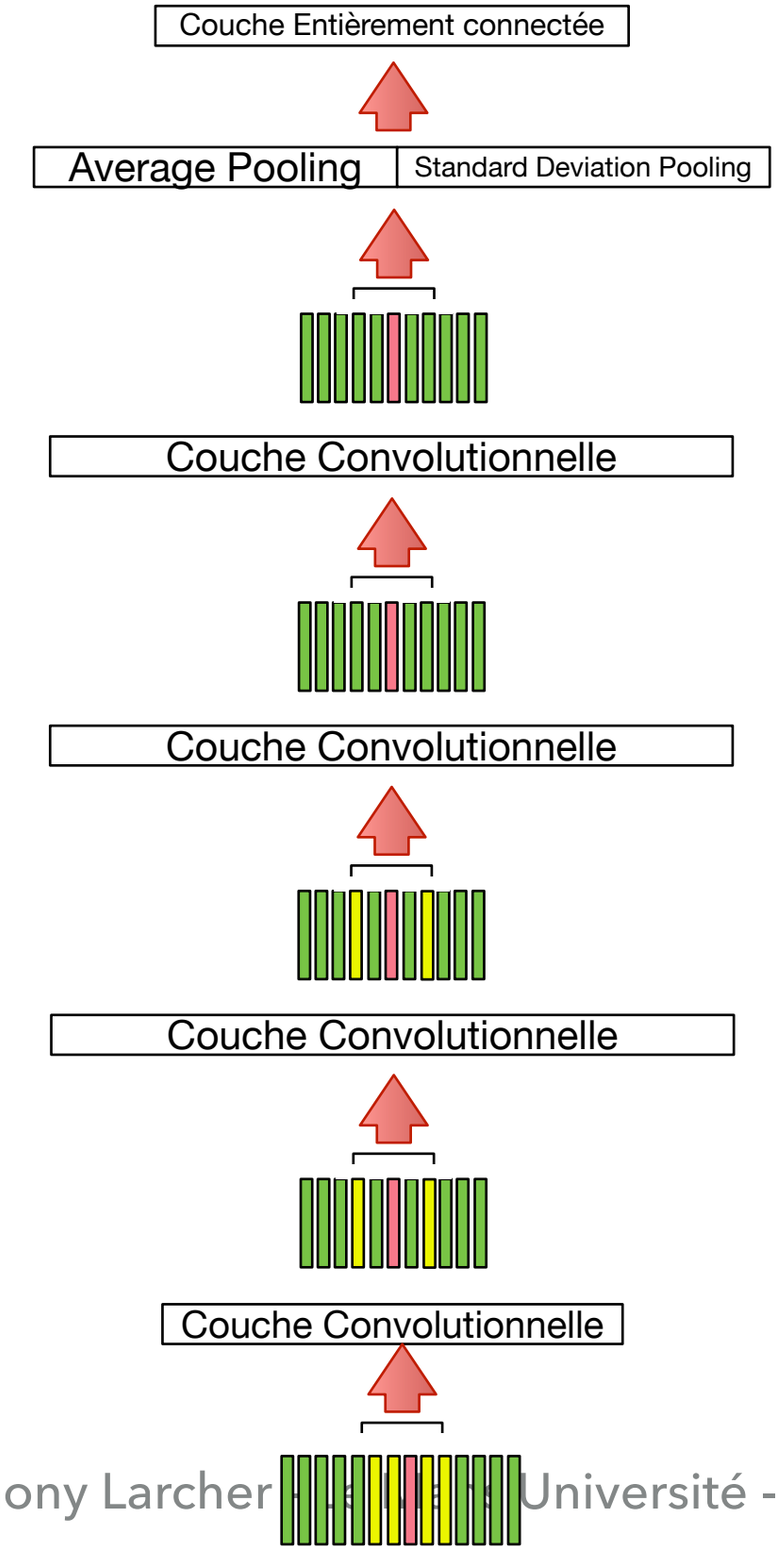
X-VECTEURS: LE RÉSEAU



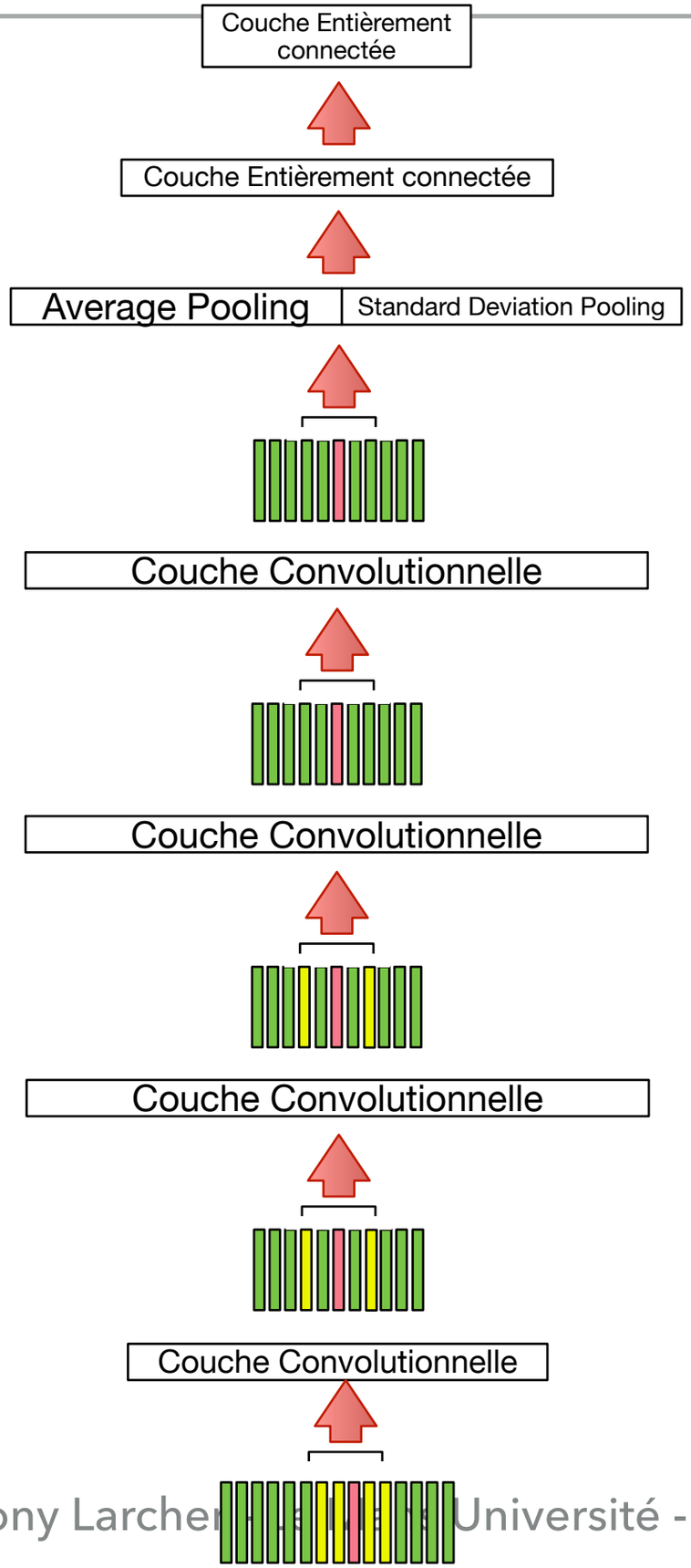
X-VECTEURS: LE RÉSEAU



X-VECTEURS: LE RÉSEAU

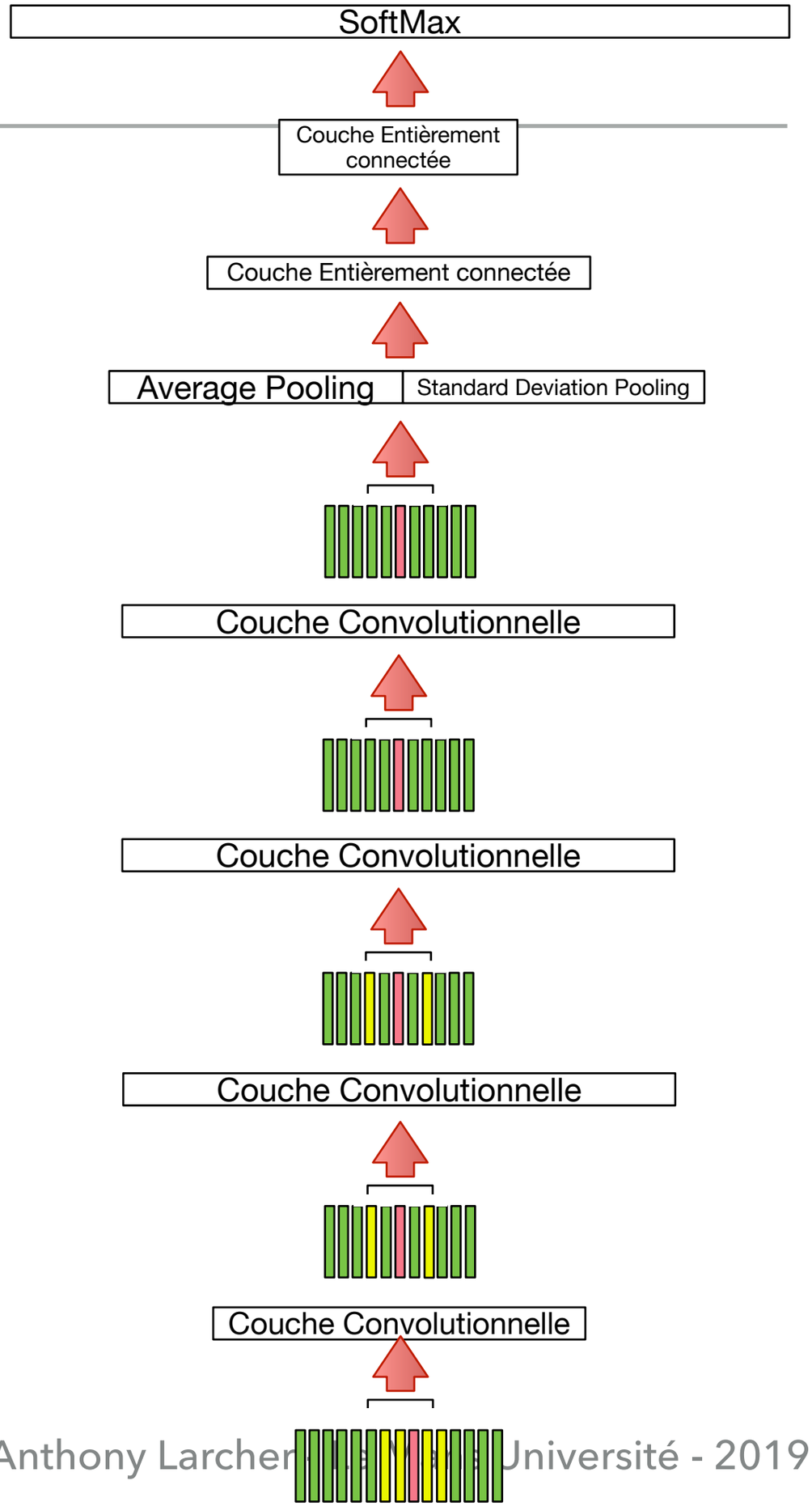


X-VECTEURS: LE RÉSEAU

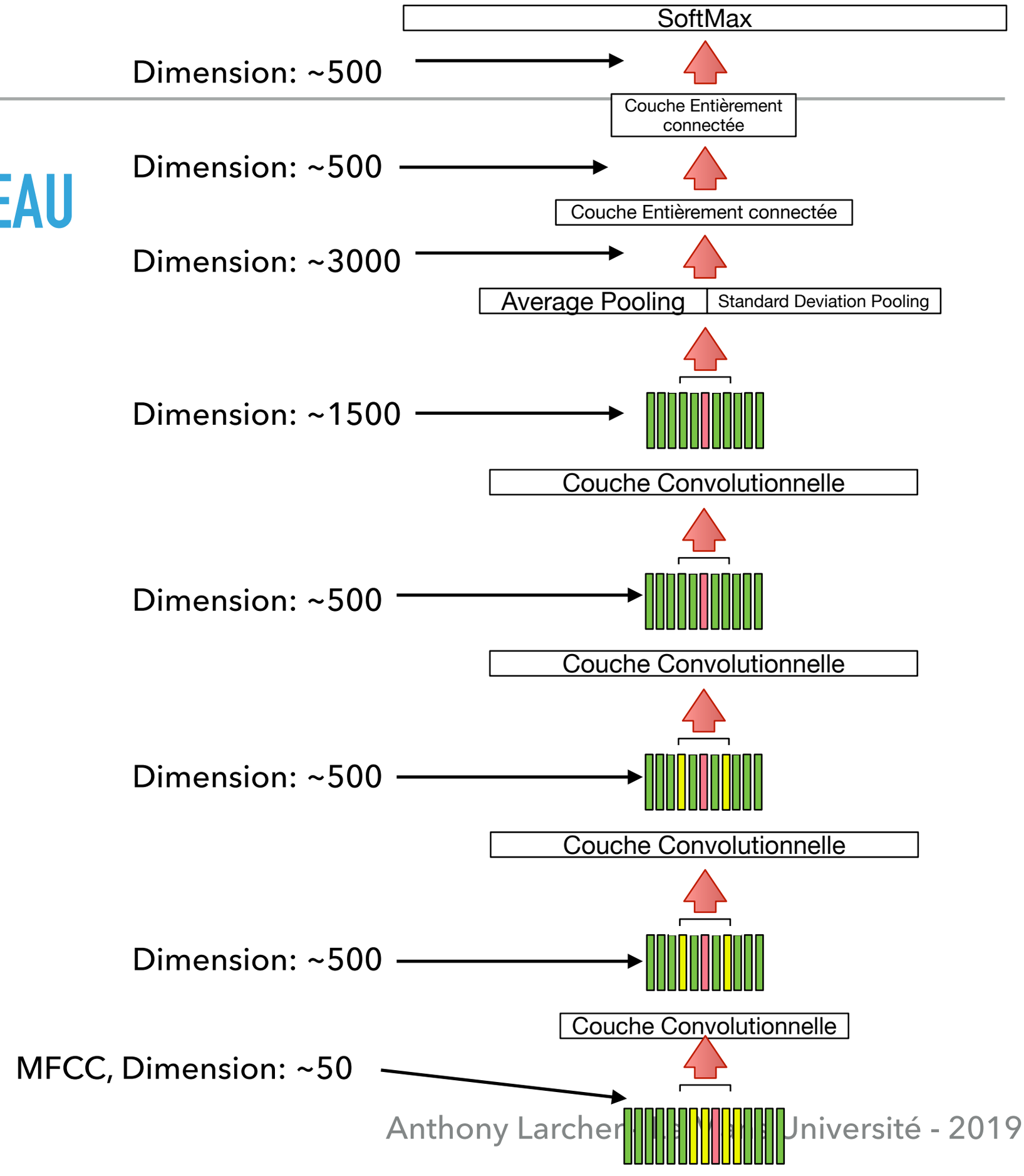


X-VECTEURS: LE RÉSEAU

- Un softmax pour quoi?
- Combien de locuteur?
- Pour quelle tâche?



X-VECTEURS: LE RÉSEAU



X-VECTEURS: LE RÉSEAU

- En pratique:

```
class Xtractor(torch.nn.Module):
    """
    Class that defines an x-vector extractor based on 5 convolutional layers and a mean standard deviation pooling
    """
    def __init__(self, spk_number, dropout):
        super(Xtractor, self).__init__()
        self.frame_conv0 = torch.nn.Conv1d(20, 512, 5, dilation=1)
        self.frame_conv1 = torch.nn.Conv1d(512, 512, 3, dilation=2)
        self.frame_conv2 = torch.nn.Conv1d(512, 512, 3, dilation=3)
        self.frame_conv3 = torch.nn.Conv1d(512, 512, 1)
        self.frame_conv4 = torch.nn.Conv1d(512, 3 * 512, 1)
        self.seg_lin0 = torch.nn.Linear(3 * 512 * 2, 512)
        self.dropout_lin0 = torch.nn.Dropout(p=dropout)
        self.seg_lin1 = torch.nn.Linear(512, 512)
        self.dropout_lin1 = torch.nn.Dropout(p=dropout)
        self.seg_lin2 = torch.nn.Linear(512, spk_number)
        #
        self.norm0 = torch.nn.BatchNorm1d(512)
        self.norm1 = torch.nn.BatchNorm1d(512)
        self.norm2 = torch.nn.BatchNorm1d(512)
        self.norm3 = torch.nn.BatchNorm1d(512)
        self.norm4 = torch.nn.BatchNorm1d(3 * 512)
        self.norm6 = torch.nn.BatchNorm1d(512)
        self.norm7 = torch.nn.BatchNorm1d(512)
        #
        self.activation = torch.nn.LeakyReLU(0.2)
```

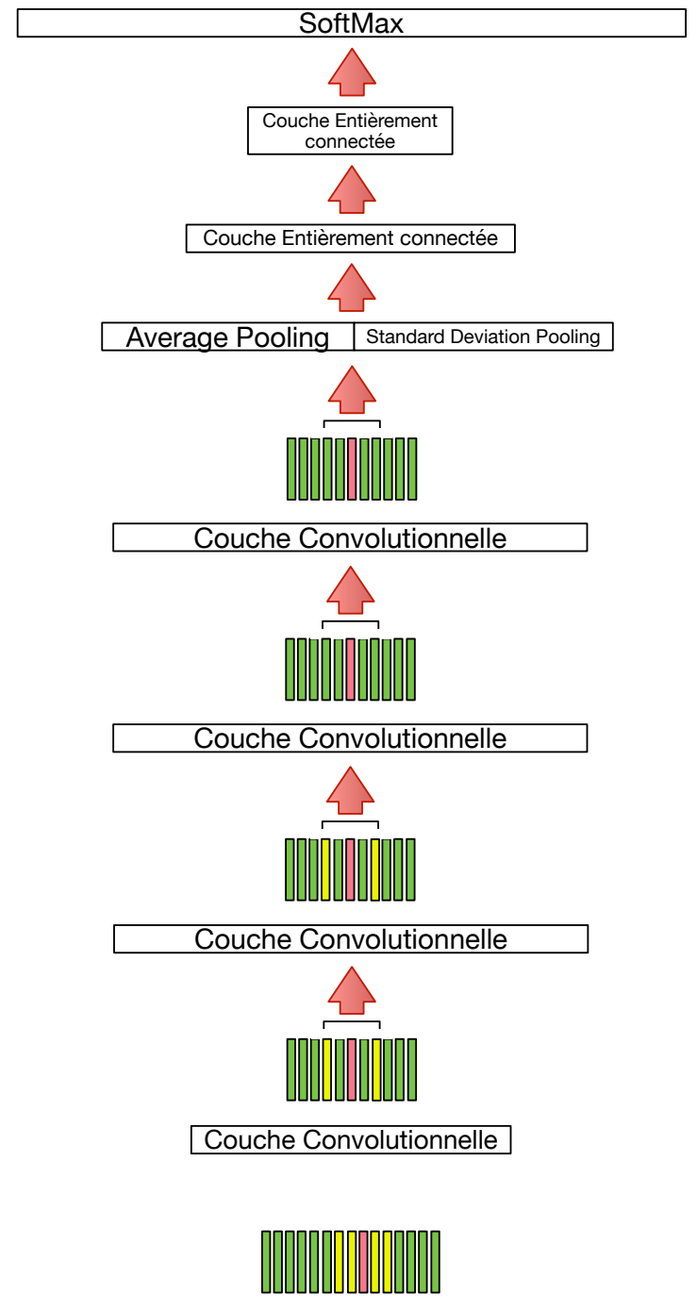
X-VECTEURS: LE RÉSEAU

- En pratique:

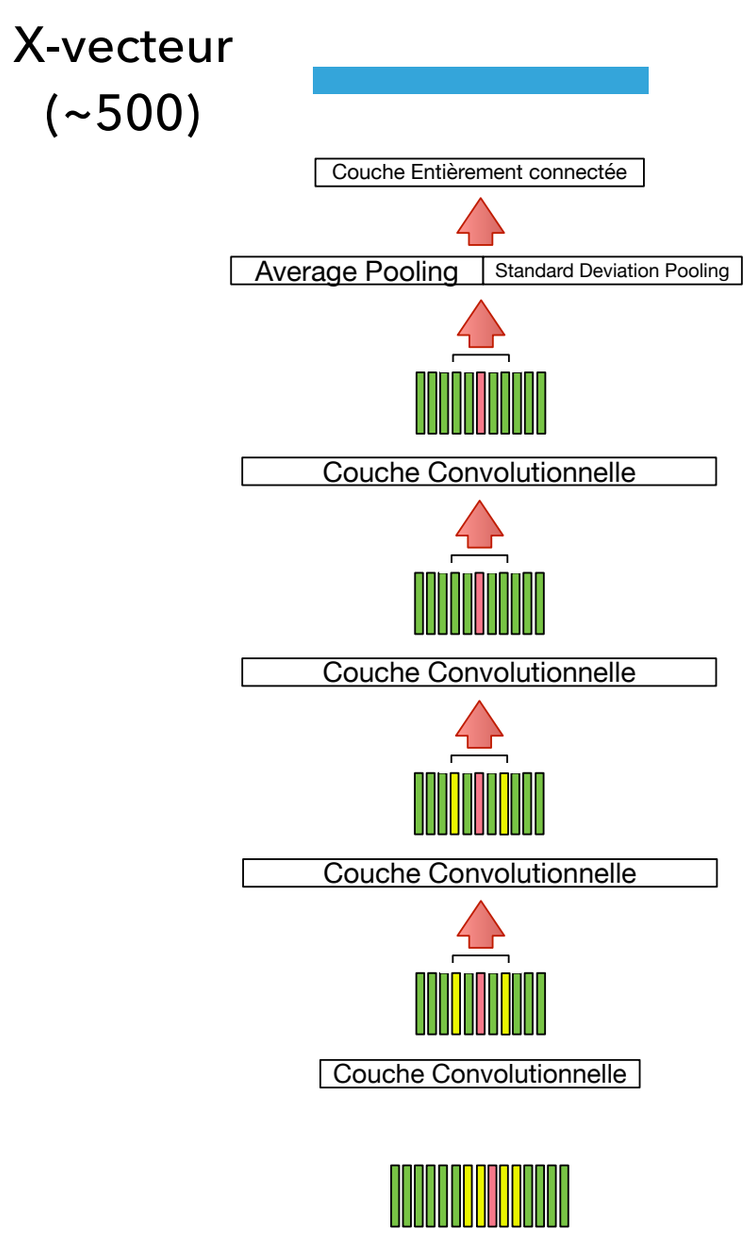
```
def forward(self, x):
    """

    :param x:
    :return:
    """
    seg_emb_0 = self.produce_embeddings(x)
    # batch-normalisation after this layer
    seg_emb_1 = self.norm6(self.activation(seg_emb_0))
    # new layer with batch Normalization
    seg_emb_2 = self.norm7(self.activation(self.seg_lin1(self.dropout_lin1(seg_emb_1))))
    # No batch-normalisation after this layer
    result = self.activation(self.seg_lin2(seg_emb_2))
    return result
```

X-VECTEURS: LE RÉSEAU

- Comment utiliser ce réseau pour des locuteurs inconnus?



X-VECTEURS: LE RÉSEAU



X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Combien de locuteurs?

> 2000

Plus = mieux

NIST-SRE: ~5000

VoxCeleb: ~7000

X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Quelle quantité de données?

>100 segments par locuteur (700 000 pour >Voxceleb)

Essayer d’équilibrer le nombre de segments

X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Quelle quantité de données?

>100 segments par locuteur (700 000 pour >Voxceleb)

Essayer d'équilibrer le nombre de segments

Pour augmenter la robustesse: ajout de bruit
(data augmentation)

Données + bruit, voix, reverb, musique… -> données x5

X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Comment « alimenter » le système?

Minibatchs de taille : 64 -> 256

X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Quelle détection de parole?

Pas trop robuste...

Hypothèse: le bruit rend le système robuste

X-VECTEURS: COMMENT APPRENDRE UN SYSTÈME

- Comment utiliser les x-vecteurs?

PLDA: Probabilistic Linear Discriminant Analysis

(Factor Analysis discriminant)

TRAITEMENT DE LA PAROLE

ET DEPUIS LES X-VECTEURS?

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Utilisation de réseaux résiduels pour des réseaux très profonds

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

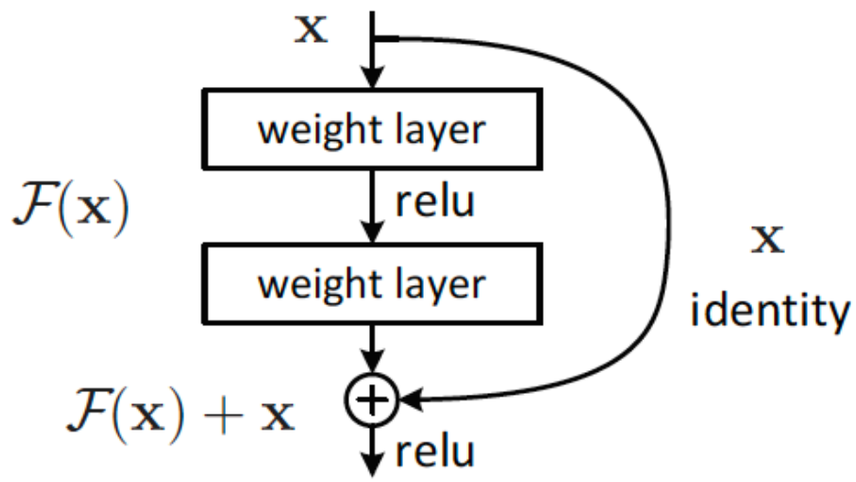


Figure 2. Residual learning: a building block.

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Utilisation de réseaux résiduels
  - Convolution 2D
  - Plus de couches

Zeinali, H., Wang, S., Silnova, A., Matějka, P., & Plchot, O. (2019). BUT System Description to VoxCeleb Speaker Recognition Challenge 2019. *arXiv preprint arXiv:1910.12592*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

Table 1: *x-vector topology proposed in [5]. K in the first layer indicates different feature dimensionalities, T is the number of training segment frames and N in the last row is the number of speakers.*

Layer	**Standard DNN**		**BIG DNN**	
	Layer context	**(Input) × output**	**Layer context**	**(Input) × output**
frame1	$[t-2, t-1, t, t+1, t+2]$	$(5 \times K) \times 512$	$[t-2, t-1, t, t+1, t+2]$	$(5 \times K) \times 1024$
frame2	$[t]$	512×512	$[t]$	1024×1024
frame3	$[t-2, t, t+2]$	$(3 \times 512) \times 512$	$[t-4, t-2, t, t+2, t+4]$	$(5 \times 1024) \times 1024$
frame4	$[t]$	512×512	$[t]$	1024×1024
frame5	$[t-3, t, t+3]$	$(3 \times 512) \times 512$	$[t-3, t, t+3]$	$(3 \times 1024) \times 1024$
frame6	$[t]$	512×512	$[t]$	1024×1024
frame7	$[t-4, t, t+4]$	$(3 \times 512) \times 512$	$[t-4, t, t+4]$	$(3 \times 1024) \times 1024$
frame8	$[t]$	512×512	$[t]$	1024×1024
frame9	$[t]$	512×1500	$[t]$	1024×2000
stats pooling	$[0, T]$	1500×3000	$[0, T]$	2000×4000
segment1	$[0, T]$	3000×512	$[0, T]$	4000×512
segment2	$[0, T]$	512×512	$[0, T]$	512×512
softmax	$[0, T]$	$512 \times N$	$[0, T]$	$512 \times N$

Zeinali, H., Wang, S., Silnova, A., Matějka, P., & Plchot, O. (2019). BUT System Description to VoxCeleb Speaker Recognition Challenge 2019. *arXiv preprint arXiv:1910.12592*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

Table 2: *The proposed ResNet34 architecture. N in the last row is the number of speakers. The first dimension of the input shows number of filter-banks and the second dimension indicates the number of frames.*

Layer name	**Structure**	**Output**
Input	–	$40 \times 200 \times 1$
Conv2D-1	3×3, Stride 1	$40 \times 200 \times 32$
ResNetBlock-1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$, Stride 1	$40 \times 200 \times 32$
ResNetBlock-2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$, Stride 2	$20 \times 100 \times 64$
ResNetBlock-3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$, Stride 2	$10 \times 50 \times 128$
ResNetBlock-4	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$, Stride 2	$5 \times 25 \times 256$
StatsPooling	–	10×256
Flatten	–	2560
Dense1	–	256
Dense2 (Softmax)	–	N
Total	–	–

Zeinali, H., Wang, S., Silnova, A., Matějka, P., & Plchot, O. (2019). BUT System Description to VoxCeleb Speaker Recognition Challenge 2019. *arXiv preprint arXiv:1910.12592*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

Table 3: *Comparison of equal error rate (EER) of various front-end speaker embedding extraction systems. The back-end classifier is fixed to cosine similarity.*

System	EER %
i-vector	13.8
i-vector/LDA	7.25
x-vector(w/o augment) [21]	11.3
x-vector(w augment) [21]	9.9

Jung, J. W., Heo, H. S., Kim, J. H., Shim, H. J., & Yu, H. J. (2019). RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. *arXiv preprint arXiv:1904.08104*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

Table 3: *Results of the systems on Voxceleb challenge. Cosine distance and PLDA are used as backends for ResNet and TDNN systems, respectively. Note that, for the open systems, VoxCeleb1 development data was used for training the embedding networks. That explains their good performance on E and H conditions where they are a subset of this development set.*

#	Fixed/Open	Acc. features	Embd NN	Backend	S-norm	Vox1 O cleaned		Vox1 E cleaned		Vox1 H cleaned	
						MinDCF	EER	MinDCF	EER	MinDCF	EER
1	Fixed	FBANK	ResNet256 + AAM	cos	yes	0.166	1.42	0.164	1.35	0.233	2.48
2	Fixed	FBANK	ResNet160 + AAM	cos	yes	0.154	1.31	0.163	1.38	0.233	2.50
3	Fixed	FBANK	TDNN + AAM	PLDA	no	0.181	1.46	0.185	1.57	0.299	2.89
4	Fixed	PLP	TDNN	PLDA	no	0.213	1.94	0.239	2.03	0.379	3.97
5	Open	FBANK	ResNet256 + AAM	cos	yes	0.157	1.22	0.102	0.81	0.164	1.50
6	Open	FBANK	TDNN	PLDA	no	0.195	1.65	0.170	1.42	0.288	2.70
7	Open	PLP	TDNN	PLDA	no	0.210	1.98	0.163	1.51	0.249	2.83
8	Fixed	Fusion 1+2+3+4 (weighted average)				0.131	1.02	0.138	1.14	0.212	2.12
9	Open	Fusion 1+2+3+4 LR				0.131	1.02	0.138	1.14	0.212	2.12
10	Open	Fusion 2+5+6+7 LR				0.118	0.96	0.098	0.80	0.160	1.51

Jung, J. W., Heo, H. S., Kim, J. H., Shim, H. J., & Yu, H. J. (2019). RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification. *arXiv preprint arXiv:1904.08104*.

TRAITEMENT DE LA PAROLE

FONCTIONS DE COÛT DISCRIMINANTES

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Quelle fonction de coût?
  - Cross-entropy avec Softmax :
    pénalise la mauvaise classification mais n
    « séparation des classes »
    (variance intra et inter classes)

$$L_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i+\mathbf{b}_{y_i}}}{\sum_{j=1}^{c}e^{\mathbf{W}_j^T\mathbf{x}_i+\mathbf{b}_j}}$$

  - Pour l'entropie croisée: un élément proche d'une autre classe mais bien classé n'est pas pénalisant

➡ Généralise mal…

Xiang, X., Wang, S., Huang, H., Qian, Y., & Yu, K. (2019). Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition. *arXiv preprint arXiv:1906.07317*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Quelle fonction de coût?
  - Cross-entropy avec A-Softmax
    On supprime le biais et on ne regarde que l'angle entre un embedding et les colonnes de W

$$L'_{\text{Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{||\mathbf{x}_i||\cos(\theta_{y_i,i})}}{\sum_{j=1}^{c}e^{||\mathbf{x}_i||\cos(\theta_{j,i})}}$$

- On impose une marge entre les classes: m>1

$$L_{\text{A-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{||\mathbf{x}_i||\phi(\theta_{y_i,i})}}{Z}$$

$$Z = e^{||\mathbf{x}_i||\phi(\theta_{y_i,i})} + \sum_{j=1,j\neq i}^{c} e^{||\mathbf{x}_i||\cos(\theta_{j,i})}$$

$$\phi(\theta_{y_i,i}) = \cos(m\theta_{y_i,i}) \leq \cos(\theta_{y_i,i})$$

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Quelle fonction de coût?
  - Cross-entropy avec AAM-Softmax
    On normalise les vecteurs (projection sur une sphère unité)
  - La distance entre 2 vecteurs est proportionnelle à l'angle

$$L_{\text{AAM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i,i}+m))}}{Z}$$

$$\phi(\theta_{y_i,i}) = \cos(\theta_{y_i,i}+m) \qquad Z = e^{s(\cos(\theta_{y_i,i}+m))} + \sum_{j=1,j\neq i}^{c} e^{s(\cos(\theta_{j,i}))}$$

s: coefficient qui assure que le gradient ne soit pas trop faible

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Quelle fonction de coût?

Table 2: *Results on the original VoxCeleb1 test set and the extented and hard test sets (VoxCeleb1-E and VoxCeleb1-H).*

	Model	Loss	Training set	EER
VoxCeleb1 test set				
Nagrani *et al.* [19]	GMM-UBM (i-vector)	-	VoxCeleb1	8.8
Cai *et al.* [15]	ResNet-34	A-Softmax	VoxCeleb1	4.40
Okabe *et al.* [21]	TDNN (x-vector)	Softmax	VoxCeleb1	3.85
Hajibabaei *et al.* [22]	ResNet-20	AM-Softmax	VoxCeleb1	4.30
Chung *et al.* [23]	ResNet-50	Softmax + Contrastive	VoxCeleb2	4.19
Xie *et al.* [24]	Thin ResNet-34	Softmax	VoxCeleb2	3.22
Ours	TDNN (x-vector)	AAM-softmax	VoxCeleb2	**2.694**
Ours	TDNN (x-vector)	AAM-softmax	VoxCeleb1 + VoxCeleb2	**2.238**
VoxCeleb1-E test set				
Chung *et al.* [23]	ResNet-50	Softmax + Contrastive	VoxCeleb2	4.42
Xie *et al.* [24]	Thin ResNet-34	Softmax	VoxCeleb2	3.13
Ours	TDNN (x-vector)	AAM-softmax	VoxCeleb2	**2.762**
VoxCeleb1-H test set				
Chung *et al.* [23]	ResNet-50	Softmax + Contrastive	VoxCeleb2	7.33
Xie *et al.* [24]	Thin ResNet-34	Softmax	VoxCeleb2	5.06
Ours	TDNN (x-vector)	AAM-softmax	VoxCeleb2	**4.732**

Xiang, X., Wang, S., Huang, H., Qian, Y., & Yu, K. (2019). Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition. *arXiv preprint arXiv:1906.07317*.

RÉSEAUX AVEC BLOCS RÉSIDUELS

- Quelle fonction de coût?

Table 3: *Results of the systems on Voxceleb challenge. Cosine distance and PLDA are used as backends for ResNet and TDNN systems, respectively. Note that, for the open systems, VoxCeleb1 development data was used for training the embedding networks. That explains their good performance on E and H conditions where they are a subset of this development set.*

#	Fixed/Open	Acc. features	Embd NN	Backend	S-norm	Vox1 O cleaned		Vox1 E cleaned		Vox1 H cleaned	
						MinDCF	EER	MinDCF	EER	MinDCF	EER
1	Fixed	FBANK	ResNet256 + AAM	cos	yes	0.166	1.42	0.164	1.35	0.233	2.48
2	Fixed	FBANK	ResNet160 + AAM	cos	yes	0.154	1.31	0.163	1.38	0.233	2.50
3	Fixed	FBANK	TDNN + AAM	PLDA	no	0.181	1.46	0.185	1.57	0.299	2.89
4	Fixed	PLP	TDNN	PLDA	no	0.213	1.94	0.239	2.03	0.379	3.97
5	Open	FBANK	ResNet256 + AAM	cos	yes	0.157	1.22	0.102	0.81	0.164	1.50
6	Open	FBANK	TDNN	PLDA	no	0.195	1.65	0.170	1.42	0.288	2.70
7	Open	PLP	TDNN	PLDA	no	0.210	1.98	0.163	1.51	0.249	2.83
8	Fixed	Fusion 1+2+3+4 (weighted average)				0.131	1.02	0.138	1.14	0.212	2.12
9	Open	Fusion 1+2+3+4 LR				0.131	1.02	0.138	1.14	0.212	2.12
10	Open	Fusion 2+5+6+7 LR				0.118	0.96	0.098	0.80	0.160	1.51

Zeinali, H., Wang, S., Silnova, A., Matějka, P., & Plchot, O. (2019). BUT System Description to VoxCeleb Speaker Recognition Challenge 2019. *arXiv preprint arXiv:1910.12592*.

TRAITEMENT DE LA PAROLE

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE

- La plupart des approches utilisent encore la PLDA
- Difficile d'utiliser un classifier binaire pour la reconnaissance du locuteur (manque d'exemples pour les tests client)
- Idée:
  - représenter un test (couple de vecteurs) par un vecteur unique
  - Classifier ces vecteurs uniques

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE

- On construit un b-vecteur en concatenant les vecteurs suivant:

$$\boldsymbol{b}_a = \boldsymbol{w}_1 \oplus \boldsymbol{w}_2, \quad (3)$$
$$\boldsymbol{b}_m = \boldsymbol{w}_1 \otimes \boldsymbol{w}_2, \quad (4)$$
$$\boldsymbol{b}_s = |\boldsymbol{w}_1 \ominus \boldsymbol{w}_2|, \quad (5)$$
$$\boldsymbol{b}_r = |\log|\boldsymbol{w}_1| \ominus \log|\boldsymbol{w}_2||, \quad (6)$$

- On utilise un SVM ou un DNN pour classifier les b-vecteurs

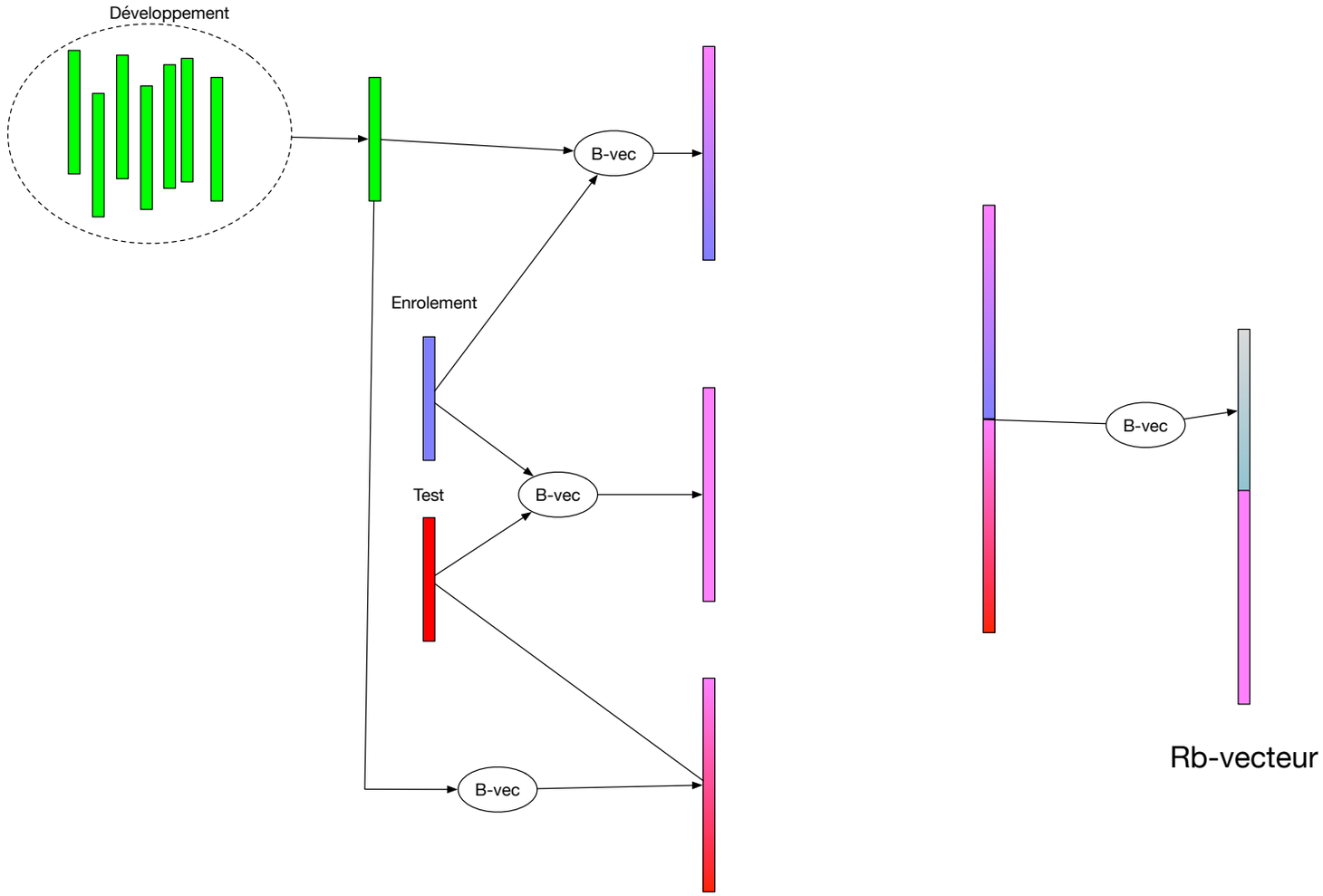
H. S.Lee,Y.Tso, Y. F.Chang, H. M. Wang andS. K.Jeng, “Speaker verification using kernel-based binary classifiers with binary operation derived features,”Acoustics, Speech and Signal Processing (ICASSP),pp. 1660-1664, 2014

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE

- Inconvénient: les b-vecteurs ne contiennent aucune information « a priori » sur la distribution des locuteurs.
- Le UBM/GMM ou la PLDA contiennent cette information
- Comment rajouter cette information?

Heo, H. S., Yang, I. H., Kim, M. J., Yoon, S. H., & Yu, H. J. (2016, March). Advanced b-vector system based deep neural network as classifier for speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5465-5469). IEEE.

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE



Heo, H. S., Yang, I. H., Kim, M. J., Yoon, S. H., & Yu, H. J. (2016, March). Advanced b-vector system based deep neural network as classifier for speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5465-5469). IEEE.

REMPLACER LA PLDA PAR UNE APPROCHE NEURONALE

Table 2. Performance in EER and minDCF of the PLDA and DNN systems (NIST08, short2-short3, male set)

		Avg	DET1	DET2	DET3	DET4	DET5	DET6	DET7	DET8
EER(%)	i-vector PLDA	4.30	**5.79**	0.81	**5.73**	7.01	4.98	**4.67**	3.14	2.30
	b-vector DNN	4.42	6.38	**0.40**	6.56	6.42	5.46	4.90	**2.94**	2.30
	Rb-vector DNN	**3.98**	5.85	**0.40**	5.98	**4.63**	**4.17**	5.46	3.23	**2.21**
minDCF	i-vector PLDA	**.0197**	**.0266**	**.0008**	**.0261**	**.0287**	.0200	**.0249**	.0172	.0138
	b-vector DNN	.0206	.0282	.0028	.0287	.0302	.0238	.0251	**.0147**	**.0112**
	Rb-vector DNN	**.0197**	.0268	**.0008**	.0273	.0292	**.0188**	.0276	.0160	**.0112**

Heo, H. S., Yang, I. H., Kim, M. J., Yoon, S. H., & Yu, H. J. (2016, March). Advanced b-vector system based deep neural network as classifier for speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5465-5469). IEEE.

TRAITEMENT DE LA PAROLE

SYSTÈMES DE BOUT-EN-BOUT

Le Mans Université

Anthony Larcher - Le Mans Université - 2017

LE COURS DE M. TAHON ÉTAIT-IL UTILE?

- Les MFCC extraient une information dans les « basses fréquences » (plus de filtres)
- Ils permettent d'extraire de l'information dans les bandes spectrales qui caractérisent la parole humaine

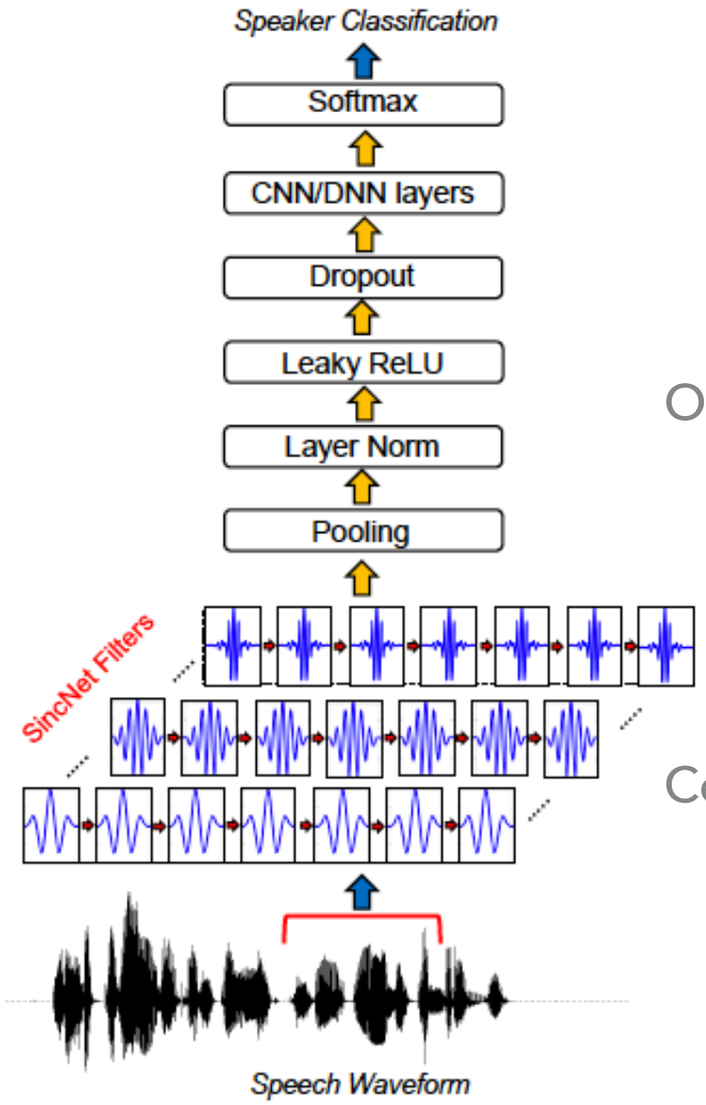
Les MFCC sont idéals pour la reconnaissance de parole

- Les MFCC extraient une information dans les « basses fréquences » (plus de filtres)
- Ils permettent d'extraire de l'information dans les bandes spectrales qui caractérisent les locuteurs

Les MFCC sont idéals pour la reconnaissance du locuteur

UN SYSTÈME END-TO-END?

Est-ce que les paramètres acoustiques (MFCC) utilisés pour reconnaître les locuteurs doivent être les mêmes que pour reconnaître le contenu phonétique?



Le signal « brut » est filtré en utilisant des filtres convolutionnels

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l]$$

On choisi un modèle qui recrée des filtres **fréquentiels** rectangulaires

$$G[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right)$$

Ces filtres sont équivalents à une convolution temporelle avec « g »:

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n)$$

Le réseau « apprend » les fréquences de coupure des filtres.

Ravanelli, M., & Bengio, Y. (2018). Speaker Recognition from raw waveform with SincNet. *arXiv preprint arXiv:1808.00158*.

RÉSEAUX ÉVOLUTIFS

Valenti, G., Delgado, H., Todisco, M., Evans, N. W., & Pilati, L. (2018). An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks. In *Odyssey* (pp. 288-295).

SYSTÈMES DE BOUT-EN-BOUT

- RAWnet: lire l'article et décrire la structure du système à partir des éléments vus en cours