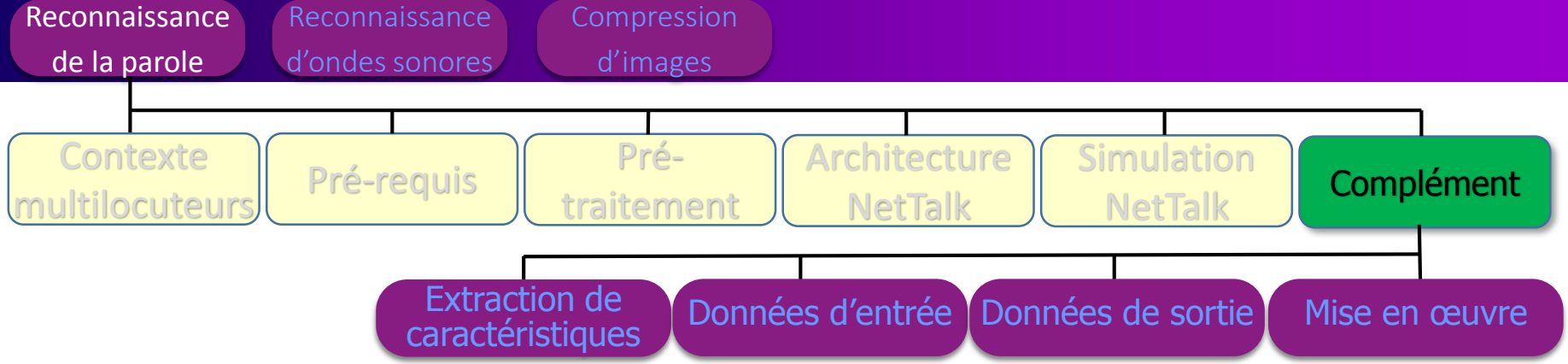
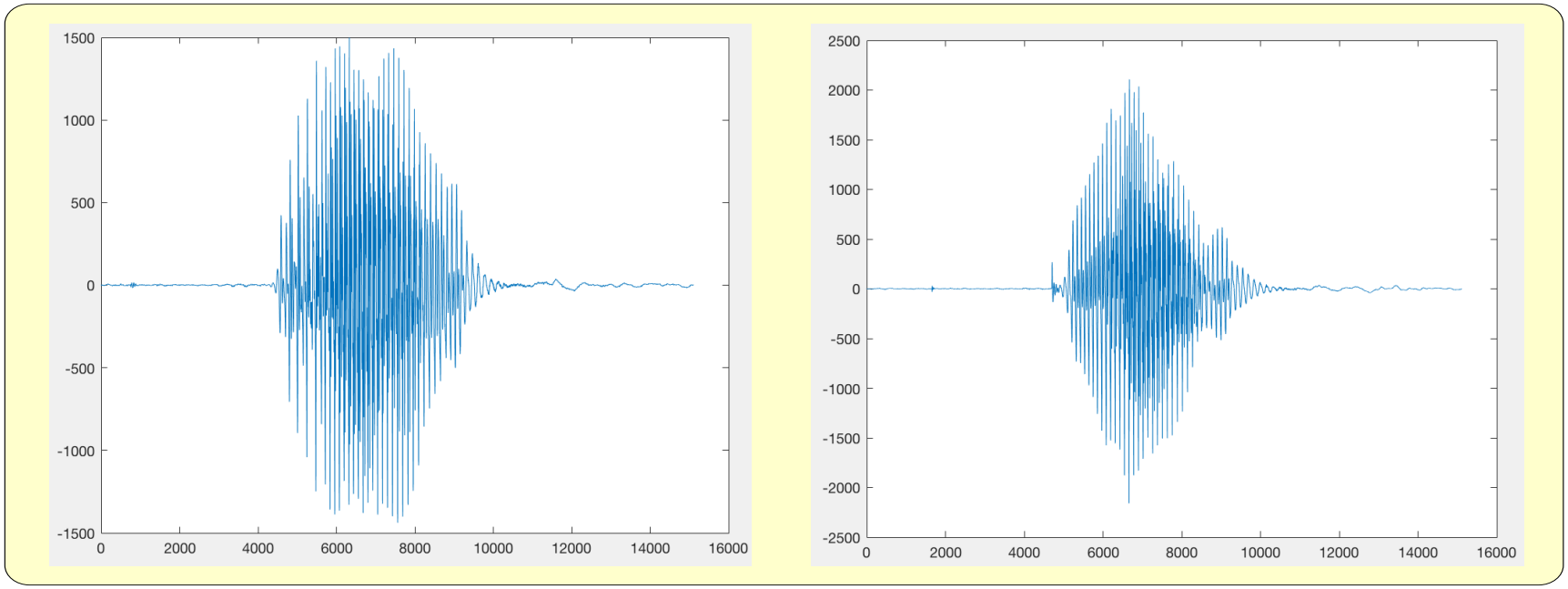
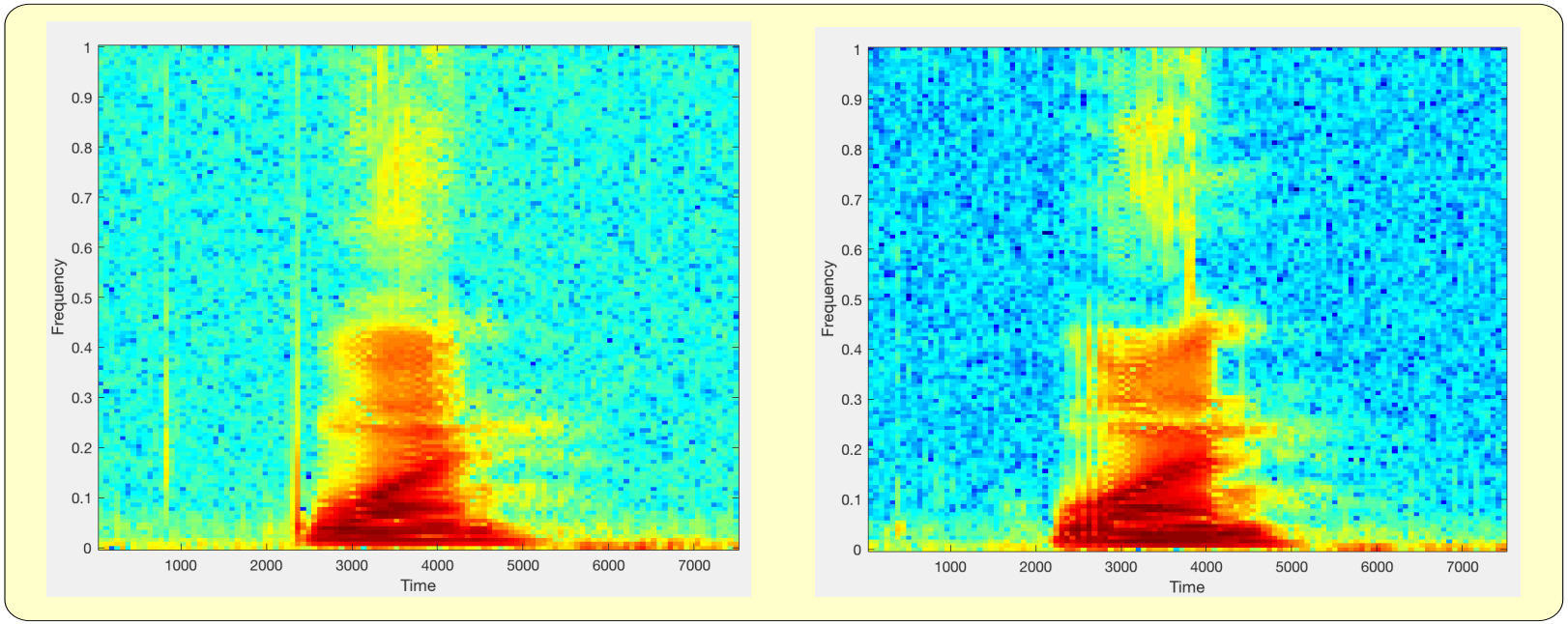


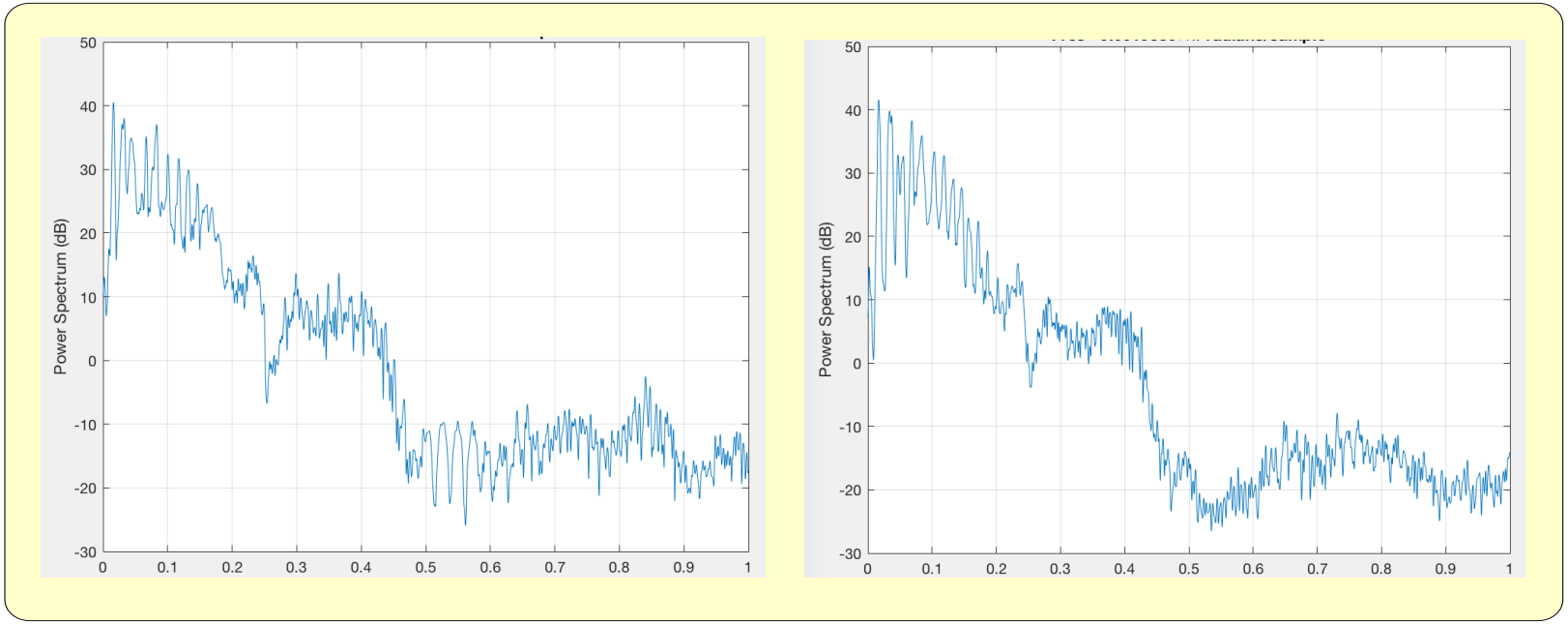

# Contexte du laboratoire 2



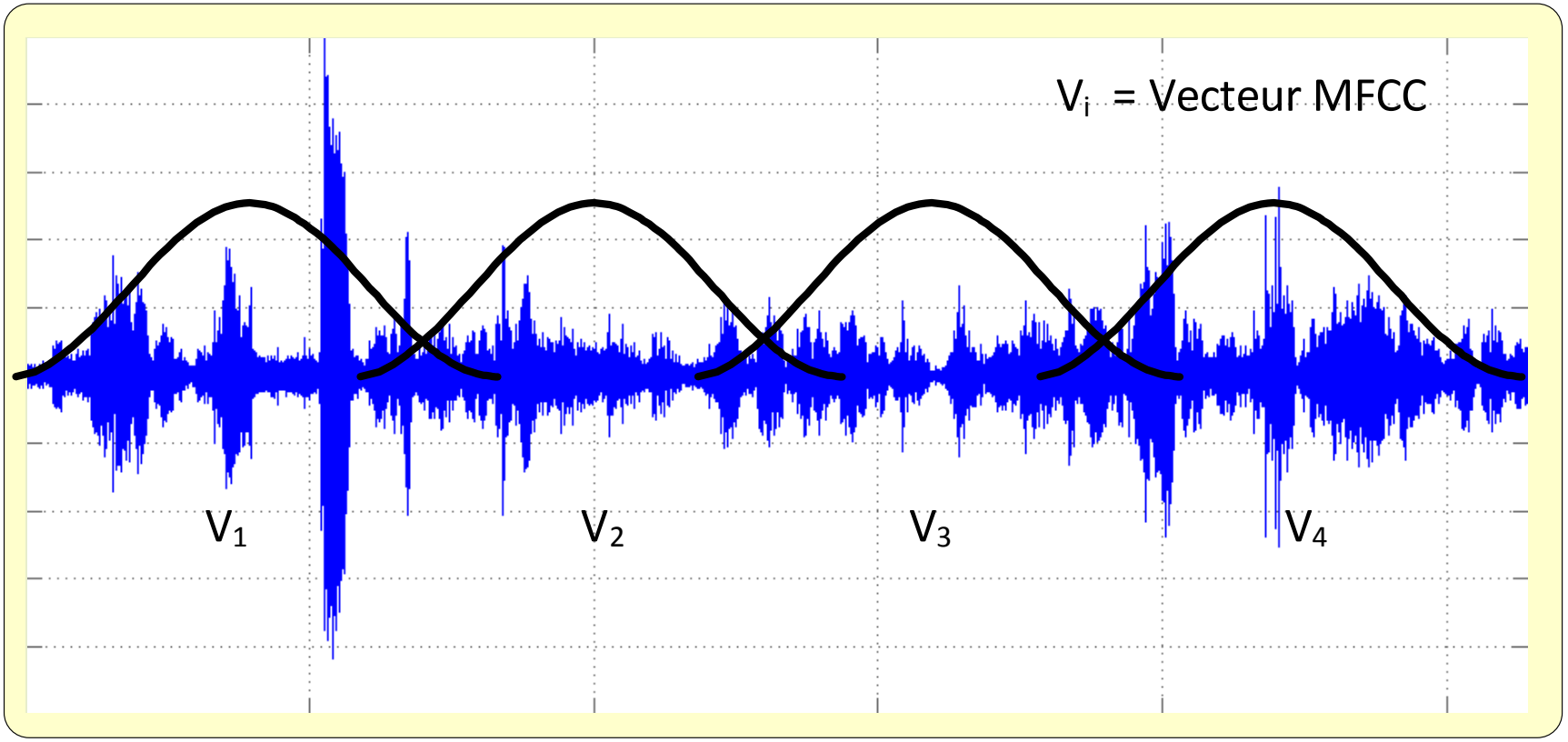

Signal audio du mot « one » prononcé par la même personne à deux intervalles de temps différents. Représentation dans le domaine temporel.




**Spectrogramme** du signal audio du mot « one » prononcé par la même personne à deux intervalles de temps différents.




**Spectre** du signal audio du mot « one » prononcé par la même personne à deux intervalles de temps différents.

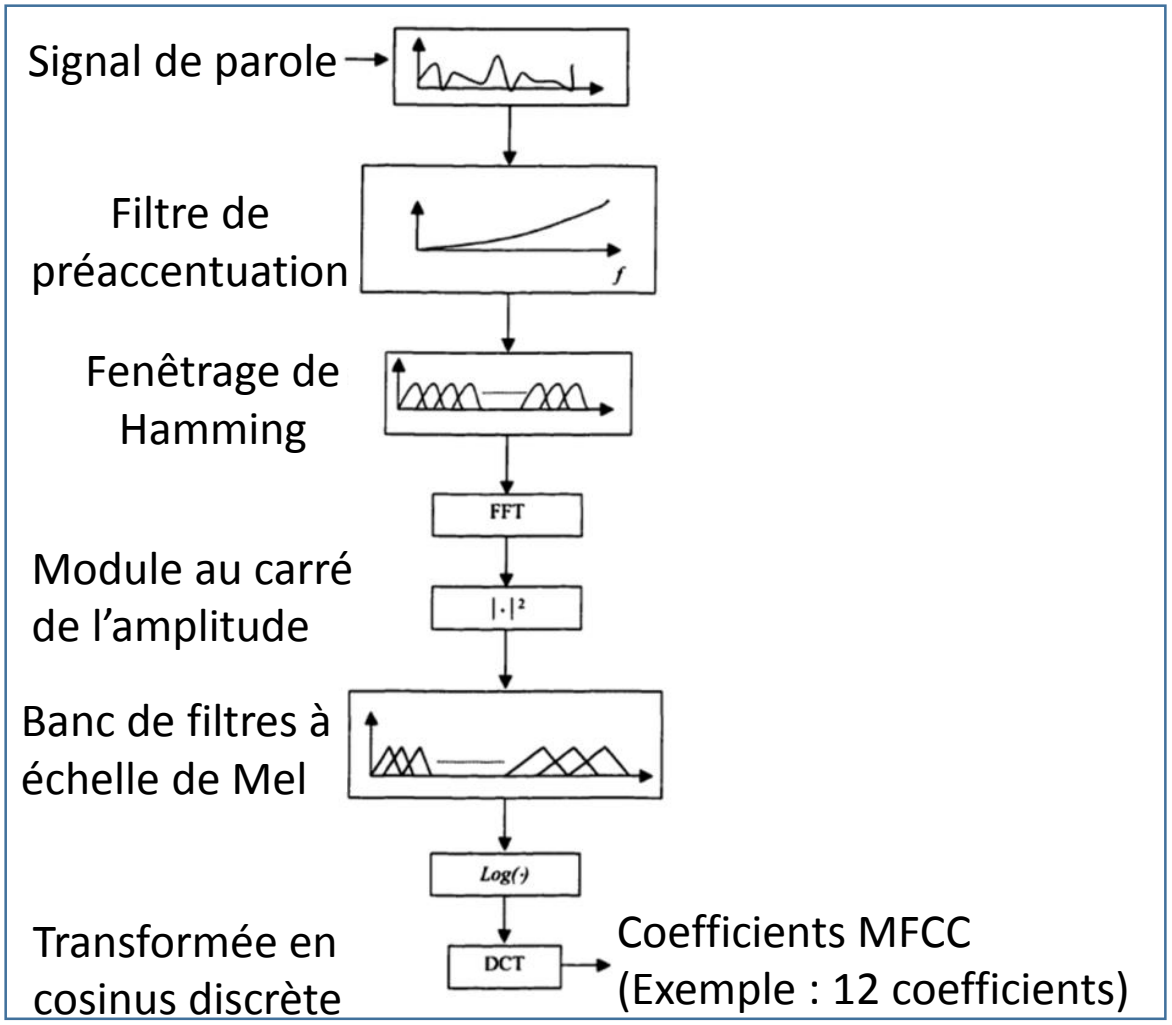


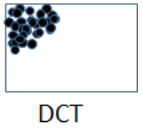

**Technique de fenêtrage**

- Délimiter le signal
- Stationnarité

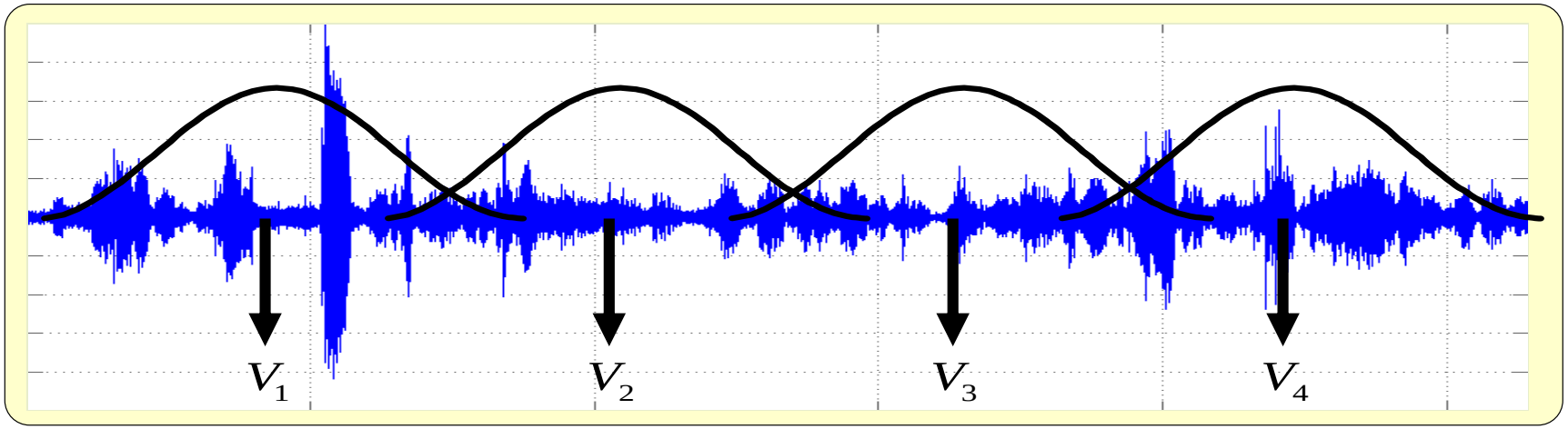
- **Calcul des MFCC** : Mel-Frequency Cepstral Coefficients
  - Le spectre de puissance du signal est calculé
  - Les coefficients cepstraux sont calculés à partir d'une transformée en cosinus discrète du spectre obtenu
  - Les bandes de fréquence de ce spectre sont espacées logarithmiquement selon l'échelle de Mel.

- **Pourquoi les MFCC** ?
  - Le signal de parole est modélisé par la convolution de la fonction de transfert du conduit vocal (filtre) avec le signal d'excitation (source).
  - La représentation cepstrale permet de dissocier la source du filtre pour estimer la fréquence fondamentale ou les formants.




**Étapes de calcul des coefficients MFCC** (Rabiner et Juang, 1993)



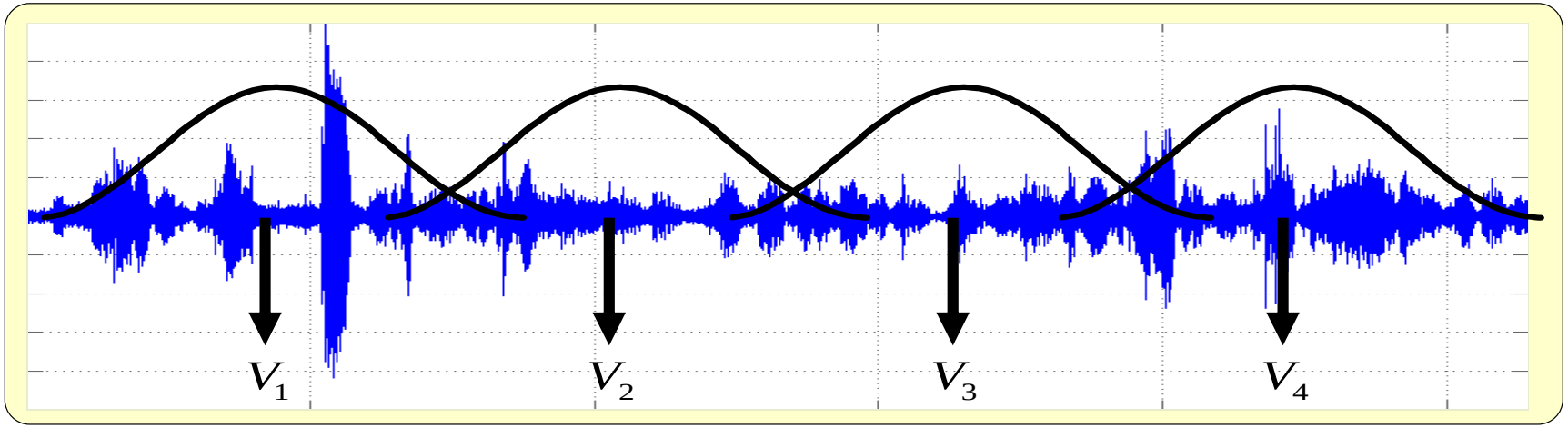

- Statiques : $S_1, S_2, \ldots, S_{12}$ (MFCC)
- Énergie statique : $E_s$
- Dynamiques : $D_1, D_2, \ldots, D_{12}$
- Énergie dynamique : $E_d$

Pour chaque trame (fenêtre d'analyse) :

$$[\underbrace{S_1, S_2, \ldots, S_{12}}_{\text{statique (mfcc)}}, \underbrace{E_s}_{\text{énergie statique}}, \underbrace{D_1, D_2, \ldots, D_{12}}_{\text{dynamique}}, \underbrace{E_d}_{\text{énergie dynamique}}]$$

(26 coefficients)

Que faire pour un signal ayant 4 trames ?



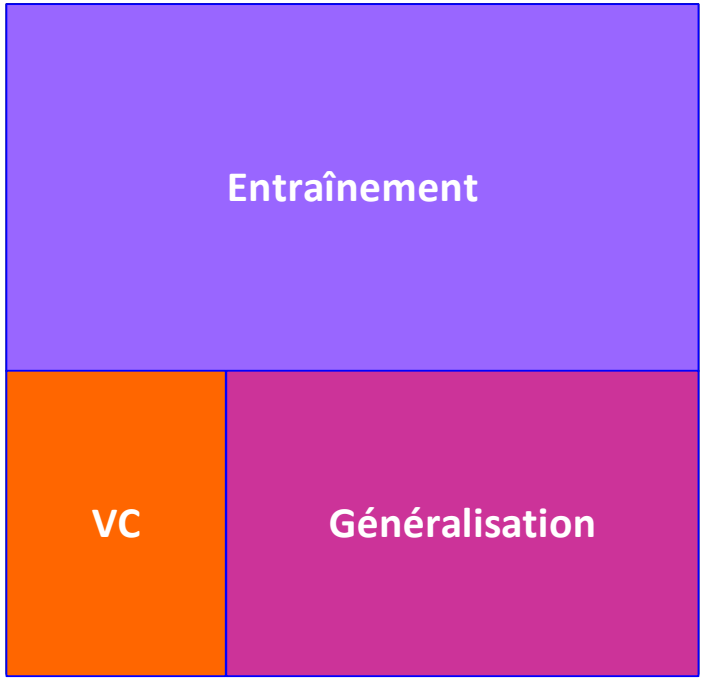

Pour un signal ayant 4 trames : $V_1, V_2, V_3, V_4$

$$V = \begin{bmatrix} S_{1,1}, S_{1,2}, \ldots, S_{1,12}, & E_{1s}, & D_{1,1}, D_{1,2}, \ldots, D_{1,12}, & E_{1d} \\ S_{2,1}, S_{2,2}, \ldots, S_{2,12}, & E_{2s}, & D_{2,1}, D_{2,2}, \ldots, D_{2,12}, & E_{2d} \\ S_{3,1}, S_{3,2}, \ldots, S_{3,12}, & E_{3s}, & D_{3,1}, D_{3,2}, \ldots, D_{3,12}, & E_{3d} \\ S_{4,1}, S_{4,2}, \ldots, S_{4,12}, & E_{4s}, & D_{4,1}, D_{4,2}, \ldots, D_{4,12}, & E_{4d} \end{bmatrix}$$

Cela correspond a une entrée de 4 x 26 paramètres

- Dans le cadre du laboratoire, commencez par les paramètres statiques (12 paramètres par trame)
- Comme travail optionnel, on pourrait utiliser tous les paramètres (statiques et dynamiques)

## Bases de données

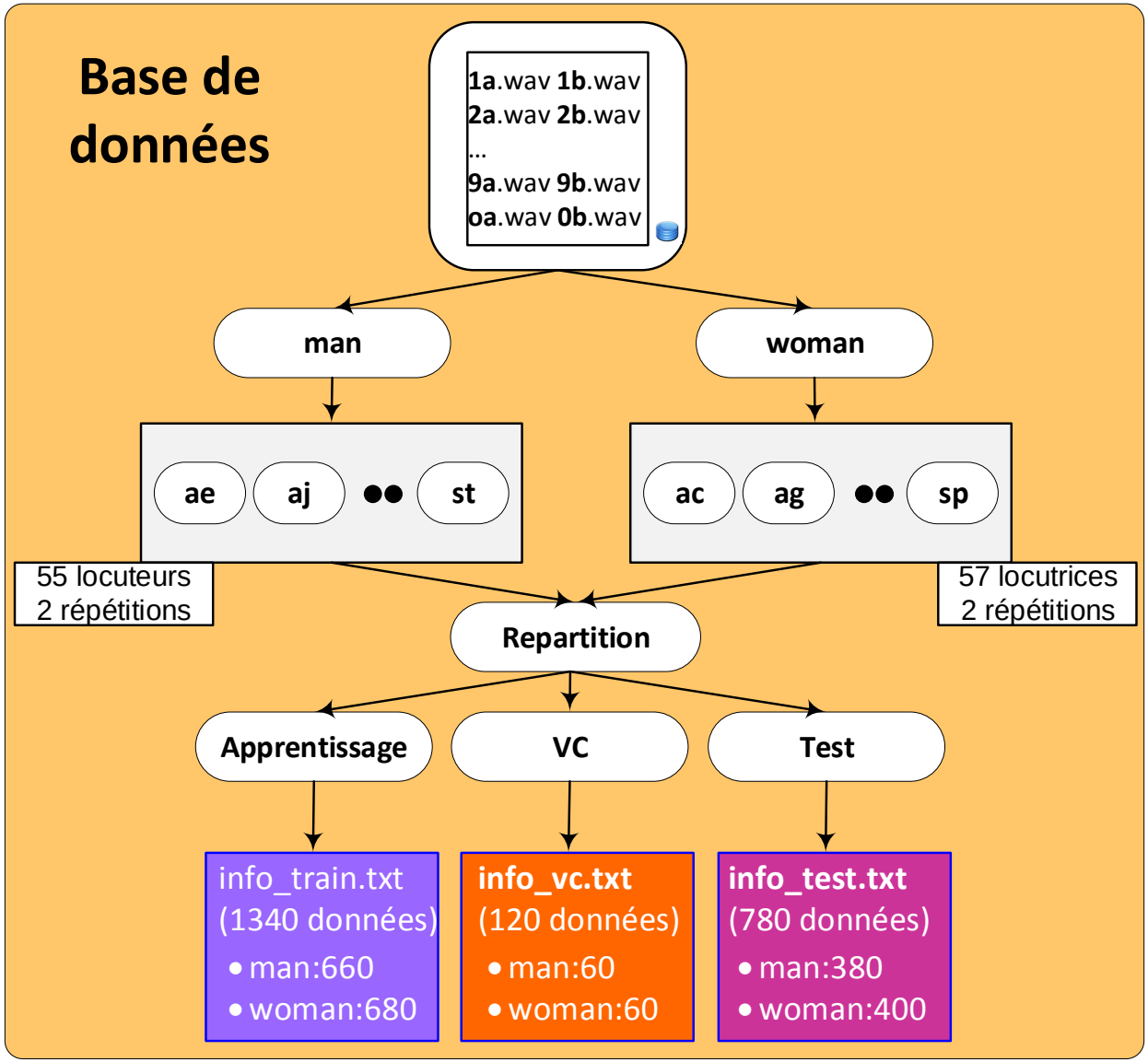



Informations sur la base de données :

- Les données sont réparties en 3 catégories :

1) Entrainement (apprentissage)
2) Validation croisée
3) Test de généralisation

- La répartition des données se trouve dans le répertoire **info_data** :
  - **info_train.txt** : contient la liste des données (fichiers) pour l'apprentissage
  - **info_vc.txt** : contient la liste des fichiers pour la validation croisée
  - **info_test.txt** : contient la liste des fichiers pour le test de généralisation


Base de données
1a.wav 1b.wav
2a.wav 2b.wav
...
9a.wav 9b.wav
0a.wav 0b.wav
man
woman
ae
aj
st
ac
ag
sp
55 locuteurs
2 répétitions
57 locutrices
2 répétitions
Repartition
Apprentissage
VC
Test
info_train.txt
(1340 données)
• man:660
• woman:680
info_vc.txt
(120 données)
• man:60
• woman:60
info_test.txt
(780 données)
• man:380
• woman:400

Entête (header)

- info_train.txt
  - 57 train/man/ha/8b.txt
  - 59 train/man/ha/4a.txt
  - 59 train/man/ha/ob.txt
  - 61 train/man/aj/2b.txt
  - 61 train/man/it/2b.txt
  - 62 train/man/aj/1b.txt
  - 62 train/man/ha/2b.txt
  - 62 train/woman/eg/8b.txt
  - 63 train/man/ha/oa.txt
  - 65 train/man/aj/1a.txt
  - 65 train/man/aj/3b.txt
  - 65 train/man/ha/4b.txt
  - 65 train/man/it/2a.txt

```
NIST_1A   1024
database_id -s8 TIDIGITS
database_version -s3 1.0
utterance_id -s6 ha_8_b
channel_count -i 1
sample_count -i 11776
sample_rate -i 20000
sample_min -i -1659
sample_max -i 2364
sample_n_bytes -i 2
sample_byte_format -s2 01
sample_sig_bits -i 16
speaker_id -s2 ha
prompt_code -s1 8
utterance_production -s1 b
recording_date -s11 15-JUL-1982
end_head
```

Nombre de trames : 57

Mot prononcé : 8 (eight)

- info_train.txt

  57 train/man/ha/8b.txt

Ligne 1 (trame 1)
-6.785384e+00 -2.439699e+00 -8.893854e+00 -6.788761e+00 5.160503e+00 3.697916e+00 7.243154e+00 2.016473e+00 4.590659e+00 6.813693e+00 -2.607599e+00 4.110450e+00 -1.134814e-01 -1.147757e-01 3.345045e-01 1.698681e-01 5.829000e-01 -7.146162e-01 -1.040522e+00 -1.475640e+00 -6.800541e-01 8.455756e-02 -1.175227e+00 -4.345238e-01 -5.065219e-01 8.969450e-03

Ligne 2 (trame 2)
-7.583665e+00 -1.479346e+00 -8.630378e+00 -2.991285e+00 4.616855e+00 4.277121e-01 1.831902e+00 -2.219958e+00 4.970704e+00 7.269855e+00 -1.422976e+00 2.214219e+00 -8.479893e-02 -4.761591e-01 -1.880742e-02 1.276574e-01 8.844274e-01 -7.473347e-01 -7.809860e-01 -7.301049e-01 -1.371514e+00 -1.439842e-01 3.125768e-01 3.292378e-01 1.056775e-01 9.104836e-03

⋮

Ligne 57 (trame 57)
-5.001660e+00 3.509410e+00 -3.024056e+00 2.012576e+00 9.698270e+00 2.662062e+00 3.227849e+00 -4.134442e+00 6.171534e+00 4.405443e+00 -6.717266e+00 -2.741660e-01 -8.662212e-02 -4.564412e-01 -2.203409e-01 5.941224e-01 1.189348e+00 9.793396e-01 -3.815369e-01 1.014225e-01 -6.284602e-01 8.625444e-01 -8.555050e-02 4.655319e-01 2.363525e-01 -8.272087e-03

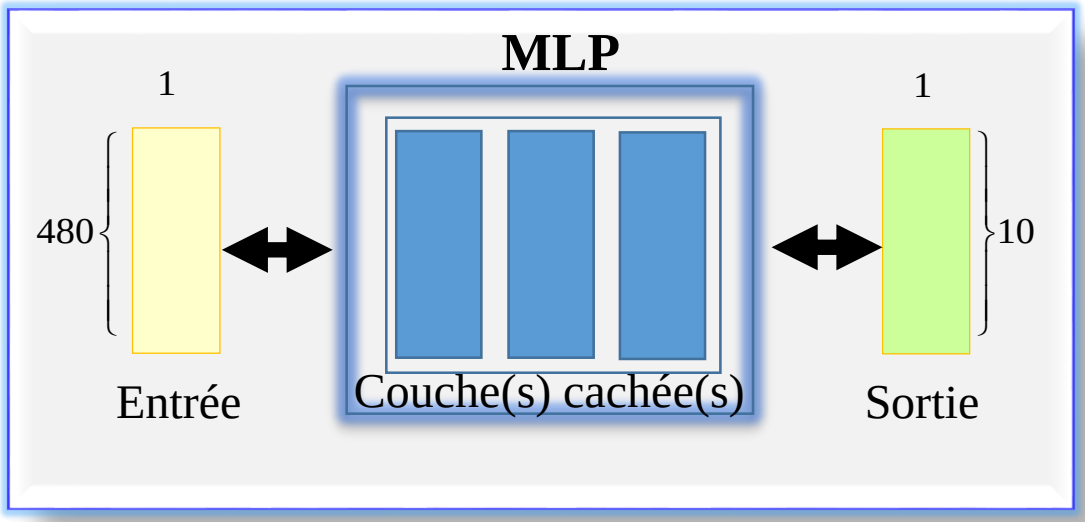
Avant conversion

- info_train.txt
  - 57 train/man/ha/8b.txt
  - 59 train/man/ha/4a.txt
  - 59 train/man/ha/ob.txt
  - 61 train/man/aj/2b.txt
  - 61 train/man/it/2b.txt
  - 62 train/man/aj/1b.txt
  - 62 train/man/ha/2b.txt
  - 62 train/woman/eg/8b.txt
  - 63 train/man/ha/oa.txt
  - 65 train/man/aj/1a.txt
  - 65 train/man/aj/3b.txt

Après conversion

- info_train.txt
  - 40 train/man/ha/8b.txt
  - 40 train/man/ha/4a.txt
  - 40 train/man/ha/ob.txt
  - 40 train/man/aj/2b.txt
  - 40 train/man/it/2b.txt
  - 40 train/man/aj/1b.txt
  - 40 train/man/ha/2b.txt
  - 40 train/woman/eg/8b.txt
  - 40 train/man/ha/oa.txt
  - 40 train/man/aj/1a.txt
  - 40 train/man/aj/3b.txt

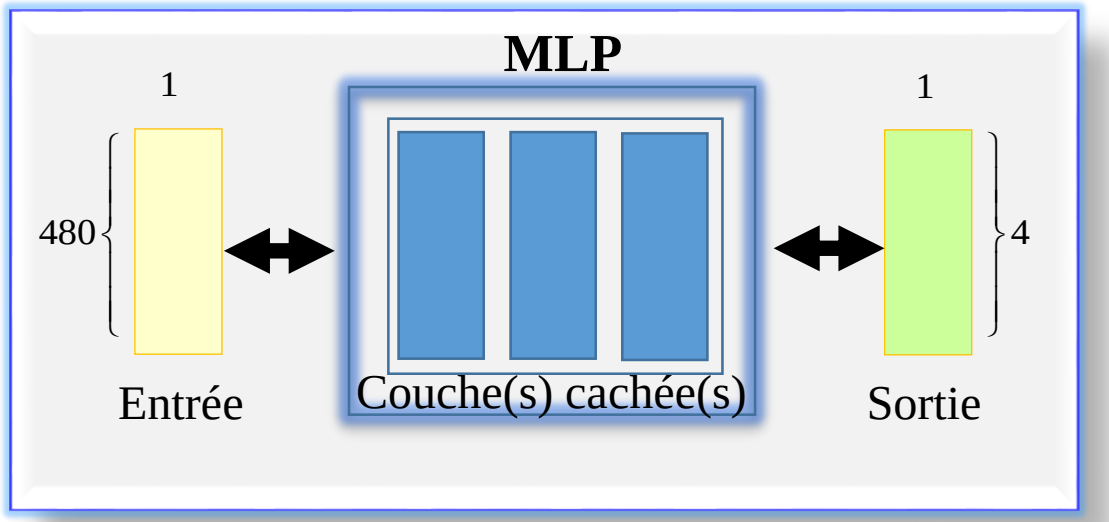
NB : Chaque fichier est une entrée.




**Système de reconnaissance de la parole**

```
NbreMots:  10
NbreOutputs:  10
1:  1 0 0 0 0 0 0 0 0 0
2:  0 1 0 0 0 0 0 0 0 0
3:  0 0 1 0 0 0 0 0 0 0
4:  0 0 0 1 0 0 0 0 0 0
5:  0 0 0 0 1 0 0 0 0 0
6:  0 0 0 0 0 1 0 0 0 0
7:  0 0 0 0 0 0 1 0 0 0
8:  0 0 0 0 0 0 0 1 0 0
9:  0 0 0 0 0 0 0 0 1 0
o:  0 0 0 0 0 0 0 0 0 1
```

**Important :** l'application est indépendante du système (NN dans notre cas)




**Système de reconnaissance de la parole**

NbreMots: 10
NbreOutputs: 4
1: 0 0 0 1
2: 0 0 1 0
3: 0 0 1 1
4: 0 1 0 0
5: 0 1 0 1
6: 0 1 1 0
7: 0 1 1 1
8: 1 0 0 0
9: 1 0 0 1
o: 1 1 0 0

**Quel est le meilleur des deux code de sortie ?**

```
NbreMots:  10
NbreOutputs:  10
1:  1  0  0  0  0  0  0  0  0  0
2:  0  1  0  0  0  0  0  0  0  0
3:  0  0  1  0  0  0  0  0  0  0
4:  0  0  0  1  0  0  0  0  0  0
5:  0  0  0  0  1  0  0  0  0  0
6:  0  0  0  0  0  1  0  0  0  0
7:  0  0  0  0  0  0  1  0  0  0
8:  0  0  0  0  0  0  0  1  0  0
9:  0  0  0  0  0  0  0  0  1  0
o:  0  0  0  0  0  0  0  0  0  1
```

```
NbreMots:  10
NbreOutputs:  4
1:  0  0  0  1
2:  0  0  1  0
3:  0  0  1  1
4:  0  1  0  0
5:  0  1  0  1
6:  0  1  1  0
7:  0  1  1  1
8:  1  0  0  0
9:  1  0  0  1
o:  1  1  0  0
```

- Chaque entrée doit avoir une taille fixe
- Si on veut utiliser 40 lignes et ne garder que les paramètres statiques :
  - Chaque forme d'entrée aurait une dimension de 40 x 12 = 480
  - Trouver une façon d'éliminer les lignes les moins significatives :
    1. utilisez l'énergie $E_s$
    2. utilisez l' énergie dynamique $E_d$
    3. toute autre méthode d'interpolation et/ou extrapolation

statique
E(statique)
dynamique
E(dynamique)

train/man/ha/8b.txt

-6.785384e+00 -2.439699e+00 -8.893854e+00 -6.788761e+00
5.160503e+00 3.697916e+00 7.243154e+00 2.016473e+00 4.590659e+00
6.813693e+00 -2.607599e+00 4.110450e+00 **-1.134814e-01** -1.147757e-
01 3.345045e-01 1.698681e-01 5.829000e-01 -7.146162e-01 -
1.040522e+00 -1.475640e+00 -6.800541e-01 8.455756e-02 -
1.175227e+00 -4.345238e-01 -5.065219e-01 **8.969450e-03**

-7.583665e+00 -1.479346e+00 -8.630378e+00 -2.991285e+00
4.616855e+00 4.277121e-01 1.831902e+00 -2.219958e+00 4.970704e+00
7.269855e+00 -1.422976e+00 2.214219e+00 **-8.479893e-02** -4.761591e-
01 -1.880742e-02 1.276574e-01 8.844274e-01 -7.473347e-01 -
7.809860e-01 -7.301049e-01 -1.371514e+00 -1.439842e-01 3.125768e-
01 3.292378e-01 1.056775e-01 **9.104836e-03**

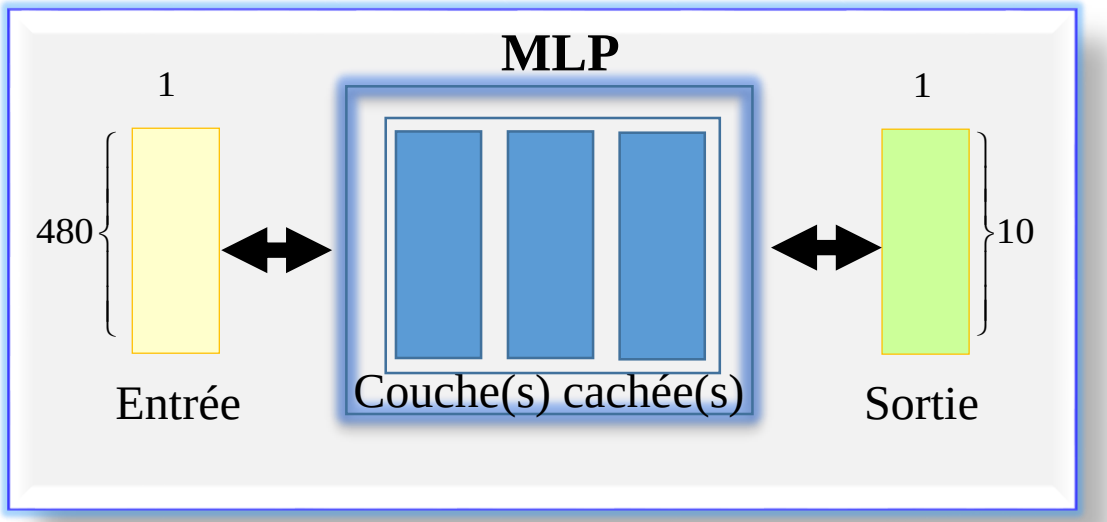
**Formulation :**

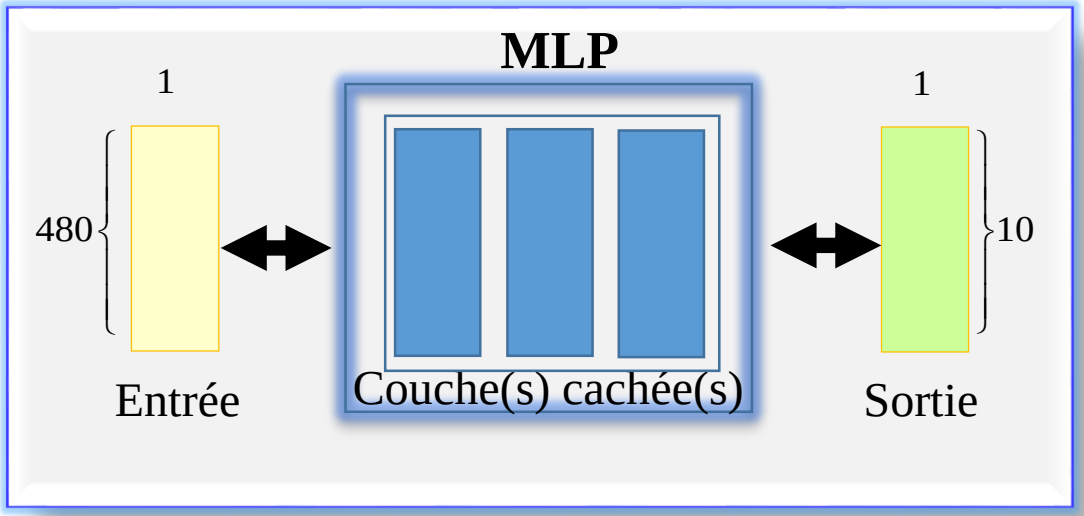
- Chaque forme d'entrée aura une dimension de 40 ligne x 12 statiques = 480 paramètres (former **1 vecteur colonne**)

1 Fichier = 1 Entrée de Dim 480

- Définir la taille du réseau de neurones (nombre de couches, nombre de neurones par couche)
- Définir particulièrement le nombre de neurones à la sortie : le plus souvent 10 neurones
- Définir le code de sortie
- Définir la fonction d'activation, le taux d'apprentissage
- Initialiser aléatoirement les poids [-0.1, +0.1]

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 5: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 6: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 7: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 9: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| o: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |






**Fonctionnement :**

Définir K, le nombre de cycles d'apprentissage pour faire le test de validation croisée.

1) Piger une entrée au hasard (eg. train/man/ha/4a.txt)
2) Dérouler l'algorithme d'apprentissage (4 phases)
3) Trouver la sortie des 10 neurones, déterminer le code correspondant
   a) Réussite : ne rien faire
   b) Échec : faire l'apprentissage (règle de delta généralisée)
4) Critère d'arrêt atteint : FIN, sinon aller au 1).

**info_train.txt** (1)
40 train/man/ha/8b.txt
40 train/man/ha/4a.txt
40 train/man/ha/ob.txt
⋮
→ aléatoire

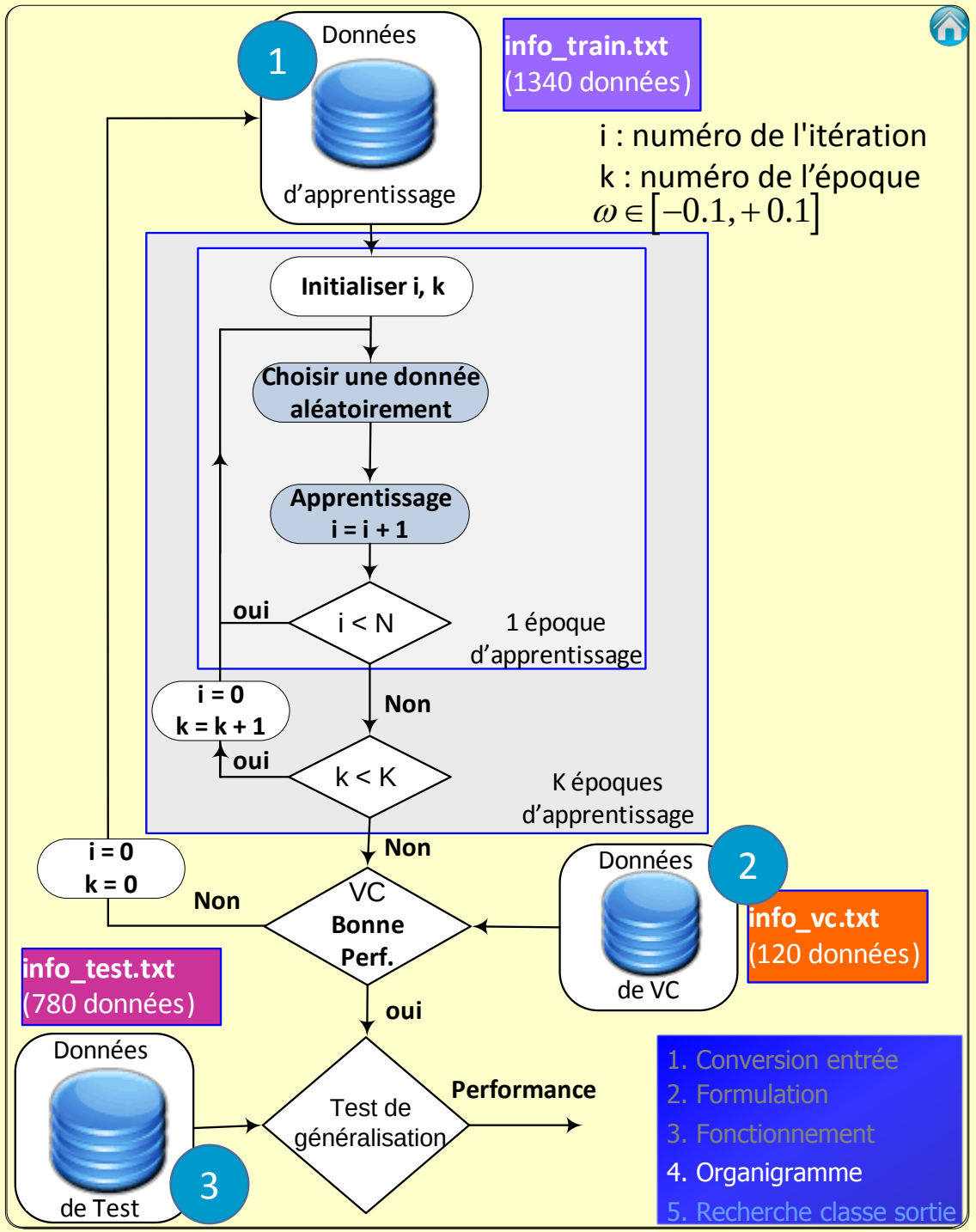
**info_VC.txt** (2)
40 vc/man/kd/2a.txt
40 vc/man/jt/2b.txt
40 vc/man/jt/ob.txt
⋮
→ séquentiel

**info_test.txt** (3)
40 test/man/nr/8b.txt
40 test/woman/ng/8a.txt
40 test/man/rd/8a.txt
⋮
→ séquentiel

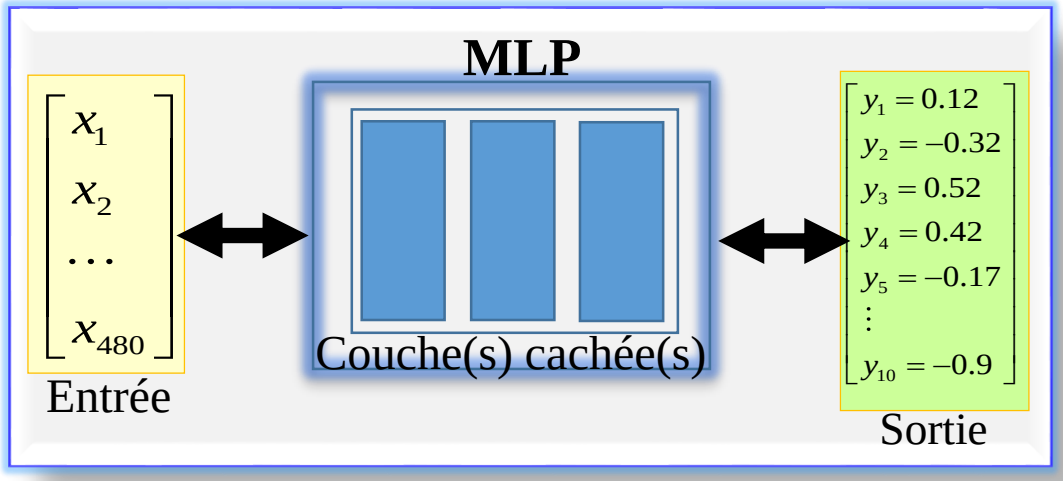
**N** = nombre de données disponibles pour l'apprentissage

1 cycle (époque) = N données disponibles pour l'apprentissage

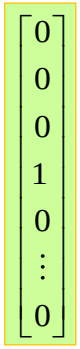
**K** = nombre de cycles d'apprentissage avant la VC



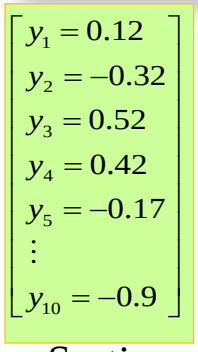
train/man/ha/4a.txt




MAX ?

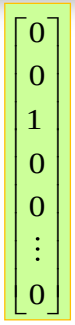



Sortie désirée (4)



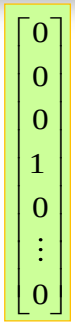

Sortie réelle

MAX

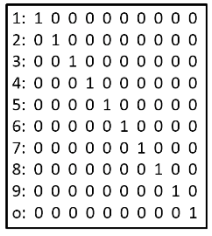



Résultat de la fonction MAX (3)

≠



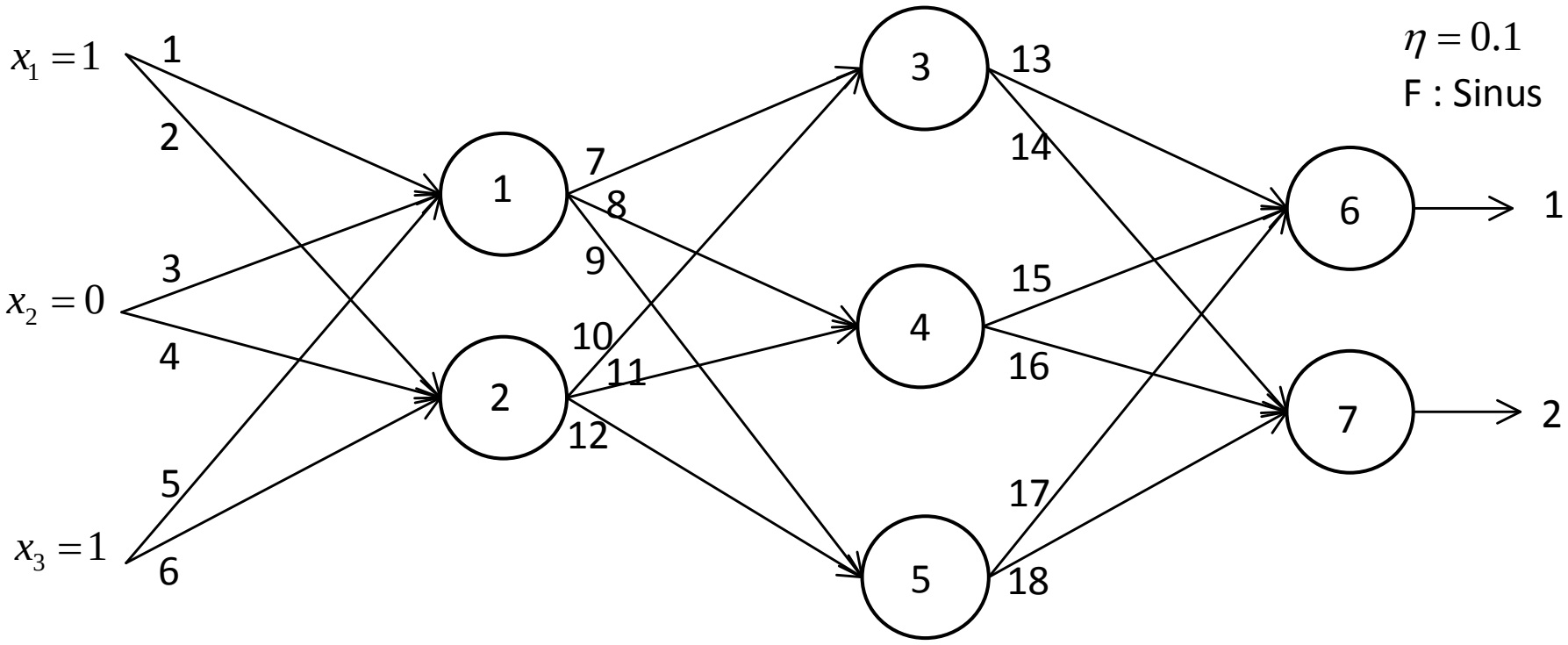

Sortie désirée (4)




- La sortie réelle étant différente de la sortie désirée ⇒ Apprendre la donnée ha/4a.txt
- **Important :** lors de la mise à jour des poids, on doit utiliser les valeurs de la sortie réelle, et non pas le résultat de la fonction MAX.

# Quelques démonstrations

# Exercice♦




# Apprentissage

Trouver les nouvelles valeurs des poids du réseau en appliquant la règle de delta généralisée

♦ Cet exercice (diapositive) n'est pas à la bonne place ! C'est juste pour ne pas modifier la pagination de vos notes de cours !

# Transformée en cosinus discrète (DCT)

- DCT est une transformation orthogonale comme la DFT, avec coefficients réels.
- Assomption de la périodicité plus réaliste que la DFT !!!
- Propriété de compaction de l'énergie : d'où son utilisation en compression

$$1\mathrm{D}: \; X(k) = \alpha(k) \sum_{n=0}^{N-1} x[n] cos\left(\frac{(2n+1)k\pi}{2N}\right), \qquad 0 \le k \le N-1$$

$$2\mathrm{D}: \; X(k_1, k_2) = \alpha(k_1)\alpha(k_2) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x[n_1, n_2]\, r(n_1, n_2, k_1, k_1)$$