



JOURNEES
TECHNIQUES
ACOUSTIQUE
ET VIBRATIONS



Estimation du coefficient d'absorption acoustique moyen par des méthodes de machine learning

Lille 11 – 12 Mars 2020

Corto Bastien

Sorbonne Université, stage de Master

Antoine Deleforge

Inria – Equipe Multispeech – Nancy

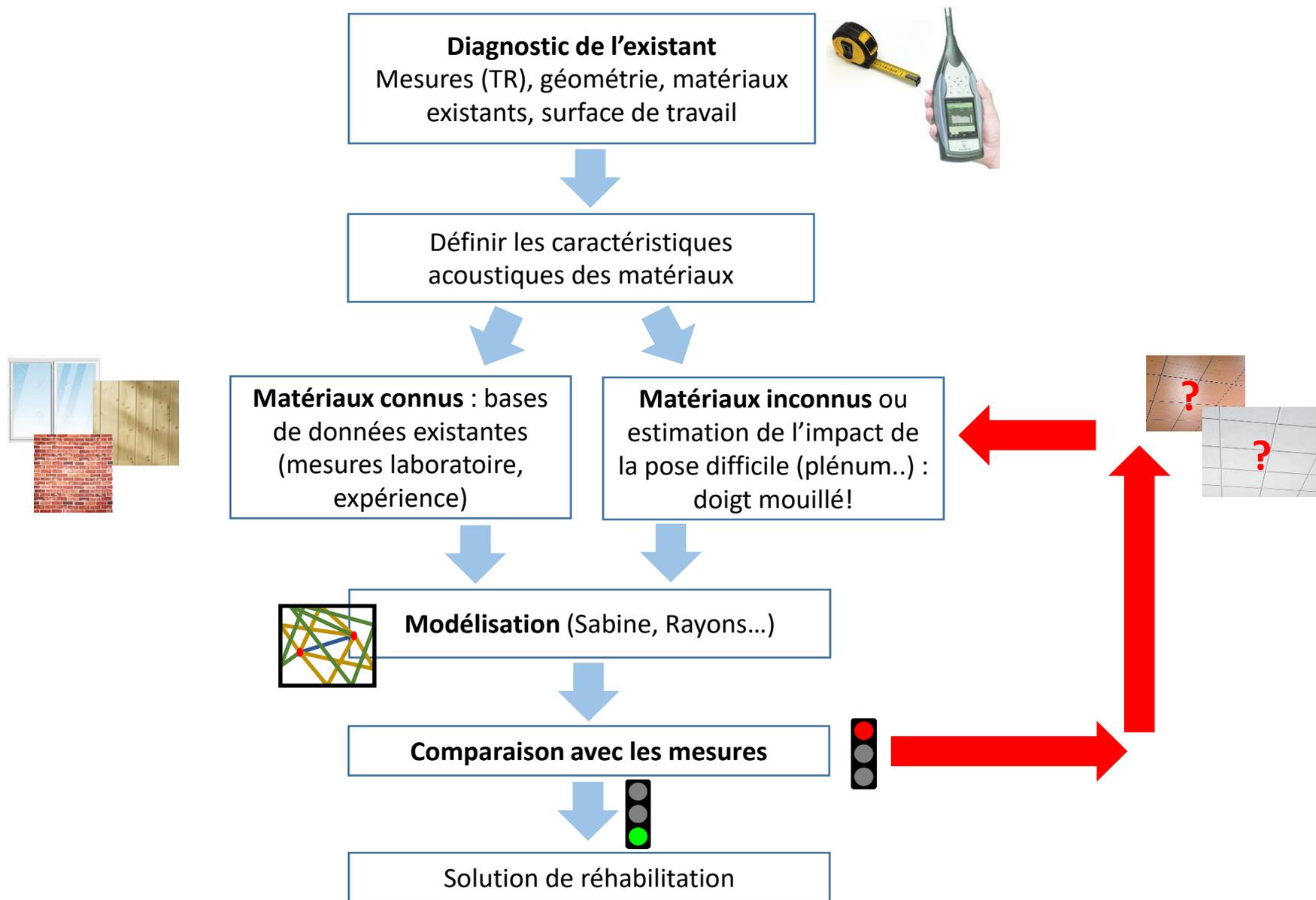
Cédric Foy

UMRAE – Strasbourg



La réhabilitation acoustique du bâtiment

Méthodologie actuelle - Problématique générale



Problème de physique inverse non-linéaire :

Est – il possible, en utilisant des **mesures ponctuelles** de champ sonore obtenues à partir de microphones dans une salle, d'estimer précisément les **paramètres acoustiques et géométriques** d'entrée de cette salle par **apprentissage automatisé** préalable sur larges jeux de données ?

Définition du réseau de neurones

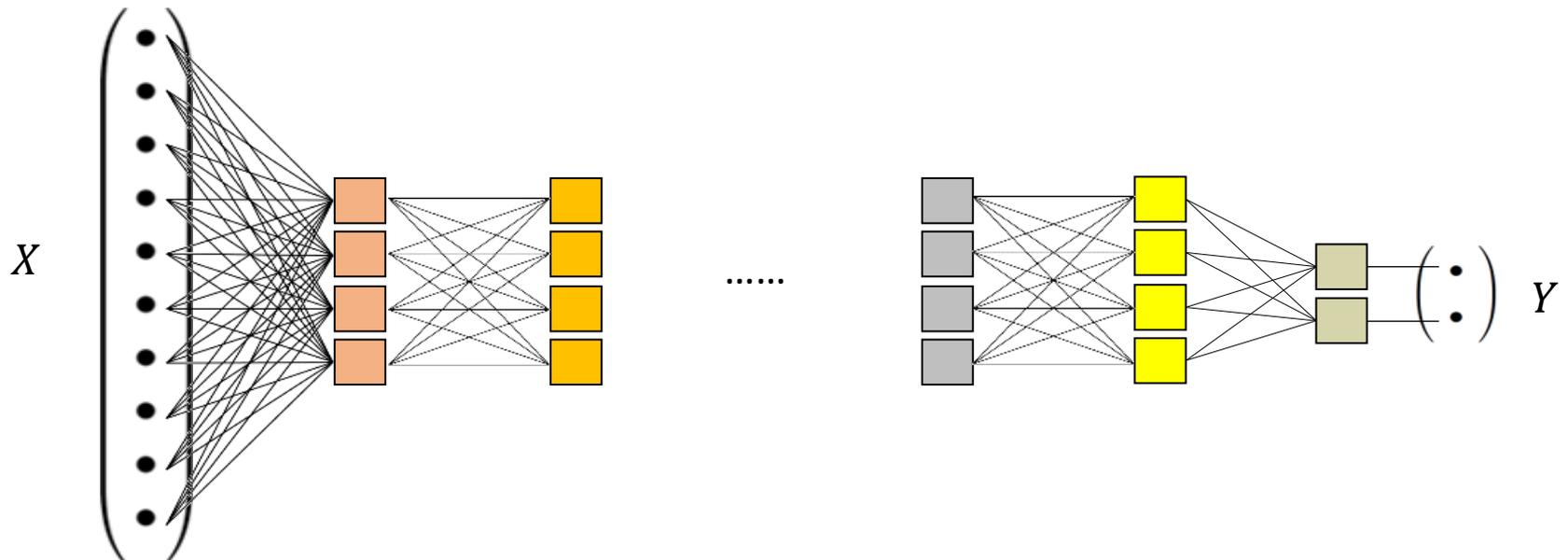
Principe - Poids et Biases - Entrée des neurones - Sortie des neurones

- Un grand nombre de données disponibles (analytique, numérique, empirique, mesures..)
- N couples de données $(X, Y) = [(\dots x_k \dots), (\dots y_k \dots)]$

Couche d'entrée (e)

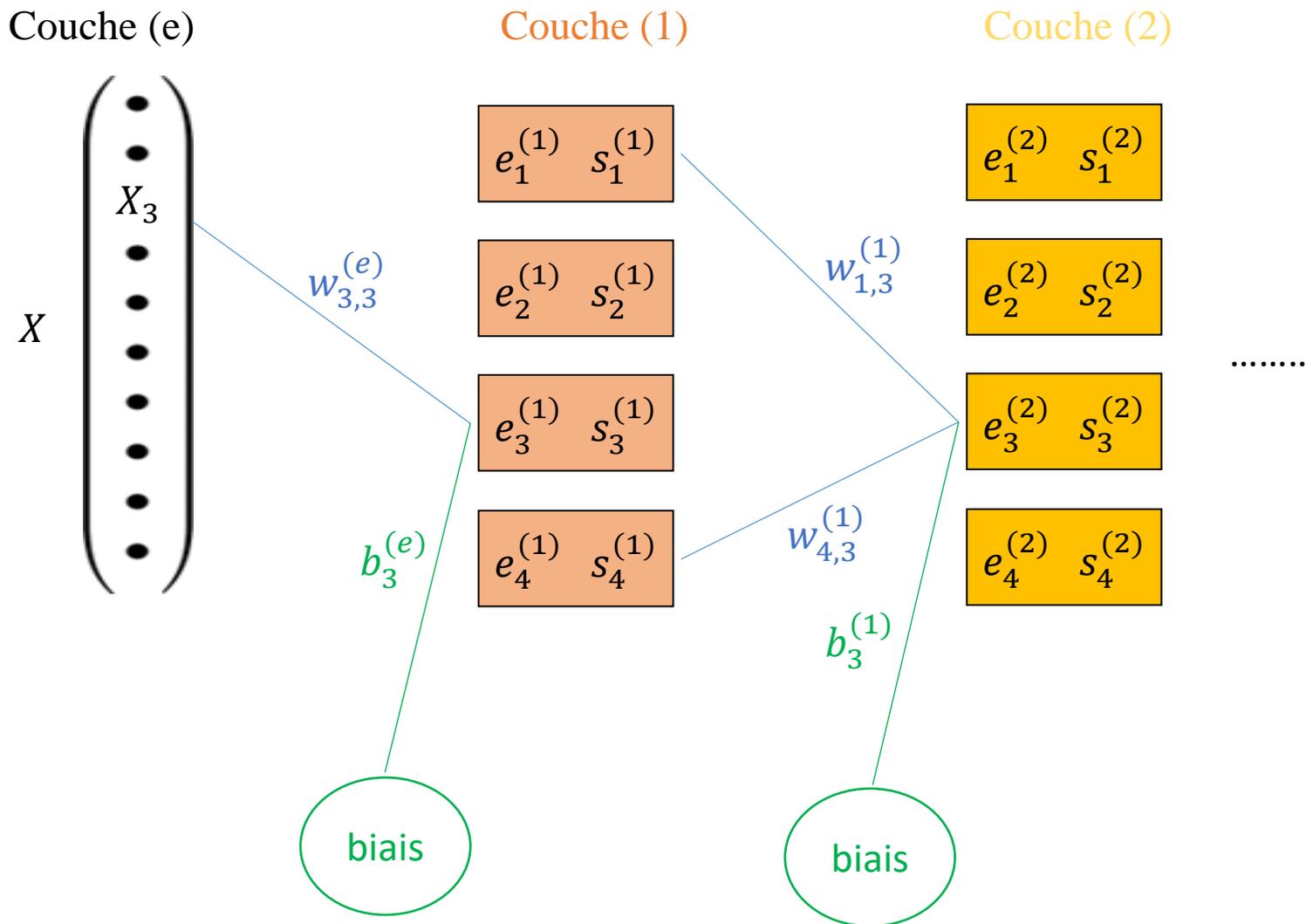
couches intermédiaires

couche de sortie (s)



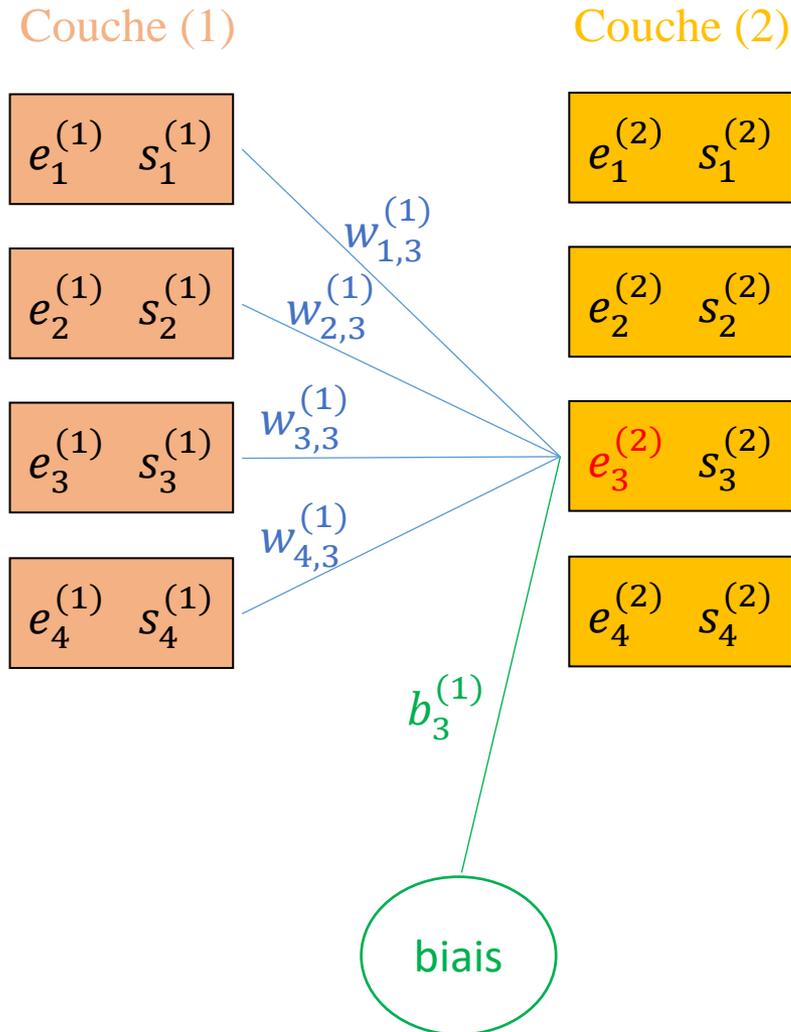
Définition du réseau de neurones

Principe - Poids et Biases - Entrée des neurones - Sortie des neurones



Définition du réseau de neurones

Principe - Poids et Biais - Entrée des neurones - Sortie des neurones



$$e_3^{(2)} = w_{1,3}^{(1)} s_1^{(1)} + w_{2,3}^{(1)} s_2^{(1)} + w_{3,3}^{(1)} s_3^{(1)} + w_{4,3}^{(1)} s_4^{(1)} + b_3^{(1)}$$

Couche (i) neurone k

$$e_k^{(i+1)} = \sum_j w_{j,k}^{(i)} s_j^{(i)} + b_k^{(i)}$$

Définition du réseau de neurones

Principe - Poids et Biais - Entrée des neurones - Sortie des neurones

Couche (2)

$$e_1^{(2)} \quad s_1^{(2)} \quad s_1^{(2)} = f^{(2)}(e_1^{(2)})$$

$$e_2^{(2)} \quad s_2^{(2)} \quad \vdots$$

$$e_3^{(2)} \quad s_3^{(2)} \quad s_3^{(2)} = f^{(2)}(e_3^{(2)})$$

$$e_4^{(2)} \quad s_4^{(2)}$$

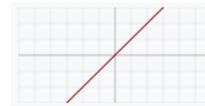
Couche (i) neurone j

$$s_j^{(i)} = f^{(i)}(e_j^{(i)})$$

- La fonction d'activation f est identique sur une couche mais peut varier d'une couche à l'autre

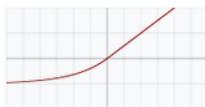
- Exemple de fonctions d'activation :

Fonction identité



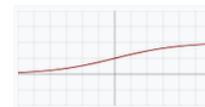
$$f^{(i)}(e_j^{(i)}) = e_j^{(i)}$$

Fonction ELU



$$f^{(i)}(e_j^{(i)}) = \begin{cases} \exp(e_j^{(i)} - 1) & \text{si } e_j^{(i)} < 0 \\ e_j^{(i)} & \text{si } e_j^{(i)} \geq 0 \end{cases}$$

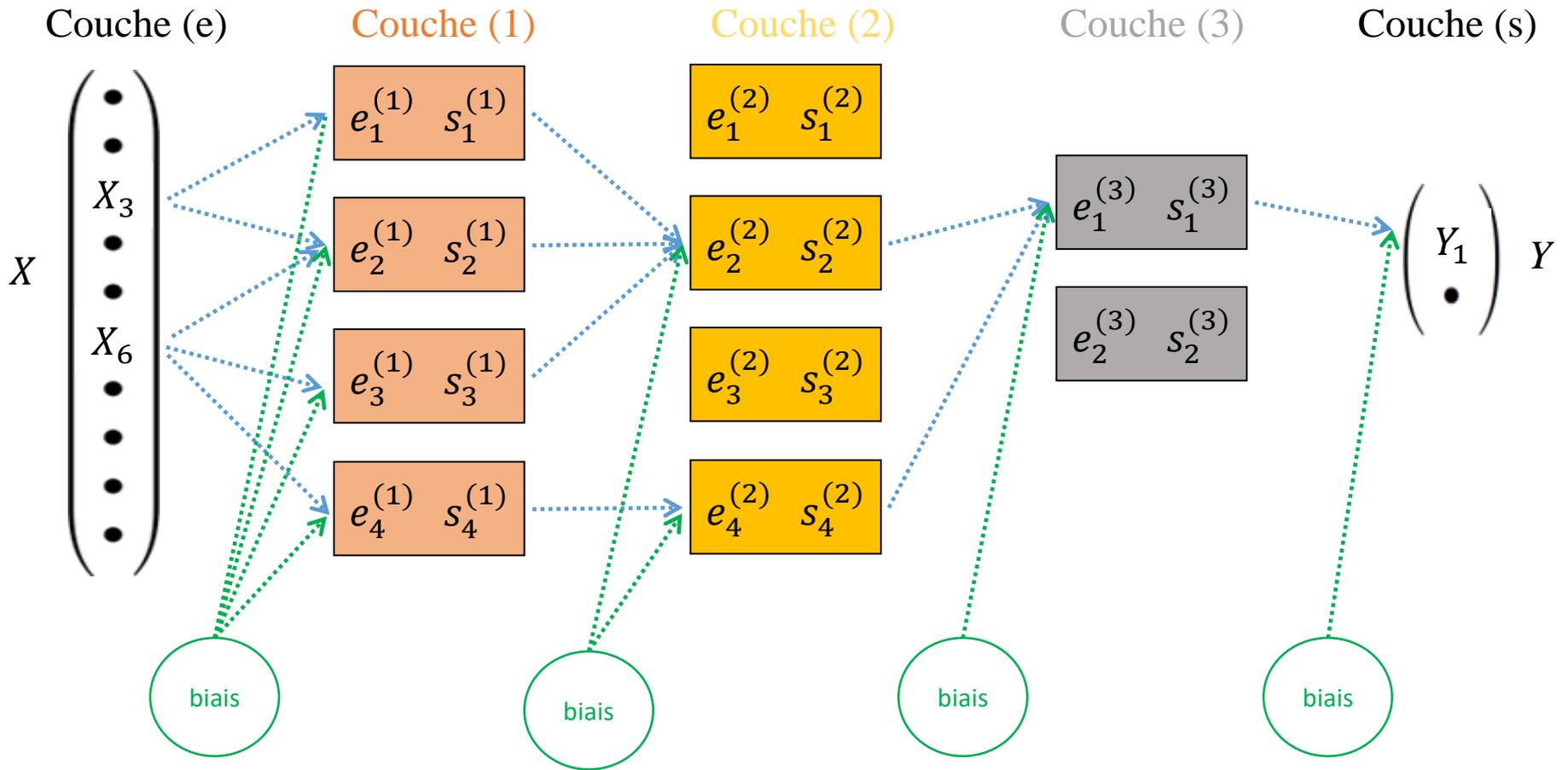
Fonction sigmoïde



$$f^{(i)}(e_j^{(i)}) = \frac{1}{1 + \exp(-e_j^{(i)})}$$

Définition du réseau de neurones

Principe - Poids et Biais - Entrée des neurones - Sortie des neurones



$$\begin{aligned}
 Y_1 &= s_1^{(3)} + b_1^{(3)} = f^{(3)}(e_1^{(3)}) + b_1^{(3)} \\
 &= f^{(3)} \left[w_{2,1}^{(2)} s_2^{(2)} + w_{4,1}^{(2)} s_4^{(2)} + b_1^{(2)} \right] + b_1^{(3)} \\
 &= f^{(3)} \left[w_{2,1}^{(2)} f^{(2)}(e_2^{(2)}) + w_{4,1}^{(2)} f^{(2)}(e_4^{(2)}) + b_1^{(2)} \right] + b_1^{(3)} \\
 &= f^{(3)} \left[w_{2,1}^{(2)} f^{(2)} \left[w_{1,2}^{(1)} s_1^{(1)} + w_{2,2}^{(1)} s_2^{(1)} + w_{3,2}^{(1)} s_3^{(1)} + b_2^{(1)} \right] + w_{4,1}^{(2)} f^{(2)} \left[w_{4,4}^{(1)} s_4^{(1)} + b_4^{(1)} \right] + b_1^{(2)} \right] + b_1^{(3)} \\
 &= f^{(3)} \left[w_{2,1}^{(2)} f^{(2)} \left[w_{1,2}^{(1)} f^{(1)}(e_1^{(1)}) + w_{2,2}^{(1)} f^{(1)}(e_2^{(1)}) + w_{3,2}^{(1)} f^{(1)}(e_3^{(1)}) + b_2^{(1)} \right] + w_{4,1}^{(2)} f^{(2)} \left[w_{4,4}^{(1)} f^{(1)}(e_4^{(1)}) + b_4^{(1)} \right] + b_1^{(2)} \right] + b_1^{(3)} \\
 &= f^{(3)} \left[w_{2,1}^{(2)} f^{(2)} \left[w_{1,2}^{(1)} f^{(1)} \left[X_3 + b_1^{(e)} \right] + w_{2,2}^{(1)} f^{(1)} \left[X_3 + X_6 + b_2^{(e)} \right] + w_{3,2}^{(1)} f^{(1)} \left[X_6 + b_3^{(e)} \right] + b_2^{(1)} \right] + w_{4,1}^{(2)} f^{(2)} \left[w_{4,4}^{(1)} f^{(1)} \left[X_6 + b_4^{(e)} \right] + b_4^{(1)} \right] + b_1^{(2)} \right] + b_1^{(3)} \quad 8
 \end{aligned}$$

Entraînement du réseau de neurones : estimer les poids et des biais

- Hypothèse : réseau de neurones choisi (nombre et tailles des couches, fonctions d'activation)
- Choix aléatoire des poids et des biais de départ
- Utilisation de N_E couples $(X, Y)_E \stackrel{\text{def}}{=} (X, C)_E$ connus (notés C pour « cible »)
- Processus itératif

L'entraînement de la validation d'un réseau de neurones

Principe - Erreur et Processus itératif - Rétro-propagation - Minimisation - Critère d'arrêt

- **itération 1** Pour chacun des N_E couples $(X, C)_E$
 - Estimation de $(Y)_E$ à partir des poids et biais de départ
 - Estimation de l'erreur ε entre les valeurs cibles C et les valeurs de sortie Y (mean squared error)

$$[\varepsilon]_{t=1} = \left[\frac{1}{\dim Y} \sum_{k=1}^{\dim Y} (Y_k - C_k)^2 \right]_{t=1}$$

- **Rétro-propagation** de la dérivée de l'erreur par rapport au poids et la dérivée de l'erreur par rapport au biais
- Choix de nouveaux poids et biais qui **minimisent l'erreur** : **méthode de descente de gradient** basée sur ces dérivées

- itération 2 $(Y)_{t=2} > [\varepsilon]_{t=2} > \text{rétro-propa.} > \text{minimisation}$

.....

- itération τ $(Y)_{t=\tau} > [\varepsilon]_{t=\tau} > \text{rétro-propa.} > \text{minimisation}$

- itération $\tau + 1$ $(Y)_{t=\tau+1} > [\varepsilon]_{t=\tau+1} > \text{rétro-propa.} > \text{minimisation}$

-

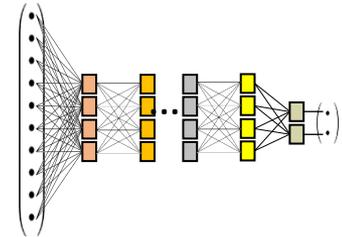
- itération finale

L'entraînement de la validation d'un réseau de neurones

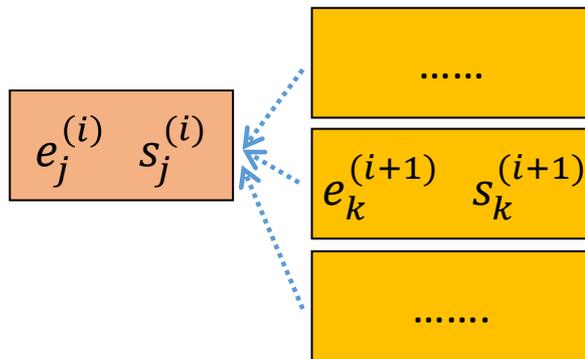
Principe - Erreur et Processus itératif - Rétro-propagation - Minimisation - Critère d'arrêt

- la fonction d'activation de la couche de sortie (s) est la fonction unité

$$\frac{\partial \varepsilon}{\partial e_k^{(s)}} = \frac{\partial \varepsilon}{\partial s_k^{(s)}} = \frac{\partial \varepsilon}{\partial Y_k} = 2(Y_k - C_k)$$



- Rétro-propagation entre deux neurones (entre les neurones de la couche (i+1) et le neurone j de la couche (i))



$$\frac{\partial \varepsilon}{\partial s_j^{(i)}} = \sum_k w_{j,k}^{(i)} \frac{\partial \varepsilon}{\partial e_k^{(i+1)}}$$

- Rétro-propagation entre l'entrée et la sortie d'un neurone (neurone j de la couche (i))



$$\frac{\partial \varepsilon}{\partial e_j^{(i)}} = \frac{\partial \varepsilon}{\partial s_j^{(i)}} \frac{\partial f_j^{(i)}(e_j^{(i)})}{\partial e_j^{(i)}}$$

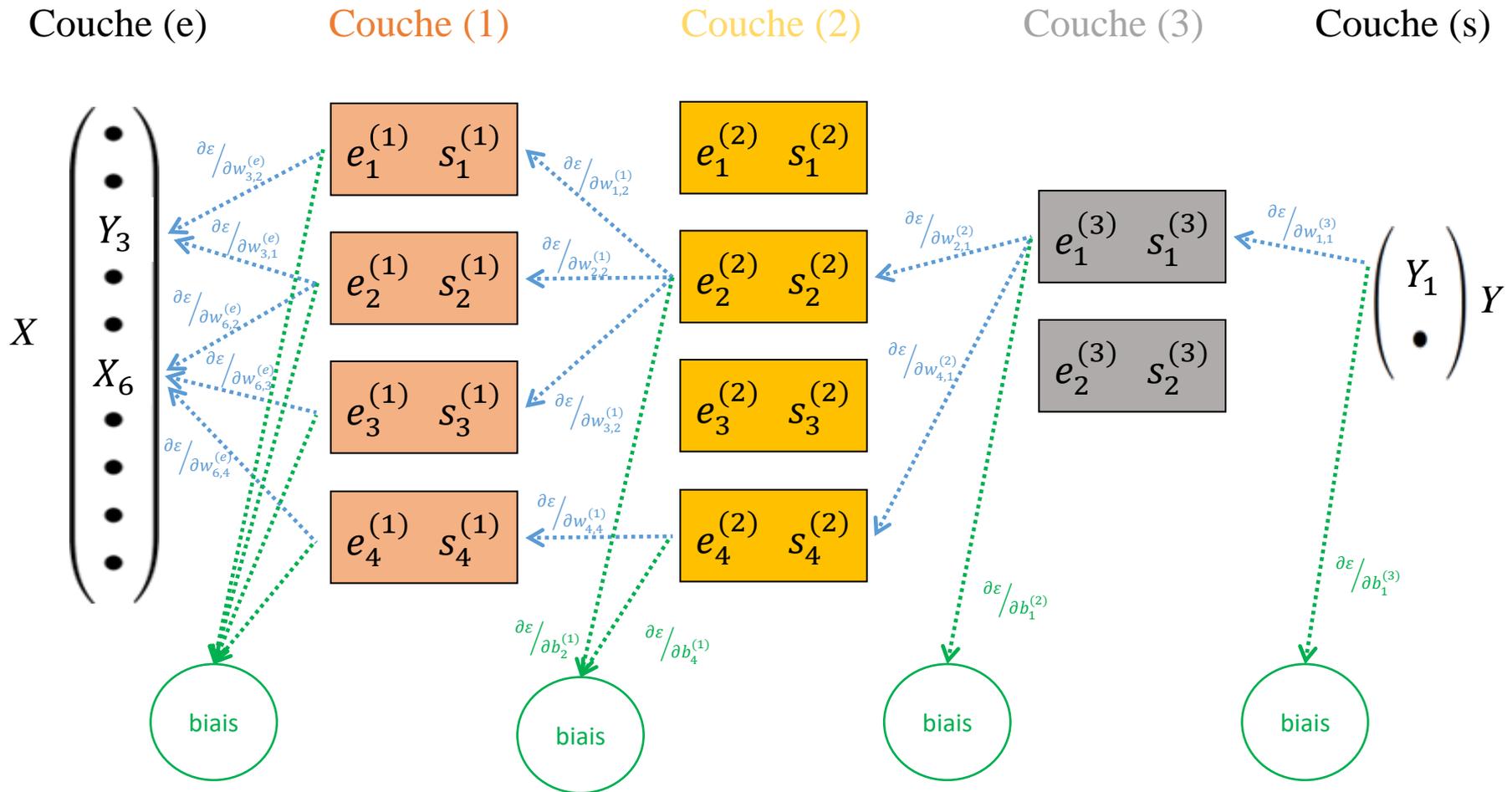
- Finalement nous pouvons estimer les gradients de ε par rapport aux poids et aux biais

$$\frac{\partial \varepsilon}{\partial w_{j,k}^{(i)}} = \frac{\partial \varepsilon}{\partial e_k^{(i+1)}} f_j^{(i)}(e_j^{(i)})$$

$$\frac{\partial \varepsilon}{\partial b_k^{(i)}} = \frac{\partial \varepsilon}{\partial e_k^{(i+1)}}$$

L'entraînement de la validation d'un réseau de neurones

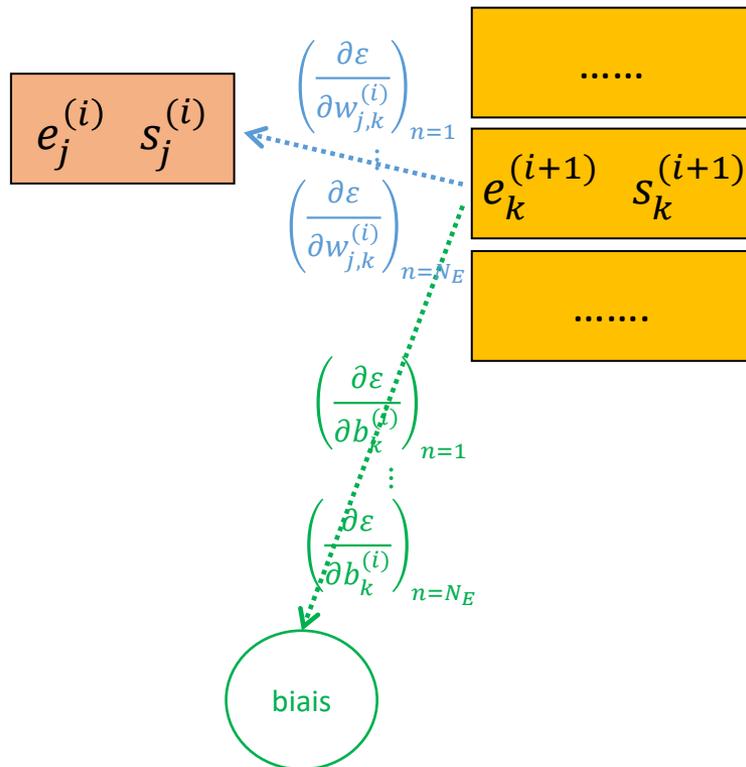
Principe - Erreur et Processus itératif - Rétro-propagation - Minimisation - Critère d'arrêt



L'entraînement de la validation d'un réseau de neurones

Principe - Erreur et Processus itératif - Rétro-propagation - Minimisation - Critère d'arrêt

La méthode de descente de gradient



- Calcul des gradients de l'erreur ε pour les N_E couples $(X, C)_E$
- Calcul de la moyenne de ces gradients

$$\overline{\partial \varepsilon_w} = \frac{1}{N_E} \sum_n \left(\frac{\partial \varepsilon}{\partial w_{j,k}^{(i)}} \right)_n \quad \overline{\partial \varepsilon_b} = \frac{1}{N_E} \sum_n \left(\frac{\partial \varepsilon}{\partial b_k^{(i)}} \right)_n$$

- Le gradient pointant dans la direction de plus forte croissance de l'erreur, on déplace le poids ou le biais dans la direction opposée pour diminuer l'erreur

$$[w_{j,k}^{(i)}]_{\tau+1} = [w_{j,k}^{(i)} - \alpha \overline{\partial \varepsilon_w}]_{\tau}$$

$$[b_k^{(i)}]_{\tau+1} = [b_k^{(i)} - \alpha \overline{\partial \varepsilon_b}]_{\tau}$$

- α : taux d'apprentissage

L'entraînement de la validation d'un réseau de neurones

Principe - Erreur et Processus itératif - Rétro-propagation - Minimisation - Critère d'arrêt

Entraînement sur les N_E couples $(X, C)_E$

itération 1
$$[\bar{\varepsilon}]_{t=1} = \left[\frac{1}{N_E} \sum_n (\varepsilon)_n \right]_{t=1}$$

itération 2
$$[\bar{\varepsilon}]_{t=2} = \left[\frac{1}{N_E} \sum_n (\varepsilon)_n \right]_{t=2}$$

itération 3
$$[\bar{\varepsilon}]_{t=3} = \left[\frac{1}{N_E} \sum_n (\varepsilon)_n \right]_{t=3}$$

.....
$$[\bar{\varepsilon}]_{t=31} = \left[\frac{1}{N_E} \sum_n (\varepsilon)_n \right]_{t=31}$$

Risque de sur-apprentissage du réseau !
Nécessité d'un couple de données
supplémentaires (validation) !

Validation sur les N_V couples $(X, C)_V$

- estimation de $(Y)_V$ à partir des poids et biais calculés sur l'entraînement
- Estimation de l'erreur moyenne sur les N_V couples pour les itérations

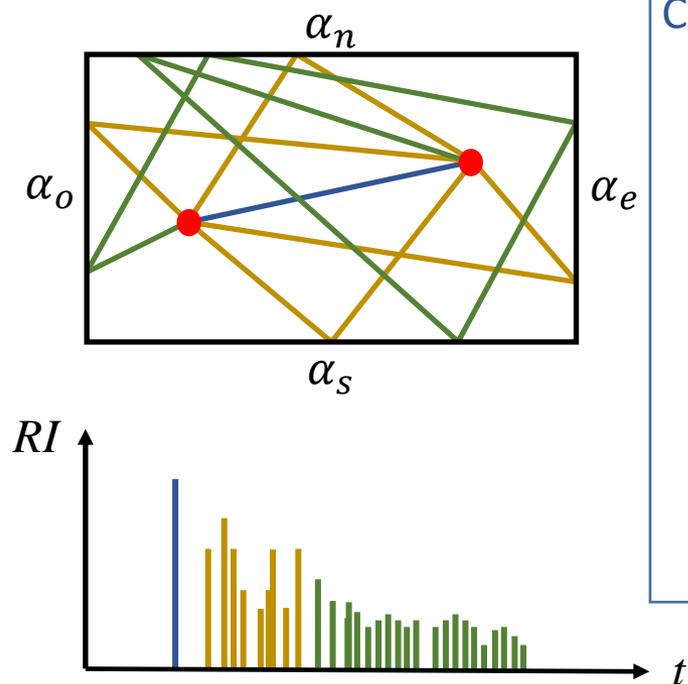
$$[\bar{\varepsilon}']_{t=1} \quad [\bar{\varepsilon}']_{t=2} \quad [\bar{\varepsilon}']_{t=3} \quad \dots \quad [\bar{\varepsilon}']_{t=\tau-3} \quad [\bar{\varepsilon}']_{t=\tau-2} \quad [\bar{\varepsilon}']_{t=\tau-1} \quad [\bar{\varepsilon}']_{t=\tau} \quad \dots$$

$$\frac{([\bar{\varepsilon}']_{t=\tau-3}; [\bar{\varepsilon}']_{t=\tau-1}) - ([\bar{\varepsilon}']_{t=\tau-2}; [\bar{\varepsilon}']_{t=\tau})}{([\bar{\varepsilon}']_{t=\tau-3}; [\bar{\varepsilon}']_{t=\tau-1})} < 0.01$$

Application de l'AI à l'acoustique du bâtiment

Base de données - Réseau et Réverbération - Cas d'études

Est-il possible de retrouver l'absorption moyenne des parois d'une salle à partir de la connaissance d'une seule de ces réponses impulsionnelles ?



Constitution des jeux de données

- Logiciel Roomsim
 - Sources-images + Rayons
 - RI : échogramme + hypothèse de phase minimum)
- Normalisation du champ direct
- Filtrage à 1000 Hz
- Echantillonnage (1024 points)
- Salles [1.5-10 : 1.5-10 : 2.5-4]
- Absorption surfacique : bases réelles (sol, murs et plafond)
- Diffusivité $s=0.1$ (comportement déterministe)
- Positions source et récepteur aléatoires (>0.5 m des parois)

Entraînement

$$N_E=15000 \text{ couples } (X, C)_E = (RI, \bar{\alpha}(\alpha_i, S_i))_E$$

Validation

$$N_V=5000 \text{ couples } (X, C)_V = (RI, \bar{\alpha}(\alpha_i, S_i))_V$$

Test

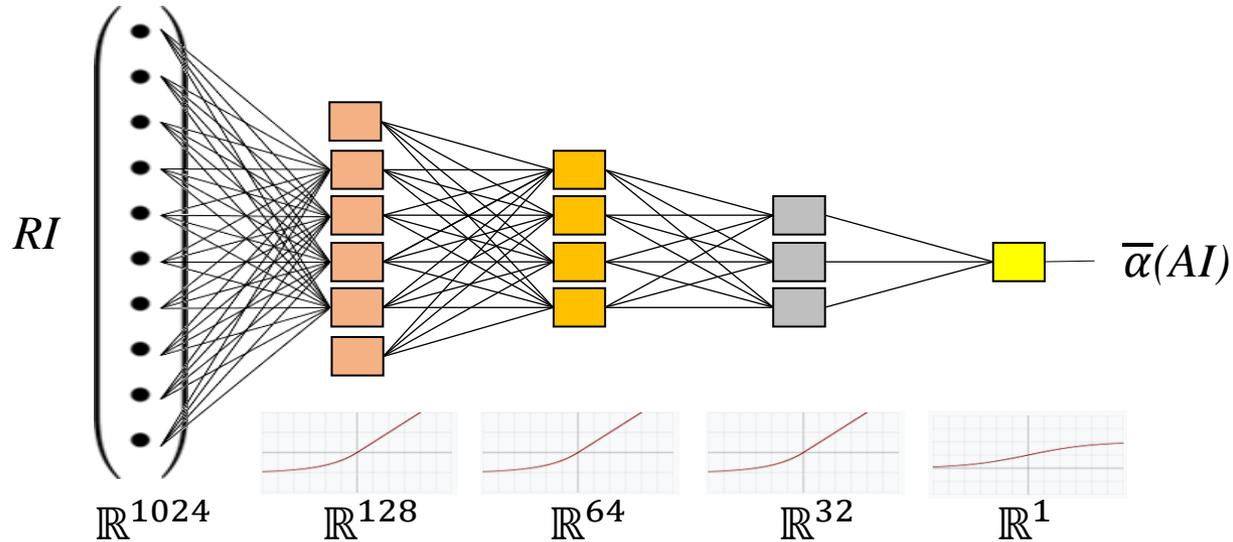
$$N=5000 \text{ couples } (X, C) = (RI, \bar{\alpha}(\alpha_i, S_i))$$

$$\bar{\alpha}(\alpha_i, S_i) = \frac{\sum_i \alpha_i S_i}{\sum_i S_i}$$

Application de l'AI à l'acoustique du bâtiment

Base de données - Réseau et Réverbération - Cas d'études

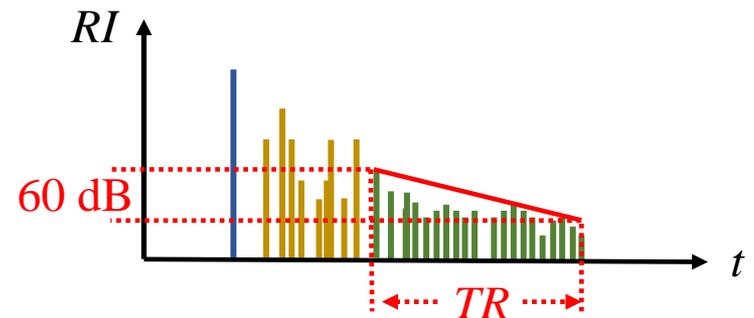
Perceptron : fully-connected, 3 couches « ELU », 1 couche « Sigmoide »



- Comparaison $(Y, C) = (\bar{\alpha}(AI), \bar{\alpha}(\alpha_i, S_i))$
- Absorption moyenne issue de la théorie de la réverbération
 - Estimation du TR via la RI (ISO 3382-1)
 - $\bar{\alpha}(TR)$:

$$TR = -\frac{0.16V}{S \ln(1 - \bar{\alpha}(TR))}$$

- $(\bar{\alpha}(TR), \bar{\alpha}(\alpha_i, S_i))$
- $\bar{\alpha}(AI)$: 1 paramètre RI $\bar{\alpha}(TR)$: 3 paramètres S, V, TR

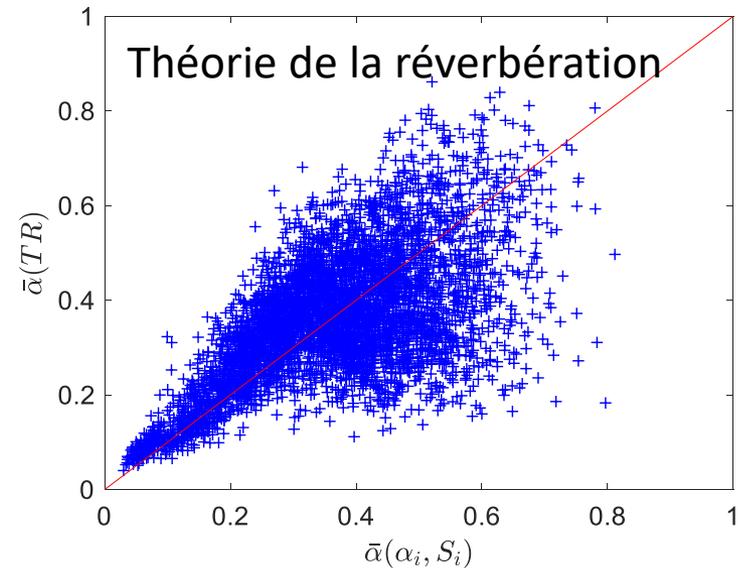
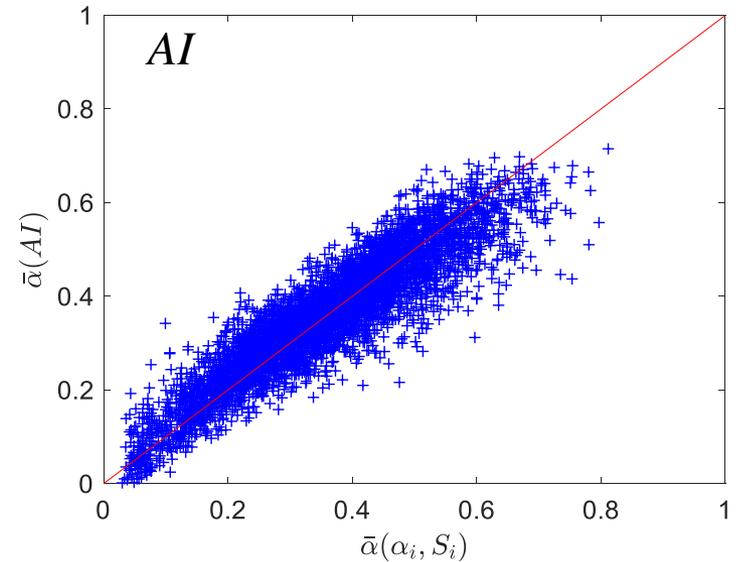
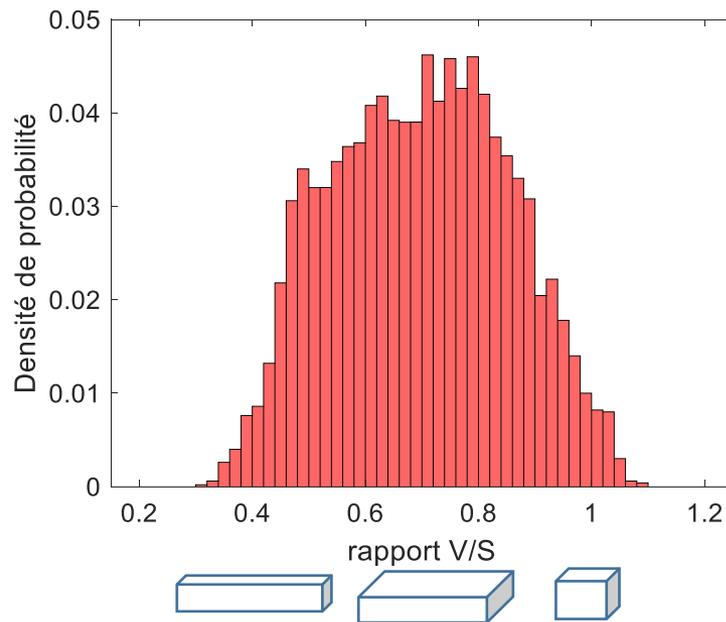


Application de l'AI à l'acoustique du bâtiment

Base de données - Réseau et Réverbération - Cas d'études

- 1 Cas du jeu de test (Test 0)

- AI >> Théorie Réverbération (TR)
- AI prend mieux en compte l'aspect salle inhomogène (champ inhomogène)
- Théorie Réverbération : Amélioration de l'utilisation du TR compliqué



Application de l'AI à l'acoustique du bâtiment

Base de données - Réseau et Réverbération - Cas d'études

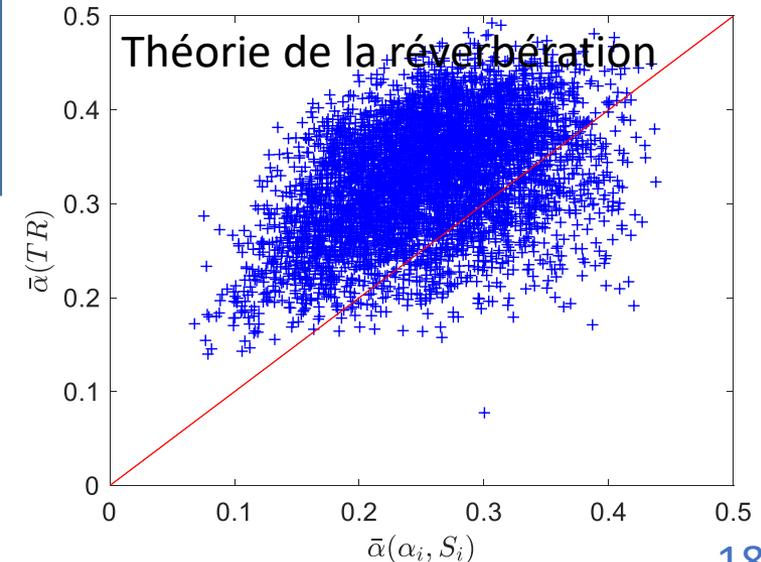
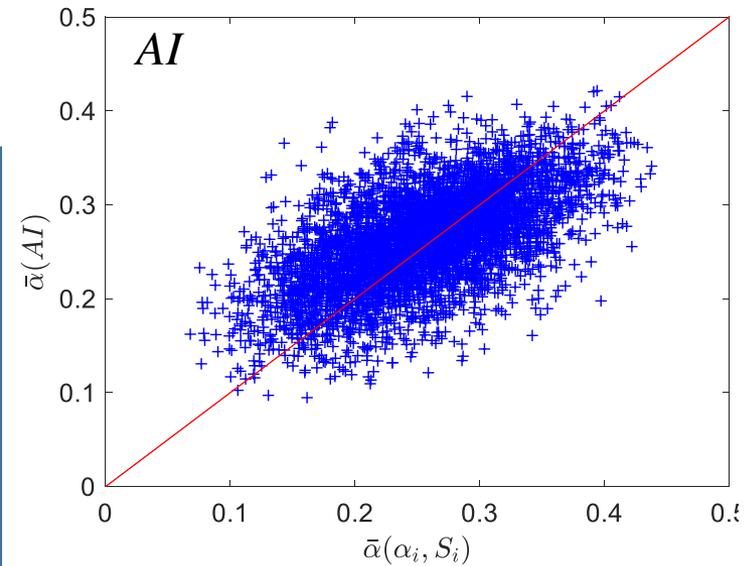
• 2 Modification de l'absorption surfacique

Test 1

- Logiciel Roomsim (images + rayons)
- Normalisation du champ direct
- Filtrage à 1000 Hz
- Echantillonnage (1024 points)
- Salles [1.5-10 : 1.5-10 : 2.5-4]
- ~~Absorption surfacique : bases réelles (sol, mur, plafond)~~
- Absorption surfacique aléatoire ($0,01 < \alpha_i < 0,5$)
sol, mur, plafond
- Diffusivité $s=0.1$
- Positions source et récepteur aléatoires

• Même avec une répartition de l'absorption surfacique peu courante, le réseau de neurones reste plus performant.

• Perspective de recherche : définir l'absorption surfacique et non l'absorption moyenne



Application de l'AI à l'acoustique du bâtiment

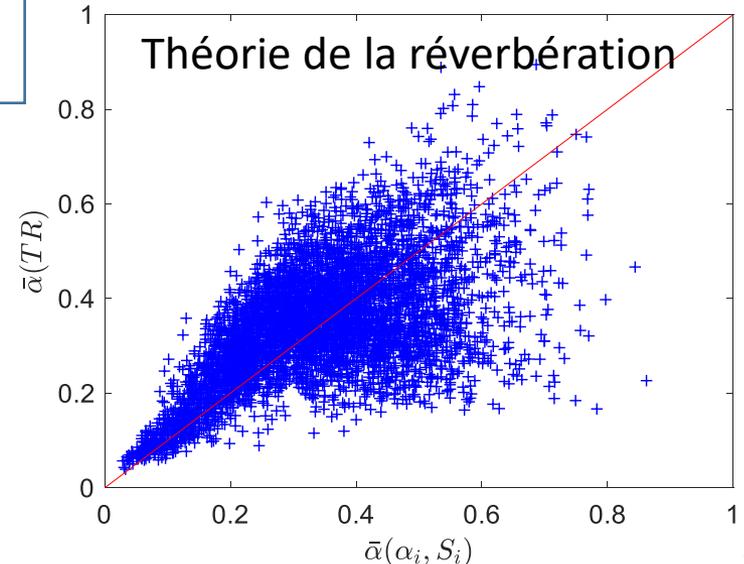
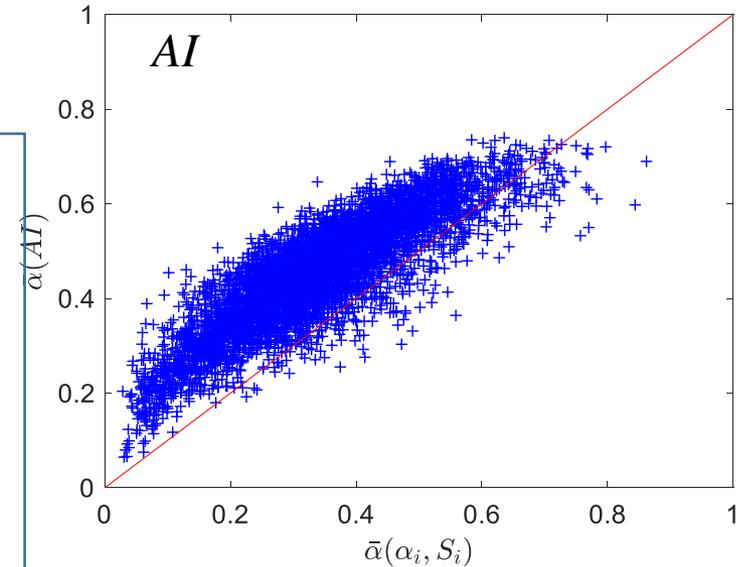
Base de données - Réseau et Réverbération - Cas d'études

3 Modification du filtrage des RI

Test 2

- Logiciel Roomsim (images + rayons)
- Normalisation du champ direct
- ~~Filtrage à 1000 Hz~~
- Filtrage à 500 Hz
- Echantillonnage (1024 points)
- Salles [1.5-10 : 1.5-10 : 2.5-4]
- Absorption : bases réelles (sol, mur, plafond)
- Diffusivité $s=0.1$
- Positions source et récepteur aléatoires

- Le réseau de neurones semble rester performant.



Application de l'AI à l'acoustique du bâtiment

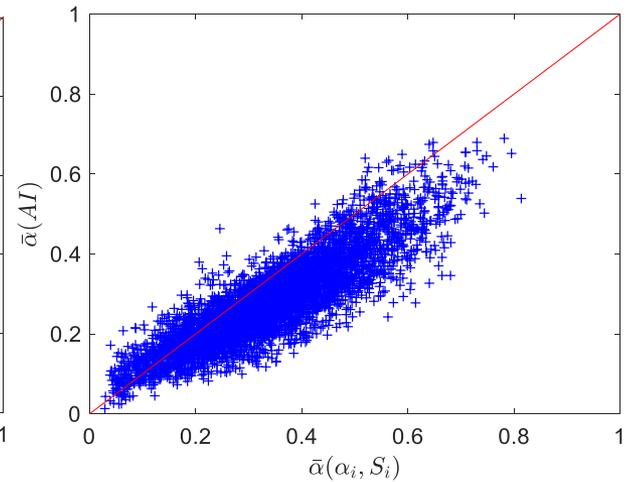
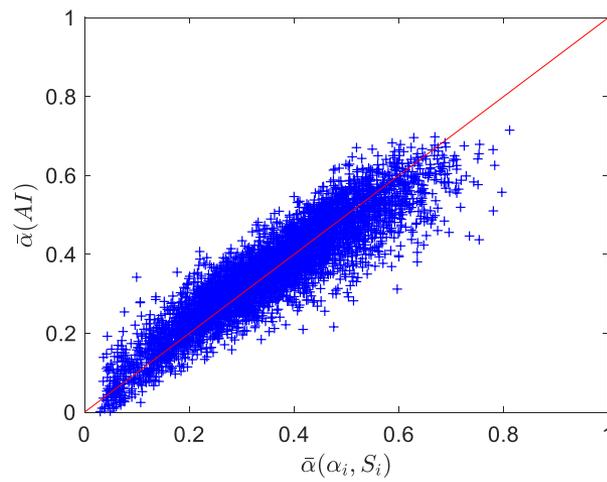
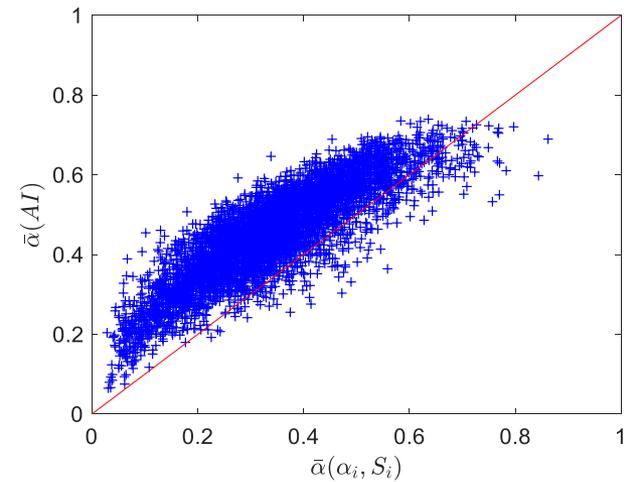
Base de données - Réseau et Réverbération - Cas d'études

- 3 Modification du filtrage des *RI*

500 Hz

1000Hz

2000Hz



Application de l'AI à l'acoustique du bâtiment

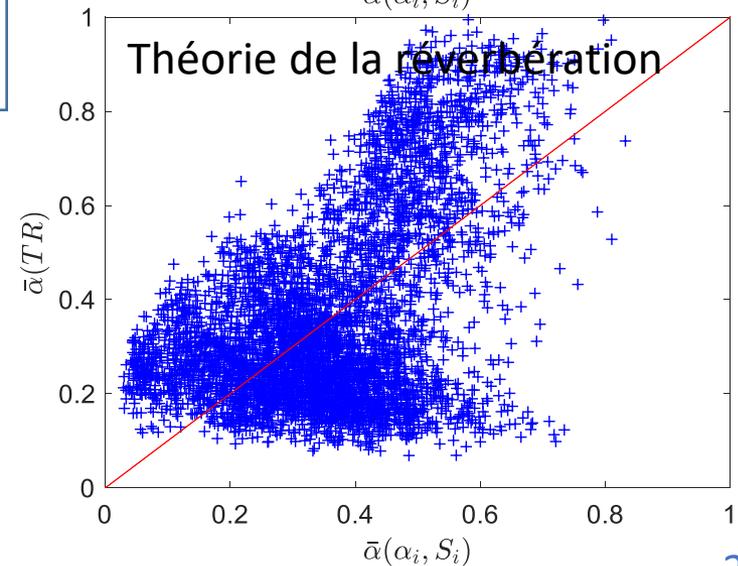
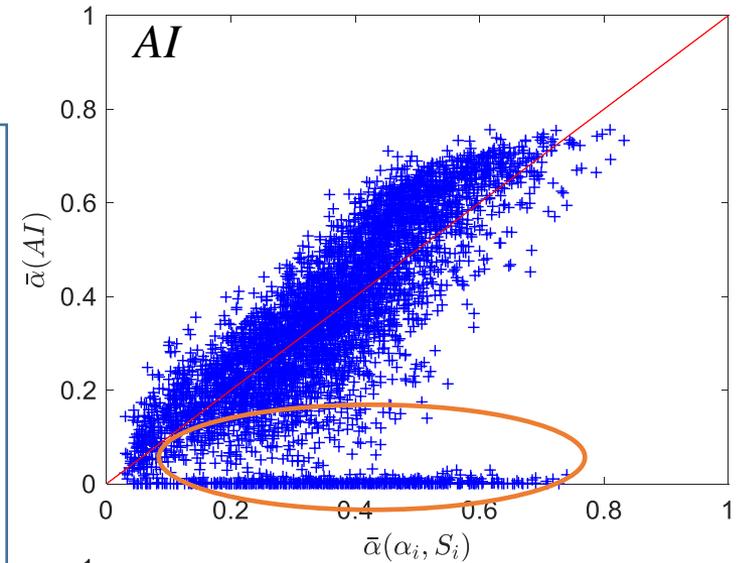
Base de données - Réseau et Réverbération - Cas d'études

4 Modification du logiciel de simulation

Test 3

- ~~Logiciel Roomsim (images + rayons)~~
- Logiciel Pyroom (images)
- Normalisation du champ direct
- Filtrage à 1000 Hz
- Echantillonnage (1024 points)
- Salles [1.5-10 : 1.5-10 : 2.5-4]
- Absorption : bases réelles (sol, mur, plafond)
- ~~Diffusivité $s=0.1$~~
- Positions source et récepteur aléatoires

- Le réseau de neurones semble être plus performant.
- Indépendance vis-à-vis du modèle
- Perspective de recherche : *RI* mesurées
- 5 Résultats des tests similaires avec le réseau de neurones (CNN : Convolutional Neural Network)



Est-il possible de retrouver l'absorption moyenne des parois d'une salle à partir de la connaissance d'une seule de ces réponses impulsionnelles ?

- Il semble que oui
- Résultat important

- Perspective de recherche

Adéquation physique – choix du réseau de neurones

Merci de votre attention