Réseau neuronal end-to-end pour musique polyphonique pour piano

(Juillet 2020)

Nicolas Deviers, pour le projet EIP Bard's Way.

I. INTRODUCCION

La transcription automatique de la musique (TAM) est un problème fondamental de la recherche d'informations musicales (RIM). L'AMT vise à générer une transcription symbolique, semblable à une partition, à partir d'un signal acoustique polyphonique. La transcription musicale est considérée comme un problème difficile, même par les experts humains, et les systèmes actuels de transcription musicale ne correspondent pas aux performances humaines. L'AMT polyphonique est un problème difficile parce que les notes d'un ou de plusieurs instruments qui sonnent simultanément provoquent une interaction complexe et un chevauchement des harmoniques dans le signal acoustique. La variabilité du signal d'entrée dépend également du type spécifique d'instrument utilisé. En outre, les systèmes AMT à polyphonie non contrainte ont un espace de sortie combinatoire très important, ce qui complique encore le problème de modélisation. Généralement, la variabilité du signal d'entrée est saisie par des modèles qui visent à apprendre les propriétés de timbre de l'instrument transcrit [2], [3], tandis que les questions relatives à un grand espace de sortie sont traitées en contraignant les modèles à avoir une polyphonie maximale [4], [5].

La majorité des systèmes AMT actuels sont basés sur le principe de la description du spectrogramme d'amplitude d'entrée comme une combinaison pondérée de spectres de base correspondant à des hauteurs de son. Les spectres de base peuvent être estimés par différentes techniques telles que la factorisation matricielle non négative (NMF) et la décomposition éparse. Les approches NMF non supervisées [6], [7] visent à apprendre un dictionnaire des spectres de tonalité à partir des exemples de formation. Cependant, les approches purement non supervisées peuvent souvent conduire à des bases qui ne correspondent pas aux hauteurs musicales, entraînant ainsi des problèmes d'interprétation des résultats au moment des tests. Ces problèmes liés aux méthodes non supervisées de factorisation des spectrogrammes sont résolus en incorporant des contraintes harmoniques dans l'algorithme d'apprentissage [8], [9]. Les techniques basées sur la factorisation des spectrogrammes ont été étendues avec l'introduction de l'analyse probabiliste des composantes latentes (PLCA) [10]. L'analyse PLCA vise à ajuster un modèle probabiliste à variables latentes à des spectrogrammes normalisés. Les modèles basés sur l'analyse PLCA sont faciles à former avec l'algorithme de maximisation des attentes (EM) et ont été étendus et appliqués de manière extensive aux problèmes d'AMT [11], [3].

Comme alternative aux techniques de factorisation des spectrogrammes, les approches discriminantes de l'AMT ont suscité un intérêt considérable. Les approches discriminatoires visent à classer directement les caractéristiques extraites des trames audios dans les hauteurs de sortie. Cette approche présente l'avantage qu'au lieu de construire des modèles générateurs spécifiques à l'instrument, des classificateurs complexes peuvent être formés en utilisant de grandes quantités de données de formation pour saisir la variabilité des entrées. Lors de l'utilisation d'approches discriminantes, la performance des classificateurs dépend des caractéristiques extraites du signal. Récemment, les réseaux de neurones ont

été appliqués aux données brutes ou aux représentations de bas niveau pour apprendre conjointement les caractéristiques et les classificateurs d'une tâche [12]. Au fil des ans, de nombreuses expériences ont été menées pour évaluer les approches discriminantes de l'AMT. Poliner et Ellis [13] utilisent des machines à vecteurs de support (SVM) pour classifier les spectres de magnitude normalisés [14] qui superposent un SVM sur un réseau de croyances profondes (DBN) afin d'apprendre les caractéristiques d'une tâche de TMA. De même, un réseau neuronal récurrent bidirectionnel (RNN) est appliqué aux spectrogrammes de magnitude pour la transcription polyphonique dans [15].

Dans les systèmes de reconnaissance vocale à grand vocabulaire, les informations contenues dans le signal acoustique ne suffisent souvent pas à elles seules à résoudre les ambiguïtés entre les sorties possibles. Un modèle de langage est utilisé pour fournir une probabilité préalable du mot courant par rapport aux mots précédents dans une phrase. Les modèles de langage statistiques sont essentiels pour la reconnaissance vocale de grands vocabulaires [16]. Tout comme la parole, les séquences musicales présentent une structure temporelle. En plus d'un modèle acoustique précis, un modèle qui capture la structure temporelle de la musique ou un modèle de langage musical (MLM), peut potentiellement aider à améliorer les performances des systèmes AMT. Contrairement à la parole, les modèles de langage ne sont pas courants dans la plupart des modèles AMT en raison du problème difficile que pose la modélisation de l'espace de sortie combinatoirement large de la musique polyphonique. Généralement, les sorties des modèles acoustiques sont traitées par des modèles de Markov cachés (HMM) à deux états, spécifiques à la hauteur de la voix, qui imposent des contraintes de lissage et de durée sur les hauteurs de sortie [3], [13]. Cependant, étendre cette méthode à la modélisation des sorties en haute dimension d'un système polyphonique d'AMT s'est avéré difficile, bien que certaines études explorent cette idée. Un réseau bayésien dynamique est utilisé dans [17], pour estimer les probabilités préalables de combinaisons de notes dans un cadre de transcription basé sur le NMF. De même, dans [18], un MLM basé sur un réseau neuronal récurrent (RNN) est utilisé pour estimer les probabilités préalables de séquences de notes, parallèlement à un modèle acoustique PLCA. Un cadre de transduction de séquences est proposé dans [19], où les modèles acoustiques et de langage sont combinés dans un seul RNN.

Les idées présentées dans cet article sont des extensions des expériences préliminaires de [20]. Nous proposons une architecture de bout en bout pour former conjointement les modèles acoustique et linguistique pour une tâche d'AMT. Nous évaluons la performance du modèle proposé sur un ensemble de données de musique polyphonique pour piano. Nous formons des modèles acoustiques de réseaux de neurones pour identifier les hauteurs de son dans un cadre audio. Les classificateurs discriminants peuvent en théorie être formés sur des mélanges complexes de sources d'instruments, sans avoir à tenir compte de chaque instrument séparément. Les classificateurs de réseaux neuronaux peuvent être directement appliqués à la représentation temps-fréquence, ce qui élimine la nécessité d'une étape d'extraction de caractéristiques distincte. En plus des architectures de réseaux neuronaux à réaction en profondeur (DNN) et RNN dans [20], nous explorons l'utilisation de réseaux neuronaux convolutionnels (ConvNets) comme modèles acoustiques. Les ConvNets ont été initialement proposés comme classificateurs pour la reconnaissance d'objets en vision par ordinateur, mais ont trouvé des applications croissantes dans la reconnaissance vocale [21], [22]. Bien que les ConvNets aient été appliqués à certains problèmes dans MIR [23], [24], ils restent inexplorés pour les tâches de transcription. Nous incluons également des comparaisons avec deux modèles acoustiques de pointe basés sur la factorisation des spectrogrammes [3], [8] qui sont populaires dans la littérature sur l'AMT. Comme nous l'avons déjà mentionné, les sorties à haute dimension du modèle acoustique posent un problème difficile pour la modélisation du langage. Nous proposons d'utiliser les RNN comme alternative aux modèles d'espace d'état comme les HMM factoriels [25] et les réseaux bayésiens dynamiques [17], pour modéliser la structure temporelle des notes en musique. Les modèles de langage basés sur les RNN ont été utilisés pour la première fois en parallèle avec un modèle acoustique PLCA [18]. Cependant, dans cette configuration, le modèle de langage est utilisé pour affiner itérativement les prédictions dans une boucle de rétroaction résultant en un modèle non causal et théoriquement insatisfaisant. Dans le cadre hybride, l'inférence approximative sur les variables de sortie est effectuée en utilisant la recherche de faisceau.

Cependant, la recherche de faisceau peut être coûteuse en termes de calcul lorsqu'elle est utilisée pour décoder de longues séquences temporelles. Nous appliquons l'algorithme efficace de recherche de faisceau par hachage proposé dans [26] pour l'inférence. Le nouvel algorithme d'inférence réduit le temps de décodage d'un ordre de grandeur et rend le modèle proposé adapté aux applications en temps réel. Nos résultats montrent que les modèles acoustiques des réseaux de neurones convolutifs surpassent les autres modèles acoustiques pour un certain nombre de mesures d'évaluation.

Nous observons également une amélioration des performances avec l'application des modèles de langage musical. Le reste de l'article est organisé comme suit : La section II décrit les modèles de réseaux de neurones utilisés dans l'expérience, la section III discute du modèle proposé et de l'algorithme d'inférence, la section IV détaille l'évaluation du modèle et les résultats expérimentaux. Les discussions, les travaux futurs et les conclusions sont présentés dans la section V.

II. CONTEXTE

Dans cette section, nous décrivons les modèles de réseaux de neurones utilisés pour la modélisation acoustique et linguistique. Bien que les réseaux neuronaux soient un concept ancien, ils ont récemment été appliqués avec succès à un large éventail de problèmes d'apprentissage machine [12]. L'une des raisons principales de leur succès récent est la disponibilité de grands ensembles de données et d'une infrastructure informatique à grande échelle [27], qui permet de former des réseaux comportant des millions de paramètres. Les paramètres de toute architecture de réseau neuronal sont généralement estimés à l'aide de techniques d'optimisation numérique. Une fois qu'une fonction de coût appropriée a été définie, les dérivés du coût par rapport aux paramètres du modèle sont trouvés à l'aide de l'algorithme de rétropropagation [28] et les paramètres sont mis à jour à l'aide de la descente de gradient stochastique (SGD) [29]. La SGD a la propriété utile que les paramètres du modèle sont mis à jour de manière itérative en utilisant de petits lots de données. Cela permet à l'algorithme d'entraînement de s'adapter à de très grands ensembles de données. La structure hiérarchique en couches des réseaux de neurones rend possible l'apprentissage de bout en bout, ce qui implique que le réseau peut être formé pour prédire les sorties à partir d'entrées de bas niveau sans extraire de caractéristiques. Ceci est à l'opposé de nombreux autres modèles d'apprentissage machine dont les performances dépendent des caractéristiques extraites des données. Leur capacité à apprendre conjointement les transformations de caractéristiques et les classificateurs rend les réseaux neuronaux particulièrement bien adaptés aux problèmes de MIR [30].

A. Modèles acoustiques

1) **Réseaux neuronaux profonds** (Deep Neural Networks)

Les DNN sont de puissants modèles d'apprentissage machine qui peuvent être utilisés pour des tâches de classification et de régression. Les DNN sont caractérisés par une ou plusieurs couches de transformations non linéaires. Formellement, une couche d'un DNN effectue la transformation suivante:

$$h_{l+1} = f(W_l h_l + b_l). \qquad (1)$$

Dans l'équation 1, Wl, bl sont la matrice de pondération et le biais pour la couche l, $0 \leq l \leq L$ et f sont des fonctions non linéaires qui sont appliquées par élément. Pour la première couche, h0 = x, où x est l'entrée. Dans toutes nos expériences, nous fixons f comme étant la fonction sigmoïde (f(x) = 1 1+e-x). La sortie de la couche finale hL est transformée en fonction du problème donné pour obtenir une

distribution de probabilité postérieure sur les variables de sortie P(y|x, θ). Les paramètres θ = {Wl , bl} L 0, sont estimées numériquement avec l'algorithme de rétropropagation et SGD.

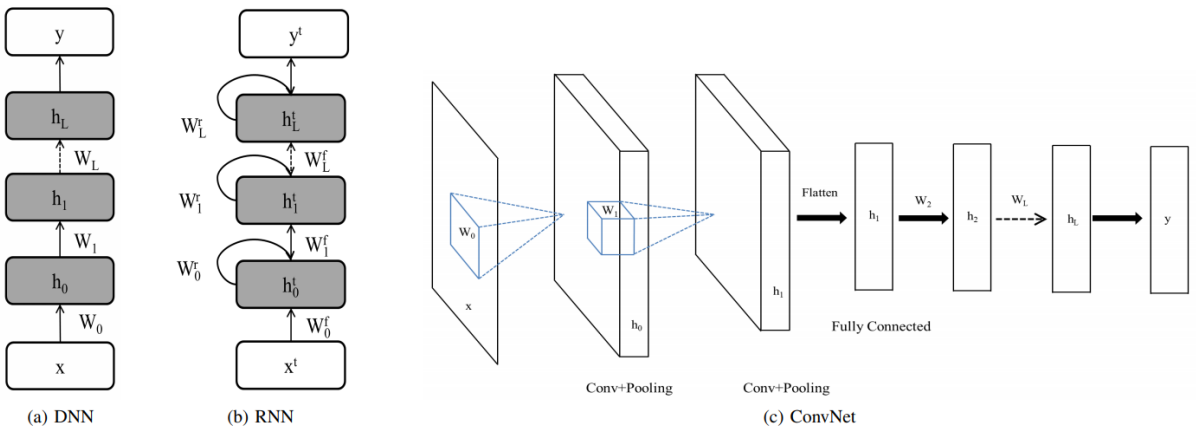


Fig. 1. Architectures de réseaux neuronaux pour la modélisation acoustique.

La figure 1a montre une représentation graphique de l'architecture DNN, les flèches en pointillés représentent les couches cachées intermédiaires. Pour la modélisation acoustique, l'entrée du DNN est un cadre de caractéristiques, par exemple un spectrogramme de magnitude ou la transformée Q constante (CQT) et le DNN est formé pour prédire la probabilité de tangages présents dans le cadre p(yt|xt) à un moment t.

2) Réseaux de neurones récurrents

Les DNN sont de bons classificateurs pour les données stationnaires, comme les images. Cependant, ils ne sont pas conçus pour prendre en compte les données séquentielles. Les RNN sont des extensions naturelles des DNN, conçus pour traiter des données séquentielles ou temporelles. Cela les rend plus adaptés aux tâches d'AMT, puisque les images d'audios consécutifs présentent des modèles temporels à court et à long terme [31]. Les RNN sont caractérisés par des connexions récursives entre les activations de la couche cachée à un moment t et les activations de la couche cachée à t-1, comme le montre la figure 1b. Formellement, la couche cachée d'un RNN au moment t effectue le calcul suivant :

$$h_{l+1}^t = f(W_l^f h_l^t + W_l^r h_l^{t-1} + b_l). \qquad (2)$$

Dans l'équation 2, W f l est la matrice de poids de l'entrée aux unités cachées, Wr l est la matrice de poids pour la connexion récurrente et bl sont les biais pour la couche l. A partir de l'équation 2, nous pouvons voir que la mise à jour récursive de l'état caché au temps t, implique que ht est implicitement une fonction de toutes les entrées jusqu'au temps t, x t 0. Tout comme les DNN, les RNN sont constitués d'une ou plusieurs couches d'unités cachées. Les sorties de la couche finale sont transformées à l'aide d'une fonction appropriée pour obtenir la distribution souhaitée sur les sorties. Les paramètres RNN θ = n W f l, Wr l, bl oL 0 sont calculés à l'aide de l'algorithme de rétropropagation dans le temps (BPTT) [32] et du SGD. Pour la modélisation acoustique, le RNN agit sur une séquence de caractéristiques d'entrée pour obtenir une distribution de probabilité sur les sorties P(yt|x t 0), où x t 0 = {x0, x1, . . . , xt}.

3) **Réseaux convolutionnels** (Convolutional Networks)

Les ConvNets sont des réseaux de neurones ayant une structure unique. Les couches convolutionnelles sont spécifiquement conçues pour préserver la structure spatiale des entrées. Dans une couche convolutive, un ensemble de poids agit sur une région locale de l'entrée. Ces poids sont ensuite appliqués de manière répétée à l'ensemble de l'entrée pour produire une carte des caractéristiques. Les couches convolutionnelles se caractérisent par le partage des poids sur l'ensemble de l'entrée. Comme le montre la figure 1c, les ConvNets sont constitués d'une alternance de couches convolutives et de couches de mise en commun, suivies d'une ou plusieurs couches entièrement connectées (comme les DNN). Formellement, l'application répétée des poids partagés au signal d'entrée constitue une opération de convolution :

$$h_{j,k} = f(\sum_{r} W_{r,j} x_{r+k-1} + b_j). \qquad (3)$$

L'entrée x est un vecteur d'entrées provenant de différents canaux, par exemple des canaux RVB pour les images. Formellement, x = {x0, x1, ...}, où chaque entrée xi représente un canal d'entrée. À chaque bande d'entrée xi est associée une matrice de pondération. Tous les poids d'une couche convolutive sont collectivement représentés par un tenseur quadridimensionnel. Étant donné une région m × n d'une carte de caractéristiques h, la fonction de regroupement max renvoie l'activation maximale dans la région. À tout moment t, l'entrée dans le ConvNet est une fenêtre de 2k + 1 trames de caractéristiques x t+k t-k. Les sorties de la couche finale donnent la distribution de la distribution postérieure P(yt|x t+k t-k).

Il existe plusieurs raisons d'utiliser les ConvNets pour la modélisation acoustique. De nombreuses expériences en MIR suggèrent qu'au lieu de classer une seule trame d'entrée, il est possible d'obtenir de meilleures précisions de prédiction en incorporant des informations sur plusieurs trames d'entrée [26], [33], [34]. Généralement, on y parvient soit en appliquant une fenêtre contextuelle autour de la trame d'entrée, soit en agrégeant les informations dans le temps en calculant des moments statistiques sur une fenêtre de trames. L'application d'une fenêtre contextuelle autour d'une trame de caractéristiques spectrales de bas niveau, comme la transformée de Fourier à temps court (STFT), conduirait à une entrée dimensionnelle très élevée, ce qui est peu pratique. Deuxièmement, la prise de la moyenne, de l'écart-type ou d'autres moments statistiques fait des hypothèses très simplistes sur la distribution des données dans le temps dans des cadres voisins. Grâce à leur architecture [12], les ConvNets peuvent être directement appliqués à plusieurs trames d'entrée pour apprendre des caractéristiques le long des axes de temps et de fréquence. De plus, en utilisant une représentation d'entrée comme le CQT, les ConvNets peuvent apprendre des caractéristiques invariantes de la hauteur de son, puisque l'espacement inter-harmonique des signaux musicaux est constant sur toute la fréquence logarithmique. Enfin, l'architecture de partage et de mise en commun du poids entraîne une réduction du nombre de paramètres du ConvNet, par rapport à un DNN entièrement connecté. C'est une propriété utile étant donné que de très grandes quantités de données étiquetées sont difficiles à obtenir pour la plupart des problèmes de MIR, y compris l'AMT.

B. Modèles de langage musical

Étant donné une séquence y = y t 0 , nous utilisons le MLM pour définir une distribution de probabilité préalable P(y). yt est un vecteur binaire à haute dimension qui représente les notes jouées à t (un pas de temps d'une représentation piano-roulement). La nature hautement dimensionnelle de l'espace de sortie fait de la modélisation de yt un problème difficile. La plupart des algorithmes de post-traitement font l'hypothèse simplificatrice que toutes les hauteurs sont indépendantes et modélisent leur évolution

temporelle à l'aide de modèles indépendants [13]. Cependant, pour la musique polyphonique, les hauteurs de son qui sont actives simultanément sont fortement corrélées (harmonies, accords). Dans cette section, nous décrivons les modèles de langage musical RNN introduits pour la première fois dans [35].

1) RNN génératif

Les RNN définis dans les sections précédentes ont été utilisés pour faire correspondre une séquence d'entrées x à une séquence de sorties y. À chaque pas de temps t, le RNN produit la distribution conditionnelle P(yt|x t 0). Toutefois, les RNN peuvent être utilisés pour définir une distribution sur une séquence y en reliant les sorties du RNN à t - 1 aux entrées du RNN à t, ce qui donne une distribution de la forme :

$$P(y) = P(y_0) \prod_{t>0} P(y_t | y_0^{t-1}) \qquad (4)$$

Bien qu'un RNN prévoie yt en fonction des entrées de dimension élevée y t-1 0 , les sorties de pas individuelles yt(i) sont indépendantes, où i est l'indice de pas (section IV-C). Comme mentionné précédemment, cela n'est pas vrai pour la musique polyphonique. Boulanger-Lewandowski et. al. [35] démontrent qu'au lieu de prédire des distributions indépendantes, les paramètres d'une distribution de sortie paramétrique plus complexe peuvent être conditionnés sur l'état caché RNN. Dans nos expériences, nous utilisons le RNN pour produire les biais d'un estimateur de distribution neuronale autorégressive (NADE) [35].

2) Estimateur de distribution autorégressif neuronal (Neural Autogressive Distribution Estimator)

Le NADE est un estimateur de distribution pour les données binaires à haute dimension [36]. Le NADE a été initialement proposé comme une alternative à la machine de Boltzmann restreinte (RBM). La NADE estime la distribution conjointe sur des variables binaires à haute dimension comme suit :

$$P(x) = \prod_i P(x_i | x_0^{i-1}).$$

Le NADE est similaire à un réseau de croyance sigmoïde entièrement visible [37], puisque la probabilité conditionnelle de xi est une fonction non linéaire de x t 0 . Le NADE calcule les distributions conditionnelles selon :

$$h_i = \sigma(W_{:,<i} x_0^{i-1} + b_h) \qquad (5)$$

$$P(x_i | x_0^{i-1}) = \sigma(V_i h_i + b_v^i) \qquad (6)$$

Où W, V sont des matrices de poids, W :,<I est une sous-matrice de W qui désigne les premières colonnes i - 1 et bh, bv sont les biais cachés et visibles, respectivement. Les gradients de la fonction de vraisemblance P(x) par rapport aux paramètres du modèle θ = {W, V, bh, bv} peuvent être trouvés exactement, ce qui n'est pas possible avec les MPR [36]. Cette propriété permet de combiner facilement le NADE avec d'autres modèles et les modèles peuvent être entraînés conjointement avec des optimiseurs basés sur des gradients.

3) RNN-NADE

Afin d'apprendre les distributions temporelles à haute dimension pour le MLM, nous combinons le NADE et un RNN, comme proposé dans [35]. Le modèle résultant donne une séquence de NADE conditionnée par un RNN, qui décrit une distribution sur des séquences de musique polyphonique. Le modèle commun est obtenu en laissant les paramètres du NADE à chaque étape de temps être une fonction de l'état caché du RNN θ t NADE = f(ht). ht est l'état caché de la couche finale du RNN (équation 2) au temps t. Afin de limiter le nombre de paramètres libres dans le modèle, nous permettons seulement aux biais du NADE d'être des fonctions de l'état caché du RNN, tandis que les paramètres restants (W, V) sont maintenus constants dans le temps. Nous calculons les biais NADE comme une transformation linéaire de l'état caché RNN plus un terme de biais ajouté [35] :

$$b_v^t = b_v + W_1 h_t \qquad (7)$$

$$b_h^t = b_h + W_2 h_t \qquad (8)$$

W1 et W2 sont des matrices de poids allant de l'état caché RNN aux biais visible et caché, respectivement. Les gradients par rapport à tous les paramètres du modèle peuvent être facilement calculés à l'aide de la règle de la chaîne et le modèle commun est formé à l'aide de l'algorithme BPTT [35].

III. MODELE PROPOSE

Dans cette section, nous examinons le modèle de réseau neuronal proposé pour l'AMT polyphonique. Comme mentionné précédemment, le modèle comprend un modèle acoustique et un modèle de langage musical. En plus des modèles acoustiques présentés dans [20], nous proposons l'utilisation de ConvNets pour identifier les hauteurs de son présentes dans le signal audio d'entrée et comparer leurs performances à celles de divers autres modèles acoustiques (section IV-F). Les modèles acoustiques et linguistiques sont combinés sous un seul objectif de formation en utilisant une architecture RNN hybride, ce qui donne un modèle de bout en bout pour l'AMT avec une polyphonie sans contrainte. Nous décrivons d'abord le modèle RNN hybride, puis l'algorithme d'inférence proposé.

A. RNN hybride

Le RNN hybride est un modèle graphique qui combine les prédictions de tout modèle acoustique arbitraire au niveau du cadre, avec un modèle de langage basé sur le RNN. Soit x = x T 0 une séquence d'entrées et soit y = y T 0 les transcriptions correspondantes. La probabilité conjointe de y, x peut être factorisée comme suit:

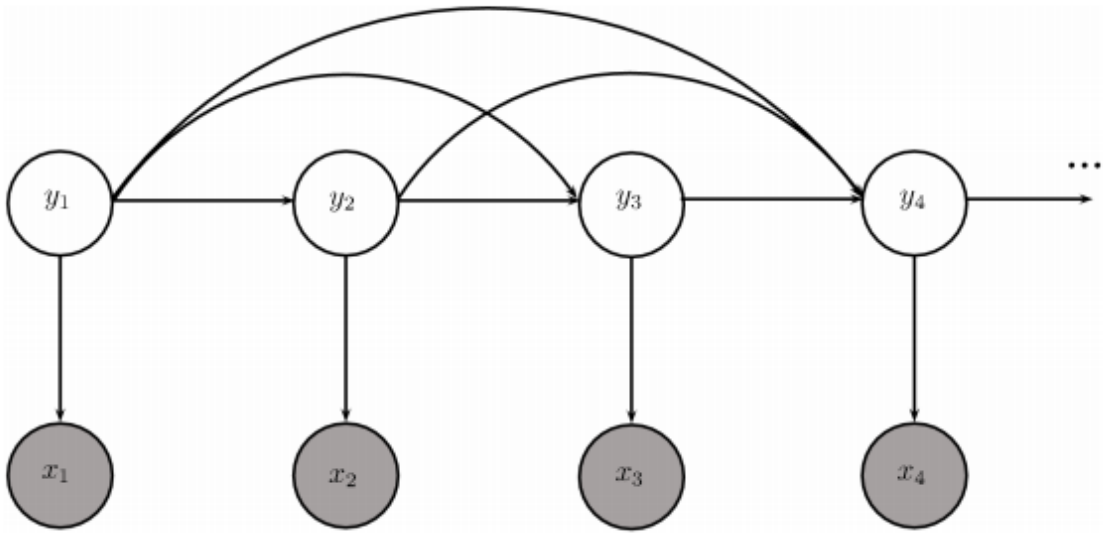


Fig. 2. Graphical Model of the Hybrid Architecture

$$P(y,x) = P(y_0 \dots y_T, x_0 \dots x_T) \tag{9}$$
$$= P(y_0)P(x_0|y_0)\prod_{t=1}^{T} P(y_t|y_0^{t-1})P(x_t|y_t).$$

La factorisation de l'équation 9 fait les hypothèses d'indépendance suivantes :

$$P(y_t|y_0^{t-1}, x_0^{t-1}) = P(y_t|y_0^{t-1}) \tag{10}$$

$$P(x_t|y_0^{t}, x_0^{t-1}) = P(x_t|y_t) \tag{11}$$

Ces hypothèses d'indépendance sont similaires aux hypothèses faites dans les HMM [38]. La figure 2 est une représentation graphique du modèle hybride. Dans l'équation 9, P(xt|yt) est la probabilité d'émission d'une entrée, étant donné la sortie yt. En utilisant la règle de Bayes, la distribution conditionnelle peut être écrite comme suit :

$$P(y|x) \propto P(y_0|x_0)\prod_{t=1}^{T} P(y_t|y_0^{t-1})P(y_t|x_t), \tag{12}$$

Où les marginaux P(yt) et les antécédents P(y0), P(x0) sont supposés être fixes sans les paramètres du modèle. Avec cette reformulation de la distribution conjointe, on observe que la distribution conditionnelle P(y|x) est directement proportionnelle au produit de deux distributions. La distribution antérieure P(yt|y t-1 0) est obtenue en utilisant un RNN génératif (Section II-B1) et la distribution postérieure sur des combinaisons de notes P(yt|xt) peut être modélisée en utilisant n'importe quel classificateur basé sur un cadre. Le modèle graphique hybride RNN est similaire à un HMM, où les probabilités de transition d'état pour le HMM P(yt|yt-1) ont été généralisées pour inclure les connexions de toutes les sorties précédentes, ce qui donne les termes P(yt|y t-1 0) dans l'équation 12. Pour le problème de la transcription automatique de la musique, la représentation temps-fréquence d'entrée

forme la séquence d'entrée x, tandis que la séquence piano-roulement de sortie y désigne les transcriptions. Les distributions antérieures P(yt|y t-1 0) sont obtenues à partir du MLM RNNNADE, tandis que les distributions postérieures P(yt|xt) sont obtenues à partir des modèles acoustiques. Les modèles peuvent ensuite être entraînés en trouvant les dérivées des objectifs des modèles acoustiques et linguistiques par rapport aux paramètres du modèle et en utilisant la descente de gradient. L'entraînement indépendant des modèles acoustiques et linguistiques est une propriété utile car les ensembles de données disponibles pour la transcription de la musique sont considérablement plus petits que les ensembles de données pour la vision et la parole par ordinateur. Cependant, il est relativement facile de trouver de grands corpus de musique MIDI sur l'internet. Par conséquent, en théorie, les MLM peuvent être formés sur de grands corpus de musique MIDI, comme c'est le cas pour les modèles de langage dans le domaine de la parole.

B. Inférence

Au moment du test, nous aimerions trouver le mode de la distribution conditionnelle de la sortie:

$$y^* = \underset{y}{\operatorname{argmax}}\, P(y|x) \qquad (13)$$

A partir de l'équation 12, nous observons que les antécédents P(yt|y t-1 0), lient les prédictions du modèle acoustique P(yt|xt) à toutes les prédictions faites jusqu'au temps t. Ce terme antérieur favorise la cohérence entre les prédictions dans le temps et permet à la structure musicologique apprise par les modèles de langage d'influencer les prédictions successives. Cependant, cette structure plus générale conduit à une procédure d'inférence (ou de décodage) plus complexe au moment du test. Cela est dû au fait qu'au temps t, l'histoire y t-1 0 n'a pas été déterminée de manière optimale. Par conséquent, le choix optimal de yt dépend de toutes les prédictions passées du modèle. Procéder avidement de manière chronologique en sélectionnant yt qui optimise P(yt|xt) ne donne pas nécessairement de bonnes solutions. Nous sommes intéressés par les solutions qui optimisent globalement p(y|x). Mais la recherche exhaustive de la meilleure séquence est insoluble puisque le nombre de configurations possibles de yt est exponentiel dans le nombre de pas de sortie (2 n pour n pas de pas). La recherche de faisceau est un algorithme de recherche de graphe qui est couramment utilisé pour décoder les sorties conditionnelles d'un RNN [39], [19], [26]. La recherche de faisceau s'étend à des séquences arbitrairement longues et le compromis entre le coût de calcul et la précision peut être contrôlé par la largeur du faisceau. L'algorithme d'inférence comprend les étapes suivantes : à tout moment t, l'algorithme maintient au maximum w solutions partielles, où w est la largeur du faisceau ou la capacité du faisceau. Les solutions dans le faisceau à t correspondent à des sous-séquences de longueur t. Ensuite, tous les descendants possibles des w solutions partielles dans le faisceau sont dénombrés et ensuite triés par ordre décroissant de log-vraisemblance. Parmi ces solutions candidates, les w solutions supérieures sont retenues comme entrées du faisceau pour une recherche plus approfondie. La recherche de faisceau peut être facilement appliquée aux problèmes pour lesquels le nombre de solutions candidates à chaque étape est limité, comme la reconnaissance vocale [40] et l'estimation d'accords audio [26]. Cependant, l'utilisation de la recherche de faisceau pour des séquences de décodage avec un grand espace de sortie est prohibitivement inefficace. Lorsque l'espace des solutions candidates est grand, l'algorithme peut être contraint de ne considérer que K nouveaux candidats pour chaque solution partielle dans le faisceau, où K est connu comme le facteur de branchement. La procédure de sélection des K candidats peut être conçue en fonction du problème donné. Pour l'architecture hybride, nous notons à partir de l'équation 12 :

$$P(y_0^t|x_0^t) \propto P(y_0^{t-1}|x_0^{t-1})P(y_t|y_0^{t-1})P(y_t|x_t) \qquad (14)$$

A l'instant t, les solutions partielles dans le faisceau correspondent à des configurations de y t-1 0 . Par conséquent, étant donné P(y t-1 0 |x t-1 0), les configurations K qui maximisent P(yt|y t-1 0)P(yt|xt) seraient un choix approprié de candidats pour yt. Cependant, pour de nombreuses familles de distributions, il pourrait ne pas être possible d'énumérer yt par ordre de probabilité décroissante. Dans [19], les auteurs proposent de former un groupe de K candidats en tirant des échantillons aléatoires des distributions conditionnelles de sortie. Cependant, l'échantillonnage aléatoire peut être inefficace et l'obtention d'échantillons indépendants peut être très coûteuse pour de nombreux types de distributions. Comme alternative, nous proposons d'échantillonner les solutions à partir de la distribution postérieure du modèle acoustique P(yt|xt) [20]. Il y a deux motivations principales pour ce faire. Premièrement, les sorties du modèle acoustique sont des probabilités de classe indépendantes. Par conséquent, il est facile d'énumérer les échantillons par ordre décroissant de log-vraisemblance [19]. Deuxièmement, nous évitons l'accumulation d'erreurs dans les prédictions RNN au fil du temps [41]. Les modèles RNN sont formés pour prédire yt, étant donné les sorties réelles y t-1 0 . Cependant, au moment du test, les sorties échantillonnées à partir du RNN sont réinjectées comme entrées au pas de temps suivant. Cet écart entre les objectifs de la formation et du test peut entraîner une accumulation d'erreurs de prédiction au fil du temps.

Algorithm 1 High Dimensional Beam Search

Find the most likely sequence y given x with a beam width w and branching factor K.

$beam \leftarrow$ new beam object
$beam$.insert$(0, \{\})$
for $t = 1$ to T **do**
  $new\_beam \leftarrow$ new beam object
  for l, s, m_a, m_l **in** $beam$ **do**
    for $k = 1$ to K **do**
      $y' = m_a$.next_most_probable()
      $l' = \log P_l(y'|s) P_a(y'|x_t) - \log P(y')$
      $m_l' \leftarrow m_l$ with $y_t := y'$
      $m_a' \leftarrow m_a$ with $x := x_{t+1}$
      new_beam.insert$(l + l', \{s, y'\}, m_a, m_l)$
  $beam \leftarrow new\_beam$
return $beam$.pop()

Bien que la génération de candidats à partir du modèle acoustique donne de bons résultats, elle nécessite l'utilisation de faisceaux de grande largeur. Cela rend la procédure d'inférence lente sur le plan du calcul et inadaptée aux applications en temps réel [20]. Dans cette étude, nous proposons d'utiliser l'algorithme de recherche de faisceau haché proposé dans [26]. La recherche de faisceau est fondamentalement limitée lors du décodage de longues séquences temporelles. Cela est dû au fait que des solutions qui diffèrent seulement à quelques pas de temps peuvent saturer le faisceau. Cela fait que l'algorithme recherche un espace très limité de solutions possibles. Ce problème peut être résolu par un élagage efficace. L'algorithme de recherche de faisceau haché améliore l'efficacité en élaguant les solutions qui sont similaires aux solutions ayant une plus grande probabilité. La métrique qui détermine la similarité des séquences peut être choisie en fonction du problème et est codée sous la forme d'une fonction de hachage sensible à la localité [26]. Dans l'algorithme 1, nous décrivons l'algorithme de recherche de faisceau utilisé pour nos expériences, tandis que l'algorithme 2 décrit l'objet faisceau de la table de hachage. Dans les algorithmes 1 et 2, s est une séquence y t 0 , l est la log-vraisemblance de s, ma, ml sont des objets acoustiques et de modèle de langage et fh est la fonction de hachage. Il existe deux différences essentielles entre l'algorithme 1 et l'algorithme de [20]. Premièrement, la file d'attente prioritaire qui stocke le faisceau est remplacée par un objet faisceau de table de hachage (voir Algorithme 2). Deuxièmement, pour chaque entrée du faisceau, nous évaluons K solutions candidates.

Cela contraste avec l'algorithme de [20], où une fois que le faisceau est plein, seules w solutions candidates sont évaluées par itération. Il pourrait sembler que l'algorithme de recherche du faisceau haché soit plus coûteux, puisqu'il évalue w ∗ K candidats au lieu de w candidats. Toutefois, en élaguant efficacement les solutions similaires, l'algorithme donne de meilleurs résultats pour des valeurs de w beaucoup plus petites, ce qui se traduit par une augmentation significative de l'efficacité (section IV-F, figure 3).

Algorithm 2 Description of beam objects given w, f_h, k

```
Initialise beam object
beam.hashQ = defaultdict of priority queues*
beam.queue = indexed priority queue of length w**
Insert l, s into beam
key= f_h(s)
queue = beam.queue
hashQ = beam.hashQ[key]
fits_in_queue = not queue.full() or l ≥queue.min()
fits_in_hashQ = not hashQ.full() or l ≥hashQ.min()
if fits_in_queue and fits_in_hashQ then
    hashQ.insert(l, s)
    if hashQ.overfull() then
        item = hashQ.del_min()
        queue.remove(item)
    queue.insert(l, s)
    if queue.overfull() then
        item = queue.del_min()
        beam.hashQ[f_h(item.s)].remove(item)
```

* A priority queue of length k maintains the top k entries at all times.

** An *indexed* priority queue allows efficient random access and deletion.

L'algorithme 2 décrit l'objet faisceau de la table de hachage. L'algorithme de recherche de faisceau haché offre plusieurs avantages par rapport à l'algorithme de recherche de faisceau standard. La notion de similarité des solutions peut être encodée sous la forme de fonctions de hachage. Pour la transcription musicale, nous choisissons la fonction de similarité comme étant les n dernières trames d'une séquence s. n = 1 correspond à une programmation dynamique comme le décodage (similaire aux HMM) où toutes les séquences ayant le même état final yt sont considérées comme équivalentes, et la séquence ayant la plus forte log-vraisemblance est retenue. n = len(séquence) c

IV. EVALUATION

Dans cette section, nous décrivons comment la performance du modèle proposé est évaluée pour une tâche de transcription polyphonique.

A. Ensemble de données

Nous évaluons le modèle proposé sur l'ensemble de données MAPS [43]. L'ensemble de données est constitué de données audio et d'annotations correspondantes pour des sons isolés, des accords et des morceaux complets de musique pour piano. Pour nos expériences, nous n'utilisons que les morceaux

de musique complets pour l'entraînement et le test des modèles acoustiques des réseaux de neurones et des MLM. L'ensemble de données se compose de 270 morceaux de musique classique et d'annotations MIDI. Il existe 9 catégories d'enregistrements correspondant à différents types de pianos et à différentes conditions d'enregistrement, avec 30 enregistrements par catégorie. 7 catégories d'audio sont produites par des synthétiseurs de piano logiciels, tandis que 2 ensembles d'enregistrements sont obtenus à partir d'un piano droit Yamaha Disklavier. L'ensemble de données comprend donc 210 enregistrements synthétisés et 60 enregistrements réels. Nous effectuons 2 séries d'investigations dans ce document. La première série d'expériences étudie l'effet des MLM RNN sur les prédictions des modèles acoustiques. Pour ce faire, nous divisons l'ensemble des données en 4 séries de tests disjoints, afin de s'assurer que les plis sont indépendants des morceaux de musique. Plus précisément, pour certains des morceaux de musique de l'ensemble de données, l'audio de chaque morceau est rendu en utilisant plus d'un piano. Par conséquent, lors de la création des splits, nous veillons à ce que les données de formation et de test ne contiennent aucun morceau qui se chevauche1 . Pour chaque fractionnement, nous sélectionnons 80 % des données pour l'entraînement (216 morceaux de musique) et le reste pour les tests (54 morceaux). Pour chaque fractionnement de formation, nous conservons 26 pistes comme ensemble de validation pour la sélection des hyperparamètres de l'algorithme de formation (section IV-D). Tous les résultats rapportés sont des valeurs moyennes des paramètres d'évaluation sur les 4 split. A partir de maintenant, cette configuration d'évaluation sera appelée Configuration 1. Bien que la configuration expérimentale ci-dessus soit utile pour étudier l'efficacité des MLM RNN, le kit de formation contient des exemples de modèles de pianos qui sont utilisés pour les tests. Cela n'est généralement pas vrai dans la pratique, où les modèles/sources des instruments au moment du test sont inconnus et ne coïncident généralement pas avec les instruments utilisés pour la formation. Une majorité d'expériences avec l'ensemble de données MAPS entraîne et teste des modèles d'instruments disjoints [3], [2], [44]. Nous réalisons donc une deuxième série d'expériences pour comparer les performances des différents modèles acoustiques de réseaux de neurones dans un cadre plus réaliste. Nous formons les modèles acoustiques en utilisant les 210 pistes créées à l'aide de pianos synthétisés (180 pistes pour la formation et 30 pistes pour la validation) et nous testons les modèles acoustiques sur les 60 enregistrements audio obtenus à partir des enregistrements de pianos Yamaha Disklavier (modèles "ENSTDkAm" et "ENSTDkCl" dans la base de données MAPS). Dans cette expérience, nous n'appliquons pas les modèles linguistiques puisque les trains et les jeux de test contiennent des morceaux de musique qui se chevauchent. En plus des modèles acoustiques du réseau de neurones, nous incluons des comparaisons avec deux modèles acoustiques non supervisés de pointe [3], [8] pour les deux expériences. Cette configuration d'évaluation indépendante de la source de l'instrument sera dorénavant appelée Configuration 2

B. Mesures

Nous utilisons des mesures basées sur des cadres et des notes pour évaluer la performance du système proposé [45]. Les évaluations basées sur les trames sont faites en comparant la sortie binaire transcrite et la vérité de base MIDI trame par trame. Pour l'évaluation basée sur les notes, le système renvoie une liste de notes, avec les hauteurs correspondantes, le temps d'apparition et de décalage. Nous utilisons la mesure, la précision, le rappel et l'exactitude de Fmeasure pour l'évaluation basée sur les images et les notes. Formellement, les mesures basées sur les trames sont définies comme suit :

$$\mathcal{P} = \sum_{t=1}^{T} \frac{TP[t]}{TP[t] + FP[t]}$$

$$\mathcal{R} = \sum_{t=1}^{T} \frac{TP[t]}{TP[t] + FN[t]}$$

$$\mathcal{A} = \sum_{t=1}^{T} \frac{TP[t]}{TP[t] + FP[t] + FN[t]}$$

$$\mathcal{F} = \frac{2 * \mathcal{P} * \mathcal{R}}{\mathcal{P} + \mathcal{R}}$$

Où TP[t] est le nombre de vrais positifs pour l'événement à t, FP est le nombre de faux positifs et FN est le nombre de faux négatifs. La sommation sur T est effectuée sur l'ensemble des données de l'essai. De même, des mesures analogues basées sur des notes peuvent être définies [45]. Un événement de note est considéré comme correct si sa hauteur prévue se situe dans une plage de ±50 ms par rapport à la vérité de base.

C. Prétraitement (Preprocessing)

Nous transformons l'audio d'entrée en une représentation temps-fréquence qui est ensuite entrée dans les modèles acoustiques. Dans [20], nous avons utilisé la transformée de Fourier à court terme (STFT) comme entrée pour les modèles acoustiques. Cependant, ici nous expérimentons avec la transformée Q constante (CQT) comme représentation d'entrée. Il y a deux raisons à cela. Premièrement, la CQT est fondamentalement mieux adaptée comme représentation temps-fréquence pour les signaux musicaux, puisque l'axe des fréquences est linéaire en hauteur [46]. Un autre avantage de l'utilisation du CQT est que la représentation résultante est beaucoup moins dimensionnelle que la STFT. Une représentation à plus faible dimension est utile lors de l'utilisation de modèles acoustiques de réseaux de neurones, car elle réduit le nombre de paramètres du modèle.

Nous réduisons l'échantillon audio de 44,1 kHz à 16 kHz. Nous calculons ensuite les CQT sur 7 octaves avec 36 cases par octave et une taille de saut de 512 échantillons, ce qui donne un vecteur d'entrée à 252 dimensions de valeurs réelles, avec une fréquence de trame de 31,25 images par seconde. En outre, nous calculons la moyenne et l'écart-type de chaque dimension sur l'ensemble de la formation et transformons les données en soustrayant la moyenne et la plongée de l'écart-type. Ces vecteurs prétraités sont utilisés comme entrées dans le modèle acoustique. Pour la formation au modèle linguistique, nous échantillonnons les transcriptions de vérité de sol MIDI des données de formation au même rythme que l'audio (32 ms). Nous obtenons des séquences de 88 vecteurs binaires dimensionnels pour l'apprentissage des modèles linguistiques RNN-NADE 8. Les 88 sorties correspondent aux notes A0-C8 sur un piano.

L'audio de test est échantillonné à une fréquence de 100 Hz, ce qui donne $100 * 30 = 3000$ images par fichier de test. Pour 54 fichiers test répartis sur 4 split, nous obtenons un total de 648 000 images au moment du test2.

D. Formation au réseau

Dans cette section, nous décrivons les détails de la procédure de formation pour les différentes architectures de modèles acoustiques et le modèle de langage RNN-NADE. Tous les modèles acoustiques ont 88 unités dans la couche de sortie, correspondant aux 88 pas de sortie. Les sorties de la couche finale sont transformées par une fonction sigmoïde et donnent des probabilités de pas indépendantes P(yt(i) = 1|x). Tous les modèles sont formés en maximisant la log-vraisemblance sur tous les exemples de l'ensemble de formation.

1) Modèles acoustiques DNN

Pour la formation DNN, nous contraignons toutes les couches cachées du modèle à avoir le même nombre d'unités afin de simplifier la recherche de bonnes architectures de modèles. Nous effectuons une recherche par grille sur les paramètres suivants : nombre de couches L ∈ {1, 2, 3, 4}, nombre d'unités cachées H ∈ {25, 50, 100, 125, 150, 200, 250}, activations des unités cachées act ∈ {ReLU, sigmoïde} où ReLU est la fonction d'activation des unités linéaires rectifiées [48]. Nous avons constaté que l'abandon [49] est essentiel pour améliorer les performances de généralisation. Un taux d'abandon de 0,3 a été utilisé pour la couche d'entrée et toutes les couches cachées du réseau. Plutôt que d'utiliser le taux d'apprentissage et les calendriers de mise à jour du momentum, nous utilisons ADADELTA [50] pour adapter l'apprentissage au fil des itérations. En plus de l'abandon, nous utilisons l'arrêt anticipé pour minimiser le suréquipement. La formation est arrêtée si le coût de l'ensemble de validation n'a pas diminué pendant 20 époques. Nous avons utilisé des mini lots de taille 100 pour les mises à jour du SGD.

2) Modèles acoustiques RNN

Pour la formation RNN, nous contraignons toutes les couches cachées à avoir le même nombre d'unités. Nous effectuons une recherche par grille sur les paramètres suivants : L ∈ {1, 2, 3}, H ∈ {25, 50, 100, 150, 200, 250}. Nous fixons les activations cachées des couches récurrentes comme étant la fonction tangente hyperbolique. Nous avons constaté que ADADELTA n'était pas particulièrement bien adapté à la formation des IA. Nous utilisons un taux d'apprentissage initial de 0,001 et le réduisons linéairement à 0 sur 1000 itérations. Nous utilisons un taux d'impulsion constant de 0,9. Les séquences de formation sont ensuite divisées en sous-séquences de longueur 100. Les mises à jour du SGD sont effectuées une sous-séquence à la fois, sans aucun mini lot. Comme pour les DNN, nous utilisons un arrêt anticipé et arrêtons l'entraînement si le coût de validation ne diminue pas après 20 itérations. Afin d'éviter l'explosion des gradients au début de l'entraînement, nous utilisons l'écrêtage des gradients [51]. Nous découpons les gradients lorsque la norme du gradient est supérieure à 5.

3) Modèles acoustiques ConvNet

L'entrée du ConvNet est une fenêtre contextuelle de cadres et la cible est le cadre central de la fenêtre [26]. Les trames au début et à la fin de l'audio sont sans rembourrage afin qu'une fenêtre contextuelle puisse être appliquée à chaque trame. Bien que le pooling puisse être effectué sur les deux axes, nous n'effectuons le pooling que sur l'axe des fréquences. Nous avons effectué une recherche de grille sur les paramètres suivants : taille de la fenêtre ws ∈ {3, 5, 7, 9}, nombre de couches convolutionnelles Lc ∈ {1, 2, 3, 4}, nombre de filtres par couche nl ∈ {10, 25, 50, 75, 100}, nombre de couches entièrement connectées Lf c ∈ {1, 2, 3}, nombre d'unités cachées dans les couches entièrement connectées H ∈ {200, 500, 1000}. Les fonctions d'activation de convolution ont été fixées comme étant les fonctions tangentes hyperboliques, tandis que toutes les activations de couches entièrement connectées ont été fixées comme étant la fonction sigmoïde. La taille de la mise en commun est fixée à P = (1, 3) pour toutes les couches convolutionnelles. L'abandon avec un taux de 0,5 est appliqué à toutes les couches convolutives. Nous avons essayé une grande permutation des formes de fenêtres pour la couche convolutive et le sous-

ensemble suivant de formes de fenêtres a donné de bons résultats : w ∈ {(3, 3),(3, 5),(5, 5),(3, 25),(5, 25),(3, 75),(5, 75)}. Nous avons observé que la performance de classification se détériorait fortement pour les filtres plus longs le long de l'axe des fréquences. Un dropout de 0,5 a été appliqué à toutes les couches entièrement connectées. Les paramètres du modèle ont été formés avec SGD et une taille de lot de 256. Un taux d'apprentissage initial de 0,01 a été linéairement réduit à 0 sur 1000 itérations. Un taux d'impulsion constant de 0,9 a été utilisé pour toutes les mises à jour. Nous avons arrêté la formation si l'erreur de validation ne diminuait pas après 20 itérations sur l'ensemble de la formation.

4) Modèles linguistiques RNN-NADE

Les modèles RNN-NADE ont été formés avec SGD et avec des séquences de longueur 100. Nous avons effectué une recherche par grille sur les paramètres suivants : nombre d'unités récurrentes HRNN ∈ {50, 100, 150, 200, 250, 300} et nombre d'unités cachées pour le NADE HNADE ∈ {50, 100, 150, 200, 250, 300}. Le modèle a été formé avec un taux d'apprentissage initial de 0,001 qui a été réduit linéairement à 0 au cours de 1000 itérations. Un taux d'impulsion constant de 0,9 a été appliqué tout au long de la formation. Nous avons sélectionné les architectures du modèle en effectuant une recherche par grille sur les valeurs des paramètres décrits plus haut dans la section. Les différents modèles ont été évalués sur une répartition train/test et l'architecture la plus performante a ensuite été utilisée pour toutes les autres expériences.

E. Approches comparatives

À des fins de comparaison, deux méthodes de transcription de musique polyphonique de pointe ont été utilisées pour les expériences [3], [8]. Dans les deux cas, la sortie d'activation de la hauteur non binaire des méthodes susmentionnées a été extraite, afin d'effectuer une comparaison approfondie avec les modèles de réseaux de neurones proposés. La méthode de détection multi-pitch de [8] est basée sur la factorisation matricielle non négative (NMF) et fonctionne en décomposant une représentation temps-fréquence d'entrée en une série de spectres de base (représentant les pitchs) et d'activations de composants (indiquant l'activité des pitchs dans le temps). Cette méthode modélise chaque spectre de base comme une somme pondérée de spectres à bande étroite représentant quelques partiels harmoniques adjacents, renforçant l'harmonicité et la fluidité spectrale. Comme représentation temps-fréquence d'entrée, une banque de filtres à largeur de bande rectangulaire équivalente (ERB) est utilisée. Comme la méthode s'appuie sur un dictionnaire de spectres harmoniques à bande étroite (fait à la main), les paramètres du système restent les mêmes pour les deux configurations d'évaluation.

Post Processing	Thresholding		HMM		Hybrid Architecture	
Acoustic Model	Frame	Note	Frame	Note	Frame	Note
Benetos [3]	64.20	65.22	64.84	66.05	65.10	66.48
Vincent [8]	58.95	**68.5**	60.37	**68.87**	59.78	**69.00**
DNN	67.54	60.02	68.32	62.26	67.92	63.18
RNN	68.38	63.84	68.09	64.50	69.25	65.24
ConvNet	**73.57**	65.35	**73.75**	66.20	**74.45**	67.05

TABLE I
F-MEASURES FOR MULTIPLE PITCH DETECTION ON THE MAPS DATASET, USING EVALUATION CONFIGURATION 1.

	$\mathcal{P}$		$\mathcal{R}$		$\mathcal{A}$	
Acoustic Model	Frame	Note	Frame	Note	Frame	Note
Benetos [3]	59.54	73.51	69.51	60.67	48.47	49.03
Vincent [8]	52.71	**79.93**	69.04	60.69	43.04	**52.92**
DNN	65.66	62.62	70.34	63.75	51.76	45.33
RNN	67.89	64.64	70.66	65.85	54.38	48.18
ConvNet	**72.45**	67.75	**76.56**	**66.36**	**58.87**	50.07

TABLE II
PRECISION, RECALL AND ACCURACY FOR MULTIPLE PITCH DETECTION ON THE MAPS DATASET USING THE HYBRID ARCHITECTURE ($w = 10, K = 4, k = 2, f_h(y_0^t) = y_t$), USING EVALUATION CONFIGURATION 1.

Acoustic Model	Benetos [3]	Vincent [8]	DNN	RNN	ConvNet
F-measure (Frame)	59.31	59.60	59.91	57.67	**64.14**
F-measure (Note)	54.29	**59.12**	49.43	49.20	54.89

TABLE III
F-MEASURES FOR ACOUSTIC MODELS TRAINED ON SYNTHESISED PIANOS AND TESTED ON REAL RECORDINGS (EVALUATION CONFIGURATION 2).

La méthode de transcription multi-instruments de [3] est basée sur le PLCA invariant par déplacement (une contrepartie convolutive et probabiliste du NMF). Dans ce modèle, la représentation temps-fréquence d'entrée est décomposée en une série de spectres de base par hauteur et par source d'instrument qui sont décalés sur la log-fréquence, ce qui permet de prendre en charge les changements d'accord et les modulations de fréquence. Les sorties comprennent la distribution d'activation des hauteurs et la contribution de la source de l'instrument par hauteur. Contrairement au modèle paramétrique de [8], les spectres de base sont pré-extraits de sons d'instruments de musique isolés. Comme dans la méthode proposée, la représentation temps-fréquence d'entrée de [3] est le CQT. Pour les recherches avec les MLM (configuration 1), les modèles PLCA sont formés sur des exemples de sons isolés provenant des 9 modèles de pianos de la base de données MAPS (afin que les expériences soient comparables à la méthode proposée). Pour la deuxième série d'expériences qui étudient les capacités de généralisation des modèles (configuration 2), le modèle acoustique PLCA est formé sur des sons isolés provenant des pianos synthétisés et testé sur des enregistrements créés à l'aide du piano Yamaha Disklavier.

F. Résultats

Dans cette section, nous présentons les résultats des expériences sur l'ensemble de données MAPS. Comme mentionné précédemment, tous les résultats sont les valeurs moyennes de diverses mesures calculées sur les 4 différents fractionnements de train/test. Les modèles acoustiques donnent une séquence de probabilités pour les différentes hauteurs actives (posteriogrammes). Les méthodes de post-traitement sont utilisées pour transformer les postériogrammes en une représentation binaire de type piano à roulette. Les différentes mesures de performance (à la fois sur la base de la trame et de la note) sont ensuite calculées en comparant les sorties des systèmes à la réalité du terrain.

Model	Architecture
DNN	$L = 3, H = 125$
RNN	$L = 2, H = 200$
ConvNet	$w_s = 7, L_c = 2, L_{fc} = 2, w_1 = (5, 25), P_1 = (1, 3)$ $w_2 = (3, 5), P_2 = (1, 3), n_1 = n_2 = 50, h_1 = 1000, h_2 = 200$
RNN-NADE	$H_{RNN} = 200, H_{NADE} = 150$

TABLE IV
MODEL CONFIGURATIONS FOR THE BEST PERFORMING ARCHITECTURES.

Nous considérons 3 types de méthodes de post-traitement. La méthode de post-traitement la plus simple consiste à appliquer un seuil aux probabilités de hauteur de son de sortie obtenue à partir du modèle acoustique. Nous sélectionnons le seuil qui maximise la mesure F sur l'ensemble de la formation et utilisons ce seuil pour les tests. Les hauteurs dont les probabilités sont supérieures au seuil sont fixées à 1, tandis que les hauteurs restantes sont fixées à 0. La deuxième méthode de post-traitement considérée utilise des HMM de hauteur individuelle pour le post-traitement, comme dans [13]. Les paramètres HMM (probabilités de transition, marges de hauteur) sont obtenus en comptant la fréquence de chaque événement sur les données de vérité de terrain MIDI. Les sorties binaires sont obtenues par décodage de Viterbi [38], où les probabilités mises à l'échelle sont utilisées comme probabilités d'émission. Enfin, nous combinons les prédictions du modèle acoustique avec les MLM RNN-NADE et obtenons des transcriptions binaires en utilisant la recherche de faisceau.

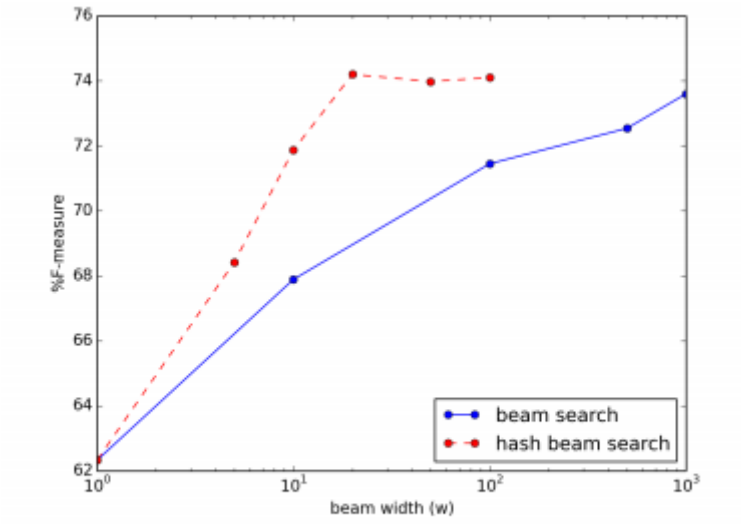


Fig. 3. Effect of beam width (w) on F-measure using evaluation Configuration 1. $k = 2, K = 4, f_h = y_t$.

Dans le tableau I, nous présentons les scores F (à la fois par trame et par note) pour tous les modèles acoustiques et les 3 méthodes de post-traitement utilisant la configuration 1. Dans le tableau, nous constatons que tous les modèles de réseaux neuronaux surpassent les modèles PLCA et NMF en termes de mesure F basée sur les trames de 3 à 9 %. Les performances des modèles acoustiques DNN et RNN sont similaires, tandis que le modèle acoustique ConvNet surpasse clairement tous les autres modèles. Les ConvNets apportent une amélioration absolue de $\sim$ 5% par rapport aux autres modèles de réseaux neuronaux, tout en surpassant les modèles de factorisation des spectrogrammes de $\sim$ 10% en termes de mesure F par trame. Pour la mesure de F basée sur les notes, les modèles RNN et ConvNet sont plus performants que le modèle acoustique DNN. Cela est dû en grande partie au fait que ces modèles incluent des informations contextuelles dans leurs entrées, ce qui lisse implicitement les prévisions de sortie.

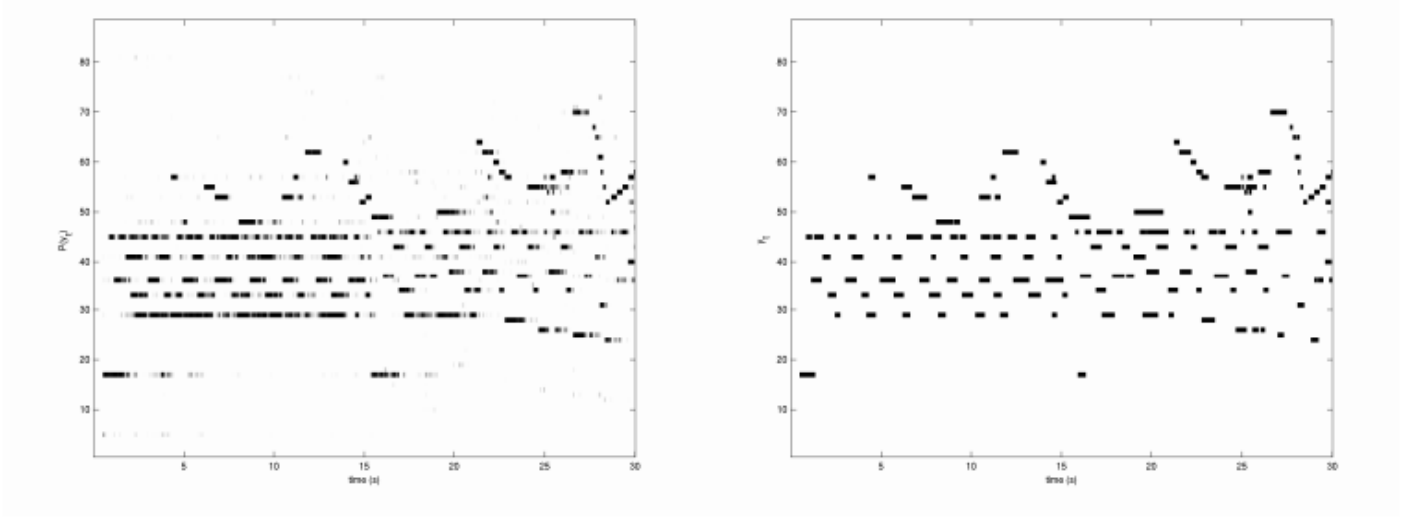
Nous comparons les différentes méthodes de post-traitement pour la configuration 1 en observant les lignes du tableau I. Nous constatons que le MLM permet d'améliorer les performances des mesures de F basées sur les trames et les notes pour tous les modèles acoustiques. L'augmentation des performances est plus importante sur la mesure de F basée sur la note. L'amélioration relative des performances est maximale pour le modèle acoustique DNN, par rapport au RNN et au ConvNet. Cela pourrait être dû au fait que l'hypothèse d'indépendance de l'équation 11 est violée par le RNN et le ConvNet, qui incluent des informations contextuelles lorsqu'ils font des prédictions. Cela conduit à ce que certains facteurs soient comptés deux fois et nous observons une plus petite amélioration des performances dans ce cas. Dans les lignes 1 et 2 du tableau I, nous observons que le MLM RNN-NADE produit une augmentation de performance pour les modèles acoustiques PLCA et NMF, bien que l'amélioration relative soit moindre par rapport aux modèles acoustiques du réseau neuronal. Cela pourrait être dû au fait que, contrairement aux modèles de réseaux de neurones, ces modèles ne sont pas entraînés à maximiser la probabilité conditionnelle des hauteurs de sortie en fonction des entrées acoustiques. Un autre facteur qui contribue à cette situation est le fait que les posteriogrammes PLCA et NMF représentent la distribution de l'énergie sur les hauteurs de ton plutôt que les probabilités explicites de hauteur de ton, ce qui fait que de nombreuses activations sont supérieures à 1. Cette divergence dans l'échelle des prédictions acoustiques et linguistiques conduit à une pondération inégale des prédictions lorsqu'elles sont utilisées dans le cadre hybride RNN. Dans le tableau I, nous observons que le modèle acoustique de [8] surpasse tous les autres modèles acoustiques en ce qui concerne la mesure de F basée sur la note, alors que la mesure de F basée sur le cadre est nettement inférieure. Cela peut être attribué à l'utilisation d'une représentation d'entrée de la banque de filtres ERB, qui offre une résolution temporelle améliorée par rapport à la CQT pour les basses fréquences.

Dans le tableau II, nous présentons des mesures supplémentaires (précision, rappel et exactitude) pour tous les modèles acoustiques après décodage avec un RNN-MLM, en utilisant la configuration 1. Nous observons que les modèles NMF et PLCA ont une précision basée sur les trames faible et un rappel élevé et l'inverse pour la précision basée sur les notes. Pour les modèles de réseaux neuronaux, nous observons de plus petites différences entre les valeurs de précision et de rappel à la fois basées sur les trames et sur les notes. Parmi tous les modèles de réseaux neuronaux, nous observons que le ConvNet surpasse tous les autres modèles pour toutes les mesures.

Dans le tableau III, nous présentons les mesures F pour les expériences où les modèles acoustiques sont formés sur des données synthétisées et testés sur des données réelles (configuration 2). Dans le tableau, nous notons que la mesure F basée sur le cadre pour les modèles DNN et RNN est similaire au modèle PLCA et au modèle de [8]. Nous notons que le ConvNet surpasse tous les autres modèles sur la mesure F basée sur la trame de $\sim 5\%$. Sur la base des évaluations basées sur les notes, nous observons que RNN et DNN sont tous deux plus performants que les autres modèles. Les performances du ConvNet sont similaires à celles du modèle PLCA, tandis que le modèle acoustique de [8] est à nouveau le plus performant pour les mesures basées sur les notes.

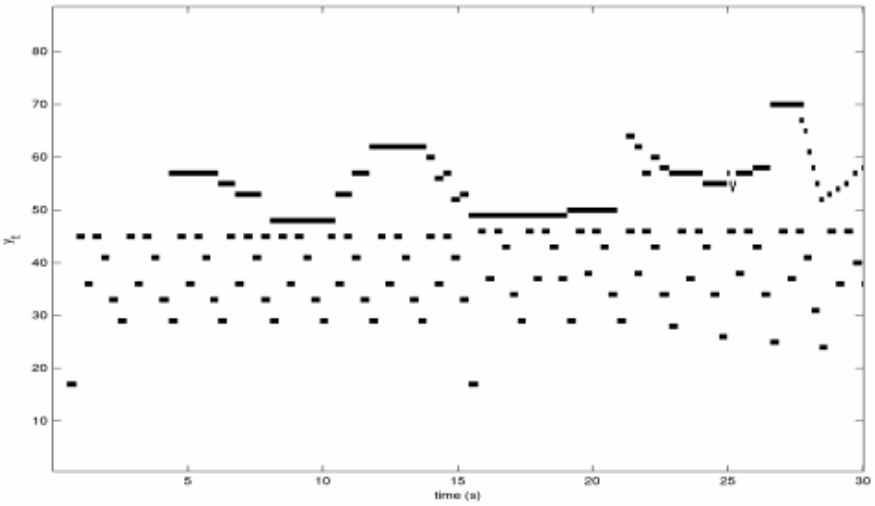
Nous discutons maintenant des détails de l'algorithme d'inférence. L'algorithme de recherche de faisceau haché à haute dimension a les paramètres suivants : la largeur du faisceau w, le facteur de ramification K, le nombre d'entrées par entrée de la table de hachage k et la métrique de similarité fh (algorithme 2). Nous avons observé qu'une valeur de $K \geq 4$ donnait de bons résultats. Des valeurs plus élevées de K ne donnent pas une augmentation significative des performances et entraînent des durées d'exécution beaucoup plus longues, c'est pourquoi nous avons fixé $K = 4$ pour toutes les expériences. Nous avons observé que de petites valeurs de k (nombre de solutions par entrée de la table de hachage), $1 \leq k \leq 4$ ont donné de bons résultats. Les précisions de décodage se détériorent fortement pour les grandes valeurs de k, comme observé dans [26]. Par conséquent, nous avons fixé le nombre d'entrées par clé de hachage $k = 2$ pour toutes les expériences. Nous avons laissé la métrique de similarité être les n derniers symboles émis, $fh(y\ t\ 0\ ) = y\ t\ t\text{-}n$. Nous avons expérimenté en faisant varier les valeurs de n et avons observé que nous étions capables d'obtenir de bonnes performances pour le petit n, $1 \leq n \leq 5$. Nous n'avons observé

aucune amélioration des performances pour le grand n, c'est pourquoi nous fixons fh(y t 0) = yt pour toutes les expériences. La figure 3 est un graphique montrant l'effet de la largeur de faisceau w sur les performances de transcription. Les résultats sont des valeurs moyennes des précisions de décodage sur 4 fractionnements. Nous comparons les performances de la recherche par faisceau haché avec la recherche par faisceau à haute dimension dans [20]. La figure 3 montre que l'algorithme de recherche de faisceau haché est capable d'améliorer les performances avec des largeurs de faisceau nettement plus petites. Par exemple, l'algorithme de recherche de faisceau à haute dimension prend 20 heures pour décoder l'ensemble du test avec w = 100, tandis que la recherche de faisceau haché prend 22 minutes, avec w = 10 et permet d'obtenir une meilleure précision de décodage.



(a) ConvNet Posteriogram

(b) ConvNet Transcription



(c) Ground Truth

Fig. 4. a) Pitch-activation (posteriogram) matrix for the first 30 seconds of track MAPS_MUS-chpn_op27_2_AkPnStgb produced by a ConvNet acoustic model. b) Binary piano-roll transcription obtained from posteriogram in a) after post processing with RNN MLM and beam search. c) Corresponding ground truth piano roll representation.

La figure 4 est une représentation graphique des sorties d'un modèle acoustique ConvNet. On observe que certaines des notes les plus longues sont fragmentées et que les décalages sont mal estimés. L'une des raisons en est que les décalages de la vérité de sol ne correspondent pas nécessairement au décalage du signal acoustique (en raison des effets de la pédale de sustain), ce qui implique des décalages bruyants

dans la vérité de sol. On observe également que le modèle ne fait pas beaucoup d'erreurs harmoniques dans ses prédictions.

V. CONCLUSION

Dans cet article, nous présentons un modèle RNN hybride pour l'AMT polyphonique de la musique pour piano. Le modèle comprend un modèle acoustique à réseau de neurones et un modèle de langage musical basé sur le RNN. Nous proposons d'utiliser un ConvNet pour la modélisation acoustique, ce qui, à la connaissance des auteurs, n'a jamais été tenté auparavant pour l'AMT. Nos expériences sur l'ensemble de données MAPS démontrent que les modèles acoustiques de réseaux de neurones, en particulier le ConvNet, surpassent deux modèles acoustiques populaires de la littérature sur l'AMT. Nous observons également que les MLM du RNN améliorent constamment les performances sur tous les paramètres d'évaluation. L'algorithme d'inférence proposé avec la recherche de faisceau de hachage est capable de produire de bonnes précisions de décodage avec des durées d'exécution nettement plus courtes, ce qui rend le modèle adapté aux applications en temps réel.

Nous discutons maintenant de certaines des limites du modèle proposé. Comme nous l'avons déjà dit, l'un des principaux facteurs de succès des réseaux neuronaux profonds est la disponibilité de très grands ensembles de données. Cependant, les ensembles de données disponibles pour la recherche sur l'AMT sont considérablement plus petits que ceux disponibles pour la parole, la vision par ordinateur et le traitement du langage naturel (NLP). Par conséquent, l'applicabilité des réseaux neuronaux profonds pour la modélisation acoustique est limitée aux ensembles de données comportant de grandes quantités de données étiquetées, ce qui n'est pas courant en AMT (du moins pour la musique non pianistique). Bien que les modèles acoustiques des réseaux neuronaux soient performants, leurs performances pourraient être encore améliorées de nombreuses façons. Des bruits ou des déformations peuvent être ajoutés aux exemples de formation pour encourager les classificateurs à être invariants aux transformations d'entrée couramment rencontrées. En outre, la représentation d'entrée CQT peut être remplacée par une représentation à plus haute résolution temporelle (comme l'ERB ou une transformation à Q variable), afin d'améliorer les performances des mesures basées sur les notes.

L'abondance des données sur les partitions musicales et les progrès récents dans les tâches de PNL avec les réseaux de neurones constituent une forte motivation pour poursuivre les recherches sur les MLM pour l'AMT. Bien que nos résultats montrent une certaine amélioration des performances de transcription avec les MLM, il reste plusieurs limites et questions ouvertes. Les MLM sont formés sur des vecteurs binaires échantillonnés à partir de la vérité de base MIDI. Selon le taux d'échantillonnage, la plupart des événements de note sont répétés plusieurs fois dans cette représentation. Les MLM sont entraînés à prédire la prochaine trame de notes, à partir d'une séquence d'entrée de combinaisons de notes binaires. Dans les cas où les mêmes notes sont répétées plusieurs fois, la log-vraisemblance peut être trivialement maximisée en répétant les entrées précédentes. Cela permet au MLM d'effectuer une opération de lissage, plutôt que d'imposer une quelconque structure musicale sur les sorties. Une solution potentielle serait d'effectuer une modélisation du langage aligné sur le rythme pour la formation et les données de test, plutôt que d'échantillonner le MIDI à une fréquence d'échantillonnage arbitraire. En outre, les RNN peuvent être étendus pour inclure des modèles de durée pour chacune de leurs sorties de hauteur, comme les HMM de second ordre. Cependant, il s'agit d'un problème difficile qui reste actuellement inexploré. Il serait également intéressant d'encourager les RNN à apprendre des modèles de notes temporels plus longs en interfacant les contrôleurs RNN avec des unités de mémoire externes [52] et également d'incorporer une notion de temps ou de mètre dans la représentation d'entrée pour les MLM.

L'effet de la tonalité sur les performances des MLM devrait être étudié plus en détail. Idéalement, les MLM devraient être invariants aux transpositions d'un morceau de musique à différentes hauteurs. La vérité de base MIDI peut être facilement transposée à n'importe quelle tonalité. Les MLM peuvent être formés sur des entrées avec des tonalités transposées ou des MLM individuels pour chaque tonalité peuvent être formés. En outre, la couche d'entrée entièrement connectée du RNN MLM peut être remplacée par une couche convolutive, avec des convolutions le long de l'axe de la hauteur de son pour encourager le réseau à être invariant aux transpositions de hauteur de son.

Une autre limitation du modèle hybride proposé est que la probabilité conditionnelle de l'équation 11 est dérivée en supposant que les prédictions au temps t sont seulement une fonction de l'entrée à t et indépendantes de toutes les autres entrées et sorties. La violation de cette hypothèse entraîne le double comptage de certains facteurs et réduit donc l'impact des MLM. Les résultats montrent clairement que les améliorations apportées par le MLM sont maximales lorsque le modèle acoustique est basé sur un cadre. Les améliorations sont comparativement plus faibles lorsqu'elles sont combinées avec les prévisions d'un modèle acoustique RNN ou ConvNet. Ceci est problématique car les modèles acoustiques ConvNet donnent les meilleures performances.

VI. BIBLIOGRAPHIE

[1] A. Klapuri and M. Davy, Signal processing methods for music transcription. Springer Science & Business Media, 2007.

[2] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, “Unsupervised transcription of piano music,” in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 1538–1546.

[3] E. Benetos and S. Dixon, “A shift-invariant latent variable model for automatic music transcription,” Computer Music Journal, vol. 36, no. 4, pp. 81–94, 2012.

[4] A. P. Klapuri, “Multiple fundamental frequency estimation based on harmonicity and spectral smoothness,” IEEE Transactions on Speech and Audio Processing, vol. 11, no. 6, pp. 804–816, 2003.

[5] V. Emiya, R. Badeau, and B. David, “Automatic transcription of piano music based on hmm tracking of jointly-estimated pitches,” in 16th European on Signal Processing Conference. IEEE, 2008, pp. 1–5.

[6] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. IEEE, 2003, pp. 177–180.

[7] S. A. Abdallah and M. D. Plumbley, “Polyphonic music transcription by non-negative sparse coding of power spectra,” in Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR), 2004, pp. 318–325.

[8] E. Vincent, N. Bertin, and R. Badeau, “Adaptive harmonic spectral decomposition for multiple pitch estimation,” IEEE Transactions on Audio, Speech, and Language Processing., vol. 18, no. 3, pp. 528–537, 2010.

[9] N. Bertin, R. Badeau, and E. Vincent, “Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription,” IEEE Transactions on Audio, Speech, and Language Processing., vol. 18, no. 3, pp. 538–549, 2010.

[10] P. Smaragdis, B. Raj, and M. Shashanka, “A probabilistic latent variable model for acoustic modeling.” Advances in models for acoustic processing, NIPS, vol. 148, 2006.

[11] G. C. Grindlay and D. P. Ellis, “A probabilistic subspace model for multi-instrument polyphonic transcription,” in Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR). International Society for Music Information Retrieval, 2010, pp. 21–26.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015. [13] G. E. Poliner and D. P. Ellis, “A discriminative model for polyphonic piano transcription,” EURASIP Journal on Applied Signal Processing, vol. 2007, no. 1, pp. 154–154, 2007.

[14] J. Nam, J. Ngiam, H. Lee, and M. Slaney, “A classification-based polyphonic piano transcription approach using learned feature representations.” in Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR), 2011, pp. 175–180.

[15] S. Bock and M. Schedl, “Polyphonic piano note transcription with recur- ¨ rent neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012, pp. 121–124.

[16] L. Rabiner and B.-H. Juang, “Fundamentals of speech recognition,” 1993.

[17] S. Raczynski, E. Vincent, and S. Sagayama, “Dynamic Bayesian networks for symbolic polyphonic pitch modeling,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 9, pp. 1830–1840, 2013.

[18] S. Sigtia, E. Benetos, S. Cherla, T. Weyde, A. S. d. Garcez, and S. Dixon, “An RNN-based music language model for improving automatic music transcription,” in Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), 2014.

[19] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Highdimensional sequence transduction,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2013, pp. 3178–3182.

[20] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d’Avila Garcez, and S. Dixon, “A Hybrid Recurrent Neural Network for Music Transcription,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brisbane, Australia, April 2015, pp. 2061–2065. 13

[21] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012, pp. 4277–4280.

[22] O. Abdel-Hamid, L. Deng, and D. Yu, “Exploring convolutional neural network structures and optimization techniques for speech recognition.” in INTERSPEECH, 2013, pp. 3366–3370.

[23] J. Schluter and S. B ¨ ock, “Improved musical onset detection with ¨ convolutional neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2014, pp. 6979–6983.

[24] E. J. Humphrey and J. P. Bello, “Rethinking automatic chord recognition with convolutional neural networks,” in Machine Learning and Applications (ICMLA), 2012 11th International Conference on, vol. 2. IEEE, 2012, pp. 357–362.

[25] E. Vincent and X. Rodet, “Music transcription with ISA and HMM,” in Independent Component Analysis and Blind Signal Separation. Springer, 2004, pp. 1197–1204.

[26] S. Sigtia, N. Boulanger-Lewandowski, and S. Dixon, “Audio chord recognition with a hybrid neural network,” in Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR), 2015.

[27] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, and Q. V. Le, “Large scale distributed deep networks,” in Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1223–1231.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Cognitive Modeling, vol. 5, p. 3, 1988.

[29] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller, “Efficient ¨ backprop,” in Neural networks: Tricks of the Trade. Springer, 2012, pp. 9–48.

[30] E. J. Humphrey, J. P. Bello, and Y. LeCun, “Feature learning and deep architectures: new directions for music informatics,” Journal of Intelligent Information Systems, vol. 41, no. 3, pp. 461–481, 2013.

[31] D. Eck and J. Schmidhuber, “Finding temporal structure in music: Blues improvisation with lstm recurrent networks,” in Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on. IEEE, 2002, pp. 747–756.

[32] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” Proceedings of the IEEE, vol. 78, no. 10, pp. 1550–1560, 1990.

[33] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Audio chord recognition with recurrent neural networks.” in Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR), 2013, pp. 335–340.

[34] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl, “Aggre- ´ gate features and adaboost for music classification,” Machine learning, vol. 65, no. 2-3, pp. 473–484, 2006.

[35] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription,” in Proceedings of the 29th International Conference on Machine Learning (ICML), 2012, pp. 1159–1166.

[36] H. Larochelle and I. Murray, “The neural autoregressive distribution estimator,” in International Conference on Artificial Intelligence and Statistics, 2011, pp. 29–37.

[37] R. M. Neal, “Connectionist learning of belief networks,” Artificial Intelligence, vol. 56, no. 1, pp. 71–113, 1992.

[38] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[39] A. Graves, “Sequence transduction with recurrent neural networks,” in Representation Learning Workshop, ICML, 2012.

[40] N. Boulanger-Lewandowski, J. Droppo, M. Seltzer, and D. Yu, “Phone sequence modeling with recurrent neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 5417–5421.

[41] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in Advances in Neural Information Processing Systems, 2015, pp. 1171–1179.

[42] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms. MIT press Cambridge, 2001, vol. 2.

[43] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 6, pp. 1643–1654, 2010.

[44] K. O’Hanlon and M. D. Plumbley, “Polyphonic piano transcription using non-negative matrix factorisation with group sparsity,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 3112–3116.

[45] M. Bay, A. F. Ehmann, and J. S. Downie, “Evaluation of multiple-F0 estimation and tracking systems.” in Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR), 2009, pp. 315–320.

[46] J. C. Brown, “Calculation of a constant q spectral transform,” The Journal of the Acoustical Society of America, vol. 89, no. 1, pp. 425– 434, 1991.

[47] E. Benetos, “Automatic transcription of polyphonic music exploiting temporal evolution,” Ph.D. dissertation, Queen Mary University of London, Dec. 2012.

[48] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[50] M. D. Zeiler, “Adadelta: An adaptive learning rate method,” arXiv preprint arXiv:1212.5701, 2012.

[51] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances in optimizing recurrent networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2013, pp. 8624–8628.

[52] E. Grefenstette, K. M. Hermann, M. Suleyman, and P. Blunsom, “Learning to transduce with unbounded memory,” arXiv preprint arXiv:1506.02516, 2015.