Optimisation de la longueur de fenêtre du spectrogramme au sein d'un réseau de neurones

Maxime LEIBER[1,2], Axel BARRAU[3], Yosra MARNISSI[2], Dany ABBOUD[2], Mohammed EL BADAOUI[4]

[1]INRIA, DI/ENS, Université PSL,
2 Rue Simone IFF, 75012 Paris, France

[2]Safran Tech, Groupe Safran,
Rue des Jeunes Bois-Châteaufort, 78772 Magny-les-Hameaux Cedex, France

[3]Offroad,
5 ter rue Parmentier, 94140 Alfortville, France

[4]Univ Lyon, UJM-St-Etienne, LASPI, EA3059, F-42023 Saint-Etienne, France

maxime.leiber@inria.fr, axel@offroad.works
yosra.marnissi@safrangroup.com, dany.abboud@safrangroup.com
mohamed.elbadaoui@univ-st-etienne.fr

Résumé – Nous présentons dans cet article une méthode permettant d'optimiser la longueur de la fenêtre de la transformée de Fourier à court terme (TFCT) dans un réseau de neurones. Plutôt que de le régler empiriquement sur une valeur entière au préalable, nous rendons ce paramètre continu par rapport auquel les valeurs de la TFCT sont différentiables. Plus précisément, nous définissons une version différentiable de la TFCT de taille fixe qui s'encapsule aisément dans tout réseau de neurones comme première couche. Nous montrons ensuite la facilité d'intégration sur une tâche de classification.

Abstract – We present in this paper a method for optimizing the window length of the widely used short-time Fourier transform (STFT) within a neural network. Rather than tuning it empirically on an integer-valued beforehand, we make this parameter continuous w.r.t. which STFT values are differentiable. More specifically, we define a differentiable version of the fixed-size STFT that is easily encapsulated in any neural network as first layer. We then show the ease of integration on a classification task.

1 Introduction

La transformée de Fourier à court terme (TFCT) est un outil omniprésent pour l'analyse des signaux non stationnaires tels que les événements transitoires comme les sons d'animaux [1], les événements discontinus comme les signaux d'électroencéphalographie [2] ainsi que les signaux multi-harmoniques typiquement mesurés sur des composants mécaniques à vitesse variable [3]. Les spectrogrammes construits à partir de la TFCT peuvent être en particulier utilisés pour une simple visualisation et une compréhension des signaux non stationnaires. Pour cela, certaines techniques de post-traitement ont été proposé dans la littérature afin d'améliorer la lisibilité des spectrogrammes telles que le reassignment [4] et le synchro-squeezing [5, 6]. Les spectrogrammes peuvent être également combinés avec des méthodes plus complexes. Par exemple, [7] combine l'analyse en composantes principales avec les spectrogrammes de signaux de vibration pour la détection de défauts mécaniques. De plus, les spectrogrammes ont été couramment utilisés pour alimenter des réseaux de neurones pour plusieurs applications comme l'identification et l'estimation [8], la reconnaissance vocale [9, 10], la détection en musique [11], la classification d'électrocardiogrammes [12], l'augmentation de données [13, 14], la séparation de sources [15] etc.

Dans cet article, nous nous intéressons particulièrement à ces dernières applications, c'est-à-dire celles combinant les réseaux de neurones et les spectrogrammes. Dans ces applications, la longueur de la fenêtre du spectrogramme est généralement un paramètre fixe, choisi empiriquement sans étude approfondie. Or, ce paramètre mérite d'être réglé avec précaution, car il détermine le compromis entre la résolution temporelle et fréquentielle du spectrogramme, conformément au principe d'incertitude de Heisenberg.

Notre principale contribution est de proposer un nouveau paradigme pour l'optimisation de la longueur de la fenêtre qui, selon nous, pourrait devenir un moyen standard de régler la longueur de la fenêtre des spectrogrammes utilisés comme entrée dans les réseaux de neurones. Il consiste à modifier la définition de l'opérateur TFCT pour faire de la longueur de la fenêtre un paramètre continu par rapport aux valeurs de la TFCT pouvant être différenciées. L'idée principale est de décomposer le paramètre *taille de fenêtre* en deux variables : un paramètre *taille de*

fenêtre numérique entier et un paramètre *résolution temporelle* continu. La formule de rétropropagation fournie dans cet article permet un réglage conjoint des poids du réseau de neurones et de la résolution du spectrogramme (via sa taille de fenêtre).

Il convient de noter que le problème de la recherche de la meilleure taille de fenêtre n'est pas nouveau et que plusieurs méthodes ont été proposées dans la littérature, comme [16–18]. Tandis que les autres approches recherchent un optimum général, notre objectif est plutôt d'optimiser la taille de fenêtre pour une tâche particulière, en minimisant le critère d'erreur du réseau de neurones donné. A notre connaissance, le seul travail qui a tenté une optimisation de la longueur de la fenêtre avec une descente de gradient est celui proposé dans [19] et qui peut être considéré comme un cas particulier de la théorie que nous proposons. Cependant, notre article va plus loin : la TFCT est mathématiquement différentiable et les calculs avec les formules de propagation et de rétropropagation sont fournis.

Cet article est organisé comme suit : dans la section 2, nous commençons par donner quelques définitions et notations. Dans la section 3, nous présentons la TFCT modifiée, qui est différentiable en fonction de la longueur de la fenêtre. Dans la section 4, nous illustrons l'efficacité de notre approche sur une tâche de classification et nous terminons notre discussion dans la section 5 avec quelques remarques finales.

2 Définitions et notations

Dans la suite de l'article, nous ferons référence à la TFCT comme une opération prenant une série temporelle unidimensionnelle $s[t]$ en entrée et renvoyant un tableau bidimensionnel $\mathcal{S}[i, f]$, chaque colonne $\mathcal{S}[i, :]$ étant la transformée de Fourier discrète (TFD) d'une portion de longueur $L = 2^n$ du signal s commençant à partir de l'indice b_i jusqu'un indice $b_i + L - 1$, multipliée par une seconde séquence h_L appelée fonction d'apodisation ('tapering function'). La TFCT peut être définie comme :

$$\begin{aligned}\mathcal{S}[i, f] &= \mathcal{F}(h_L s[b_i : b_i + L - 1])[f] \\ &= \sum_{k=0}^{L-1} h_L[k] s[b_i + k] e^{-2j\pi kf/L},\end{aligned} \tag{1}$$

où $\mathcal{F}(\cdot)$ désigne l'opérateur de la transformée de Fourier discrète (TFD), i est un entier exprimant l'indice de la tranche et b_i le point de départ associé. Les indices de départ b_i des intervalles de temps sur lesquels les spectres sont calculés sont équidistants, il suffit donc de définir le premier indice b_0 et l'espacement Δb entre b_i et b_{i+1}. Cet espacement est généralement défini en pourcentage de la L, par un rapport α appelé *chevauchement* : $\Delta b = \lfloor \alpha L \rfloor$ où $\lfloor \cdot \rfloor$ désigne la partie entière. Enfin, bien que le choix de la forme de la fenêtre soit important, nous ne nous concentrons pas sur ce critère dans cet article. Plusieurs choix de fonction d'apodisation h_L peuvent être rencontrés, l'un des plus courants étant la *fenêtre de Hann* :

$$h_L[k] = \frac{1}{2} - \frac{1}{2} \cos\left(\frac{2\pi k}{L - 1}\right). \tag{2}$$

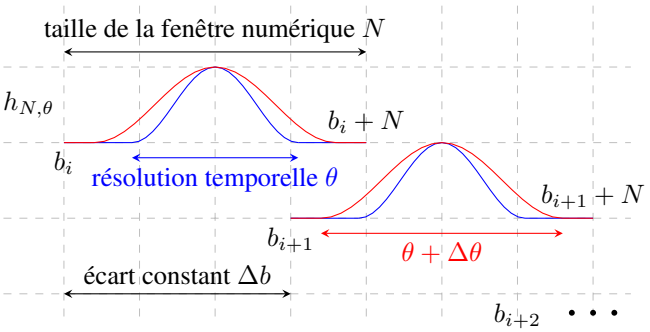

FIGURE 1 – TFCT différentiable : d'une part, la taille N du sous-signal sur lequel la TFD est calculée est fixe, d'autre part, le support θ de la fonction d'apodisation, qui détermine en fait la résolution temporelle est libre de varier. La différenciation sera faite en fonction de ce dernier paramètre.

Après avoir clarifié les notations et les terminologies, nous pouvons passer au coeur de notre contribution : construire une TFCT dont la longueur de fenêtre L est un paramètre continu par rapport auquel les valeurs du spectrogramme sont différentiables.

3 TFCT différentiable

Construire une version différentiable de la TFCT signifie écrire une formule similaire à l'Eq. (1) où L devient un paramètre continu et $\mathcal{S}[i, f]$ est différentiable par rapport à L. On voit que chaque terme de la somme est déjà différentiable en fonction de L vu comme un paramètre continu. Si les indices b_i sont supposés statiques, le seul obstacle est la présence de L comme borne de la somme, ce qui suggère de décomposer L en une *taille de fenêtre numérique* N et une *résolution temporelle* θ :

$$\mathcal{S}[i, f] = \sum_{k=0}^{N-1} h_{N,\theta}[k] s[b_i + k] e^{-2j\pi kf/N}, \tag{3}$$

où $h_{N,\theta}$ est une fonction d'apodisation définie sur $[0, N - 1]$ mais prenant des valeurs non nulles uniquement dans l'intervalle $[\frac{N-1-\theta}{2}, \frac{N-1+\theta}{2}]$ où elle vaut :

$$h_{N,\theta}(k) = h_\theta\left(k + \frac{\theta - N + 1}{2}\right). \tag{4}$$

Le paramètre θ a une signification très proche de celle de L : il définit la longueur temporelle de la tranche de signal sur laquelle un spectre local est calculé. La différence avec la TFCT classique est que la tranche est remplie de zéros afin d'être toujours de taille N. Par conséquent, la résolution en fréquence des spectres locaux n'est plus imposée par θ mais par le second paramètre $N > \theta$. Cette résolution en fréquence n'est pas réaliste car le signal a été rempli de zéros : augmenter N n'apporte pas plus d'informations. Le paramètre N ne doit être considéré que comme une borne supérieure de la résolution temporelle θ, rendant la différenciation possible. Calculons maintenant la différentielle de notre STFT proposée par rapport à la résolution temporelle θ. Nous obtenons directement :

$$\frac{\partial \mathcal{S}(i,f)}{\partial \theta} = \sum_{k=0}^{N-1} \exp\left(-2j\pi \frac{kf}{N}\right) \frac{\partial h(k)}{\partial \theta} s[b_i + k] \quad (5)$$

où nous reconnaissons le spectrogramme de s avec la fonction d'apodisation $h' = \frac{\partial h_{N,\theta}(k)}{\partial \theta}$ au lieu de $h_{N,\theta}$:

$$\frac{\partial \mathcal{S}[s]}{\partial \theta} = \mathcal{S}_{h'}[s] \quad (6)$$

Ce dernier résultat permet de dériver des formules compactes de rétropropagation par gradient. Rappelons que la rétropropagation par gradient consiste à calculer la dérivée d'une fonction $\mathcal{L}$ par rapport à l'entrée d'une fonction g étant donnée sa dérivée par rapport à la sortie de g, c'est-à-dire à calculer $\frac{\mathcal{L}}{I}$ étant donné $\frac{\mathcal{L}}{\mathcal{S}}$ pour $\mathcal{S} = g(I)$. Dans notre cas, nous obtenons directement :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{i=1}^{\mathbf{T}} \sum_{f=0}^{N-1} \frac{\partial \mathcal{L}}{\partial \mathcal{S}(i,f)} \frac{\partial \mathcal{S}(i,f)}{\partial \theta} \\ &= \left\langle \frac{\partial \mathcal{L}}{\partial \mathcal{S}}, \mathcal{S}_{h'}[s] \right\rangle, \end{aligned} \quad (7)$$

où $\langle \cdot, \cdot \rangle$ représente le produit scalaire de Froebenius de deux matrices : $\{A, B\} = \sum_{i,j} A_{i,j} B_{i,j}$.

Nous avons supposé que les positions des tranches $[b_i, b_i + k]$ étaient indépendantes de θ, ce qui signifie que le nombre de colonnes du spectrogramme est constant. Ce cas est utile lorsque le spectrogramme est l'entrée d'un autre algorithme qui prend en entrée un spectrogramme de taille fixe, comme les réseaux de neurones.

4 Expérience

Dans cette section, nous montrons comment il est facile de brancher notre TFCT modifiée à n'importe quel réseau de neurones existant. Le but est de montrer qu'il est possible d'optimiser conjointement les poids du réseau avec la taille de fenêtre du spectrogramme et que celle-ci convergera vers une taille de fenêtre minimisant la fonction coût du réseau de neurones.

TABLE 1 – Erreurs d'entraînement, de validation et de test du réseau de convolutions pour différentes tailles de fenêtres fixes.

taille de fenêtre	entraînement	validation	test
10	0.3672	1.1442	1.0391
20	0.0304	0.4443	0.3177
30	0.0089	0.2408	0.0306
40	0.0064	0.2550	0.1249
50	0.0061	0.4859	0.2868

Nous avons choisi une tâche de classification simple de reconnaissance de chiffres parlés avec la base de données Free Spoken Digit Dataset (FSDD). L'objectif est de trouver automatiquement la meilleure taille de fenêtre ainsi que les meilleurs poids de réseau pour l'ensemble des données considérées minimisant l'entropie croisée $\mathcal{L}$ entre la prédiction du réseau et la vérité terrain :

$$\mathcal{L}^{\theta} = \frac{1}{N} \sum_{n=1}^{N} -log\left(\frac{\exp\left(y_{gt}^{\hat{n}}\right)}{\sum_{c=1}^{C} \exp\left(y_{c}^{\hat{n}}\right)}\right) \quad (8)$$

où N est le nombre d'échantillons du lot de données considéré, C est le nombre de classes, $y_c^{\hat{n}}$ la prédiction du réseau du n^e échantillon associée à la classe c et $y_{gt}^{\hat{n}}$ la prédiction du réseau du n^e échantillon pour la classe correspondante à la vérité terrain.

Nous avons entraîné, pour plusieurs tailles de fenêtre préalablement fixées, un réseau de neurones convolutifs à 2 couches afin de comparer la précision du réseau en fonction de la taille de fenêtre. Nous voyons dans le tableau 1 que le paramètre de taille de fenêtre est très important car la précision d'un même réseau entraîné pour différentes valeurs de taille de fenêtre varie fortement. De plus, quelle que soit la valeur initiale de la taille de fenêtre lors d'une optimisation conjointe avec les poids du réseau, le paramètre de taille de fenêtre converge vers une valeur qui minimise la fonction coût du réseau neuronal telle qu'affichée sur la Fig. 2 et 3. En effet, à la fin de la descente de gradient, la longueur de fenêtre atteint la valeur 34.9 alors que nous avons commencé l'apprentissage avec une taille de fenêtre de 200.

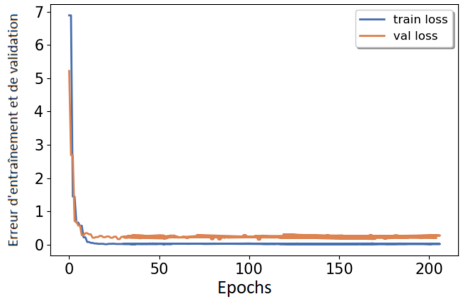

FIGURE 2 – Erreur d'entraînement et de validation par époque. L'erreur diminue en optimisant conjointement la longueur de la fenêtre et les poids du réseau de neurones.

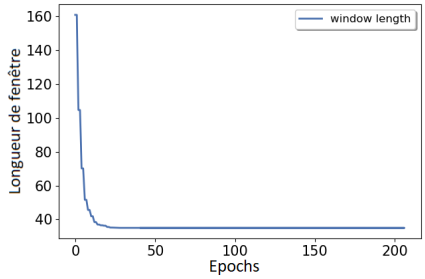

FIGURE 3 – Longueur de fenêtre par époque. Pendant l'apprentissage, le paramètre continu de résolution temporelle θ converge vers une valeur optimale.

Cette simulation simple prouve l'efficacité de notre procédure de rétropropagation basée sur une version différentiable de la TFCT. Cela illustre une méthodologie générale d'ajustement de la longueur de la fenêtre applicable à tout réseau de neurones

impliquant des spectrogrammes ou melspectrogrammes : remplacer l'étape de calcul du spectrogramme par le calcul de notre spectrogramme différentiable, puis optimiser conjointement la longueur de la fenêtre avec les poids du réseau de neurones par descente de gradient en fonction de la formule de rétropropagation que nous avons introduite.

5 Conclusion

Nous avons présenté une modification de la TFCT rendant cette opération différentiable par rapport au paramètre de longueur de fenêtre et nous avons donné la formule de rétropropagation induite. A notre connaissance, cela n'avait pas été fait jusqu'à présent. L'application principale de cette contribution est de revisiter la combinaison des spectrogrammes et des réseaux de neurones : au lieu de donner directement un spectrogramme (ou un melspectrogramme) comme entrée au réseau, on peut maintenant donner directement le signal temporel comme entrée au réseau ayant notre spectrogramme différentiable comme
première couche et la longueur de la fenêtre comme paramètre continu et le laisser optimiser cette longueur de fenêtre avec tous les poids du réseau. Une de nos perspectives est de proposer une fonction de fenêtrage paramètrique comme une somme de cosinus dont les paramètres seraient optimisés conjointement avec la taille de fenêtre.

Références

[1] Stafford, K. M., Fox, C. G., and Clark, D. S. (1998). Long-range acoustic detection and localization of blue whale calls in the northeast Pacific Ocean. The Journal of the Acoustical Society of America, 104(6), 3616-3625.

[2] Ramos-Aguilar, R., Olvera-López, J. A., and Olmos-Pineda, I. (2017). Analysis of EEG signal processing techniques based on spectrograms. Research in Computing Science, 145, 151-162.

[3] Leclère, Q., André, H., and Antoni, J. (2016). A multi-order probabilistic approach for Instantaneous Angular Speed tracking debriefing of the CMMNO 14 diagnosis contest. Mechanical Systems and Signal Processing, 81, 375-386.

[4] Flandrin, P., Auger, F., and Chassande-Mottin, E. (2018). Time-frequency reassignment : from principles to algorithms. In Applications in time-frequency signal processing (pp. 179-204). CRC Press.

[5] Thakur, G., Brevdo, E., Fučkar, N. S., and Wu, H. T. (2013). The synchrosqueezing algorithm for time-varying spectral analysis : Robustness properties and new paleoclimate applications. Signal Processing, 93(5), 1079-1094.

[6] Auger, F., Flandrin, P., Lin, Y. T., McLaughlin, S., Meignen, S., Oberlin, T., and Wu, H. T. (2013). Time-frequency reassignment and synchrosqueezing : An overview. IEEE Signal Processing Magazine, 30(6), 32-41.

[7] Yu, H., Guo, Q., Hu, J., and Xu, A. (2006). Rolling bearings fault diagnosis based on adaptive gaussian chirplet spectrogram and independent component analysis. In International Conference on Natural Computation (pp. 321-330).

[8] O'Shea, T. J., Roy, T., and Erpek, T. (2017). Spectral detection and localization of radio events with learned convolutional neural features. In 2017 25th European Signal Processing Conference (EUSIPCO) (pp. 331-335).

[9] Badshah, A. M., Ahmad, J., Rahim, N., and Baik, S. W. (2017). Speech emotion recognition from spectrograms with deep convolutional neural network. In 2017 international conference on platform technology and service (PlatCon) (pp. 1-5).

[10] Gong, Y., Chung, Y. A., and Glass, J. (2021). Ast : Audio spectrogram transformer.

[11] Schlüter, J., and Böck, S. (2014). Improved musical onset detection with convolutional neural networks. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6979-6983).

[12] Huang, J., Chen, B., Yao, B., and He, W. (2019). ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. IEEE Access, 7, 92871-92880.

[13] Park, D. S., Chan, W., Zhang, Y., Chiu, C. C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment : A simple data augmentation method for automatic speech recognition.

[14] Park, D. S., Zhang, Y., Chiu, C. C., Chen, Y., Li, B., Chan, W., ... and Wu, Y. (2020). Specaugment on large scale datasets. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6879-6883).

[15] Défossez, A. (2021). Hybrid Spectrogram and Waveform Source Separation. arXiv preprint arXiv :2111.03600.

[16] Zhong, J., and Huang, Y. (2010). Time-frequency representation based on an adaptive short-time Fourier transform. IEEE Transactions on Signal Processing, 58(10), 5118-5128.

[17] Kwok, H. K., and Jones, D. L. (2000). Improved instantaneous frequency estimation using an adaptive short-time Fourier transform. IEEE Transactions on Signal Processing, 48(10), 2964-2972.

[18] Czerwinski, R. N., and Jones, D. L. (1997). Adaptive short-time Fourier analysis. IEEE Signal Processing Letters, 4(2), 42-45.

[19] Zhao, A., Subramani, K., and Smaragdis, P. (2021). Optimizing Short-Time Fourier Transform Parameters via Gradient Descent. In 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 736-740).