

# Réseaux de neurones récurrents et LSTM

Maxime Amossé, Vincent Auriau, Laurent Beaughon, Marc Bélicard,  
Yaqine Héchaïchi, Julien Hemery, Hugo Hervieux, Sylvain Pascou,  
Thaïs Rahoul, Pierre Vigier  
encadrés par Arpad Rimmel et Joanna Tomasik



CentraleSupélec

7 juin 2017

- 1 Introduction
- 2 Ressources
- 3 Principes généraux des réseaux de neurones
- 4 Traitement de séquences, introduction aux réseaux récurrents
- 5 Génération de séquences avec des LSTM
- 6 Génération de musique
  - Génération de spectres audio
  - Génération de midi
  - Génération de partitions
- 7 Conclusion

# Introduction

- Étudier les publications originelles

# Objectifs du projet

- Étudier les publications originelles
- Implémenter un réseau simple

- Étudier les publications originelles
- Implémenter un réseau simple
- Étudier les algorithmes de traitement de séquence

- Étudier les publications originelles
- Implémenter un réseau simple
- Étudier les algorithmes de traitement de séquence
- Implémenter ces algorithmes

- Étudier les publications originelles
- Implémenter un réseau simple
- Étudier les algorithmes de traitement de séquence
- Implémenter ces algorithmes
- Découvrir la cellule LSTM et l'implémenter



## Ressources

- Réunion hebdomadaire
- Suivi des encadrants
- Répartition des tâches

- Python et Numpy, C++ et Eigen
- Github et TravisCI
- Zotero
- LaTeX

# Principes généraux des réseaux de neurones

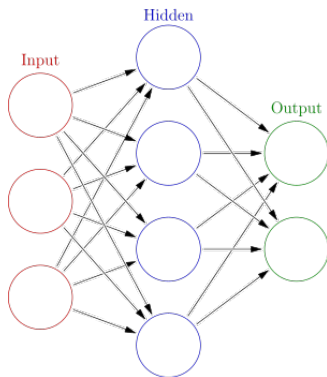


FIGURE – Exemple de réseau simple

# Propagation

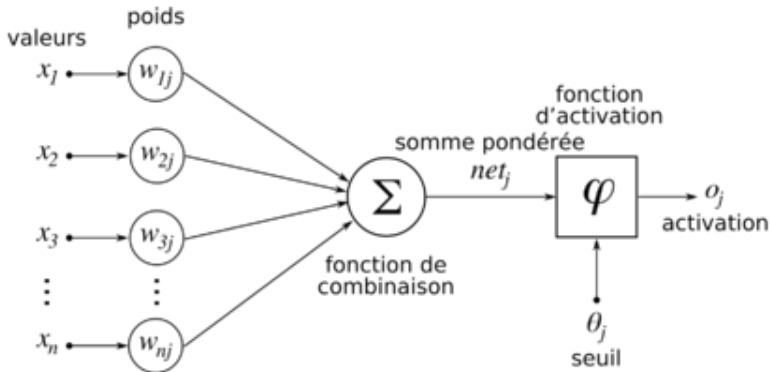


FIGURE – Propagation dans une cellule simple

Source : Wikipedia.org

# Rétropropagation et méthode du gradient

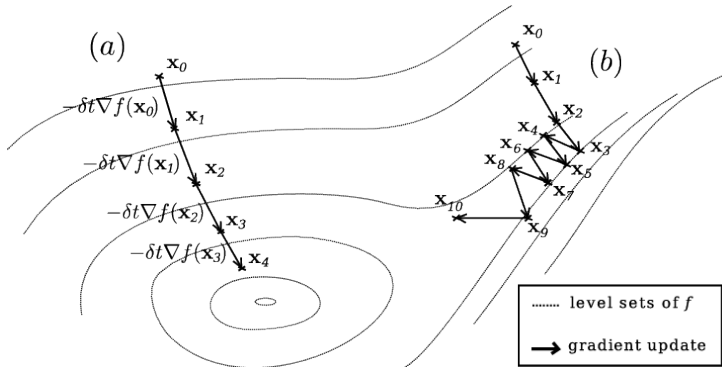


FIGURE – Exemples de descente du gradient

Source : <http://ludovicarnold.altevista.org/teaching/optimization/gradient-descent/>

# Problème type : MNIST



FIGURE – Exemple de données du MNIST



# Traitement de séquences, introduction aux réseaux récurrents

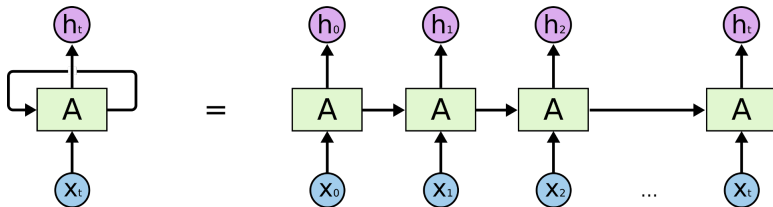


FIGURE – Dépliement dans l'espace

Source : <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

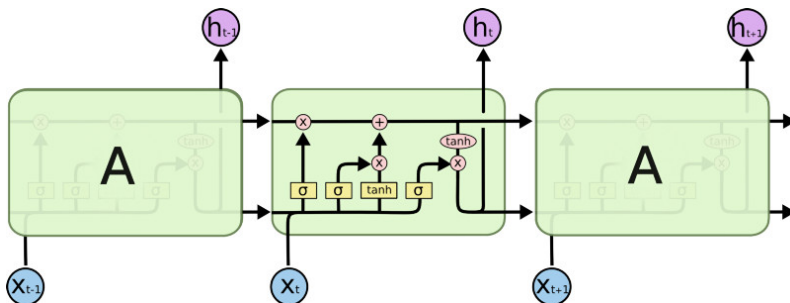


FIGURE – Dépliement dans l'espace d'une cellule LSTM

Source : <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# Génération de séquences avec des LSTM

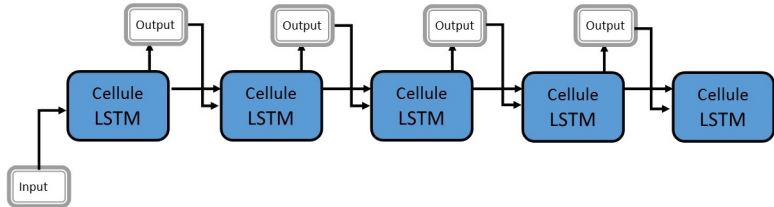


FIGURE – Principe de génération de séquences

# Exemple : génération de texte

Third Servan :

Of many bald with him fire, read now ?

Second Murderer :

Out ! where he wal'd apt thou, myself !

O brother's maliss and trunks and Caubble subject.

Now i' the fill in thy noble devart wagains to argon me thy  
commanded ?

LADY ANNE :

Sir, af you have fellow's their eyes live ?

# Génération de musique

<b>Génération</b>	<b>de spectres</b>	<b>MIDI</b>	<b>de partitions</b>
Pas de temps	Libres	Libres	Contraints
Hauteurs	Libres	Contraintes	Contraintes

FIGURE – Comparaison des trois approches



# Génération de spectres audio

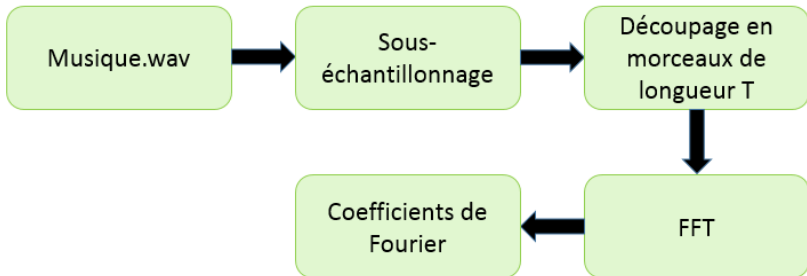


FIGURE – Création du dataset

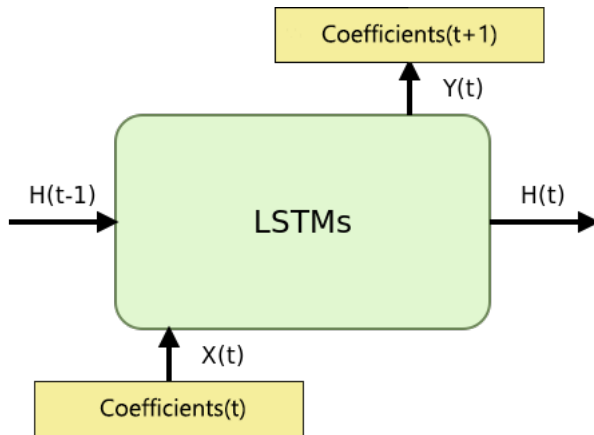


FIGURE – Fourier RNN

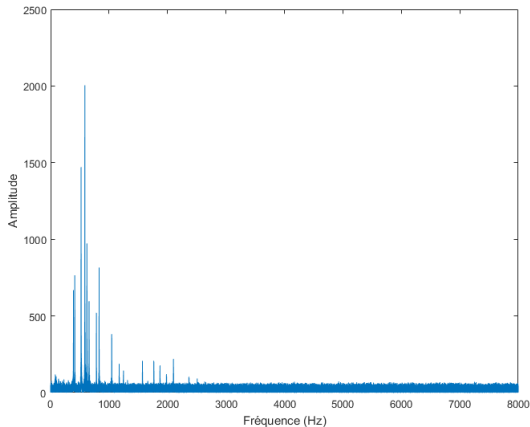


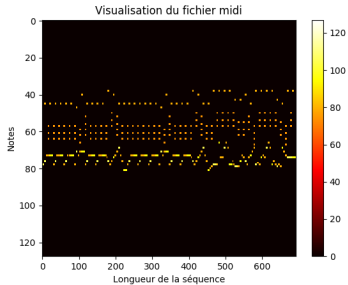
FIGURE – Spectre du signal généré

# Génération de midi

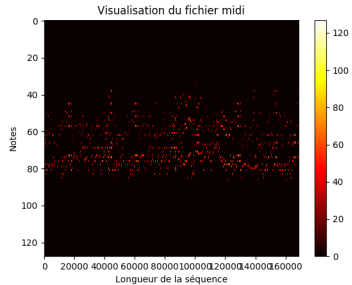
Octave Number	Hauteurs											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

Commandes :

- *note\_on note velocity time*
- *note\_off note velocity time*



(a) Jig



(b) Mozart

FIGURE – Visualisation de fichiers midi

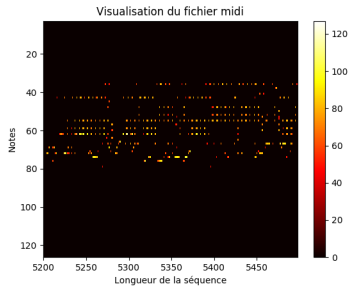


FIGURE – Jig générée



# Génération de partitions

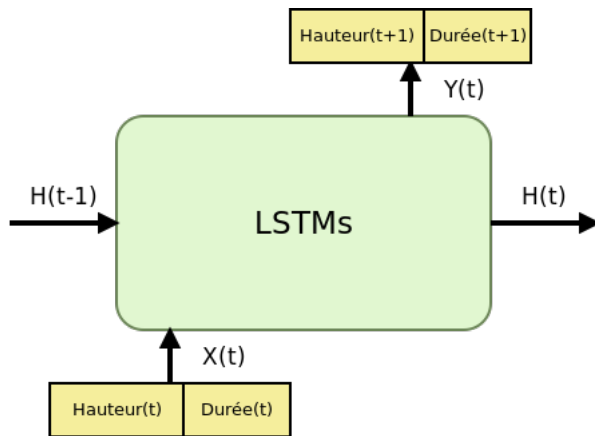


FIGURE – Note RNN



FIGURE – Une partition générée par Note RNN

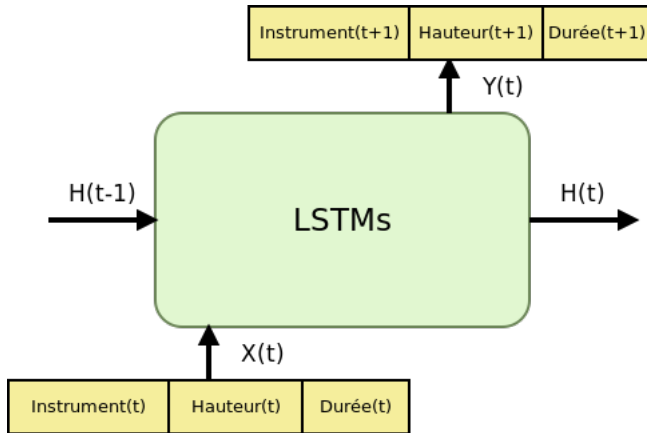


FIGURE – Series RNN



FIGURE – Une partition générée par Series RNN

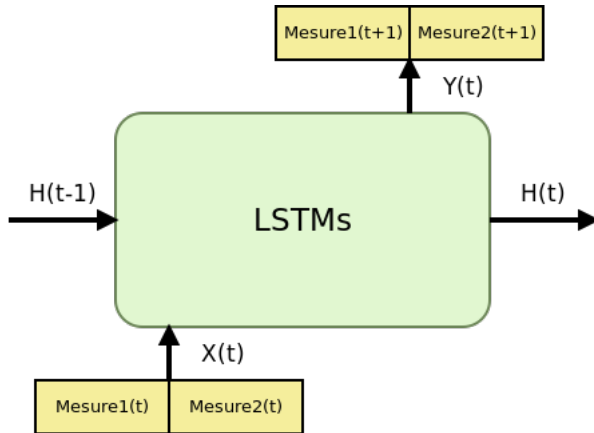


FIGURE – Measure RNN

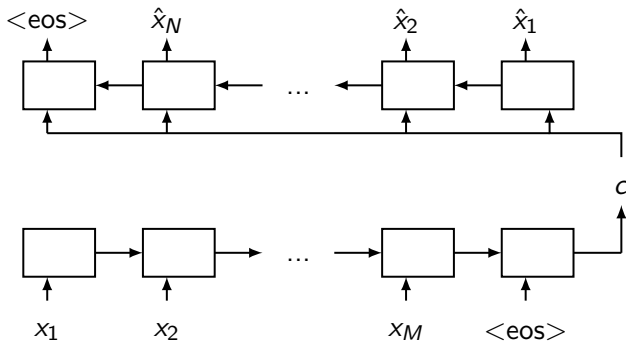


FIGURE – Réseau encodeur de mesures



FIGURE – Une partition générée par Measure RNN



# Conclusion