دانشگاه شهید بهشتی
دانشکده مهندسی و علوم کامپیوتر

تشخیص آکوردهای موسیقی با استفاده از روش‌های مبتنی بر یادگیری عمیق

پروژه کارشناسی مهندسی کامپیوتر

دانشجو:
روژین انصاری داخل

استاد راهنما:
دکتر یاسر شکفته

شهریور ۱۴۰۴

چکیده

تشخیص خودکار آکورد در موسیقی موضوعی است که اخیراً مورد توجه بسیاری از پژوهشگران در زمینه پردازش موسیقی قرار گرفته است؛ چرا که این مبحث، یکی از موضوعات پایه در موسیقی است و اساس تشکیل هارمونی و شکل‌گیری موسیقی می‌باشد. بنابراین، پیش‌رفت در این زمینه نه تنها در خلق موسیقی به صورت خودکار، بلکه در زمینه‌های مختلف، مانند نت نویسی خودکار و تحلیل و پردازش ساختار موسیقی کاربرد دارد.

در سال‌های اخیر، پیشرفت‌های قابل توجهی در این حوزه با بهره‌گیری از یادگیری عمیق حاصل شده است. روش‌های اولیه مبتنی بر استخراج دستی ویژگی‌ها مانند کرومای کلاسیک و مدل‌های آماری نظیر HMM اگرچه آغازگر این مسیر بودند، اما در دقت و تعمیم‌پذیری محدودیت‌های جدی داشتند. پس از آن، شبکه‌های کانولوشنی (CNN) توانستند با یادگیری ویژگی‌ها به‌صورت مستقیم از داده‌های صوتی، وابستگی به دانش تخصصی افراد خبره در استخراج ویژگی را کاهش دهند. ترکیب CNN با مدل‌های بازگشتی (RNN/LSTM) نیز به بهبود درک وابستگی‌های زمانی انجامید و روش‌هایی چون Hybrid RNN و CNN+CRF دقت بالاتری در شناسایی پیوستگی آکوردها ارائه کردند. با این حال، این روش‌ها هنوز در تحلیل وابستگی‌های زمانی میان آکوردها ضعیف هستند، و به همین دلیل دقت آن‌ها در تشخیص آکورد پایین می‌باشد. بنابراین، پیاده‌سازی مدلی که بتواند با درنظر گرفتن وابستگی‌های کوتاه مدت و بلند مدت آکوردها به هم، پیش‌بینی دقیق‌تری داشته باشد، کمک شایانی به پیشرفت در موضوعات مرتبط به موسیقی و یادگیری عمیق می‌کند.

در این پژوهش، مدلی مبتنی بر معماری Bidirectional Transformer (BTC) ارائه شده است که با استفاده همزمان از اطلاعات گذشته و آینده، وابستگی‌های کوتاه‌مدت و بلندمدت میان آکوردها را در نظر می‌گیرد. در نتیجه، پس از پیاده‌سازی و ارزیابی مدل، مشاهده می‌شود که مدل دقت بیش‌تری نسبت به مدل‌های پیش از خود دارد.

واژگان کلیدی: تشخیص آکورد موسیقی، یادگیری عمیق، Bidirectional Transformer، پردازش سیگنال صوتی، Attention Mechanism.

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فهرست کلمات اختصاری

Abbreviation	Full Form	Page Number
HMM	Hidden Markov Model	1
CNN	Convolutional Neural Network	1
CRF	Conditional Random Field	1
RNN	Recurrent Neural Network	1
BTC	Bidirectional Transformer for Chord recognition	1
LSTM	Long Short-Term Memory	1
MIR	Music Information Retrieval	2
CQT	Constant-Q Transform	3
WCSR	Weighted Chord Symbol Recall	4
NLP	Natural Language Processing	11
ETD	Extended Training Data	12
DNN	Deep Neural Network	14
MIREX	Music Information Retrieval Evaluation eXchange	15
ReLU	Rectified Linear Unit	15
OR	Overlap Ratio	17

WAOR	Weighted Average Overlap Ratio	20
BLSTM	Bidirectional Long Short-Term Memory	21

فصل اول: کلیّات

۱-۱ مقدمه

مسئله تشخیص آکورد، یکی از مباحث پایه در موسیقی می‌باشد، که در روش‌های سنتی، به صورت دستی و توسط افراد خبره انجام می‌شود. این روش‌ها زمان‌بر هستند و امکان خطای انسانی در تشخیص آن‌ها وجود دارد. بر همین اساس، در دو دهه‌ی اخیر پژوهش‌های گسترده‌ای در زمینه‌ی خودکارسازی این فرآیند انجام شده است. با این حال، بسیاری از روش‌های اولیه که مبتنی بر ویژگی‌های دستی و مدل‌های آماری مانند HMM بودند، تنها به دقتی در حدود ۶۵ تا ۷۲ درصد دست یافتند. با بهبودهای بعدی، دقت به حدود ۷۴ درصد رسید که همچنان با نیازهای واقعی فاصله داشت. با ورود یادگیری عمیق، شبکه‌های کانولوشنی (CNN) توانستند این مقدار را به حدود ۷۷–۷۸ درصد ارتقا دهند و استفاده از معماری‌های ترکیبی (مانند CNN+CRF) باعث افزایش دقت تا حدود ۸۰ درصد شد. اما این روش‌ها همچنان دارای دقت کافی نیستند و برخی از ویژگی‌های آکوردها مانند وابستگی زمانی آن‌ها را به اندازه کافی مدل نمی‌کنند. به همین دلیل، تلاش در ارائه مدلی که دقیق‌تر باشد، در زمینه‌های مختلفی از تحلیل موسیقی و خلق آن به صورت خودکار کمک کننده است.

در این پژوهش، سعی می‌شود مدلی پیاده‌سازی شود، که با در نظر گرفتن وابستگی زمانی بین آکوردها و مدل کردن آن، عملکرد بهتری را نسبت به مدل‌های پیش از خود ارائه دهد.

۱-۲ بیان مسئله

تشخیص آکورد یکی از موضوعات بنیادین در تحلیل موسیقی است که بسیاری از پژوهش‌ها و کاربردهای عملی به آن وابسته‌اند. این موضوع، افرادی مانند موسیقی‌دانان، پژوهشگران حوزه‌ی Music Information Retrieval (MIR)، توسعه‌دهندگان سیستم‌های توصیه‌گر موسیقی، نرم‌افزارهای آموزشی و سامانه‌های تولید موسیقی خودکار را شامل می‌شود. در روش‌های سنتی، این وظیفه به‌صورت دستی و توسط افراد خبره انجام می‌شد که فرآیندی زمان‌بر و پرخطا بود و امکان استفاده‌ی گسترده در مقیاس بزرگ را نداشت.

در سال‌های اخیر و با رشد پلتفرم‌های موسیقی دیجیتال، حجم داده‌های موسیقایی به‌شدت افزایش یافته و نیاز به سیستم‌های خودکار و دقیق برای پردازش این داده‌ها بیشتر احساس می‌شود. با وجود پیشرفت‌های قابل‌توجه در حوزه‌ی یادگیری ماشین و یادگیری عمیق، مسئله‌ی دقت پایین در شناسایی آکوردهای پیچیده (مانند هفتم‌ها، وارونگی‌ها و آکوردهای

توسعه‌یافته) همچنان باقی است. همچنین، مدل‌سازی وابستگی زمانی بین آکوردها در بسیاری از روش‌ها به‌صورت کامل و دقیق انجام نشده و این موضوع منجر به پرش‌های ناگهانی یا پیش‌بینی‌های ناپایدار در خروجی می‌شود.

راه‌حل‌های موجود مانند مدل‌های مبتنی بر ویژگی‌های دستی و HMMها، یا حتی مدل‌های عمیق‌تر مانند CNN و Hybrid RNN، اگرچه توانسته‌اند دقت کلی را به حدود ۷۷ تا ۸۳ درصد در حالت آکوردهای ساده افزایش دهند، اما در مواجهه با واژگان بزرگ‌تر یا آکوردهای کم‌تکرار ضعف جدی دارند. حتی در پیشرفته‌ترین روش‌ها، مانند مدل‌های CNN+CRF یا Structured RNN، دقت در سطح آکوردهای هفتم از ۷۳ درصد فراتر نرفته است. این آمار نشان می‌دهد که هنوز فاصله‌ی قابل‌توجهی تا یک سامانه‌ی پایدار و کاربردی وجود دارد.

از این رو، چالش اصلی در این زمینه، طراحی مدلی است که بتواند هم‌زمان وابستگی‌های کوتاه‌مدت و بلندمدت میان آکوردها را مدل کند، واژگان گسترده‌تری از آکوردها را پشتیبانی کند و در عین حال پایداری بیشتری در پیش‌بینی مرزهای زمانی آکوردها داشته باشد. در این پروژه، معماری پیشنهادی بر پایه‌ی Bidirectional Transformer طراحی شده است که با استفاده از سازوکار توجه (Self-Attention) امکان درک بهتر ساختارهای هارمونی در موسیقی را فراهم می‌کند. به این ترتیب، رویکرد ارائه شده می‌تواند شکاف موجود در پژوهش‌های پیشین را پوشش داده و دقت و تعمیم‌پذیری بالاتری در مقایسه با روش‌های قبل ارائه دهد.

۳-۱ کلیات روش پیشنهادی

هدف پروژه

هدف اصلی این پروژه، طراحی و پیاده‌سازی یک مدل یادگیری عمیق برای تشخیص خودکار آکورد موسیقی است که بتواند با در نظر گرفتن وابستگی‌های کوتاه‌مدت و بلندمدت میان آکوردها، دقت بالاتری نسبت به مدل‌های موجود ارائه دهد. این سیستم باید علاوه بر شناسایی آکوردهای پایه (ماژور و مینور)، توانایی تشخیص آکوردهای پیچیده‌تر (مانند هفتم و وارونگی‌ها) را نیز داشته باشد.

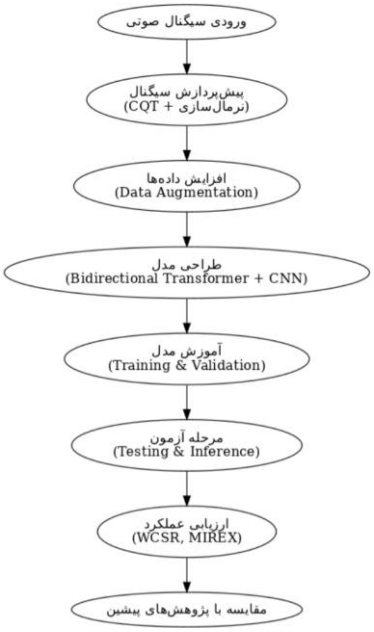
روش پیشنهادی

برای رسیدن به این هدف، رویکردی مبتنی بر Bidirectional Transformer (BTC) پیشنهاد می‌شود. در این روش، سیگنال صوتی ابتدا به نمایش زمان–فرکانس مناسب (CQT) تبدیل شده و سپس پس از افزایش داده‌ها (Data Augmentation)، به مدل داده می‌شود. مدل پیشنهادی با بهره‌گیری از لایه‌های توجه دوطرفه (Bi-directional Self-Attention)، قادر است وابستگی‌های زمانی آکوردها را هم از گذشته و هم از آینده به‌طور هم‌زمان در نظر بگیرد. ترکیب این قابلیت با بلوک‌های کانولوشنی محلی باعث می‌شود مدل علاوه بر درک ساختار هارمونیک کلی، جزئیات محلی و مرزبندی آکوردها را نیز با دقت بالاتری تشخیص دهد.

مراحل کلی روش پیشنهادی

۱. جمع‌آوری و آماده‌سازی داده‌ها: استفاده از دیتاست‌های استاندارد و نگاشت برچسب‌ها به کلاس‌های هدف.
۲. پیش‌پردازش سیگنال: محاسبه‌ی CQT از سیگنال صوتی و انجام نرمال‌سازی.
۳. افزایش داده‌ها (Data Augmentation): شیفت دادن فرکانس داده‌ها برای افزایش تنوع داده و بهبود تعمیم‌پذیری مدل.
۴. طراحی معماری مدل: ترکیب بلوک‌های کانولوشنی و لایه‌های توجه دوطرفه در ساختار BTC.
۵. آموزش مدل: بهینه‌سازی پارامترها با استفاده از مجموعه‌های آموزشی و اعتبارسنجی، و جلوگیری از بیش‌برازش با تکنیک‌هایی مانند Dropout.
۶. مرحله استنتاج و آزمایش: اجرای مدل روی مجموعه‌ی آزمون و ارزیابی عملکرد با معیارهایی مانند WCSR.
۷. مقایسه با روش‌های پیشین: تحلیل نتایج و مقایسه‌ی دقت مدل با مدل‌های مرجع در پژوهش‌های قبلی.

شکل (۱-۱)، نقشه این مراحل را نشان می‌دهد:



شکل ۱ مراحل کلی روش پیشنهادی

۱-٤ ساختار پروژه

در فصل‌های بعد، ابتدا برخی از مفاهیم اولیه که دانستن آن‌ها برای درک بهتر کارکرد مدل ضروریست را بیان کرده، و بعد به بررسی مقالات و پژوهش‌هایی که پیش از این در این زمینه شکل گرفته‌اند می‌پردازیم؛ سپس، ساختار و نحوه پیاده‌سازی مدل را شرح می‌دهیم و در نهایت، پس از توضیح معیارهای ارزیابی استفاده شده و مجموعه‌دادگان، نتایج بدست آمده از ارزیابی مدل را بیان کرده و به مقایسه آن با پژوهش‌های پیشین می‌پردازیم.

فصل دوم: مفاهیم پایه

۲-۱ مقدمه

در این فصل برخی از مفاهیم پایه و اولیه، که دانستن آن‌ها برای درک بهتر عملکرد روش ارائه شده ضروریست، بیان می‌شود. این بخش شامل مفاهیم پایه موسیقی، مفاهیم مربوط به بخش یادگیری عمیق و پردازش سیگنال و ارتباط میان آن‌ها می‌باشد.

بدیهیست که برخی از موضوعات ارائه شده در این قسمت، به ویژه در موارد مربوط به موسیقی، نسبت به توضیحات موجود در این مقاله بسیار پیچیره‌تر است و بیان کامل آن ضرورتی ندارد. به همین دلیل، این موارد به صورت مختصر و در حد نیاز برای درک روش ارائه شده توضیح داده شده‌اند.

۲-۲ مفاهیم پایه موسیقی

۲-۲-۱ زیر و بمی صدا

Pitch یا همان زیر و بمی صدا، ویژگی ادراکی صدا است که اساس آن فرکانس پایه (Fundamental Frequency) می‌باشد. به بیان ساده، هرچه فرکانس یک صدا بیشتر باشد، زیرتر شنیده می‌شود و هرچه کمتر باشد، بم‌تر درک می‌شود.

در موسیقی هر نت مانند C (نت دو) نماینده یک بازه فرکانسی می‌باشد، که این فرکانس همان pitch مربوط به آن نت است؛ برای مثال، نت A4 (نت «لا» در اکتاو چهارم) دارای فرکانس ۴۴۰ هرتز می‌باشد. علائم نگارشی نت‌ها علاوه بر فرکانس، اطلاعات دیگری همچون مدت زمان کشش آن نت را نمایش می‌دهند. اما از آنجایی که در این پژوهش ورودی مدل فایل‌های صوتی است و مدل با فرکانس صدا آموزش می‌بیند، مفهوم pitch کارامدتر است.

۲-۲-۲ اکتاو

اکتاو (Octave) در موسیقی به فاصله‌ی بین دو Pitch گفته می‌شود که یکی دارای دو برابر فرکانس دیگری است. به بیان ساده، اگر یک نت فرکانس f داشته باشد، نت متناظر آن در یک اکتاو بالاتر فرکانسی برابر 2f خواهد داشت و در یک اکتاو پایین‌تر فرکانس آن f/2 خواهد بود.

از نظر شنیداری، دو نت با فاصله‌ی یک اکتاو، بسیار شبیه به هم درک می‌شوند. به همین دلیل است که در بازه فرکانسی نت‌ها، پس از دوبرابر شدن فرکانس آن نت، نام آن نت یکسان می‌ماند، اما pitch آن تغییر می کند و زیرتر می‌شود. یعنی A4 (440Hz) و A5 (880Hz) هردو نت «لا» هستند ولی با اختلاف یک اکتاو (2f).

در سیستم کوک مساوی (Equal Temperament) که متداول‌ترین سیستم کوک در موسیقی غربی است، هر اکتاو به ۱۲ نیم‌پرده تقسیم می‌شود. به این ترتیب، نسبت فرکانسی بین هر نیم‌پرده تقریبا برابر است با 1.05946.

کمی بعد تر نشان داده می‌شود که در این پژوهش برای پیش‌پردازش فایل‌های صوتی و تبدیل آن‌ها به vector هایی از فرکانس، اعداد انتخاب شده در الگوریتم‌ها به این تقسیم ۱۲تایی مرتبط هستند.

۳-۲-۲ آکورد

آکورد به ترکیب هم‌زمان سه یا چند Pitch گفته می‌شود که با هم نواخته یا شنیده می‌شوند و هسته‌ی اصلی هارمونی موسیقی را تشکیل می‌دهند. آکوردها بر اساس تعداد نت‌ها و فاصله‌ی بین آن‌ها دسته‌بندی می‌شوند. ابتدایی‌ترین آن‌ها تریادها (Triads) هستند که شامل سه نت می‌شوند. هر آکورد سه صدایی (Triad) دارای این اجزا می‌باشد که تشخیص آن به تشخیص نوع آکورد منجر می‌شود:

۱. نت ریشه (Root): نت پایه آکورد که نام آکورد از آن گرفته می‌شود.
۲. نت سوم (Third): این نت، تعیین کننده کیفیت یا همان نوع آکورد می‌باشد. یعنی فاصله‌ی این نت نسبت به ریشه مشخص می‌کند آکورد ماژور است یا مینور.
۳. نت پنجم (Fifth): این نت تقویت‌کننده‌ی ساختار آکورد است؛ و ثبات و انسجام هارمونیک ایجاد می‌کند.

انواع اصلی تریادها بر اساس فاصله‌ها:

۱. ماژور (Major Triad): در این تریاد نت سوم با نت ریشه فاصله سوم بزرگ (یعنی به اندازه ۴ نیم‌پرده) و نت پنجم فاصله پنجم درست (۷ نیم‌پرده) را داراست.
۲. مینور (Minor Triad): در این آکورد نت سوم به اندازه ۳ نیم‌پرده و نت پنجم به اندازه ۷ نیم پرده با ریشه فاصله دارد.

۳. افزوده (Augmented Triad): در این آکورد نت سوم به اندازه ۴ نیم‌پرده و نت پنجم به اندازه ۸ نیم‌پرده با ریشه فاصله دارد.

۴. کاسته (Diminished Triad): در این آکورد نت سوم ۳ نیم‌پرده و نت پنجم ۶ نیم‌پرده با ریشه فاصله دارد.

برای مثال، در فایل‌های annotation در مجموعه داده استفاده شده، C:maj به معنای آکورد دو ماژور (C Major) است، که در آن نت ریشه همان نت C، و نت سوم و پنجم به ترتیب E و G می‌باشند؛ که در آن‌ها فواصل گفته شده دیده می‌شود.

در موسیقی همچنین آکوردهای چهار صدایی (Tetrads) هم استفاده می‌شوند که با افزودن فاصله‌ی هفتم نسبت به ریشه ساخته می‌شوند. این آکوردها شامل هفتم ماژور (Major 7th)، هفتم نمایان (Dominant 7th)، و هفتم مینور (Minor 7th) هستند. انواع دیگری از آکوردها نیز وجود دارد که در سبک‌های مختلف استفاده می‌شوند و هر کدام بر اساس قوانین خاصی بر اساس فاصله نت‌ها تشکیل می‌شوند. برای سیستم‌های تشخیص آکورد خودکار، شناسایی این فواصل کلیدی است؛ زیرا الگوریتم باید بتواند هم نت ریشه و هم کیفیت آکورد (بر اساس فاصله‌ها) را تشخیص دهد. مدل معرفی شده در این پژوهش قابلیت یادگیری و استخراج این ویژگی‌ها را دارد. با این حال، در این مدل و بسیاری از مدل‌های مشابه، دسته بندی کلاس‌ها به ۲۵ کلاس ماژور، مینور و بدون آکورد خلاصه شده است؛ زیرا بیشتر موسیقی‌های غربی متکی بر آکوردهای ماژور و مینور هستند و این دسته‌بندی مانع گرایش و سوگیری مدل به سمت آکوردهای پرتکرار می‌شود.

۴-۲-۲ هارمونی

در موسیقی هارمونی به مطالعه و کاربرد توالی آکوردها و روابط میان آن‌ها می‌پردازد. به زبان ساده، توالی آکوردها پشت سر هم به شیوه‌ای که از قوانین ساختار موسیقی پیروی کند، هارمونی نام دارد. از دیدگاه پردازش سیگنال و یادگیری ماشین، هارمونی به‌معنای الگوهای زمانی—فرکانسی است که بیانگر تغییرات پی‌درپی آکوردها هستند. ویژگی‌ها و قوانین هارمونی در موسیقی باعث می‌شوند که نیاز به توجه به آکوردهای آینده و گذشته در ورودی برای تشخیص آکورد مورد نظر بوجود بیاید و به همین دلیل است که الگوریتم‌هایی لازم هستند که بتوانند رابطه زمانی میان آکوردها را به خوبی مدل کنند.

۲-۳ مفاهیم پایه یادگیری عمیق و پردازش سیگنال

۲-۳-۱ تبدیل Q ثابت (Constant-Q Transform یا CQT) [1]

تبدیل Q ثابت یک تبدیل زمان–فرکانس است که مانند تبدیل فوریه سیگنال را به دامنه فرکانس می‌برد، با این تفاوت که محور فرکانس در آن به صورت لگاریتمی (هم‌خوان با فواصل موسیقایی) نمونه‌برداری می‌شود .به بیان ساده، CQT از یک بانک فیلتر با فرکانس‌های مرکزی هندسی فاصله‌گذاری‌شده استفاده می‌کند (مثلاً هر اکتاو به تعداد ثابتی باند تقسیم می‌شود). منظور از Q ثابت این است که نسبت فرکانس مرکزی به عرض‌باند هر فیلتر (عامل Q) در تمام باندهای فرکانسی یکسان است. در نتیجه فاصله فرکانسی بین هر دو فیلتر متوالی تابعی ثابت از فرکانس خود آنهاست. برای مثال اگر B=12 انتخاب شود (۱۲ باند در هر اکتاو)، فرکانس‌های مرکزی فیلترها دقیقاً منطبق بر نت‌های موسیقی با فاصله نیم‌پرده خواهند بود (هر ۱۲ باند یک اکتاو را تشکیل می‌دهند). با تنظیم مناسب پایین‌ترین فرکانس f0 (مثلاً نت پایه)، می‌توان کاری کرد که باند k-ام تبدیل مستقیماً متناظر با نت موسیقایی شماره k باشد. ویژگی مهم دیگر CQT این است که قدرت تفکیک زمانی-فرکانسی آن وابسته به فرکانس است: هرچه فرکانس یک باند بالاتر باشد، پنجره زمانی کوتاه‌تر و تفکیک زمانی بهتر است؛ برعکس، برای فرکانس‌های پایین پنجره تحلیل بلندتر و تفکیک فرکانسی بالاتر خواهد بود. این رفتار شبیه به سیستم شنوایی انسان و نیز منطبق بر روند معمول موسیقی (دقت بیشتر در فرکانس‌های پایین و تحرک بیشتر در فرکانس‌های بالا) است. در CQT محور فرکانس به صورت خطی در مقیاس موسیقایی است (فواصل مساوی نمایانگر نسبت‌های فرکانسی یکسان، مثلاً هر واحد معادل یک نیم‌پرده)، در نتیجه الگوهای صوتی مستقل از کوک و گام موسیقی راحت‌تر شناسایی می‌شوند. برای نمونه، یک شبکه عصبی کانولوشنی (CNN) می‌تواند روی طیف‌نگار CQT آموزش ببیند و به دلیل یکنواختی محور زیر و بمی، الگوهای آکورد یا ملودی را حتی در صورت انتقال گام (Transpose) تشخیص دهد.

۲-۳-۲ Data augmentation [2] [3]

تقویت داده یا همان Data Augmentation ، تکنیکی است که در بسیاری از پژوهش‌ها و پروژه‌های مرتبط با یادگیری عمیق استفاده می‌شود تا با استفاده از آن، بتوان مجموعه داده را گسترش داد و باعث می‌شود تنوع داده‌ها نیز افزایش پیدا کند. در این تکنیک، با روش‌های متفاوت، می‌توان از مجموعه داده موجود، داده‌های جدیدی را تولید کرد. برخی از این روش‌ها شامل استفاده از رابطه میان داده‌ها یا قوانین و اطلاعات پایه در زمینه مورد بحث پژوهش می‌باشد. در این پژوهش، قبل از مرحله CQT، داده صوتی خام که ورودی مرحله پیش‌پردازش می‌باشد، کمی به بالا یا پایین شیفت داده می‌شود. در

نتیجه این شیفت، داده با فرکانس‌های جدید تولید شده که می‌تواند در آموزش مدل استفاده شود و باعث افزایش دقت مدل شود. این داده‌های افزایش یافته، فقط در مرحله آموزش استفاده می‌شوند و در مرحله آزمون، فقط داده‌های اصلی مورد استفاده قرار می‌گیرند.

۳-۳-۲ [4] Transformers

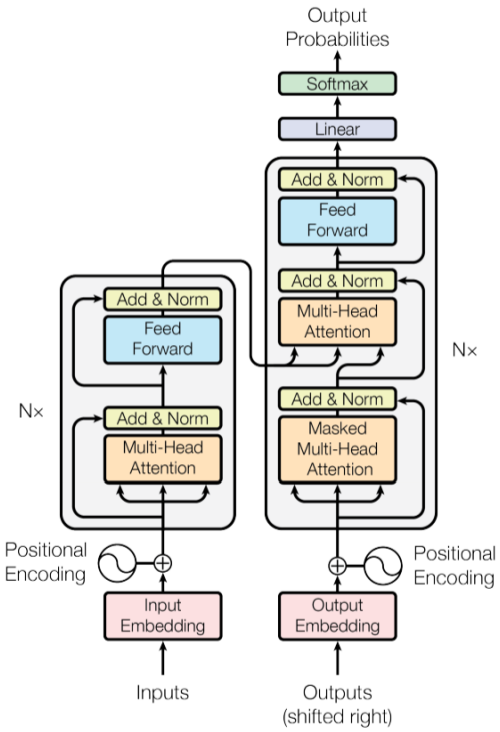
مدل Transformer یک معماری نوین در حوزه یادگیری عمیق است که برای پردازش داده‌های ترتیبی (Sequence) معرفی شده است. این مدل در سال ۲۰۱۷ توسط محققان گوگل و با انتشار مقاله‌ی "Attention Is All You Need" معرفی گردید. تفاوت اساسی این مدل با مدل‌های قبلی (مانند RNN) در این است که ترنسفورمر برای مدل‌سازی وابستگی‌های دور و نزدیک در دنباله‌ها، تنها از مکانیزم توجه (Attention) استفاده می‌کند و وابستگی به شبکه‌های بازگشتی یا CNN را به طور کامل حذف کرده است. این رویکرد موجب شده که مدل به صورت موازی قابل آموزش باشد و در مقایسه با مدل‌های ترتیبی قدیمی، سرعت آموزش بسیار بالاتری داشته باشد. مدل ترنسفورمر علاوه بر داشتن کارایی بالا، دقت بی‌سابقه‌ای را نیز در مسائل مختلف نشان داده و مبنای توسعه‌ی مدل‌های بزرگی در حوزه‌های گوناگون قرار گرفته است.

برخی از مثال‌های این بخش در مورد پردازش زبان است تا درک کارکرد مدل ملموس‌تر باشد. در ادامه، کاربرد این مدل در موارد مربوط به موسیقی و عملکرد آن در مدل ارائه شده در این پژوهش شرح داده می‌شود.

معماری کلی مدل Transformer

شکل (۲-۱) نمایی کلی از معماری مدل Transformer را نمایش می‌دهد. معماری مدل Transformer از دو بخش encoder (رمزگذار) و decoder (رمزگشا) تشکیل شده است. هر دو بخش شامل چند لایه‌ی تکرارشونده هستند. در رمزگذار، هر لایه دارای یک مکانیزم Multi-Head Self-Attention و یک شبکه‌ی Feed-Forward نقطه‌ای است. در رمزگشا نیز همین ساختار تکرار می‌شود، با این تفاوت که علاوه بر این دو زیرلایه، یک زیرلایه‌ی توجه Encoder-Decoder نیز اضافه می‌شود تا رمزگشا بتواند از اطلاعات رمزگذار استفاده کند. برای جلوگیری از نگاه به آینده در حین تولید خروجی، در بخش Self-Attention رمزگشا از ماسک‌گذاری استفاده می‌شود. در زیرلایه‌ی توجه Encoder-Decoder، بردارهای Query از لایه‌ی قبلی رمزگشا گرفته می‌شوند و بردارهای Key و Value از خروجی رمزگذار دریافت می‌شوند تا رمزگشا

بتواند به بخش‌های مرتبط ورودی توجه کند. همچنین، در تمامی زیرلایه‌ها، Residual Connections و نرمال‌سازی لایه‌ای به کار گرفته می‌شود تا پایداری گرادیان‌ها و یادگیری بهتر مدل تضمین گردد.



شکل ۲ - معماری کلی ترنسفورمر [4]

مفهوم Attention: [5] [6] [7]

ایده‌ی اصلی مکانیزم Attention این است که زمانی که مدل می‌خواهد یک خروجی تولید کند (مثلاً یک کلمه در ترجمه ماشینی)، لازم نیست به همه‌ی بخش‌های ورودی به یک اندازه نگاه کند، بلکه می‌تواند روی بخش‌های مهم‌تر و مرتبط‌تر تمرکز کند. برای این منظور، این مکانیزم برای هر بخش از ورودی یک وزن اهمیت محاسبه می‌کند و سپس ترکیبی وزن‌دار از ورودی‌ها می‌سازد. در این روش برای هر ورودی، سه بردار تعریف می‌شود. بردار Query، که بیانگر این است که دنبال چه اطلاعاتی هستیم، بردار Key که نشان می‌دهد هر ورودی چه اطلاعاتی دارد، و بردار Value که همان محتوای اصلی ورودی است. وزن توجه با مقایسه Query و Keyها (با ضرب داخلی) به دست می‌آید. سپس این وزن‌ها روی Valueها اعمال می‌شوند و ترکیب نهایی همان چیزی است که مدل استفاده می‌کند. خروجی Attention در واقع برداری غنی‌شده است که نشان می‌دهد برای پیش‌بینی فعلی، کدام قسمت‌های ورودی اهمیت بیشتری داشته‌اند.

:Self-Attention

در حالت کلی، Attention بین دو توالی مختلف به کار می‌رود. یعنی Queryها از یک دنباله (مثلاً جمله‌ی مقصد در ترجمه) می‌آیند، ولی Key و Valueها از دنباله‌ی دیگری (مثلاً جمله‌ی ورودی مبدأ) گرفته می‌شوند. اما در مکانیزم self-attention که در این معماری استفاده شده، همه این مقادیر از یک توالی مشترک گرفته می‌شوند. یعنی مدل به جای اینکه به ورودی دیگری نگاه کند، روی خود دنباله‌ی فعلی تمرکز می‌کند، و هر کلمه در جمله می‌تواند به همه‌ی کلمات دیگر جمله (و خودش) توجه کند. برای مثال در جمله‌ی انگلیسی "The animal didn't cross the street because it was too tired" ، برای فهمیدن منظور it، مکانیزم Self-Attention می‌تواند تشخیص دهد که باید روی کلمه‌ی animal تمرکز بیشتری کند.

:Multi-Head Attention

در مکانیزم Attention معمولی یا Self-Attention، مدل فقط یک بار بین Query و Keyها امتیاز توجه محاسبه می‌کند و یک ترکیب خروجی به دست می‌آید. اما گاهی یک بار انجام دادن این فرایند کافی نیست؛ چون ممکن است روابط مختلفی بین کلمات وجود داشته باشد (مثلاً رابطه‌ی دستوری، معنایی، یا وابستگی بلندمدت). برای همین، در Multi-Head Attention به جای یک بار، چندین بار Attention به صورت موازی انجام می‌شود.

بدین منظور، ابتدا Query، Key و Value به چند فضای برداری کوچکتر نگاشت می‌شوند. سپس روی هر کدام یک Attention جداگانه (که به آن Head می‌گویند) محاسبه می‌شود، و هر Head یک خروجی مستقل به دست می‌آورد. در نهایت خروجی همه‌ی Headها به هم الحاق شده و دوباره با یک لایه‌ی خطی ترکیب می‌شود تا خروجی نهایی ساخته شود. در نتیجه هر Head می‌تواند روی نوع متفاوتی از روابط تمرکز کند و مدل نمایشی غنی‌تر و متنوع‌تر از داده به دست می‌آورد. از آنجایی که بعد از تقسیم، هر Head در بعد کوچکتری کار می‌کند، هزینه‌ی محاسباتی تقریباً مثل یک Attention معمولی باقی می‌ماند.

:Encoder

رمزگذار (Encoder) مجموعه‌ای از لایه‌هاست که ورودی (مثلاً جمله‌ی مبدأ در ترجمه ماشینی) را به یک نمایش میانی برداری تبدیل می‌کنند. هر لایه‌ی رمزگذار شامل Multi-Head Self-Attention و شبکه‌ی Feed-Forward است، که عبارت‌های ورودی را نسبت به لایه‌ی قبلی غنی‌تر می‌کند. در شروع Encoder، ابتدا کلمات ورودی به بردارهای Embedding تبدیل می‌شوند. سپس این بردارها با بردارهای مکان (Positional Encoding) جمع می‌شوند تا موقعیت ترتیبی کلمات نیز در نمایش آن‌ها لحاظ شود. خروجی نهایی Encoder یک دنباله برداری به طول ورودی است که نمایش غنی‌شده‌ی معنایی-دستوری ورودی می‌باشد.

:Decoder

رمزگشا (Decoder) نیز پشته‌ای از لایه‌هاست که وظیفه‌ی تولید خروجی (مثلاً جمله‌ی ترجمه‌شده) را بر عهده دارند. هر لایه‌ی Decoder نیز شامل سه زیرلایه است: ابتدا یک زیرلایه Masked Multi-Head Self-Attention که به بخش تولیدشده‌ی فعلی خروجی توجه می‌کند (و با ماسک کردن مانع نگاه به آینده می‌شود)؛ سپس یک زیرلایه Attention Encoder-Decoder که به خروجی کامل Encoder توجه می‌کند و اطلاعات ورودی را وارد فرایند Decode می‌کند، و در نهایت یک زیرلایه Feed-Forward. در Decoder نیز مانند Encoder پس از تعبیه‌ی کلمات خروجی ، این بردارهای تعبیه با positional encoding جمع می‌شوند و به عنوان ورودی اولیه‌ی Decoder استفاده می‌گردند. به این ترتیب، هر لایه‌ی Decoder با استفاده از اطلاعاتی که از خروجی Encoder و وضعیت فعلی خروجی دارد، کلمه‌ی بعدی را پیش‌بینی می‌کند.

:Positional Encoding

مدل Transformer برخلاف RNNها و CNNها، هیچ مکانیزم ذاتی برای درک ترتیب ورودی ندارد زیرا هیچ حلقه یا کانولوشنی در ساختارش به کار نرفته است. به همین دلیل، لازم است اطلاعات موقعیت کلمات در توالی به نحوی به مدل اضافه شود. برای این منظور، از کدگذاری موقعیت بر پایه‌ی توابع سینوسی و کسینوسی استفاده شده است. بدین منظور، برای هر موقعیت مکانی یک بردار ثابت ایجاد می‌شود که مقادیر آن با توجه به توابع تناوبی با فرکانس‌های مختلف تنظیم شده

است. این بردارهای موقعیتی در ابتدای شبکه به بردار کلمه‌ی متناظر افزوده می‌شوند. به این ترتیب مدل می‌تواند براساس الگوهای سینوسی، فاصله و ترتیب نسبی کلمات را تشخیص دهد (زیرا اختلاف بردارهای موقعیتی دو کلمه اطلاعات فاصله‌ی بین آن‌ها را در بر دارد).

[8] :Residual Connections and Normalization

یکی از عوامل کلیدی موفقیت مدل Transformer استفاده از Residual Connections و نرمال‌سازی لایه‌ای در هر لایه است. این فرایند بدین شکل است که خروجی هر زیرلایه با ورودی آن جمع زده می‌شود (عملیات skip connection یا add) و سپس نتیجه به یک لایه Layer Normalization اعمال می‌گردد. این ساختار دو مزیت مهم دارد: نخست، جریان گرادیان در شبکه‌های عمیق را بهبود می‌دهد و از محو شدن یا انفجار گرادیان‌ها جلوگیری می‌کند؛ دوم، با اضافه کردن مستقیم ورودی به خروجی هر لایه، شبکه می‌تواند در صورت نیاز آن ورودی خام را نیز حفظ کند و یادگیری لایه‌های بالاتر راحت‌تر صورت می‌گیرد. Layer Normalization نیز با نرمال کردن خروجی هر لایه (با میانگین صفر و واریانس واحد) به تسریع همگرایی و پایداری آموزش کمک می‌کند. در Transformer پس از هر زیرلایه (چه Self-Attention و چه Feed-Forward) این مرحله‌ی Add & Norm وجود دارد که برای عمق ۶ لایه (در مدل اصلی) نقش حیاتی در آموزش پایدار شبکه دارد.

روند آموزش مدل Transformer

مدل Transformer به صورت نظارت‌شده (Supervised) برای وظایف مختلف قابل آموزش است. برای مثال در مسأله‌ی ترجمه، داده‌های موازی (جمله‌ی زبان مبدأ و جمله‌ی زبان مقصد) به مدل داده می‌شود و مدل یاد می‌گیرد که جملات ترجمه کند. تابع هزینه‌ی معمول در این حالت Cross-Entropy بین توالی خروجی پیش‌بینی‌شده و خروجی صحیح است. طی آموزش، مدل کلمات خروجی را یکی‌یکی تولید می‌کند و در هر گام از کلمات درست قبلی به عنوان ورودی Decoder استفاده می‌کند. سپس با مقایسه‌ی کلمه‌ی پیش‌بینی‌شده با کلمه‌ی مرجع، خطا محاسبه و سپس Backpropagation انجام می‌گیرد تا وزن‌ها به‌روزرسانی شوند.

کاربردهای مدل Transformer در حوزه‌های مختلف [9] [10] [11]

از زمان معرفی، مدل Transformer انقلابی در حوزه‌های گوناگون ایجاد کرده و به سرعت به معماری استاندارد برای بسیاری از مسائل تبدیل شده است. این مدل ابتدا در زمینه پردازش زبان طبیعی (NLP) استفاده شد و سپس کاربرد آن در حوزه‌های گوناگون گسترش یافت. برای مثال، مدل ترنسفورمر در زمینه ترجمه، خلاصه‌سازی متون، مدل‌های زبانی و درک متن، گفت‌وگو های هوشمند و بسیاری از زمینه‌های مرتبط، عملکرد خیلی خوبی داشته است.

کاربردهای Transformer در پردازش موسیقی و صوت:

معماری Transformer نه تنها در زبان طبیعی، بلکه در پردازش توالی‌های موسیقی و صوت نیز کاربردهای چشمگیری داشته است. موسیقی نیز می تواند به عنوان یک زبان ترتیبی دیده شود که در آن نت‌ها و آکوردها با ترتیب زمانی مشخصی دنبال هم می‌آیند و ساختارهای بلندمدت (تم‌ها، تکرار ملودی‌ها و ...) نقش مهمی در آن دارند. به همین خاطر، محققان به سراغ استفاده از ترنسفورمر برای مدل‌سازی موسیقی رفته‌اند. برای مثال، مدل Music Transformer توسط Huang و همکاران او در سال ۲۰۱۸ معرفی شد که یک نسخه‌ی بهبودیافته از ترنسفورمر برای تولید موسیقی با ساختار بلندمدت است [12]. همچنین در یک سناریوی ترانسفورمر دنباله‌به‌دنباله (seq2seq)، Music Transformer قادر به تولید همراهی (accompaniment) برای ملودی ورودی شد، به این معنی که می‌تواند بر اساس ملودی داده‌شده یک بخش هارمونی مکمل بنویسد.

۲-٤ جمع‌بندی

در این فصل به بررسی برخی از مفاهیم پایه مورد نیاز برای درک کارکرد مدل معرفی شده در این پژوهش پرداختیم. این مفاهیم شامل تعاریف مربوط به موسیقی (زیر و بمی صدا، اکتاو، آکورد و هارمونی)، و مفاهیم پایه یادگیری عمیق و پردازش سیگنال از جمله تبدیل Q ثابت (CQT)، افزایش داده و مدل پایه Transformer می‌باشند.

در فصل بعد، برخی از پژوهش‌های پیشین را بررسی کرده و نقاط قوت و ضعف هر کدام را بیان می‌کنیم.

فصل سوم: پژوهش‌های مرتبط

۳-۱ مقدمه

جهت پیاده‌سازی یک ابزار تشخیص آکورد خودکار با استفاده از روش‌های یادگیری عمیق، لازم است پژوهش‌هایی که پیش از این در این زمینه انجام شده را بررسی نماییم، تا با مطالعه مزایا و معایب هر یک از آن‌ها و جمع‌بندی اطلاعات بدست آمده، شکاف موجود در روش‌های گذشته را بیابیم و به راه حلی کارآمد برای این کاربرد دست پیدا کنیم.

بدین منظور، در این بخش، به ترتیب سال چاپ مقالات، از قدیمی‌ترین روش‌ها شروع کرده و به معرفی روش، مزایا و معایب هر کدام به طور خلاصه می‌پردازیم.

۳-۱-۱ شناسایی خودکار آکورد با استفاده از شبکه‌های عصبی کانولوشنی[13]

در این مقاله، رویکردی مبتنی بر یادگیری داده‌محور به عنوان روش حل مسئله تشخیص خودکار آکوردهای موسیقی ارائه شده است. در این رویکرد از شبکه‌های عصبی کانولوشنی (CNN) استفاده شده تا به کمک این روش ساختارهای سنتی چندمرحله‌ای کنار گذاشته شوند.

در روش‌های مرسوم تشخیص خودکار آکورد پیش از این مقاله، از ویژگی‌های دست‌ساز مانند ویژگی کروما‌ی مبتنی بر طیف و مدل‌های مخفی مارکوف (HMM) استفاده شده است. این روش‌ها به سقف عملکرد خود رسیده‌اند و به شدت به تنظیمات دستی و تخصصی وابسته‌اند. همچنین، ویژگی‌های کروما در بازنمایی انواع مختلف آکورد محدود هستند و فرضیات آماری آن‌ها (مانند توزیع گوسین) همیشه با داده‌های واقعی هم‌خوانی ندارد. بنابراین، در این مقاله یک مدل یادگیری عمیق مبتنی بر شبکه عصبی کانولوشنی، که به صورت مستقیم قطعات ۵ ثانیه‌ای از طیف زمان-فرکانس (pitch spectrogram) را به برچسب آکورد تبدیل می‌کند ارائه شده، و مدل به‌صورت End-to-End آموزش می‌بیند و دیگر نیازی به استخراج ویژگی دستی ندارد.

نمایش ورودی (Input Representation)

در این مقاله از تبدیل Q ثابت (Constant-Q Transform یا CQT) برای تبدیل سیگنال صوتی به یک نمایش زمان–فرکانس استفاده شده است. CQT، طیفی خطی در فضای فرکانس ایجاد می‌کند که متناسب با زیر و بمی (pitch)

موسیقی است. خروجی این تبدیل شامل ۲۵۲ باند فرکانسی در بازه ۲۷/۵ تا ۱۷۶۰ هرتز است. نرخ نمونه‌برداری ابتدا به ۷۰۴۰ هرتز کاهش یافته و سپس طیف‌ها با نرخ ۴ فریم بر ثانیه استخراج شده‌اند.

افزایش داده (Extended Training Data - ETD)

در این روش دو تکنیک مهم برای بهبود داده‌های آموزشی استفاده شده است:

انتقال گام (Transposition): جابه‌جایی کل طیف به بالا یا پایین برای شبیه‌سازی همان نمونه در کلیدهای دیگر (۱۲ برابر کردن داده‌ها). در واقع در این مرحله از خاصیت موسیقیایی آکوردها استفاده شده تا تعداد نمونه‌ها افزایش پیدا کند.

نرمال‌سازی کنتراست (Contrast Normalization): برای کنترل بهره (gain) در فرکانس‌های مختلف.

معماری شبکه:

معماری شبکه از چند لایه کانولوشنی (Convolutional) و دو لایه Fully Connected تشکیل شده است. لایه‌های کانولوشنی به شناسایی الگوهای زمانی و فرکانسی کمک می‌کنند. در نهایت، لایه SoftMax خروجی را به یکی از ۲۵ کلاس آکورد (۱۲ ماژور، ۱۲ مینور، بدون آکورد) نگاشت می‌کند.

سه نوع معماری ساده، متوسط، و پیچیده، با اندازه‌های مختلف برای شبکه آزمایش شده‌اند، که در هرکدام دو حالت با کرنل‌های کشیده‌شده در زمان یا فرکانس بررسی شدند.

استراتژی آموزش:

برای جلوگیری از بایاس کلاس‌ها، در هر مرحله‌ی آموزش، دسته‌هایی با توزیع یکنواخت از کلاس‌ها ساخته شده‌اند. در این مدل از گرادیان نزولی تصادفی با مینی‌بچ‌ها استفاده شده و معیار خطا، منفی لگاریتم درست‌نمایی (Negative Log-Likelihood) است. مدل‌ها به مدت ۶۰۰۰ تکرار آموزش داده شدند و بهترین وزن‌ها بر اساس عملکرد روی مجموعه اعتبارسنجی ذخیره شدند. برای تقسیم داده به بخش‌های آموزش/اعتبارسنجی/آزمون، از یک الگوریتم ژنتیک استفاده شده تا اطمینان حاصل شود که توزیع انتقال آکوردها در هر بخش مشابه است.

مزایا و معایب رویکرد ارائه شده:

این روش نسبت به روش‌های سنتی تشخیص آکورد این مزایا را داراست:

-حذف ویژگی‌های دستی: برخلاف روش‌های سنتی که به ویژگی‌هایی مثل کرومای دستی متکی بودند، CNN ویژگی‌ها را به‌صورت خودکار از داده‌ها یاد می‌گیرد.

-یکپارچگی مدل: فرایند استخراج ویژگی و طبقه‌بندی در یک مدل واحد انجام می‌شود؛ یعنی مدل مستقیماً از طیف صوتی به برچسب آکورد می‌رسد.

-استفاده مناسب از زمینه زمانی یا همان Temporal context: ورودی مدل، قطعه‌های ۵ ثانیه‌ای از طیف هستند که باعث می‌شود مدل نسبت به روش‌های فریم‌به‌فریم، درک بهتری از تغییرات هارمونیک در طول زمان داشته باشد.

-تقویت داده با انتقال گام آکوردها: با انتقال گام داده‌های آموزشی، مدل در برابر تغییرات کلید مقاوم شده و بیش‌برازش کاهش یافته است.

معایب:

-ساده‌سازی برچسب‌ها: برای کاهش پیچیدگی، تمامی آکوردها به ۲۵ کلاس (۱۲ ماژور، ۱۲ مینور، ۱ بی‌آکورد) کاهش یافته‌اند. این باعث از دست رفتن اطلاعات دقیق‌تر آکوردها مثل هفتم، ششم و سایر آکوردهای مهم موسیقی شده است.

-نیاز به داده برچسب دار: مدل به داده‌های برچسب‌خورده با کیفیت نیاز دارد. نویسندگان اشاره می‌کنند که خطا یا ابهام در برچسب‌ها (مثلاً نگاشت نادرست آکوردهای پیچیده به ماژور/مینور) می‌تواند مانع یادگیری مناسب شود.

-تعمیم‌پذیری محدود در مورد آکوردهای نادر: با وجود افزایش داده از طریق ETD، عملکرد مدل روی آکوردهای کم‌تکرار یا پیچیده همچنان چالش‌برانگیز است.

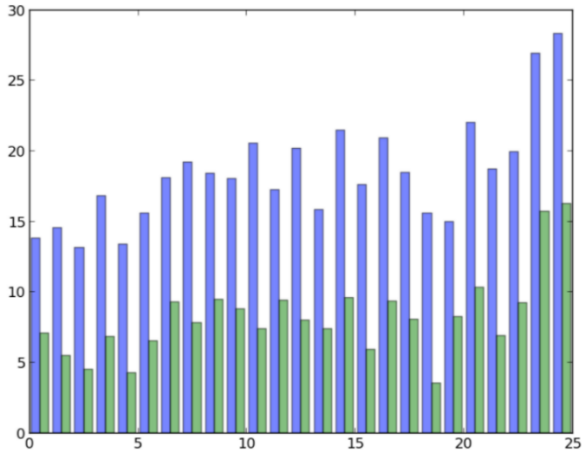
- عدم مدل‌سازی مستقیم توالی و تداوم زمانی: برخلاف روش‌هایی مثل HMM یا RNN که ساختار دنباله‌ای داده‌ها را مدل می‌کنند، CNN در این مقاله ساختار توالی زمانی را به‌صورت صریح مدل نمی‌کند.

مجموعه داده:

در این مقاله، نویسندگان از یک مجموعه داده‌ی ترکیبی استفاده کرده‌اند و با دقت زیادی روش تقسیم و آماده‌سازی داده را طراحی کرده‌اند تا یادگیری بهینه و نتایج قابل اعتماد به‌دست آید. این مجموعه داده شامل The Beatles Dataset، RWC Pop Dataset (شامل ۱۰۰ قطعه پاپ ژاپنی)، و US Pop Dataset (شامل ۱۹۴ قطعه از آهنگ‌های پاپ آمریکایی) می‌شود. در مجموع، حدود ۵۰۰۰۰ نمونه آکورد برچسب‌خورده در این مجموعه‌ها وجود دارد. بیش از ۸۰۰ برچسب آکورد متفاوت در ابتدا وجود داشته، و برای ساده‌سازی، تمام برچسب‌ها به ۲۵ کلاس (۱۲ آکورد ماژور، ۱۲ مینور و یک کلاس بدون آکورد) نگاشت شده‌اند.

تحلیل نتیجه نهایی رویکرد معرفی شده:

نمودار زیر برای هر کلاس آکورد، اختلاف بین دقت مدل روی داده‌های آموزش و آزمون را نمایش می‌دهد. رنگ آبی برای حالت بدون ETD و رنگ سبز برای حالت دارای ETD می‌باشد.



شکل ۳ - اختلاف بین دقت مدل روی داده‌های آموزش و آزمون [13]

در حالت بدون ETD، اختلاف بین دقت آموزش و آزمون برای بسیاری از کلاس‌ها بالا است، که نشان‌دهنده بیش‌برازش (Overfitting) است. اما با اعمال ETD، این اختلاف به‌صورت یکنواخت کاهش می‌یابد، که به معنی بهبود تعمیم‌دهی مدل است.همچنین تفاوت عملکرد در آکوردهای رایج و نادر، به‌وضوح کم‌تر شده است.

مقایسه عملکرد مدل پیشنهادی با روش‌های پیشین در شناسایی آکورد روی دیتاست

در مقاله، مقایسه‌ی مستقیم عددی بین مدل پیشنهادی و روش‌های قبلی ارائه نشده، اما نویسندگان به‌طور غیرمستقیم عملکرد مدل خود را با بهترین روش‌های پیشین مقایسه کرده‌اند. بر اساس اطلاعات مقاله و نتایج گزارش‌شده در مطالعات پیشین، می‌توان یک جدول مقایسه‌ای تقریبی ارائه داد که مدل CNN این مقاله را در کنار روش‌های مرسوم تا آن زمان قرار می‌دهد:

جدول ۱ - مقایسه عملکرد مدل با روش‌های مرسوم پیش از خود

مدل	نوع ویژگی‌ها	مدل طبقه‌بندی	دقت آزمون	سال
Sheh & Ellis	Chroma	HMM	65-70%	2003
Mauch & Dixon	Chroma + Bass	DBN	72%	2010
Cho & Bello	Smoothed Chroma	GMM + Viterbi	74%	2011
مقاله مورد بررسی	Pitch Spectrogram (CQT)	CNN (End-to-End)	77-78%	2012

۲-۱-۳ تشخیص آکورد با استفاده از شبکه عصبی بازگشتی ترکیبی (Hybrid Recurrent Neural Network) [14]

این مقاله، معماری نوینی برای تشخیص آکورد موسیقی ارائه می‌دهد که در آن مدل‌های مرسوم مارکوف پنهان (HMM) با مدل‌های زبانی مبتنی بر شبکه‌های عصبی بازگشتی (RNN) جایگزین شده‌اند. در این رویکرد، شبکه‌های عصبی عمیق (DNN) به صورت مستقیم ویژگی‌های متمایزکننده را از تبدیل زمان-فرکانس سیگنال صوتی یاد می‌گیرند و مدل RNN نیز وابستگی زمانی میان آکوردها را مدل‌سازی می‌کند. روش‌های سنتی تشخیص آکورد بر پایه ویژگی‌های دستی طراحی‌شده

(مانند کروماى کلاسیک) و مدل‌هاى احتمالاتى مانند HMM هستند. در این مقاله با بهره‌گیرى کامل از یادگیرى عمیق، هم مرحله استخراج ویژگى و هم مدل‌سازى دنباله آکوردها را به‌صورت خودکار انجام مى‌شود.

نوآورى اصلى مقاله، معرفى الگوریتم جست‌وجوى پرتو هش‌شده (Hashed Beam Search) است که مصرف حافظه و زمان پردازش را به‌طور چشمگیرى کاهش داده و این سامانه را براى کاربردهاى بلادرنگ (real-time) مناسب مى‌سازد. نتایج آزمایش‌ها نشان مى‌دهد که این مدل عملکردى هم‌تراز با بهترین مدل‌هاى ارزیابى MIREX دارد.

معمارى پیشنهادى

معمارى پیشنهادى این مقاله از دو بخش اصلى تشکیل شده است:

۱) مدل آکوستیک مبتنى بر شبکه عصبى عمیق (DNN)

۲) مدل زبانى مبتنى بر شبکه عصبى بازگشتى (RNN)

این دو مدل در قالب یک چارچوب ترکیبى (Hybrid RNN) به هم متصل مى‌شوند و هدف آن‌ها پیش‌بینى توالى آکوردها از سیگنال صوتى خام است.

مدل آکوستیک(Acoustic Model)

استخراج ویژگى‌ها:

- ورودى مدل، سیگنال صوتى با نرخ نمونه‌بردارى ۱۱/۰۲۵ کیلوهرتز است.
- تبدیل CQT (Constant-Q Transform) روى سیگنال انجام مى‌شود. (۷ اکتاو با ۲۴ فیلتر در هر اکتاو داریم. در نتیجه، ۱۶۸ ویژگى براى هر فریم زمانى وجود دارد)
- براى در نظر گرفتن بافت زمانى اطراف هر فریم، از پنجره زمانى (Context Window) استفاده مى‌شود. پنجره به طول ۷ فریم (۳ فریم قبل و بعد از فریم مرکزى).

ساختار شبکه DNN:

شبکه شامل ۳ لایه مخفی است، و تعداد نرون‌ها در تمام لایه‌ها برابر است. از تابع فعال‌ساز ReLU در لایه‌ها استفاده شده و لایه خروجی SoftMax دارد که توزیع احتمال بر روی ۲۵ کلاس آکورد (۱۲ ماژور، ۱۲ مینور، ۱ کلاس بدون آکورد) را خروجی می‌دهد.

برای جلوگیری از بیش‌برازش (Overfitting)، از تکنیک Dropout با نرخ ۰.۳ استفاده شده است. همچنین، برای بهبود یادگیری، الگوریتم ADADELTA جهت تنظیم خودکار نرخ یادگیری به‌کار گرفته شده. بهترین مدل بر اساس عملکرد روی مجموعه اعتبارسنجی (Validation) انتخاب می‌شود و در نهایت، از فعال‌سازی لایه پنهان آخر به‌عنوان ویژگی‌های جدید برای ورودی مدل آکوستیک اصلی استفاده می‌شود.

مدل زبانی (Language Model)

ساختار RNN:

یک شبکه بازگشتی دو لایه‌ای با سلول‌های حافظه LSTM استفاده شده، که این شبکه وظیفه مدل‌سازی توالی آکوردها را دارد. برخلاف مدل‌های مارکوف (HMM) که تنها وابستگی یک‌مرحله‌ای دارند، LSTM می‌تواند وابستگی‌های بلندمدت را یاد بگیرد. خروجی RNN در هر گام زمانی، توزیع احتمالاتی شرطی بر اساس تاریخچه آکوردهاست.

مدل ترکیبی (Hybrid RNN Architecture):

این مدل، تلفیق مدل آکوستیک و مدل زبانی می‌باشد. فرمول اصلی این مدل بدین شکل است:

$$P(z_1)P(x_1|z_1)\prod_{t=2}^{T}P(z_t|\mathcal{A}_t)P(x_t|z_t)$$

X(t): ویژگی‌های آکوستیکی فریم tام.

Z(t): برچسب آکورد در فریم tام.

A(t): تاریخچه آکوردها تا فریم t-1.

ویژگی مهم این مدل این است که فقط به x(t) نیاز دارد و مستقل از آکوردهای گذشته است. مدل زبانی P(z(t)|A(t)) را با RNN یاد می‌گیرد.

مدل آکوستیک و مدل زبانی به‌طور مستقل با بیشینه‌سازی تابع درست‌نمایی (log-likelihood) آموزش داده می‌شوند. مدل آکوستیک با داده‌های دارای برچسب صوتی، و مدل زبانی با داده‌های متنی آکورد آموزش داده می‌شود.

الگوریتم استنتاج (Inference Algorithm):

جستجوی پرتو (Beam Search):

چون مدل RNN حافظه دارد و خروجی‌اش به تمام آکوردهای قبلی وابسته است، پیش‌بینی به صورت مرحله‌ای انجام می‌شود. از Beam Search برای یافتن بهترین توالی استفاده می‌شود. در هر مرحله، فقط w مسیر با بالاترین احتمال نگه داشته می‌شود (beam width).

جستجوی پرتو هش‌شده (Hashed Beam Search):

برای جلوگیری از تکرار مسیرهای مشابه در beam، از hash function برای تعیین شباهت بین مسیرها استفاده شده است. اگر دو مسیر انتهای مشابه (مثلاً دو آکورد آخر یکسان) داشته باشند، فقط یکی نگه داشته می‌شود. این کار باعث صرفه‌جویی زیاد در حافظه و زمان شده و کارایی را حفظ می‌کند.

مزایای روش ارائه شده در مقاله:

- یادگیری ویژگی‌ها به صورت خودکار: در این روش نیازی به استخراج ویژگی‌های دستی مانند chroma نیست؛ چرا که شبکه عصبی عمیق (DNN) مستقیماً از تبدیل CQT ویژگی‌های مفید را یاد می‌گیرد.
- مدل‌سازی وابستگی زمانی دقیق‌تر: استفاده از RNN با سلول‌های LSTM امکان یادگیری وابستگی‌های بلندمدت بین آکوردها را فراهم می‌کند.
- معماری ترکیبی قابل انعطاف: می‌توان از منابع متنی بدون فایل صوتی برای آموزش مدل زبانی استفاده کرد. بنابراین نیاز به داده‌های برچسب‌خورده صوتی کمتر می‌شود.
- قابلیت استفاده بلادرنگ: الگوریتم Hashed Beam Search حافظه و زمان محاسباتی را به شدت کاهش می‌دهد. بنابراین امکان پیاده‌سازی در سیستم‌های تعاملی وجود دارد.

معایب روش ارائه شده در مقاله:

- نیاز به داده‌ی زیاد برای آموزش: DNNها و RNNها نیاز به حجم قابل توجهی از داده برای یادگیری مناسب دارند. با توجه به محدود بودن منابع صوتی برچسب‌خورده، آموزش کامل ممکن است دشوار باشد، که این تا حدی به علت قابلیت انعطاف مدل ترکیبی با استفاده از داده‌های متنی جبران می‌شود.
- عدم مدل‌سازی دقیق طول آکوردها: مدل زبانی (RNN) گذر بین آکوردها را خوب مدل می‌کند، اما طول ماندگاری هر آکورد (duration) را به‌صورت صریح مدل نمی‌کند. این موضوع می‌تواند باعث ناپایداری در پیش‌بینی مرزهای زمانی آکوردها شود.
- پیچیدگی نسبی در پیاده‌سازی: مدل ترکیبی نیاز به هماهنگی دقیق بین دو مدل (آکوستیک و زبانی) و الگوریتم Beam Search دارد.
- وابستگی به کیفیت زمان‌بندی برچسب‌ها: چون آموزش مدل آکوستیک بر پایه برچسب‌های فریم به فریم است، اشتباه در زمان‌بندی آکوردها در برچسب‌ها می‌تواند مدل را گمراه کند.

در مجموع، این روش با تلفیق یادگیری عمیق و مدل‌سازی ساختار زمانی، گامی مؤثر در جهت بهبود دقت سیستم‌های تشخیص آکورد برداشته است؛ اما برای بهره‌برداری کامل از ظرفیت‌های آن، دسترسی به داده‌های با کیفیت و منابع محاسباتی کافی الزامی است.

مجموعه داده

منبع اصلی مورد استفاده این مقاله مجموعه داده MIREX است. این مجموعه داده شامل دو بخش اصلی است:

الف) داده‌های Beatles, Queen و Zweieck:

در مجموع شامل ۲۱۷ قطعه موسیقی می‌باشد، که در آن برچسب‌های آکورد با دقت زمانی بالا در دسترس هستند. این داده‌ها معمولاً به عنوان معیار استاندارد در پژوهش‌های مربوط به تشخیص آکورد استفاده می‌شوند. برچسب‌ها توسط موسیقی‌دانان خبره تهیه شده‌اند و کیفیت آن‌ها بالاست.

ب) نسخه‌ی خلاصه‌شده از مجموعه داده‌ی Billboard:

شامل ۷۴۰ قطعه موسیقی از سبک‌های مختلف پاپ/راک است و نسخه‌ای خلاصه‌شده و پردازش‌شده از مجموعه کامل Billboard Annotated Dataset می‌باشد. این مجموعه شامل طیف متنوع‌تری از قطعات و ساختارهای آکوردی است.

تقسیم‌بندی داده برای آموزش و تست:

برای این کار از روش Cross-validation با ۴ دسته (۴-fold) استفاده شده است. در هر تکرار، ۳/۴ داده برای آموزش و ۱/۴ برای تست در نظر گرفته می‌شود. همچنین، بخشی از داده‌ی آموزشی (۲۰٪) به‌صورت جداگانه برای اعتبارسنجی (validation) استفاده شده است.

پیش‌پردازش برچسب‌ها:

تمام برچسب‌های آکورد به فرم ساده‌شده major/minor نگاشته شده‌اند:

۱۲ آکورد ماژور + ۱۲ آکورد مینور + یک کلاس بدون آکورد

این ساده‌سازی به دلیل کم بودن نمونه‌های آکوردهای پیچیده در داده‌ها انجام شده تا مدل پایدارتر آموزش ببیند.

نتایج:

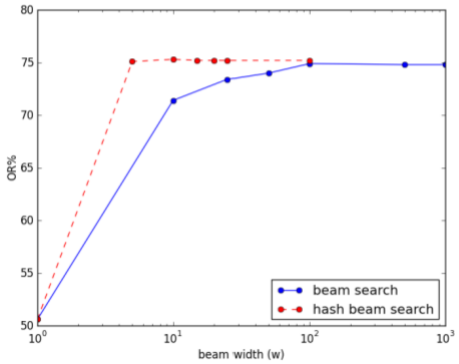
برای ارزیابی عملکرد مدل‌های مختلف از جمله مدل DNN، LSTM و Hybrid RNN، از دو معیار استاندارد Overlap ratio و Weighted Average Overlap Ratio استفاده شده است. در این جدول نتایج قابل مشاهده است:

جدول ۲ - نتایج ارزیابی عملکرد مدل‌های مختلف

مدل	OR (%)	WAOR(%)
DNN با ورودی CQT	57.0	56.5
DNN با ویژگی یادگرفته شده	69.8	69.1
DNN با ویژگی + پنجره زمانی	72.9	72.5
Hybrid RNN با ویژگی	73.4	73.5
Hybrid RNN با ویژگی + پنجره زمانی	75.5	75.0

در نتیجه، استفاده از ویژگی‌های یادگرفته‌شده عملکرد را به‌شدت بهتر کرده. همچنین، استفاده از پنجره زمانی باعث بهبود بیشتر شده، و استفاده از مدل ترکیبی (Hybrid) با LSTM، بهترین نتیجه را ارائه داده است.

در ادامه دو نمودار که داخل مقاله ارائه شده را نیز بررسی خواهیم کرد.

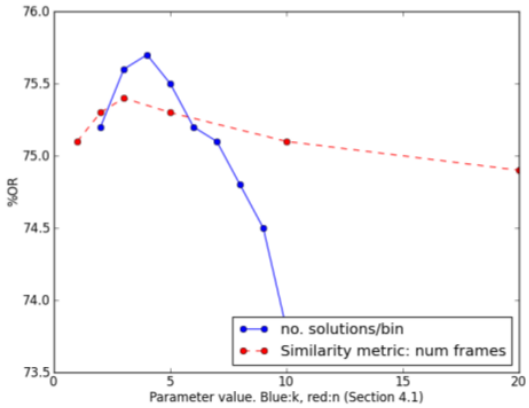


شکل ۴ - اثر Beam width در Beam Search [14]

محور افقی: عرض پرتو (beam width)

محور عمودی: OR (درصد پیش‌بینی صحیح آکوردها)

الگوریتم Hashed Beam Search با beam width بسیار کمترعملکردی معادل یا بهتر از Beam Search معمولی با عرض بزرگ دارد. این نشان می‌دهد که Hashed Beam Search به‌طور مؤثر مسیرهای تکراری را حذف کرده و جستجوی بهینه‌تری انجام می‌دهد. نمودار شکل ۵ اثر پارامترهای Hashing در Beam Search را نشان می‌دهد.



شکل ۵ - اثر پارامترهای Hashing در Beam Search [14]

محور افقی: مقدار n یا k (بسته به خط)

محور عمودی: OR (درصد پیش‌بینی صحیح آکوردها)

با افزایش n، (تعداد فریم قبلی برای hash)، تا حدی نتیجه پایدار می‌ماند اما بعد از $n = 4$ یا ۵، کاهش در دقت مشاهده می‌شود. همچنین، مقدار بهینه k تقریباً بین ۲ تا ۵ است. برای $k > 5$، دقت افت می‌کند چون beam با مسیرهای بسیار مشابه اشباع می‌شود.

جمع‌بندی:

نتایج تجربی نشان می‌دهد که این سیستم دقتی در سطح روش‌های برتر موجود در زمان خود را داراست و در عین حال از مزایای سادگی، انعطاف‌پذیری و کارایی بالا برخوردار است.

۳-۱-۳ مدل شنیداری عمیق کانولوشنی برای تشخیص آکورد موسیقی[15]

این مقاله یک معماری یادگیری عمیق کاملاً کانولوشنی برای تشخیص خودکار آکوردهای موسیقی معرفی می‌کند که ترکیبی از شبکه‌های عصبی کانولوشنی (CNN) و میدان تصادفی شرطی (CRF) برای پیش‌بینی ساختاریافته است. برخلاف روش‌های سنتی که به ویژگی‌های دستی مانند chroma وابسته‌اند، این سیستم به‌صورت end-to-end آموزش داده می‌شود و ویژگی‌های معنادار موسیقایی را مستقیماً از ورودی طیف‌نگار (spectrogram) یاد می‌گیرد. شبکه CNN ویژگی‌های زمانی-فرکانسی را استخراج می‌کند و CRF با مدل‌سازی وابستگی زمانی بین آکوردها، پیوستگی و گذارهای طبیعی‌تر را فراهم می‌سازد. نتایج آزمایش‌ها با استفاده از اعتبارسنجی متقاطع روی چند مجموعه داده از جمله Isophonics، RWC و Robbie Williams نشان می‌دهد که این مدل عملکردی هم‌سطح یا بهتر از سیستم‌های برتر ارزیابی‌شده در رقابت MIREX دارد، خصوصاً برای آکوردهای ماژور و مینور.

معماری پیشنهادی

این مقاله یک مدل end-to-end برای تشخیص آکورد ارائه می‌دهد که دو جزء اصلی دارد:

- شبکه عصبی کانولوشنی (CNN)

هدف این شبکه یادگیری ویژگی‌های زمانی-فرکانسی موسیقایی به‌طور خودکار از طیف‌نگار (spectrogram) صوتی، بدون نیاز به ویژگی‌های دستی مانند chroma می‌باشد.

ورودی: طیف‌نگار CQT (Constant-Q Transform) لگاریتمی‌شده.

پوشش فرکانسی: ۶ اکتاو با ۳۶ فیلتر در هر اکتاو که در نتیجه برای هر فریم زمانی ۲۱۶ ویژگی می‌دهد.

نرمال‌سازی: میانگین صفر و واریانس یک، همراه با Batch Normalization برای تسریع همگرایی.

ساختار CNN:

این شبکه کاملاً کانولوشنی است و هیچ لایه Fully Connected یا Pooling در آن وجود ندارد. در این بخش از معماری فقط از لایه‌های ۱D convolution روی بُعد زمان استفاده شده و از ایده Network in Network الهام گرفته شده تا ویژگی‌های موضعی پیچیده‌تری استخراج شود. تابع فعال‌سازی Relu بعد از هر لایه اعمال می‌شود و لایه خروجی توزیع احتمال روی ۲۵ کلاس (۱۲ آکورد ماژور، ۱۲ مینور، ۱ بدون آکورد) را نتیجه می‌دهد. مزیت این طراحی این است که برخلاف معماری‌هایی که از pooling استفاده می‌کنند، این طراحی برای هماهنگی زمانی فریم‌ها با برچسب‌های آکورد بسیار مناسب است.

۲- میدان تصادفی شرطی خطی (CRF)

هدف این بخش از معماری مدل‌سازی ساختار زمانی در توالی آکوردها و افزایش پیوستگی در پیش‌بینی‌ها می‌باشد.

CRF بعد از خروجی CNN قرار می‌گیرد و با استفاده از احتمالات CNN برای هر فریم، احتمال کلی دنباله آکوردها را محاسبه می‌کند. در این بخش، ماتریس گذار (Transition Matrix) نشان‌دهنده احتمال تغییر از یک آکورد به آکورد دیگر است. آموزش CRF همراه با CNN به‌صورت مشترک و end-to-end انجام می‌شود.

تابع هدف برای آموزش ترکیب درست‌نمایی منفی (Negative Log-Likelihood) برای CRF به همراه منظم‌ساز L1 برای جلوگیری از بیش‌برازش در ماتریس گذار می‌باشد.

برای آموزش از الگوریتم بهینه‌سازی Adam استفاده می‌شود. نرخ یادگیری اولیه ۰.۰۱ است و Batch size دنباله‌هایی با ۱۰۲۴ فریم می‌باشد. اگر عملکرد اعتبارسنجی تا ۵ دوره متوالی بهبود نیابد، آموزش متوقف می‌شود.

فرآیند پیش‌بینی:

ورودی: طیف‌نگار صوتی یک قطعه موسیقی

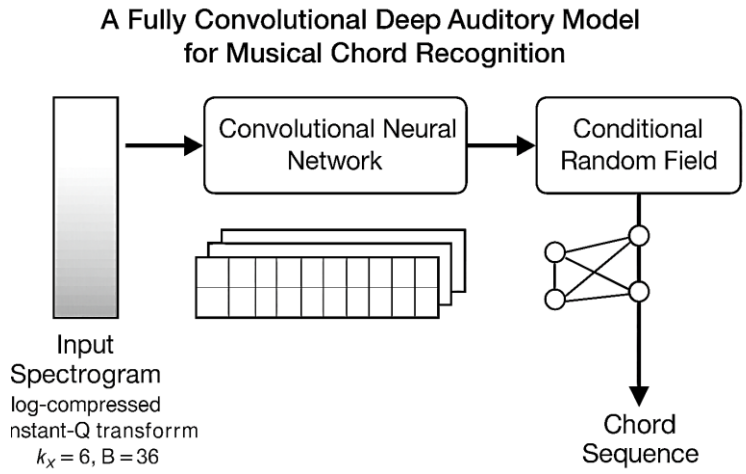
این ورودی از یک شبکه CNN عبور می کند و این شبکه حتمال آکورد برای هر فریم را استخراج می‌کند. سپس مدل CRF روی این احتمالات اعمال می‌شود و اصلاح نهایی توالی آکوردها با درنظرگرفتن وابستگی‌های زمانی صورت می‌گیرد.

جدول (۳)، نکات کلیدی در طراحی معماری این مدل را نشان می‌دهد:

جدول ۳ - نکات کلیدی در طراحی معماری مدل شنیداری عمیق کانولوشنی

مؤلفه	هدف	مزیت
CNN بدون لایه dense	استخراج ویژگی موضعی	کاهش تعداد پارامتر و افزایش هم‌راستایی زمانی
بدون pooling	حفظ طول توالی	دقت بالا در مرزهای زمانی آکوردها
CRF	مدل‌سازی ساختار زمانی	پیوستگی بیشتر در خروجی‌ها
آموزش مشترک CNN وCRF	یادگیری انتها به انتها	بهینه‌سازی همزمان ویژگی‌ها و ساختار توالی

در شکل زیر نیز طرحی کلی از این معماری دیده می‌شود:



شکل ۶ = طرحی کلی از معماری مدل شنیداری عمیق کانولوشنی [15]

مزایای روش ارائه شده:

روش ارائه‌شده در این مقاله با بهره‌گیری از شبکه عصبی کانولوشنی کاملاً کانولوشنی (Fully Convolutional CNN) و میدان تصادفی شرطی (CRF)، مزایای قابل‌توجهی را فراهم می‌کند. این مدل به‌صورت انتها به انتها آموزش داده می‌شود و بدون نیاز به استخراج ویژگی‌های دستی (مانند chroma)، مستقیماً از طیف‌نگار صوتی ویژگی‌های مؤثر را یاد می‌گیرد. حذف لایه‌های fully-connected و pooling باعث حفظ دقیق ترتیب زمانی فریم‌ها می‌شود که برای مرزبندی آکوردها بسیار حیاتی است. همچنین، استفاده از CRF پس از CNN باعث می‌شود که گذارهای بین آکوردها با در نظر گرفتن وابستگی‌های زمانی طبیعی‌تر و موسیقایی‌تر صورت گیرد، که دقت پیش‌بینی و پیوستگی زمانی خروجی‌ها را بهبود می‌دهد. این مدل با دقت بالا و پیچیدگی محاسباتی نسبتاً پایین، عملکردی رقابتی با روش‌های برتر همزمان با خود را داراست.

معایب روش ارائه شده:

با وجود مزایای متعدد، این روش نیز دارای محدودیت‌هایی است. مهم‌ترین آن، تمرکز فقط بر روی آکوردهای ماژور و مینور است که باعث می‌شود تنوع واقعی آکوردها در موسیقی پوشش داده نشود. همچنین، اگرچه CRF وابستگی‌های زمانی را مدل‌سازی می‌کند، اما هنوز مدل نمی‌تواند طول دقیق ماندگاری آکوردها را به‌صورت صریح کنترل کند (یعنی duration modeling صریح ندارد). از نظر پیاده‌سازی، آموزش همزمان CNN و CRF نیاز به تنظیم دقیق پارامترها و منابع محاسباتی بیشتر نسبت به روش‌های ساده‌تر دارد. همچنین، این مدل برای عملکرد بهینه نیازمند داده‌های برچسب‌خورده با دقت زمانی بالا است که تهیه آن در مقیاس بزرگ کار ساده‌ای نیست.

مجموعه داده:

در این مقاله، برای آموزش و ارزیابی مدل تشخیص آکورد، از یک مجموعه داده‌ی ترکیبی و متنوع استفاده شده است که شامل قطعات موسیقی از منابع مختلف با سبک‌های گوناگون است. این تنوع به مدل اجازه می‌دهد تا عملکرد خود را روی طیف وسیعی از ساختارهای موسیقایی نشان دهد.

سه مجموعه داده شامل موارد زیر می‌شوند:

- مجموعه داده Isophonics: این مجموعه شامل ۲۱۷ قطعه موسیقی از سه مجموعه مشهور The Beatles، Queen، Zweieck می‌شود.
- RWC Popular Music: این مجموعه نیز شامل ۱۰۰ آهنگ پاپ از خوانندگان ژاپنی و آمریکایی می‌باشد. محتوای این مجموعه با هدف تحقیقاتی گردآوری شده و تنوع سبک و ساختار آکوردی دارد.
- Robbie Williams Dataset: شامل ۶۵ قطعه موسیقی از خواننده پاپ بریتانیایی Robbie Williams

روش استفاده از داده‌ها:

در این متد از روش ۸-fold cross-validation برای ارزیابی استفاده شده است. در هر مرحله، داده‌ها به ۸ قسمت تقسیم می‌شوند؛ ۷ قسمت برای آموزش و ۱ قسمت برای آزمون. این روش از نظر آماری دقیق‌تر از تقسیم ساده train/test است و عملکرد مدل را به‌شکل عمومی‌تری نشان می‌دهد.

برچسب‌گذاری آکوردها:

تمام قطعات دارای برچسب‌های زمانی دقیق برای آکوردها هستند.

برچسب‌ها به ۲۵ کلاس محدود شده‌اند: ۱۲ آکورد ماژور، ۱۲ مینور و ۱ بدون آکورد.

نکته مهم در مورد تفاوت این مدل با مدل‌های قبل از خود که از داده‌های مشترک برای آموزش و تست استفاده می‌کردند (مثل برخی روش‌های MIREX)، این است که در این مقاله ارزیابی به‌صورت cross-validation بدون تداخل بین داده‌های آموزش و تست انجام شده که اعتبار نتایج را افزایش می‌دهد. همچنین تنوع مجموعه داده‌ها از نظر سبک، ساختار آکورد و خواننده باعث شده مدل توانایی تعمیم به شرایط واقعی‌تری داشته باشد.

نتایج بدست آمده:

در این قسمت با استفاده از معیار ارزیابی WCSR یا همان Weighted Chord Symbol Recall که هم‌ارز با WAOR است، عملکرد این مدل را بررسی می کنیم. این معیار میزان هم‌پوشانی زمانی بین آکوردهای پیش‌بینی‌شده و برچسب واقعی را اندازه‌گیری می‌کند. وزن‌دهی صورت گرفته بر اساس مدت زمان هر آکورد می‌باشد.

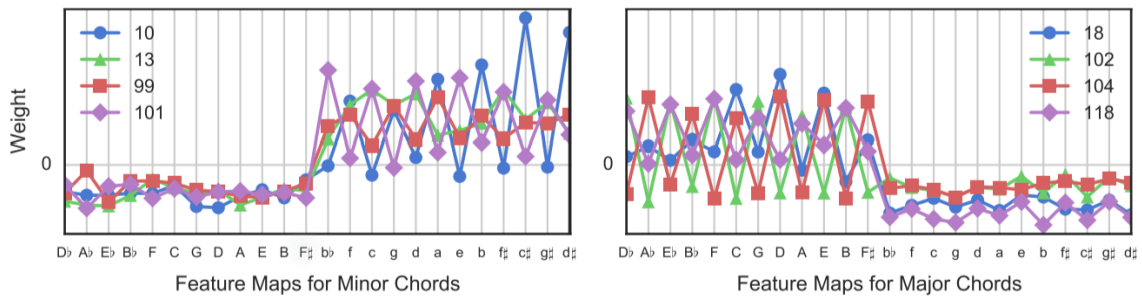
نتایج عددی بدست آمده در جدول (۴) نشان داده شده است:

جدول ۴ - نتایج بدست آمده از مدل شنیداری عمیق کانولوشنی

مجموعه داده	دقت WCSR (٪)
Isophonics	82.9
RWC Popular	82.8
Robbie Williams	82.5

این نتایج نشان می‌دهد که مدل پیشنهادی عملکردی بسیار پایدار و دقیق روی مجموعه داده‌های متنوع دارد.

در شکل (۷)، دو نمودار Feature Map برای آکوردهای ماژور و مینور نشان داده شده:



شکل ۷ - Feature Map برای آکوردهای ماژور و مینور [15]

این نمودارها نشان می‌دهند که شبکه CNN به‌صورت خودکار توانسته نگاشت‌های ویژگی تخصصی برای دسته‌بندی آکوردها یاد بگیرد، بدون اینکه به ویژگی‌های دستی وابسته باشد. یعنی بعضی کانال‌های شبکه به‌صورت طبیعی به ماژور و برخی دیگر به مینور حساس شده‌اند، که یک مزیت بزرگ مدل‌های End-to-End Learning محسوب می‌شود.

محور افقی:

نشان‌دهنده نت‌های پایه (root note) است. حرف‌های کوچک آکورد مینور و حرف‌های بزرگ آکورد ماژور را نشان می‌دهند.

محور عمودی (Weight):

نشان‌دهنده وزن (weight) اختصاص‌یافته به هر نت توسط برخی نگاشت‌های ویژگی خاص در لایه‌های شبکه CNN است. این وزن‌ها به‌طور غیرمستقیم نشان می‌دهند که کدام نت‌ها برای شناسایی آکوردها مهم‌تر هستند.

هر خط مربوط به یک Feature Map خاص در CNN است؛ مثلاً شماره‌های ۱۰، ۱۳، ۹۹، ۱۰۱ برای مینور، و ۱۸، ۱۰۲، ۱۰۴، ۱۱۸ برای ماژور هستند. این نگاشت‌ها بخشی از لایه‌های کانولوشن هستند که نقش در استخراج ویژگی از طیف‌نگار دارند.

تفسیر نمودار سمت چپ (مینور):

وزن‌ها در محدوده نت‌های مینور (مانند a، e، d) نسبت به سایر نت‌ها بالاتر هستند. این نشان می‌دهد که بعضی نگاشت‌ها (مثلاً شماره ۱۰ و ۹۹) مخصوص تشخیص آکوردهای مینور آموزش دیده‌اند.

تفسیر نمودار سمت راست (ماژور):

برعکس، در این نمودار، نت‌های ماژور (مثل C، G، D، A) وزن‌های بالاتری دارند. برخی Feature Mapها مانند شماره ۱۰۴ پاسخ شدیدی به نت‌های ماژور نشان می‌دهند و تقریباً در نواحی مینور بی‌اثرند.

مدل پیشنهادی CNN+CRF موفق شد در وظیفه تشخیص آکورد، دقتی در حدود ۸۲.۵٪ تا ۸۲.۹٪ (WCSR) روی سه مجموعه داده مختلف کسب کند. این عملکرد با برترین روش‌های موجود در آن زمان در رقابت MIREX برابری می‌کند یا از آن‌ها پیشی می‌گیرد. همچنین استفاده از CRF موجب بهبود پیوستگی زمانی در پیش‌بینی آکوردها نسبت به خروجی خام CNN شده‌است.

۴-۱-۳ تشخیص خودکار آکورد با مدل MIDI-trained Deep Feature and BLSTM-CRF [12]

این مقاله یک سامانه پیشرفته برای تشخیص آکوردهای موسیقی ارائه می‌دهد که از ترکیب یک استخراج‌کننده ویژگی مبتنی بر شبکه عصبی DRN آموزش‌دیده با داده‌های MIDI، و یک مدل دنباله‌ای BLSTM-CRF برای برچسب‌گذاری آکورد استفاده می‌کند. برخلاف روش‌های مرسوم که تنها از صدا (audio) استفاده می‌کنند، در این پژوهش از داده‌های MIDI هم برای آموزش دقیق‌تر استفاده شده و یک بازنمایی صوتی ۳۶-بعدی طراحی شده است. در مرحله پس از پردازش، یک تصمیم‌گیر ساده اما مؤثر برای شناسایی آکوردهای پیچیده‌تر از آکوردهای مینور و ماژور، مانند آکوردهای هفتم و وارون نیز ارائه می‌شود که دقت و پوشش سیستم را تا ۱۸۱ نوع آکورد افزایش می‌دهد.

داده‌های MIDI:

داده‌های MIDI (Musical Instrument Digital Interface) فایل‌هایی هستند که به‌جای ذخیره صدای واقعی، اطلاعات ساختاری موسیقی مانند نت‌ها، شدت، طول زمان و سازها را به‌صورت دیجیتال ثبت می‌کنند. این داده‌ها به دلیل ساختار دقیق و وضوح زمانی بالا، برای آموزش مدل‌های یادگیری ماشین بسیار مفیدند، زیرا امکان تبدیل مستقیم به نت‌های فعال در هر لحظه را فراهم می‌کنند. در این مقاله، نویسندگان از بیش از ۱۲ هزار فایل MIDI (از دیتاست‌هایی مانند Lakh و RWC) استفاده کرده‌اند و با تبدیل آن‌ها به فایل صوتی هم‌تراز، یک مجموعه آموزش عظیم و دقیق برای یادگیری ویژگی‌های هارمونیک طراحی کرده‌اند.

معماری پیشنهاد شده:

در این مقاله یک معماری سه‌مرحله‌ای معرفی شده است که با هدف افزایش دقت در تشخیص آکوردهای ساده و پیچیده (از جمله آکوردهای هفتم و وارونه) طراحی شده است.

معماری کلی شامل این سه بخش است:

بخش اول: استخراج ویژگی (Feature Extraction) با شبکه‌ی DRN

هدف از این بخش، یادگیری یک بازنمایی عمیق از ساختار هارمونیک موسیقی با استفاده از داده‌های MIDI می‌باشد. در این شبکه از یک شبکه عصبی Deep Residual Network با ۵ لایه fully-connected (هرکدام ۱۰۲۴ نورون) استفاده شده است. برای لایه‌های میانی از تابع فعال‌سازی tanh و برای لایه‌های خروجی از تابع sigmoid استفاده شده، و میان لایه‌ها نیز shortcut connection وجود دارد.

ورودی: ورودی این بخش، طیف‌نگار (Spectrogram) به‌دست‌آمده از تبدیل CQT بر روی فایل صوتی (با ۲۴ باند در هر اکتاو) می‌باشد. این ورودی به مقیاس لگاریتمی برده شده و نرمال‌سازی میانگین-واریانس روی آن انجام می‌شود.

خروجی: بردار ویژگی ۳۶-بعدی برای هر فریم زمانی

این طراحی باعث می‌شود که اطلاعاتی مانند ریشه، ساختار آکورد و وارونگی‌ها به‌صورت ضمنی در ویژگی‌ها لحاظ شود.

بخش دوم: طبقه‌بندی توالی با BLSTM (Bidirectional LSTM)

در این بخش مدل‌سازی وابستگی‌های زمانی بین فریم‌های پیوسته برای تشخیص دقیق‌تر آکوردها صورت می‌گیرد. این قسمت از معماری شامل یک شبکه LSTM دوطرفه (BLSTM) با ۱۲۸ نورون در جهت جلو و ۱۲۸ نورون در جهت عقب می‌باشد.

خروجی: توزیع احتمال ۲۵-بعدی برای هر فریم

برای جلوگیری از بیش‌برازش (overfitting)، روی خروجی LSTM از dropout با احتمال ۰.۵ استفاده شده است.

بخش سوم: رمزگشایی توالی برچسب‌ها با CRF (Conditional Random Field)

این قسمت برای افزایش پیوستگی زمانی و کاهش پرش‌های ناگهانی در توالی آکوردها طراحی شده است.

CRF به‌جای تصمیم‌گیری مستقل برای هر فریم، به توالی کلی نگاه می‌کند و توالی برچسبی بهینه را با الگوریتم Viterbi استخراج می‌کند.

مرحله نهایی: پردازش برای تشخیص آکوردهای پیچیده (هفتم و وارونگی)

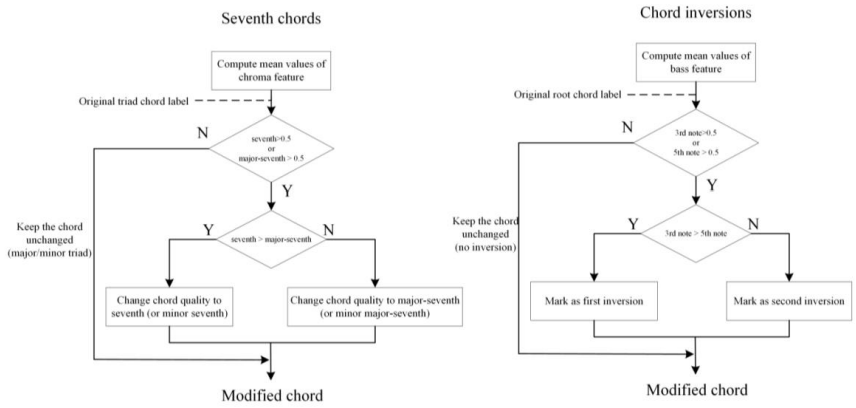
برای این منظور، ابتدا مدل فقط آکوردهای ساده را پیش‌بینی می‌کند. سپس، با بررسی میانگین ویژگی‌ها در نواحی خاص (سوم، پنجم، هفتم، نت باس) تشخیص داده می‌شود که:

آیا آکورد دارای نت هفتم هست؟

آیا وارونگی (inversion) رخ داده است؟

پس در نتیجه، سیستم می‌تواند حداکثر تا ۱۸۱ نوع آکورد را شناسایی کند (با احتساب وارونگی‌ها و آکوردهای هفتم).

این فرایند در شکل (۸) به خوبی نمایش داده شده:



شکل ۸ - فرآیند تشخیص آکورد در مدل MIDI-trained Deep Feature and BLSTM-CRF [12]

در جدول (۵) یک جمع‌بندی کلی از معماری صورت گرفته است:

جدول ۵ - معماری مدل MIDI-trained Deep Feature and BLSTM-CRF

مرحله	مدل	نقش
۱	DRN	استخراج ویژگی عمیق از طیف‌نگار
۲	BLSTM	طبقه‌بندی آکورد در سطح فریم با در نظر گرفتن توالی
۳	CRF	افزایش پیوستگی زمانی خروجی‌ها
۴	Post-processing	تشخیص آکوردهای پیچیده و وارونه

مزایای معماری پیشنهاد شده:

- استفاده از داده‌های MIDI برای آموزش، امکان یادگیری در مقیاس وسیع بدون نیاز به برچسب‌گذاری دستی را فراهم می‌کند.
- استخراج ویژگی‌های دقیق و ساختاری (pitch class، نت باس، نت بالا) از طریق شبکه DRN.
- مدل BLSTM وابستگی زمانی را به‌خوبی یاد می‌گیرد و باعث پیوستگی در توالی آکوردها می‌شود.
- CRF در مرحله‌ی رمزگشایی باعث حذف پیش‌بینی‌های ناپایدار و افزایش دقت زمانی خروجی می‌شود.

- توانایی تشخیص آکوردهای پیچیده (هفتم و وارونگی) با استفاده از منطق ساده در مرحله پس‌پردازش.
- پشتیبانی از واژگان بزرگ آکورد تا ۱۸۱ نوع بدون نیاز به افزایش تعداد کلاس‌ها در شبکه عصبی.

معایب معماری پیشنهاد شده:

- پیچیدگی بالا در معماری شامل چند مرحله مجزا
- مدل فقط آکوردهای ساده را مستقیماً پیش‌بینی می‌کند؛ آکوردهای پیچیده به پس‌پردازش وابسته‌اند.
- تشخیص آکوردهای هفتم و وارونه به آستانه‌گذاری ساده متکی است که ممکن است در شرایط مرزی یا نویزی دقت نداشته باشد.
- نیاز به پردازش دقیق و هماهنگ بین MIDI و فایل صوتی (هم‌ترازسازی زمان‌بندی) برای استخراج ویژگی.
- نیاز به منابع محاسباتی بالا برای آموزش و تنظیم اجزای مختلف شبکه

مجموعه داده:

در این مقاله، دو دسته داده‌ی مجزا برای آموزش دو بخش مختلف مدل استفاده شده‌اند: یکی برای آموزش استخراج‌کننده ویژگی (Feature Extractor) و دیگری برای آموزش طبقه‌بند دنباله‌ای (BLSTM-CRF).

آموزش استخراج‌کننده ویژگی (DRN):

-۱۲،۰۰۰ فایل MIDI به‌صورت تصادفی از دیتاست Lakh MIDI Dataset

-۲۱۰ فایل MIDI از مجموعه داده‌ی RWC (Classical, Jazz, Popular)

روش آماده‌سازی:

فایل‌های MIDI با استفاده از نرم‌افزار Direct MIDI to MP3 Converter و Chorium Soundfont به فایل صوتی تبدیل شده‌اند؛ سپس طیف‌نگار CQT برای هر فایل محاسبه شده و به‌صورت لگاریتمی و نرمال‌شده به شبکه DRN داده شده است. برچسب‌ها شامل سه بردار ۱۲بعدی هستند. در نهایت برای هر فریم صوتی، یک بردار ویژگی ۳۶بعدی ایجاد می‌شود.

آموزش مدل BLSTM-CRF:

برای آموزش طبقه‌بند دنباله‌ای که وظیفه‌ی پیش‌بینی توالی آکوردها را دارد، از فایل‌های صوتی واقعی به همراه برچسب‌های دقیق زمانی آکورد استفاده شده است.

منابع:

-RWC Popular Music Dataset

-USPOP Dataset

-دیتاستی شامل موسیقی پاپ آمریکایی با برچسب‌های دقیق آکورد

این مقاله با بهره‌گیری از MIDI به‌عنوان منبع آموزشی غنی و دقیق، توانسته یک ویژگی‌نگار قوی بسازد که برای سیستم‌های دیگر معمول نیست. در مرحله‌ی نهایی، سیستم توانایی پوشش ۶۱ تا ۱۸۱ نوع آکورد را دارد، در حالی که مدل‌های مرسوم اغلب به ۲۵ کلاس (ماژور/مینور/بدون آکورد) محدود هستند.

نتایج بدست آمده:

در این مقاله بخش نتایج به ارزیابی عملکرد مدل در دو سطح پرداخته است:

۱- ارزیابی تشخیص آکوردهای ساده (ماژور و مینور)

استفاده از CRF در مرحله‌ی خروجی باعث کاهش نوسان بین آکوردهای متوالی شده و دقت کلی سیستم را افزایش داده است. در مقاله، داده‌های عددی (مانند دقت درصدی یا WAOR) برای مقایسه مستقیم گزارش نشده‌اند و ارزیابی بیشتر کیفی است.

۲- ارزیابی پس‌پردازش و تشخیص آکوردهای پیچیده

پس از پیش‌بینی آکوردهای پایه، مدل از یک روش ساده‌ی آستانه‌گذاری (thresholding) روی ویژگی‌های استخراج‌شده استفاده می‌کند تا آکوردهای هفتم (seventh chords) را شناسایی کند و وارونگی‌های آکورد (first/second inversion) را تشخیص دهد.

در بخش نتیجه‌گیری، نشان داده شده که ترکیب یک استخراج‌کننده ویژگی مبتنی بر داده‌های MIDI با مدل دنباله‌ای BLSTM-CRF، یک چارچوب مؤثر برای تشخیص آکوردهای موسیقی فراهم می‌کند. این سیستم توانسته با یادگیری بازنمایی‌های هارمونیک دقیق، آکوردهای ساده را با دقت بالا شناسایی کند و از طریق یک فرآیند پس‌پردازش هدفمند،

آکوردهای پیچیده‌تری مانند آکوردهای هفتم و وارونگی‌ها را نیز به‌درستی تشخیص دهد. بدون نیاز به افزایش تعداد کلاس‌های خروجی شبکه، این روش قادر است واژگان آکورد را تا ۱۸۱ نوع گسترش دهد و در عین حال پیوستگی زمانی و دقت ساختاری پیش‌بینی‌ها را حفظ کند. این مقاله نشان می‌دهد که ترکیب یادگیری عمیق با داده‌های غنی MIDI و طراحی مرحله‌بندی‌شده، رویکردی کارآمد برای حل مسئله تشخیص آکورد در موسیقی است.

۳-۱-۵ تولید موسیقی به کمک transformer [12]

در مقاله Music Transformer: Generating Music with Long-Term Structure، پژوهشگران Google Brain مدلی به نام Music Transformer معرفی می‌کنند که بر پایه‌ی معماری Transformer و با استفاده از self-attention نسبی (relative attention) قادر است موسیقی‌هایی با ساختار بلندمدت و منسجم تولید کند. این مدل نسبت به نسخه‌های قبلی Transformer، با بهینه‌سازی فضای حافظه، امکان آموزش روی دنباله‌های طولانی (تا هزاران گام زمانی) را فراهم می‌کند. با استفاده از داده‌های MIDI و بازنمایی رویدادمحور (event-based)، Music Transformer قادر است ملودی‌هایی با تکرار الگوهای ساختاری، جمله‌بندی‌های موسیقایی و حتی همراهی (accompaniment) تولید کند. نتایج کمی (پایین‌تر بودن perplexity) و کیفی (ارزیابی انسانی) نشان می‌دهند که نسخه‌ی مجهز به attention نسبی، نسبت به LSTM و نسخه پایه Transformer عملکرد برتری دارد.

معماری:

Transformer بر پایه‌ی مدل Transformer استاندارد ساخته شده، اما با یک نوآوری کلیدی. آن هم این است که به جای self-attention مطلق از relative self-attention استفاده شده است.

۱- نوع ورودی مدل (بازنمایی داده‌ها)

ورودی مدل دنباله‌ای از رویدادهای موسیقایی است (نه طیف‌نگار یا موج صوتی). در این معماری از فرمت رویدادمحور (Event-based MIDI Representation) استفاده شده. هر رویداد به یک توکن گسسته تبدیل می‌شود. این دنباله مانند متن در NLP به مدل داده می‌شود.

مثال‌هایی از این رویدادها:

Velocity Change ،Time Shift ،Note Off ،Note On

۲- ساختار مدل (Decoder-Only Transformer)

در این معماری فقط از بخش Decoder مدل Transformer استفاده شده (بدون Encoder).

لایه‌های موجود در این قسمت از معماری شامل این موارد می‌شوند:

جدول ۶ - لایه‌های موجود در معماری transformer

لایه	مشخصات
Self-Attention Layer	با relative attention
Feedforward Network	دو لایه‌ی fully-connected با ReLU
Layer Normalization	بین هر بخش قرار می‌گیرد
Dropout	برای جلوگیری از overfitting

- نوآوری اصلی: Relative Positional Attention

در مدل Transformer استاندارد، از موقعیت‌های مطلق برای تمایز بین توکن‌ها استفاده می‌شود. اما این روش مشکل‌ساز است؛ زیرا در موسیقی، الگوهای تکراری معمولاً وابسته به فاصله‌ی نسبی بین نت‌ها هستند، نه مکان مطلق. راه حل این موضوع این است که مدل به‌جای موقعیت مطلق، یاد بگیرد که چه فاصله نسبی بین دو توکن وجود دارد. این کار با اضافه کردن biasهای قابل یادگیری برای موقعیت‌های نسبی به attention انجام می‌شود.

همچنین، در مدل Music Transformer، یکی از مشکلات مهم در پیاده‌سازی Relative Self-Attention، مصرف بالای حافظه بوده. نویسندگان مقاله راهکاری دقیق و مؤثر برای کاهش مصرف حافظه ارائه داده‌اند که در ادامه به آن می‌پردازیم:

شرح مسئله:

در attention نسبی، برای هر موقعیت i ر دنباله، مدل باید موقعیت نسبی‌اش را با تمام موقعیت‌های j (پیشین و پسین) محاسبه کند. در پیاده‌سازی مستقیم، ماتریس attention باید برای همه جفت‌های موقعیتی (i,j) محاسبه و ذخیره شود.

ین باعث می‌شود پیچیدگی حافظه به‌صورت O(L^2D) باشد، که در آن L، طول توالی (تعداد توکن‌ها) و D بُعد بردار ویژگی‌ها می‌باشد. در نتیجه، برای توالی‌های موسیقی با چند هزار رویداد، این مقدار حافظه غیرعملی می‌شود.

روش حل: پیاده‌سازی فشرده با ساختار شیفت‌یافته (skewing)

ایده این راهکار این است که به‌جای تولید مستقیم یک ماتریس ۲بُعدی بزرگ برای موقعیت‌های نسبی، مدل ابتدا بردارهای attention را برای تمام فاصله‌های نسبی ممکن یاد می‌گیرد. سپس با استفاده از عملیات شیفت ماتریس (skewing) و broadcasting، بردارهای attention به‌صورت هوشمند در مکان درست قرار می‌گیرند. در نتیجه، دیگر نیازی به ساخت ماتریس بزرگ دوبُعدی نیست. بنابراین محاسبه attention با موقعیت نسبی در عمل به پیچیدگی خطی O(LD) کاهش می‌یابد.

این روش اجازه می‌دهد Music Transformer دنباله‌هایی بسیار طولانی (مثلاً بالای ۲۰۰۰ رویداد) را با مصرف حافظه محدود آموزش دهد.

نکات دیگری نیز درباره این مدل وجود دارد که شامل این موارد می‌باشند:

۱- مدل یاد می‌گیرد که در هر گام، توکن بعدی را پیش‌بینی کند. این کار با تابع هزینه Cross-Entropy Loss بین توکن واقعی و پیش‌بینی‌شده صورت می‌گیرد.

۲- در این مدل از دیتاست‌های MIDI استفاده شده که در ادامه بیش‌تر به آن می‌پردازیم.

۳- مراحل آموزش این مدل بدین شکل است: یادگیری توالی کامل موسیقی انجام می‌شود، سپس مدل بدون نیاز به اطلاعات هارمونیک یا تئوریک، فقط روی دنباله توکن‌ها آموزش می‌بیند، و در نهایت برای افزایش کیفیت از تکنیک‌های teacher forcing و dropout استفاده می‌شود.

۴- فرآیند تولید موسیقی نیز بدین شکل است که مدل توکن‌ها را یکی‌یکی تولید می‌کند، و توالی خروجی به‌راحتی قابل تبدیل به فایل MIDI و سپس به فایل صوتی است.

۵- مدل می‌تواند قطعات موسیقی منسجم، بلندمدت، و با ساختار تکرارشونده تولید کند، چیزی که در RNNها و LSTMها به دلیل محدودیت حافظه دشوار است.

مزایای مدل Music Transformer:

مدل Music Transformer توانسته است با بهره‌گیری از Relative Self-Attention ، وابستگی‌های بلندمدت و تکرارهای ساختاری در موسیقی را به‌خوبی مدل‌سازی کند. این ویژگی باعث می‌شود مدل بتواند قطعاتی تولید کند که دارای انسجام موسیقایی، جمله‌بندی مشخص، و همراهی هارمونیک طبیعی باشند.

همچنین، برخلاف RNNها و LSTMها که به‌دلیل حافظه محدود در پردازش دنباله‌های بلند ناتوان هستند، این مدل با بهینه‌سازی مصرف حافظه به O(LD)، امکان پردازش و تولید دنباله‌هایی با هزاران رویداد موسیقایی را فراهم می‌کند. پیاده‌سازی ساده‌تر، موازی‌سازی بهتر در GPU و کیفیت تولید بالا از دیگر مزایای این مدل هستند.

معایب:

با وجود دقت و قدرت تولید بالا، Music Transformer دارای محدودیت‌هایی نیز هست. نخست اینکه، مدل به‌صورت کاملاً داده‌محور آموزش می‌بیند و فاقد درک صریح از قوانین هارمونی یا ساختار موسیقی کلاسیک است؛ بنابراین در مواردی ممکن است نتایج غیرمعتبر یا ناموزون تولید کند. دوم اینکه به دلیل وابستگی به داده‌های MIDI، عملکرد مدل به کیفیت و تنوع این داده‌ها وابسته است و در سبک‌های کمتر دیده‌شده ممکن است خروجی ضعیف‌تر باشد. همچنین، تولید مرحله‌ای توکن‌ها در زمان inference هنوز نسبتاً کند است و نیاز به تنظیم دقیق (مثل تنظیم طول توالی یا early stopping) برای تولید موسیقی با کیفیت دارد.

مجموعه داده:

در این مقاله، برای آموزش و ارزیابی مدل از مجموعه داده‌های مختلف موسیقی به‌صورت MIDI استفاده شده است. این مجموعه داده‌ها هم شامل موسیقی کلاسیک ساختارمند هستند و هم قطعات پیچیده‌تری از موسیقی مدرن.

مجموعه داده‌های استفاده‌شده:

۱- JSB Chorales Dataset:

-شامل موسیقی چهارصدایی نوشته‌ی باخ (Bach Chorales).

-این مجموعه داده ساختار موسیقایی منظم و جمله‌بندی واضح دارد و دنباله‌ها نسبتاً کوتاه‌تر هستند.

-دیتاست فوق، برای مدل‌سازی دقیق وابستگی‌های هارمونیک و بررسی توانایی مدل در حفظ ساختار ملودیک استفاده شده است.

۲- MuseNet Internal Dataset

-مجموعه‌ای بزرگ از داده‌های MIDI با طول بالا که حاوی موسیقی‌هایی از سبک‌ها و سازهای مختلف است.

-شامل داده‌هایی با همراهی، تکرار بخش‌های صوتی، و تغییرات دینامیکی زیاد است. تغییرات دینامیکی به معنای تغییر در بلندی و آرومی صدا می‌باشد.

آماده‌سازی داده:

تمامی فایل‌های MIDI به فرمت Event-based تبدیل شده‌اند، و پس از تبدیل، داده‌ها به توکن‌های گسسته برای استفاده در مدل Transformer آماده شده‌اند. دنباله‌ها برای آموزش با طول‌های نسبتاً بلند نگه داشته شده‌اند تا ساختار بلندمدت حفظ شود.

برخلاف بسیاری از مقالات تشخیص آکورد که داده‌های صوتی مثل Isophonics یا RWC را استفاده می‌کنند، این مقاله فقط روی داده‌های MIDI تمرکز کرده است. از آن جایی که ورودی مدل موسیقی رویدادی است، این نوع داده به‌صورت طبیعی مناسب مدل‌های زبانی مثل Transformer است. همچنین، در آموزش مدل، از teacher forcing و تکنیک‌های regularization برای جلوگیری از یادگیری نادرست استفاده شده است.

نتایج:

در این مقاله عملکرد مدل به صورت کیفی (perplexity) و کمی (ارزیابی انسانی) در مقایسه با مدل‌های پیشین مانند LSTM و Transformer استاندارد تحلیل شده است:

نتایج کمی (Quantitative Evaluation):

-معیار: Perplexity

Perplexity معیاری رایج در مدل‌های زبانی است که نشان می‌دهد مدل چقدر در پیش‌بینی توکن بعدی دچار عدم قطعیت است. در این معیار هرچه عدد پایین‌تر باشد یعنی مدل دقیق‌تر است.

در جدول (۷) نتایج ارزیابی مدل دیده می‌شود.

جدول ۷ - نتایج ارزیابی مدل تولید موسیقی با Transformer

مدل	داده Chorales	داده‌های طولانی (Internal)
LSTM	6.67	
Transformer پایه	5.44	3.77
Music Transformer	5.20	3.31

در نتیجه، مدل Music Transformer با attention نسبی (Rel) بهترین عملکرد را از نظر perplexity دارد.

نتایج کیفی (Qualitative Evaluation):

-روش ارزیابی انسانی

-از شنوندگان موسیقی خواسته شد تا سه خروجی بدون برچسب را از سه مدل مختلف گوش کنند و به آن‌ها از یک تا سه امتیاز دهند.

نتیجه میانگین نمرات در این جدول دیده می‌شود:

جدول ۸ - نتیجه میانگین نمرات در روش ارزیابی انسانی مدل تولید موسیقی با Transformer

مدل	میانگین نمره انسانی
LSTM	1.53
Transformer پایه	۲.۰۵
Music Transformer	۲.۴۲

مدل Music Transformer نه‌تنها از نظر آماری بهتر است، بلکه از دیدگاه شنوندگان نیز موسیقی منسجم‌تر، طبیعی‌تر و زیباتر تولید می‌کند.

مدل‌های قبلی مانند LSTM یا Transformer مطلق، موسیقی‌هایی تولید می‌کردند که پس از چند ثانیه به‌هم‌ریخته یا تصادفی می‌شدند. اما Music Transformer قادر است موتیف‌های تکراری، فراز و فرود ملودیک، همراهی و حتی الگوهای چندبخشی را حفظ کند. این ویژگی‌ها به‌خصوص در خروجی‌های طولانی کاملاً مشهود است.

در نتیجه، مدل پیشنهادی (Music Transformer با attention نسبی) در هر دو روش ارزیابی نسبت به مدل‌های قبلی عملکرد بهتری دارد. این به‌دلیل توانایی مدل در یادگیری وابستگی‌های بلندمدت و تکرارهای ساختاری در موسیقی است که در مدل‌های دیگر محدود بود.

۳-۲ جمع‌بندی

در یک نگاه کلی، مسیر پیشرفت پژوهش‌ها در حوزه‌ی تشخیص خودکار آکورد از روش‌های ساده‌ی آماری به سمت مدل‌های عمیق و پیچیده حرکت کرده است. در ابتدا، سیستم‌ها بر پایه‌ی ویژگی‌های دستی مانند کرومای کلاسیک و مدل‌های احتمالاتی مثل HMM طراحی می‌شدند که اگرچه ساده بودند اما دقت پایینی داشتند. سپس، شبکه‌های کانولوشنی (CNN) معرفی شدند که ویژگی‌ها را مستقیماً از طیف‌نگار صوتی یاد می‌گرفتند و نیاز به مهندسی دستی را حذف کردند، هرچند محدود به ساده‌سازی برچسب‌ها (ماژور/مینور) بودند. پس از آن، مدل‌های ترکیبی DNN+RNN مطرح شدند که علاوه بر یادگیری ویژگی‌های آکوستیک، توالی آکوردها را نیز با LSTM مدل کردند و با الگوریتم‌هایی مثل Beam Search و نسخه‌ی بهینه‌تر آن (Hashed Beam Search) دقت و کارایی بالاتری یافتند. در ادامه، رویکردهای کاملاً کانولوشنی همراه با CRF ارائه شد که پیوستگی زمانی خروجی‌ها را بهبود دادند. برای گسترش واژگان آکورد (مثلاً آکوردهای هفتم و وارونه)، مدل‌های مبتنی بر داده‌های MIDI + BLSTM-CRF مطرح شدند که توانستند تا بیش از ۱۸۰ نوع آکورد را تشخیص دهند. نهایتاً، ورود Transformerها با توجه نسبی، امکان مدل‌سازی وابستگی‌های بلندمدت در موسیقی و تولید ساختارهای منسجم‌تر را فراهم کرد. به‌طور کلی، نقاط قوت این مسیر شامل حذف ویژگی‌های دستی، یادگیری انتهابه‌انتها، درک بهتر توالی زمانی و توانایی پشتیبانی از واژگان بزرگ است؛ در حالی که نقاط ضعف اصلی همچنان شامل نیاز به داده‌ی برچسب‌خورده‌ی زیاد، ساده‌سازی بیش‌ازحد برچسب‌ها، مشکل در مدل‌سازی دقیق طول آکورد و دشواری در تعمیم به سبک‌های متنوع می‌باشد.

در فصل بعد، روش پیاده‌سازی مدلی مبتنی بر معماری Bi-directional Transformer، که عملکرد بهتری از مدل‌های پیش از خود دارد ارائه می‌شود و نتایج ارزیابی آن و مقایسه‌اش با مدل‌های ذکر شده بررسی می‌شود.

فصل چهارم: روش پیشنهادی و نتیجه‌گیری

٤-١ مقدمه

در این پروژه، با استفاده از معماری Bi-directional Transformer، مدلی طراحی شده که می‌تواند روی مجموعه‌داده ورودی آموزش ببیند و پس از آن، به ازای فایل موسیقی ورودی، لیستی از آکوردهای آن قطعه موسیقی را به عنوان خروجی مدل ارائه دهد.

در این فصل، ابتدا ساختار این مدل بررسی می‌شود. سپس مراحل طی شده جهت پیاده‌سازی این پروژه، از مرحله جمع‌آوری داده تا بدست آوردن خروجی نهایی شرح داده می‌شود و در نهایت، معیارهای ارزیابی و مجموعه‌دادگان مورد استفاده بیان شده و نتیجه نهایی مدل و دستاوردهای پژوهش تحلیل می‌شود.

٤-٢ ساختار روش پیشنهادی

روش پیشنهادی این پژوهش با هدف بهبود فرآیند تشخیص خودکار آکورد طراحی شده است. ساختار کلی این روش بر پایه معماری Bidirectional Transformer است که امکان مدل‌سازی وابستگی‌های بلندمدت میان دنباله‌های زمانی را فراهم می‌سازد.

١-١-١ مزایای روش نسبت به کارهای پیشین:

١) درک بهتر وابستگی‌های بلند مدت آکوردها به یکدیگر: CNNها عمدتاً محلی‌اند و برای دیدن زمینه‌ی بلندمدت باید لایه‌ها یا هسته‌های آن‌ها زیاد شوند؛ RNN/LSTMها هم وابسته به پردازش ترتیبی و مستعد محوشدن گرادیان‌اند. اما Self-Attention در ترنسفورمر، وابستگی بین هر دو فریم را با مسیر محاسباتی کوتاه و به‌صورت موازی مدل می‌کند. همچنین، یک تفاوت مهم این transformer با مدل اولیه ارائه شده در مقاله اصلی، Bidirectional بودن آن است؛ که یعنی مدل، برای تصمیم‌گیری درباره‌ی فریم t، هم از گذشته‌ی نزدیک/دور و هم از آینده‌ی نزدیک/دور همان قطعه بهره می‌گیرد. بنابراین نسخه‌ی Bidirectional به‌طور خاص در تشخیص آکورد مفید است؛ زیرا همان طور که در فصل دوم توضیح داده شد، توالی‌های هارمونیک غالباً به آکوردهای قبل و بعد وابسته‌اند.

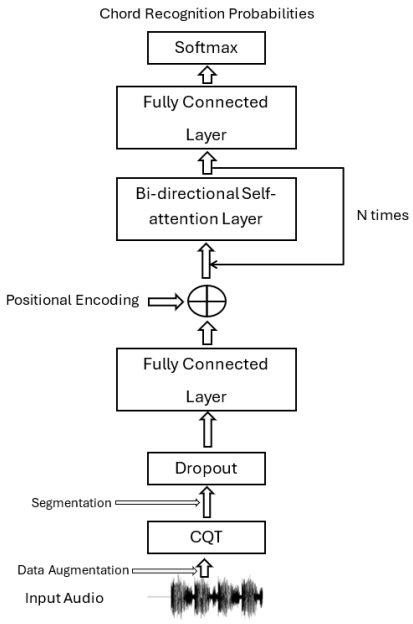
۲) کاهش خطاهای Fragmentation در مرز آکوردها: به‌دلیل مشاهده‌ی هم‌زمان گذشته و آینده، مدل مرزهای واقعی آکورد را بهتر تمایز می‌دهد و احتمال پرش بی‌دلیل از یک آکورد به آکورد دیگر کاهش می‌یابد. اهمیت این موضوع به این علت است که در موسیقی، برخی از نت‌ها در آن فریم زمانی خاص، مربوط به آن آکورد نیستند و این موضوع ممکن است در مدل‌های دیگر باعث تشخیص غلط آکورد در آن لحظه شود. اما این مدل به علت نگاه همزمان کوتاه‌مدت و بلندمدت خود احتمال خطا را کاهش می‌دهد. (CQT نیز با هم‌تراز کردن محورِ فرکانس با فواصل موسیقایی به این پایداری کمک می‌کند.)

۳) موازی‌سازی و پایداری آموزش: بر خلاف RNNها که ذاتاً دنباله‌وار آموزش می‌بینند، ترنسفورمر کل توالی را به‌صورت موازی پردازش می‌کند و با Residual Connections و Layer Normalization آموزش عمیق و پایدار دارد؛ بنابراین هم سرعت آموزش بیشتر است و هم تنظیم ابرپارامترها ساده‌تر.

۴) انعطاف معماری و تکرارپذیری بالا: در این معماری، همین ساختار را می‌توان با اندازه‌گیری sliding window تعداد لایه‌ها/Headها و ابعاد نهفته، برای سناریوهای آفلاین یا نزدیک به‌زمان‌واقعی تنظیم کرد؛ همچنین به‌سادگی با داده‌های دیگر (یا حتی داده‌های نمادین) سازگار می‌شود و قابل بازتولید در فریم‌ورک‌های مختلف است.

۱-۲-۴ معماری کلی مدل

در ادامه این بخش ابتدا نمای کلی سیستم توضیح داده می‌شود و سپس هر جزء اصلی آن به‌صورت جداگانه معرفی خواهد شد. در شکل زیر، ساختار کلی روش پیشنهادی نشان داده شده است:



شکل ۹ – معماری کلی مدل BTC

همان‌طور که در نمودارها مشخص است، ابتدا فرآیند Data Augmentation روی داده خام اولیه انجام می‌شود. سپس داده افزایش یافته وارد الگوریتم CQT می‌شود تا از حالت خام اولیه به یک بردار ویژگی (فرکانس) تبدیل شود، تا مدل بتواند با این بردار آموزش ببیند. پس از آن مرحله Segmentation اتفاق می‌افتد، و سپس بعد از لایه Fully connected، ترتیب فریم‌ها در مرحله Positional encoding مدل می‌شود تا ویژگی متوالی بودن آکوردها حفظ شود. سپس داده وارد بخش اصلی معماری، یعنی لایه‌های Bi-directional Self-attention می‌شود و نتیجه تخمین خروجی بعد از یک لایه Fully connected دیگر و SoftMax، به عنوان خروجی قرار می‌گیرد. در ادامه، اجزای مختلف این معماری شرح داده شده است.

پیش‌پردازش داده ورودی:

Data Augmentation:

در مرحله آموزش، به منظور افزایش تنوع داده و بهبود تعمیم‌پذیری مدل، از تکنیک افزایش داده مبتنی بر تغییر Pitch (Pitch-shifting) استفاده شده است. در این روش، سیگنال صوتی خام به‌صورت تصادفی بین ۵ نیم‌پرده پایین‌تر تا ۶

نیم‌پرده بالاتر شیفت داده می‌شود و در همان حال برچسب‌های آکورد نیز متناسب با این تغییر بازنویسی می‌شوند (برای مثال، آکورد C:maj پس از تغییر دو نیم‌پرده‌ای به D:maj تبدیل می‌شود). این کار باعث می‌شود که مدل در مواجهه با تفاوت‌های اجرای قطعات مقاومت بیشتری پیدا کند. پس از این تغییر، داده‌های افزوده‌شده دقیقاً همان مسیر پیش‌پردازش اصلی را طی می‌کنند. بدین ترتیب، فرآیند افزایش داده تنها در مرحله آموزش به کار می‌رود و داده‌های اعتبارسنجی و آزمون بدون هیچ‌گونه تغییر مورد استفاده قرار می‌گیرند.

Data Segmentation:

از آنجایی که مدل نمی‌تواند یک فایل صوتی کامل را پردازش کند، سیگنال صوتی ابتدا به بخش‌های کوتاه‌تر تقسیم می‌شود تا امکان پردازش مؤثرتر فراهم گردد. در این پژوهش هر قطعه صوتی به بخش‌های ۱۰ ثانیه‌ای تقسیم شده و بین هر دو بخش، ۵ ثانیه هم‌پوشانی در نظر گرفته می‌شود. این هم‌پوشانی باعث می‌شود که مدل در نواحی مرزی آکوردها اطلاعات کافی از قبل و بعد داشته باشد و قطع یا از دست رفتن داده در نقاط تغییر آکورد به حداقل برسد. علاوه بر این، تقسیم‌بندی ثابت موجب می‌شود که ورودی‌ها طول یکنواختی داشته باشند و برای پردازش دسته‌ای (Batch Processing) مناسب باشند. این مرحله به‌عنوان بخشی از فرآیند پیش‌پردازش، پایه‌ی تولید نمایش طیفی (CQT) و در نهایت آموزش مدل تشخیص آکورد را تشکیل می‌دهد.

تبدیل CQT (Constant-Q Transform):

در CQT سیگنال به باندهای فرکانسی تقسیم می‌شود که به‌صورت لگاریتمی روی محور فرکانس چیده شده‌اند (مشابه نت‌های موسیقی). این باندها در فرکانس‌های پایین باریک‌تر هستند در نتیجه وضوح فرکانسی بالاتر است. در فرکانس‌های بالا هم باندها پهن‌تر هستند و این یعنی وضوح زمانی بالا می‌باشد.

در این پژوهش، پارامترهای CQT به این صورت تنظیم شده‌اند: سیگنال صوتی پس از نمونه‌برداری در ۲۲,۰۵۰ هرتز به قطعات ۱۰ ثانیه‌ای با ۵ ثانیه هم‌پوشانی تقسیم می‌شود (Segmentation). سپس برای هر قطعه، CQT روی ۶ اکتاو از نت C1 (≈32.7 Hz) محاسبه می‌شود و در هر اکتاو ۲۴ bin در نظر گرفته می‌شود؛ در نتیجه هر فریم دارای ۱۴۴ bin فرکانسی خواهد بود. hop size برابر ۲۰۴۸ نمونه است که معادل ≈۹۳ میلی‌ثانیه در نرخ ۲۲.۰۵ kHz است؛ بنابراین هر

قطعه ۱۰ ثانیه‌ای شامل حدود ۱۰۸ فریم می‌شود و در نهایت شکل ورودی به مدل به صورت ماتریسی با ابعاد تقریبی (۱۰۸ × ۱۴۴) خواهد بود. دامنه طیف محاسبه‌شده با تابع لگاریتمی به مقیاس log-magnitude برده شده و برای جلوگیری از ناپایداری یک ε کوچک اضافه می‌شود. در نهایت، برای یکنواخت‌سازی داده‌ها، روی تمام داده‌های آموزشی میانگین و واریانس جهانی محاسبه و به‌عنوان معیار z-normalization استفاده می‌شود؛ این مقادیر سپس برای نرمال‌سازی داده‌های اعتبارسنجی و آزمون نیز به کار می‌روند. این ترکیب از پارامترها تعادلی میان وضوح زمانی ($\approx$۹۳ ms) و وضوح فرکانسی (۲۴ بین/اکتاو) برقرار می‌کند و اطلاعات لازم برای تشخیص تغییرات آکورد را به‌خوبی فراهم می‌آورد.

خلاصه نمایش ورودی:

- نرخ نمونه‌برداری (Sample Rate): یعنی تعداد نمونه‌های صوتی در هر ثانیه، که در این پژوهش عدد استاندارد 22,050 Hz برای این پارامتر انتخاب شده است.
- Hop Size: فاصله‌ی زمانی بین دو فریم که به صورت متوالی تحلیل می‌شوند؛ در نرخ ۲۲,۰۵۰ هرتز، ۲۰۴۸ نمونه معادل حدود ۰.۰۹۳ ثانیه است؛
- محدوده اکتاوهای مورد بررسی از C1 تا C7 انتخاب شده که می‌شود شش اکتاو. این محدوده به خوبی نت‌های بم و زیر موسیقی را در بر می‌گیرد.
- تعداد bin در هر اکتاو: تعداد تقسیمات هر اکتاو را مشخص می‌کند. از آنجایی که هر اکتاو ۱۲ نت دارد، در این پژوهش از ۲۴ بین در هر اکتاو استفاده شده است (نیم‌پرده تقسیم‌شده به دو بخش) تا تمایزهای ظریف‌تر مثل تفاوت ماژور و مینور بهتر تشخیص داده شوند.
- Feature dimension: تعداد کل بین‌های فرکانسی در هر فریم؛ هر اکتاو ۲۴ عدد bin دارد و در کل ۶ اکتاو در نظر گرفته شده است، پس تعداد کل binها می‌شود ۱۴۴ عدد.
- مقیاس‌بندی لگاریتمی دامنه (Log Amplitude Scaling): شدت صدا (Amplitude) دامنه‌ی بسیار وسیعی دارد (صداهای خیلی بلند یا خیلی آرام). گرفتن لگاریتم باعث فشرده‌سازی این بازه می‌شود و مقایسه

هارمونیک‌ها آسان‌تر خواهد شد. همچنین با نحوه‌ی درک شنیداری انسان هم‌خوانی دارد، چرا که ما بلندی صدا را به صورت لگاریتمی درک می‌کنیم.

- Z-Normalization: ابتدا میانگین (μ) و انحراف معیار (σ) روی داده‌های آموزش محاسبه می‌شود. سپس هر مقدار با روش Z-Normalization نرمال‌سازی می‌شود تا ویژگی‌ها میانگین صفر و واریانس یک داشته باشند و تفاوت‌های ناشی از شدت صدای ضبط‌ها یا تفاوت در تنظیم موسیقی روی مدل اثر نگذارد.

مدل BTC

پس از اتمام مراحل پیش‌پردازش داده، وارد بخش اصلی معماری مدل می‌شویم. ورودی این بخش در هر قطعه ۱۰ ثانیه‌ای، دنباله‌ای از فریم‌های ۱۴۴ بعدی به طول تقریبی ۱۰۸ می‌باشد.

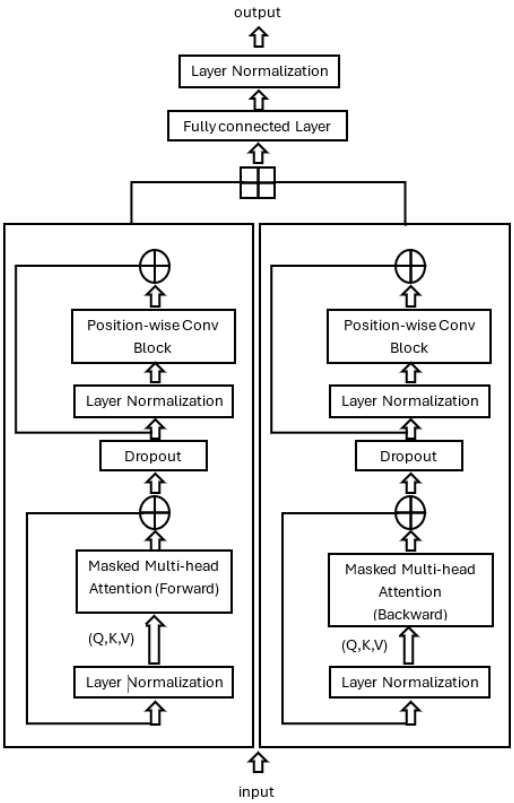
Positional encoding:

در این معماری، همانند ترنسفورمر اصلی، از کدگذاری موقعیتی سینوسی (Sinusoidal Positional Encoding) برای نمایش ترتیب فریم‌ها استفاده می‌شود. از آنجا که مکانیزم Self-Attention ذاتاً نسبت به ترتیب داده‌ها بی‌تفاوت است، لازم است اطلاعات موقعیت زمانی به بردار ویژگی‌ها اضافه شود تا مدل بتواند توالی فریم‌های CQT را درک کند. برای این منظور، به هر بردار ورودی قبل از ورود به اولین لایه توجه، یک بردار موقعیتی با استفاده از توابع سینوس و کسینوس با فرکانس‌های متفاوت اضافه می‌شود. این کدگذاری به مدل امکان می‌دهد که هم موقعیت مطلق هر فریم و هم روابط نسبی میان فریم‌ها را بیاموزد و در نتیجه بتواند تغییرات هارمونیک و مرزهای آکوردها را دقیق‌تر تشخیص دهد.

Bi-directional Self-attention Layers:

در معماری BTC، یکی از بخش‌های کلیدی مدل، لایه‌ی Bi-directional Masked Multi-head Self-Attention است که به‌صورت تخصصی برای تحلیل دنباله‌های زمانی موسیقی طراحی شده است. به کمک این لایه‌ها مدل می‌تواند همزمان از اطلاعات گذشته و آینده برای تصمیم‌گیری در مورد آکورد هر فریم استفاده کند. این لایه در شبکه N بار پیمایش می‌شود.

معماری داخلی این‌لایه در شکل (۱۰) قابل مشهاده است:



شکل ۱۰ - Bi-directional Masked Multi-head Self-Attention

هر لایه شامل دو بلوک attention مجزا است: نخست، Forward-masked Attention که فریم جاری را قادر می‌سازد تنها به فریم‌های آینده نگاه کند؛ دوم، Backward-masked Attention که فریم جاری را محدود می‌کند تا تنها به فریم‌های گذشته رجوع نماید. خروجی این دو بلوک پس از محاسبه با یکدیگر ترکیب (Concatenate) شده و سپس توسط یک لایه خطی به بُعد اصلی مدل بازنگاشت می‌گردد. این طراحی باعث می‌شود که مدل بتواند به‌طور همزمان از سرنخ‌های هارمونیک پیشین و نشانه‌های موسیقایی آتی استفاده کند، مرز تغییرات آکورد را دقیق‌تر تشخیص دهد و وابستگی‌های کوتاه‌مدت و بلندمدت را به شکل متوازن در نظر بگیرد.

Position-wise convolutional block

در معماری BTC به جای استفاده از شبکه‌ی FFN در ترنسفورمر اصلی، از Position-wise Convolutional Block استفاده شده است. این بلوک شامل یک لایه Conv1D با اندازه کرنل ۳، گام ۱ و پدینگ ۱ است که باعث می‌شود طول توالی ثابت بماند. پس از کانولوشن، یک تابع فعال‌سازی ReLU و سپس Dropout اعمال می‌شود.

هدف از این طراحی آن است که مدل علاوه بر وابستگی‌های بلندمدت که توسط لایه‌های Self-Attention پوشش داده می‌شوند، بتواند تغییرات محلی و کوتاه‌مدت در مرز آکوردها را نیز در نظر بگیرد. در واقع، این بلوک کمک می‌کند که پیش‌بینی‌های فریم‌به‌فریم روان‌تر شوند، مرز بین آکوردها دقیق‌تر شناسایی شود و نویز یا پرش‌های ناگهانی در برچسب‌گذاری کاهش یابد. به این ترتیب، بلوک کانولوشنی نقش مکملی برای مکانیزم توجه دارد و به مدل اجازه می‌دهد همزمان هم همبستگی‌های کلی (global context) و هم نشانه‌های محلی (local context) را استخراج کند.

Residual, normalization, dropout:

در معماری BTC همانند ترنسفورمر استاندارد، هر زیربخش از لایه‌ها شامل یک Residual Connection، نرمال‌سازی لایه‌ای و Dropout است. Residual Connection کمک می‌کند که گرادیان‌ها در طول شبکه‌های عمیق پایدار بمانند و مشکل ناپدید شدن یا انفجار گرادیان کاهش یابد. لایه نرمال‌سازی نیز باعث تثبیت توزیع ویژگی‌ها در طول آموزش می‌شود و یادگیری سریع‌تر و پایدارتر را امکان‌پذیر می‌سازد. در نهایت، Dropout به‌عنوان یک روش منظم‌سازی (Regularization) برای جلوگیری از بیش‌برازش به‌کار می‌رود و با حذف تصادفی بخشی از واحدها در هنگام آموزش، شبکه را وادار می‌کند تا ویژگی‌های عمومی‌تر و مقاوم‌تری بیاموزد. این سه مؤلفه به‌طور هماهنگ باعث افزایش پایداری و قابلیت تعمیم مدل در فرآیند تشخیص آکورد می‌شوند.

۴-۳ پیاده‌سازی روش پیشنهادی

۴-۳-۱ جمع‌آوری مجموعه‌دادگان

جهت پیاده‌سازی مدل پیشنهادی، پس از مطالعه و پژوهش، فاز اول پیاده‌سازی، یعنی جمع‌آوری مجموعه داده‌ها جهت استفاده در آموزش و آزمون مدل انجام شد. همان‌طور که در بخش مجوزها نیز بیان شد، مجموعه داده‌های معتبر در زمینه

پژوهش‌های مرتبط با موسیقی، فقط شامل فایل‌های برچسب‌گذاری می‌باشند. بنابراین، ابتدا فایل‌های صوتی‌ای که در دسترس بودند تهیه شدند و سپس فایل‌های برچسب مربوط به هر قطعه، به صورت دستی با استفاده از دانش موسیقی بررسی شده و در صورت وجود مغایرت در فایل صوتی و فایل برچسب، همه موارد به دقت اصلاح شدند. همچنین، اصلاحاتی همچون تطبیق نام فایل‌های صوتی و فایل‌های برچسب‌گذاری نیز پیش از شروع کار با داده‌ها انجام شده‌است.

۴-۳-۲ بستر توسعه پروژه

به منظور پیاده‌سازی و آزمایش مدل پیشنهادی، پروژه بر پایه‌ی زبان Python توسعه داده شد. برای بخش‌های مختلف کار، از کتابخانه‌های متناسب استفاده گردید؛ به‌طور مشخص، PyTorch برای تعریف و آموزش شبکه‌های عصبی عمیق، Librosa برای پردازش سیگنال صوتی و محاسبه‌ی ویژگی‌های طیفی مانند CQT، و mir_eval برای محاسبه معیارهای ارزیابی استاندارد از جمله WCSR مورد استفاده قرار گرفتند. مدیریت و اجرای کدها در محیط‌های متنوعی انجام شد تا انعطاف‌پذیری بیشتری در توسعه و آزمایش فراهم گردد؛ از جمله محیط Visual Studio Code (VSCode) برای توسعه و ویرایش کد، Google Colab برای اجرای آزمایش‌ها در بستر ابری، و Anaconda در سیستم‌عامل ویندوز برای مدیریت بسته‌ها و اجرای محلی. علاوه بر این، از GitHub به‌عنوان مرجع نگهداری و نسخه‌بندی کدها استفاده شد تا امکان دسترسی سایر پژوهشگران و بررسی جزئیات پیاده‌سازی فراهم باشد.

۴-۳-۳ پیش‌پردازش داده‌ها

در مرحله‌ی پیش‌پردازش، داده‌های صوتی خام آماده‌سازی شدند تا به شکلی استاندارد و قابل استفاده برای مدل درآیند. ابتدا تمام فایل‌های صوتی به نرخ نمونه‌برداری ۲۲،۰۵۰ هرتز نمونه‌برداری شدند تا یکدستی داده‌ها تضمین گردد. سپس هر قطعه‌ی صوتی به بازه‌های زمانی ۱۰ ثانیه‌ای تقسیم شد که دارای ۵ ثانیه هم‌پوشانی بودند؛ این کار باعث شد مدل در مرز تغییر آکوردها اطلاعات کافی از قبل و بعد در اختیار داشته باشد.

برای استخراج ویژگی‌ها، از تبدیل CQT (Constant-Q Transform) استفاده شد که متناسب با ساختار هارمونیک موسیقی عمل می‌کند. در این تبدیل، بازه‌ی ۶ اکتاو از نت C1 تا C7 با تفکیک ۲۴ بین در هر اکتاو پوشش داده شد که در مجموع ۱۴۴ بعد فرکانسی در هر فریم تولید می‌کند. اندازه‌ی گام (hop size) برابر ۲۰۴۸ نمونه (≈ ۹۳ میلی‌ثانیه) در نظر گرفته شد که منجر به حدود ۱۰۸ فریم برای هر بازه‌ی ۱۰ ثانیه‌ای شد.

```
if self.feature_name == FeatureTypes.cqt:
    # print(pid, "make feature")
    feature = librosa.cqt(song_seq, sr=sr, n_bins=feature_config['n_bins'],
                          bins_per_octave=feature_config['bins_per_octave'],
                          hop_length=feature_config['hop_length'])
```

شکل ۱۱ – کد تبدیل CQT (Constant-Q Transform)

پس از محاسبه‌ی CQT، مقادیر طیفی با تابع لگاریتمی به مقیاس log-magnitude منتقل شدند تا دامنه‌ی دینامیکی داده‌ها فشرده شود. سپس برای حذف اثر تفاوت بلندی صدا و شرایط ضبط، تمام ویژگی‌ها با روش z-normalization نرمال شدند؛ بدین ترتیب که میانگین و انحراف معیار از داده‌های آموزش محاسبه شد و همین مقادیر برای نرمال‌سازی داده‌های اعتبارسنجی و آزمون به کار رفت.

در بخش آموزش، برای افزایش تنوع داده‌ها از روش افزایش داده (Data Augmentation) مبتنی بر تغییر گام (Pitch-shifting) استفاده شد. سیگنال‌های صوتی در بازه‌ی ۵ نیم‌پرده پایین‌تر تا ۶ نیم‌پرده بالاتر تغییر داده شدند و برچسب‌های آکورد نیز متناسب با این تغییر اصلاح شدند. بدین منظور، از کتابخانه‌ی pyrubberband برای تغییر گام (Pitch-shift) استفاده شده و برچسب‌های آکورد هم متناسب با این تغییر به‌روزرسانی می‌شوند.

```
# stretch original sound and chord info
x = pyrb.time_stretch(original_wav, sr, stretch_factor)
x = pyrb.pitch_shift(x, sr, shift_factor)
audio_length = x.shape[0]
chord_info['start'] = chord_info['start'] * 1/stretch_factor
chord_info['end'] = chord_info['end'] * 1/stretch_factor
```

شکل ۱۲ – بخشی از کد Data Augmentation

۴-۳-۴ آموزش مدل

پس از آماده‌سازی داده‌ها، پیاده‌سازی و اجرای فاز آموزش مدل آغاز شد. برای آموزش مدل، پیاده‌سازی در قالب یک حلقه‌ی آموزشی (Training Loop) انجام شد که مراحل زیر را شامل می‌شود:

تعریف مدل:

ابتدا ساختار مدل BTC پیاده‌سازی شد. بدین منظور، از یکسری مدل‌های پایه و کتابخانه‌های مختلف استفاده شده و بخش‌های مختلف آن پیاده‌سازی شد. این بخش‌ها شامل لایه‌های Bi-directional Masked Multi-head Attention (برای نگاه به گذشته و آینده)، بلوک‌های کانولوشن یک‌بعدی (Conv1D)، و لایه‌ی خروجی Fully Connected با ابعاد برابر تعداد کلاس‌ها (۲۵ یا ۱۷۰) می‌شوند. تابع بهینه‌ساز Adam و تابع هزینه Cross-Entropy loss نیز برای فرایند آموزش پیاده‌سازی شد. در شکل (۱۳) بخشی از کد BTC مشاهده می‌شود.

```python
class self_attention_block(nn.Module):
    def __init__(self, hidden_size, total_key_depth, total_value_depth, filter_size, num_heads,
                 bias_mask=None, layer_dropout=0.0, attention_dropout=0.0, relu_dropout=0.0, attention_map=Fals
        super(self_attention_block, self).__init__()

        self.attention_map = attention_map
        self.multi_head_attention = MultiHeadAttention(hidden_size, total_key_depth, total_value_depth,hidden_s
        self.positionwise_convolution = PositionwiseFeedForward(hidden_size, filter_size, hidden_size, layer_co
        self.dropout = nn.Dropout(layer_dropout)
        self.layer_norm_mha = LayerNorm(hidden_size)
        self.layer_norm_ffn = LayerNorm(hidden_size)

    def forward(self, inputs):
        x = inputs

        # Layer Normalization
        x_norm = self.layer_norm_mha(x)

        # Multi-head attention
        if self.attention_map is True:
            y, weights = self.multi_head_attention(x_norm, x_norm, x_norm)
        else:
            y = self.multi_head_attention(x_norm, x_norm, x_norm)

        # Dropout and residual
        x = self.dropout(x + y)

        # Layer Normalization
        x_norm = self.layer_norm_ffn(x)

        # Positionwise Feedforward
        y = self.positionwise_convolution(x_norm)

        # Dropout and residual
        y = self.dropout(x + y)
```

شکل ۱۳ - بخشی از کد BTC

حلقه‌ی آموزش (Training Loop):

در فایل train.py، حلقه‌ی آموزش به شکل زیر پیاده‌سازی شده است:

- Forward Pass: عبور Batch از مدل و محاسبه‌ی خروجی SoftMax.

- Loss Calculation: محاسبه‌ی خطا بین خروجی مدل و برچسب مرجع هر فریم.
- Backward Pass: محاسبه‌ی گرادیان‌ها
- Optimization Step: به‌روزرسانی وزن‌ها
- Zero Gradients: صفر کردن گرادیان‌ها

اعتبارسنجی و توقف زودهنگام:

در پایان هر دوره (Epoch)، مدل روی داده‌های Validation ارزیابی می‌شود. در صورت عدم بهبود دقت به مدت ۱۰ دوره متوالی، آموزش متوقف می‌گردد (Early Stopping). همچنین در صورت افت عملکرد، نرخ یادگیری با ضریب ۰.۹۵ کاهش پیدا می‌کند (Learning Rate Decay).

ذخیره‌سازی مدل:

بهترین مدل بر اساس عملکرد روی مجموعه‌ی اعتبارسنجی ذخیره می‌شود تا در مرحله‌ی ارزیابی نهایی (Testing) مورد استفاده قرار گیرد.

```python
for epoch in range(restore_epoch, config.experiment['max_epoch']):
    # Training
    model.train()
    train_loss_list = []
    total = 0.
    correct = 0.
    second_correct = 0.
    for i, data in enumerate(train_dataloader):
        features, input_percentages, chords, collapsed_chords, chord_lens, boundaries = data
        features, chords = features.to(device), chords.to(device)

        features.requires_grad = True
        features = (features - mean) / std

        # forward
        features = features.squeeze(1).permute(0,2,1)
        optimizer.zero_grad()
        prediction, total_loss, weights, second = model(features, chords)

        # save accuracy and loss
        total += chords.size(0)
        correct += (prediction == chords).type_as(chords).sum()
        second_correct += (second == chords).type_as(chords).sum()
        train_loss_list.append(total_loss.item())

        # optimize step
        total_loss.backward()
        optimizer.step()

        current_step += 1

    # logging loss and accuracy using tensorboard
    result = {'loss/tr': np.mean(train_loss_list), 'acc/tr': correct.item() / total, 'top2/tr': (
    for tag, value in result.items(): tf_logger.scalar_summary(tag, value, epoch+1)
    logger.info("training loss for %d epoch: %.4f" % (epoch + 1, np.mean(train_loss_list)))
    logger.info("training accuracy for %d epoch: %.4f" % (epoch + 1, (correct.item() / total)))
```

شکل ۱۴ - بخشی از کد حلقه Training

۵-۳-۴ تست مدل و ارزیابی آن

در بخش تست (Inference و Evaluation) در این پروژه، مدل آموزش‌دیده روی داده‌های جدید اعمال شده و در نهایت، خروجی به‌صورت فایل برچسب تولید شد. به‌طور خلاصه، ابتدا مدل ذخیره‌شده (best_model.pth) در حالت eval بارگذاری شد و داده‌های صوتی مانند مرحله‌ی آموزش، پیش‌پردازش شدند (CQT با ۶ اکتاو، ۲۴ بین در هر اکتاو، hop=2048، log-scaling و z-normalization، اما بدون افزونگی داده (augmentation)). سپس هر کلیپ صوتی به مدل داده شده و خروجی SoftMax برای هر فریم به برچسب آکورد نگاشت شد و با هم‌پوشانی کلیپ‌ها میانگین‌گیری صورت گرفت تا دنباله‌ی نهایی ساخته شود. نتایج فریمی به بازه‌های زمانی پیوسته (lab files) تبدیل و در مسیر خروجی ذخیره شدند. برای ارزیابی، eval_chords_wcsr.py اجرا شد، فایل‌های مرجع و تخمینی با یکدیگر مقایسه شده و معیار Weighted Chord Symbol Recall (WCSR) محاسبه شد. این ارزیابی در قالب ۵-fold cross validation صورت گرفت و معیارهایی مانند Root، Thirds، Triads، Sevenths، Tetrads، Maj–Min و MIREX گزارش شده‌اند. بخشی از کدهای قسمت تست و ارزیابی در تصاویر زیر قابل مشاهده است:

```python
# Load model
if os.path.isfile(model_file):
    checkpoint = torch.load(model_file, map_location=torch.device('cpu'))
    mean = checkpoint['mean']
    std = checkpoint['std']
    model.load_state_dict(checkpoint['model'])
    logger.info("restore model")

# Audio files with format of wav and mp3
audio_paths = get_audio_paths(args.audio_dir)

# Chord recognition and save lab file
for i, audio_path in enumerate(audio_paths):
    logger.info("======== %d of %d in progress ========" % (i + 1, len(audio_paths)))
    # Load mp3
    feature, feature_per_second, song_length_second = audio_file_to_features(audio_path, config)
    logger.info("audio file loaded and feature computation success : %s" % audio_path)
```

شکل ۱۵ - کدهای قسمت تست و ارزیابی

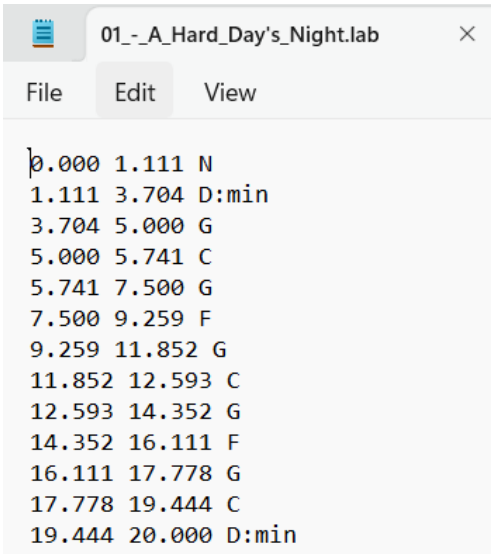
```python
for key, rpath, epath in pairs:
    try:
        ref_i, ref_l = load_lab_strict(rpath)
    except Exception as e:
        bad.append((key, "REF_PARSE", str(e))); continue
    try:
        est_i, est_l = load_lab_strict(epath)
    except Exception as e:
        bad.append((key, "EST_PARSE", str(e))); continue

    aligned = safe_align(ref_i, ref_l, est_i, est_l)
    if aligned is None:
        bad.append((key, "ALIGN", "no overlapping time span after clipping")); continue
    ref_i2, ref_l2, est_i2, est_l2 = aligned

    try:
        scores = mir_eval.chord.evaluate(ref_i2, ref_l2, est_i2, est_l2)
    except Exception as e:
        bad.append((key, "EVAL", str(e))); continue
```

شکل ۱۶ - کدهای قسمت تست و ارزیابی

در شکل زیر تصویری از بخشی از خروجی تولید شده برای یک قطعه دیده می‌شود. مدل در نهایت به ازای هر فایل صوتی، یک فایل برچسب و یک فایل midi که قابل شنیدن است و به ترتیب آکوردها در آن شنیده می‌شوند را به عنوان خروجی در یک فایل ذخیره می‌کند.



شکل ۱۷ - بخشی از خروجی تولید شده برای یک قطعه

٤-٤ روش ارزیابی

پس از پیاده‌سازی مدل، روش پیشنهادی با مجموعه داده‌های معتبر به وسیله معیارهای ارزیابی گوناگون سنجیده شده تا کارایی آن و میزان دقت آن مورد بررسی قرار بگیرد. در ادامه این مجموعه داده، نحوه تقسیم‌بندی آن برای آموزش و آزمون و همچنین معیارهای ارزیابی استفاده شده شرح داده می‌شود و پس از آن، نتایج بدست آمده از ارزیابی مدل بازگو می‌شود.

۴-۴-۱ مجموعه داده مورد استفاده

مجموعه داده مورد استفاده در این پژوهش، ترکیبی از چند مجموعه داده مختلف معتبر است که به طور معمول در زمینه پژوهش‌های تشخیص خودکار آکورد مورد استفاده قرار می‌گیرد. در این مجموعه داده‌ها، فایل‌های برچسب‌گذاری به صورت رایگان در اختیار پژوهش‌گران قرار داده شده، اما به علت قوانین copyright، تهیه فایل‌های صوتی مربوط به آن‌ها بر عهده خود پژوهش‌گران است.

فایل‌های برچسبی که در ارزیابی مدل ارائه شده استفاده می‌شوند، بر اساس استاندارد Harte طراحی شده‌اند که نخستین بار در مجموعه داده‌ی Isophonics معرفی شد. این فایل‌ها توسط افراد متخصص در حوزه موسیقی تهیه شده‌اند. در این استاندارد، هر فایل برچسب (معمولاً با پسوند .lab یا .txt) شامل سطرهایی است که هر سطر سه بخش دارد:

start_time end_time chord_label

این بخش‌ها به ترتیب زمان شروع و پایان و نام هر آکورد را نمایش می‌دهند. این استاندارد از یک گرامر متنی رسمی برای نمایش آکوردها استفاده می‌کند که شامل نت ریشه (Root)، کیفیت (Quality) یا همان نوع آکورد، نت باس (Bass) و گاهی افزودنی‌ها یا تغییرات است. مزیت این روش آن است که هم انسان و هم ماشین می‌توانند آن را به راحتی بخوانند و پردازش کنند. برای مثال:

```
0.000  2.345  N
2.345  5.678  C:maj
5.678  8.910  G:maj
8.910  12.000 A:min7
```

این یعنی از ثانیه ۰ تا ۲/۳۴۵ ثانیه، هیچ آکوردی شنیده نمی‌شود، سپس C ماژور، بعد G ماژور و سپس A مینور۷ شنیده می‌شود. توضیحات لازم برای آکوردها و انواع آن‌ها در فصل مفاهیم پایه شرح داده شده است.

Isophonics Dataset (۳۰۰ قطعه)

این مجموعه شامل قطعاتی از گروه‌هایی مانند The Beatles، Carole King، Queen و Zweieck است که به‌صورت دقیق و فریم‌به‌فریم برچسب‌گذاری شده‌اند. این مجموعه یکی از استانداردترین منابع در ارزیابی الگوریتم‌های تشخیص آکورد است. Isophonics Dataset دارای بخش‌های مختلف است، که شامل The Beatles (۱۸۰ قطعه)، Carole King، Queen و بخش‌های دیگر می‌شود. در ارزیابی این مدل از زیرمجموعه‌هایی از این بخش‌ها استفاده شده است.

Robbie Williams Dataset (۶۵ قطعه)

Robbie Williams Dataset یکی از دیتاست‌های خاص در حوزه‌ی تشخیص خودکار آکورد و تحلیل هارمونی موسیقی پاپ است. این مجموعه به طور ویژه شامل آثار خواننده بریتانیایی، Robbie Williams می‌باشد و با هدف بررسی عملکرد الگوریتم‌های تشخیص آکورد برای آثار یک هنرمند مشخص (single-artist dataset) ساخته شده است. این ویژگی که مجموعه برای کارهای یک هنرمند خاص است باعث می‌شود مدل‌های تشخیص آکورد روی رپرتوار یک هنرمند آموزش و آزمون شوند و امکان تحلیل سبک‌شناسی (stylistic analysis) فراهم گردد.

USPop Dataset (بخش انتخابی از USPop2002) (۲۰۰ قطعه)

مجموعه داده US Pop Dataset یکی از منابع مهم و پرکاربرد در حوزه تشخیص خودکار آکورد به شمار می‌رود. این مجموعه شامل ۱۹۴ قطعه موسیقی پاپ آمریکایی است که به‌صورت دستی و با دقت بالا توسط کارشناسان موسیقی آکوردگذاری شده‌اند. اهمیت این دیتاست در آن است که نمونه‌هایی از موسیقی پاپ معاصر ایالات متحده را در بر دارد و به همین دلیل مکمل ارزشمندی برای دیگر مجموعه‌های پرکاربرد مانند The Beatles Dataset (متمرکز بر آثار بیتلز در دهه ۶۰ میلادی) و RWC Pop Dataset (شامل قطعات ضبط‌شده در شرایط کنترل‌شده در سبک پاپ ژاپنی و غربی) محسوب می‌شود.

تقسیم داده‌ها برای آموزش و آزمون

برای اطمینان از تعمیم‌پذیری نتایج، از اعتبارسنجی ۵-بخشی (۵-fold cross validation) استفاده شده است. در این روش، کل داده‌ها به پنج بخش تقسیم می‌شوند و در هر بار چهار بخش برای آموزش و یک بخش برای آزمون در نظر گرفته می‌شود؛ نکته‌ی مهم این است که هیچ آهنگی همزمان در داده‌ی آموزش و ارزیابی ظاهر نمی‌شود تا از نشت داده جلوگیری گردد. بنابراین ۸۰ درصد داده برای آموزش و ۲۰ درصد برای آزمون استفاده می‌شود.

۲-۴-۴ معیارهای ارزیابی

در این پژوهش برای ارزیابی مدل معیارهای مختلفی استفاده شده که هر کدام برای ارزیابی عملکرد مدل در یک زمینه کاربرد دارند. این معیارها میزان دقت مدل را در تشخیص درست آکوردها بررسی می‌کنند.

(WCSR) Weighted Chord Symbol Recall

در این پژوهش برای ارزیابی عملکرد مدل از معیار Weighted Chord Symbol Recall استفاده شده است. این معیار نشان می‌دهد چه بخشی از مدت‌زمان کل قطعه‌ها به‌درستی برچسب‌گذاری شده است. فرمول آن به صورت زیر می‌باشد:

$$WCSR = tc/ta * 100(\%)$$

در این فرمول tc مجموع مدت‌زمان بخش‌هایی از موسیقی است که آکورد آن‌ها درست تشخیص داده شده است، و ta مدت زمان کل قطعه می‌باشد. به عبارت دیگر، WCSR نسبت زمان پیش‌بینی‌های صحیح مدل به کل زمان موسیقی را اندازه می‌گیرد.

برای محاسبه‌ی نمره‌ها، از کتابخانه‌ی استاندارد mir_eval استفاده شده است. خروجی مدل (پیش‌بینی آکوردها) به فایل‌های برچسب متنی (lab) تبدیل می‌شوند و سپس توسط mir_eval با برچسب‌های مرجع مقایسه می‌گردند.

برخی از معیارهایی که مبتنی بر WCSR هستند در ارزیابی این مدل استفاده شده‌اند. این معیارها شامل این موارد می‌شوند:

- ریشه (Root): مدل فقط باید نت ریشه (Root) آکورد را درست حدس بزند، بدون توجه به کیفیت (Maj, Min, 7 و ...).
- Thirds: هم ریشه و هم فاصله‌ی سوم (Major/Minor) درست باشند. این معیار حساس به تشخیص درست کیفیت ماژور یا مینور است.
- Triads: ریشه + سوم + پنجم درست باشند. یعنی آکورد در سطح تریاد (سه‌صدایی) درست تشخیص داده شود.
- Sevenths: ریشه + سوم + پنجم + هفتم درست باشند. برای آکوردهای هفت‌صدایی (مثل maj7، min7).
- Tetrads: تمام اجزای اصلی آکورد (ریشه، سوم، پنجم، هفتم) باید دقیقاً درست باشند. این معیار دقیق‌ترین و سخت‌گیرانه‌ترین معیار بین تمام موارد ذکر شد.
- Maj–Min: تمام آکوردها به ۲۵ کلاس ساده‌سازی می‌شوند (۱۲ ماژور، ۱۲ مینور و N). سپس دقت مدل روی این واژگان کوچک اندازه‌گیری می‌شود. این معیار به‌طور گسترده در رقابت‌های MIREX استفاده می‌شود.

۳-۴-۴ مجوزها

در مجموعه‌دادگان مورد استفاده در این پژوهش، فایل‌های برچسب‌گذاری به صورت رایگان در اختیار پژوهش‌گران قرار داده شده‌اند. اما فایل‌های صوتی قطعات مربوط به آن‌ها به علت copyright در این مجموعه‌ها قرار داده نشده‌اند و تهیه آن‌ها بر عهده پژوهشگران می‌باشد.

٤-٥ نتایج

در نتیجه ارزیابی مدل ارائه شده در این پژوهش روی مجموعه داده تست (بخشی از مجموعه داده isophonics)، دقت مدل در دو بخش مختلف بررسی شد. در بخش اول، آکوردها به ۲۵ کلاس خلاصه شدند (آکوردهای ماژور، مینور و بدون آکورد)، و در بخش دوم، ۱۷۰ کلاس آکورد که شامل آکوردهای پیچیده‌تر نیز می‌شوند مورد آزمون و ارزیابی قرار گرفتند. نتیجه ارزیابی دقت مدل در جداول زیر قابل مشاهده است.

جدول ۹ - نتایج ارزیابی مدل در حالت ۲۵ کلاس

معیار ارزیابی	mean	median
root	87.87	90.47
min/Maj	87.24	89.83

ردیف این جدول نشان می‌دهد که مدل در تشخیص نت پایه آکورد عملکرد بسیار خوبی نشان داده است. در ردیف دوم هم عملکرد مدل در تشخیص کیفیت یا همان نوع آکورد دیده می‌شود. از آن‌جایی که تفاوت minor یا Major بودن آکوردها صرفا به اندازه یک نیم‌پرده می‌باشد، تشخیص و تمایز آن‌ها برای مدل سخت‌تر است و دقت مدل در معیار min/Maj کمی پایین تر از معیار root می‌باشد. اما به طور کلی مدل در این زمینه عملکرد بسیار خوبی نشان داده‌است.

جدول ۱۰ - نتیجه ارزیابی مدل در حالت large vocabulary (۱۷۰ کلاس)

معیار ارزیابی	mean	median
root	87.22006741	89.84007253
thirds	85.63261304	88.32603172
triads	83.5436467	86.77759252
sevenths	77.60543022	81.52725856
tetrads	73.81340309	78.07449792
mirex	86.01105947	88.86405132
majmin	86.64584134	89.26407052

در سطر اول این جدول دیده می‌شود که مدل به صورت میانگین در تشخیص صحیح نت پایه دقت ۸۷.۲ درصد داشته. همچنین در سطر بعد هم دیده می‌شود مدل در تشخیص نت سوم آکورد نیز دقت بالایی دارد. در سطر بعد دقت مدل در تشخیص درست نت پایه، سوم و پنجم بررسی شده است. در این معیار، دقت کمتر از معیارهای قبلی است که طبیعی است، چون با افزایش پیچیدگی تعریف درست آکورد، احتمال خطا بیشتر می‌شود. اما مهم‌ترین و استانداردترین معیار، MIREX score هست، که با میانگین ۸۶ درصد، عملکرد خوب مدل را به نمایش می‌گذارد.

نکته قابل توجه در این دو جدول این است که میانه در همه‌ی معیارها کمی بالاتر از میانگین است. این موضوع نشان می‌دهد که اکثر آهنگ‌ها دقت خوبی دارند و چند نمونه‌ی با دقت پایین، میانگین را کاهش داده‌اند.

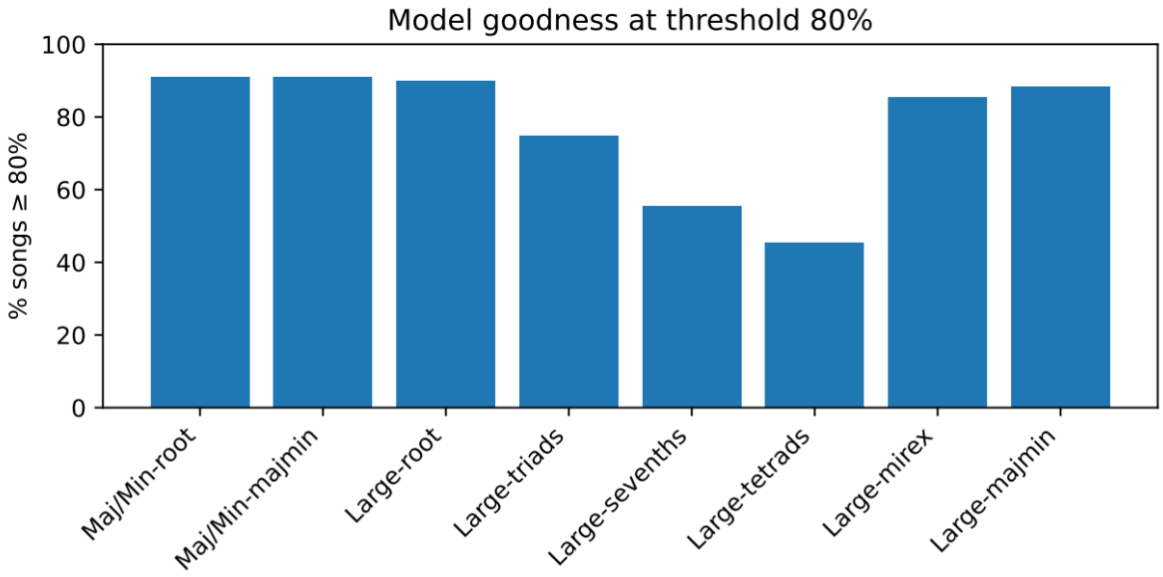
در جدول زیر نیز به طور کلی عملکرد مدل با مدل‌های پیش از خود مقایسه شده است:

جدول ۱۱ - جدول مقایسه‌ی عملکرد مدل با مدل‌های قبلی (WCSR ٪)

مدل	Root	Maj/Min	MIREX	توضیحات
مدل ارائه شده در حالت Maj/Min	87.9	87.2	-	-
مدل در حالت Large Vocab	87.2	86.6	86.0	-
McFee & Bello (2017, CNN+CRF)	83-84	81-82	80-81	مدل کانولوشنی با CRF، MIREX
Sigtia et al. (2015, Hybrid RNN)	82-83	80-81	79-80	ترکیب CNN + RNN، روی Beatles
Cho & Bello (2019, Large-vocab CNN)	85-86	83-84	82-83	یادگیری ساختارمند برای واژگان بزرگ
Humphrey & Bello (2012, NMF)	75-77	72-74	70-72	روش‌های قدیمی‌تر مبتنی بر NMF

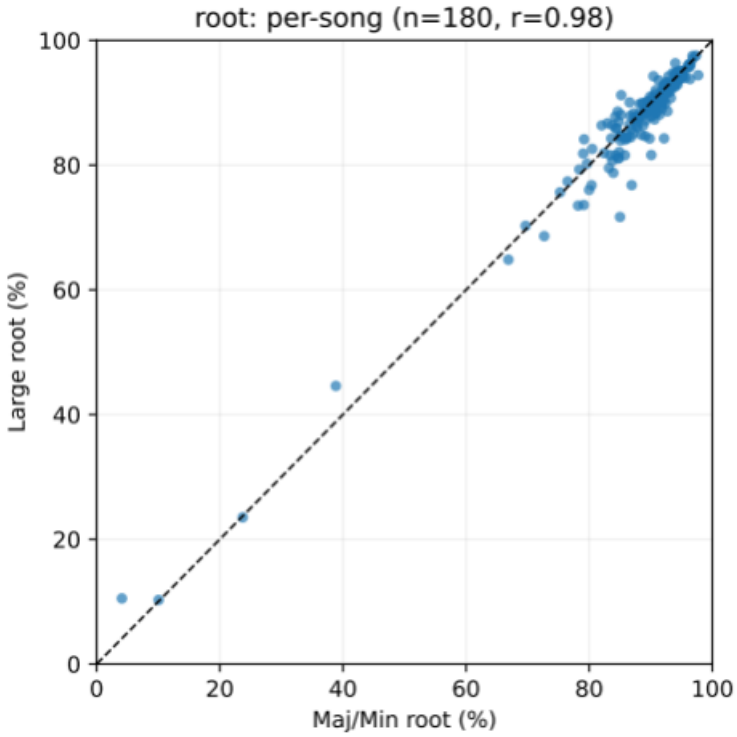
این جدول نشان می‌دهد که عملکرد مدل به طور کلی در همه حالات از مدل‌های قبل از خود بهتر است. همچنین، ثبات (میانه بالاتر از میانگین) در نتایج این مدل نشان می‌دهد که مدل روی اکثر قطعات خوب عمل کرده، در حالی که مقالات قبلی پراکندگی بیشتری داشتند.

در ادامه، با کمک نمودارهای مختلف، عملکرد مدل به صورت شهودی تفسیر می‌شود. در بخش پیوست نیز نمودارهای بیش‌تری وجود دارند تا عملکرد مدل را از جنبه‌های مختلف بررسی کنند.



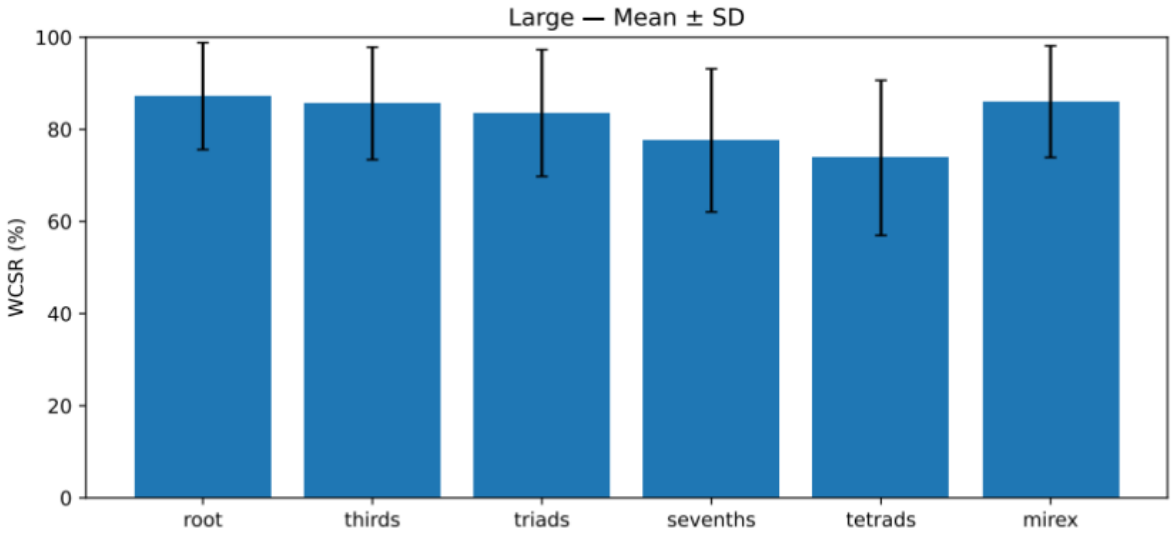
شکل ۱۸ - جدول مقایسه‌ی عملکرد مدل با مدل‌های قبلی (WCSR ٪)

این نمودار، درصد آهنگ‌هایی که WCSR آن‌ها بالاتر از ۸۰٪ است را برای معیارهای مختلف نشان می‌دهد. بیشتر از ۸۰٪ آهنگ‌ها در معیارهای Root و Maj/Min بالای ۸۰٪ دقت دارند؛ اما برای معیارهای سخت‌تر (Sevenths, Tetrads) این درصد کمتر می‌باشد.



شکل ۱۹ - پراکندگی Root accuracy برای هر آهنگ

شکل (۱۹) پراکندگی Root accuracy برای هر آهنگ را نمایش می‌دهد. در این نمودار محور افقی برای حالت Maj/Min و محور عمودی برای حالت Large Vocab می‌باشد. خط چین x=y نشان می‌دهد که امتیاز هر دو آهنگ در آن نقطه یکی است. از آن‌جایی که تقریباً تمام نقاط نزدیک خطاند، دقت ریشه در دو حالت برای تک‌تک آهنگ‌ها تقریباً یکسان است. این نشان می‌دهد که مدل ثبات بالایی دارد.



شکل ۲۰ - میانگین و انحراف معیار WCSR برای معیارهای مختلف در حالت Large Vocab

شکل (۲۰) به طور کلی میانگین و انحراف معیار WCSR برای معیارهای مختلف در حالت Large Vocab را نمایش می‌دهد. در شکل دیده می‌شود که Root و Maj/Min بالاترین دقت (بالای ۸۵٪) را دارند. اما دقت در Sevenths و Tetrads افت می‌کند. بنابراین دقت عملکرد مدل با افزایش پیچیدگی آکورد کمی کاهش پیدا می‌کند.

به طور کلی، مدل توانسته به دقت بالاتری نسبت به مدل‌های پیش از خود دست یابد. عملکرد مدل در تشخیص نت پایه بهترین بوده، و هرچه پیچیدگی آکوردها بیش‌تر شده، دقت مدل به مقدار کمی پایین آمده است. زیرا هرچه آکوردها پیچیده‌تر می‌شوند، تمایز آن‌ها از هم سخت‌تر می‌شود. همچنین، به علت تعداد کم‌تر آن‌ها در قطعات موسیقی بررسی شده، مدل در حالت Large Vocab ممکن است کمی به سمت آکوردهای پرتکرار تر گرایش پیدا کند. بنابراین یک مجموعه‌داده که متوازن‌تر باشد نیاز است تا مدل در این زمینه نیز بهبود یابد. اما با این حال حتی در زمینه Large Vocab هم عملکرد مدل نسبت به مدل‌های پیشین بهبود یافته است.

همچنین، همان‌طور که قبلا بیان شد، در مجموعه‌دادگان، فایل‌های صوتی در دسترس نبودند، و در این پژوهش، دسترسی به فایل‌های صوتی منابعی که قطعات خاص‌تری با آکوردهای پیچیده‌تر داشتند (مانند RWC Pop Dataset که روی قطعات پاپ ژاپنی تمرکز دارد)، ممکن نبود. دسترسی به قطعاتی از این قبیل و آموزش مدل با آن‌ها می‌تواند دقت مدل را روی آکوردهای متنوع‌تر بالا ببرد.

٦-٤ جمع‌بندی

پس از پیاده‌سازی مدل BTC و آموزش آن با داده‌های مربوطه، مدل روی داده‌های تست آزمایش شد، و با ارزیابی به وسیله معیارهای معتبر در زمینه پروژه‌های تشخیص آکورد، عملکرد آن بررسی و با مدل‌های پیشین مقایسه شد. در نتیجه این ارزیابی، مشخص شد که مدل عملکرد دقیق‌تری نسبت به مدل‌های پیش از خود پیدا کرده است. همچنین، ثبات مدل در تشخیص آکوردها نسبت به مدل‌های قبل بیش‌تر است؛ این بدین معنا است که اکثر آهنگ‌ها دقت خوبی دارند و فقط بعضی از آن‌ها دارای دقت کم‌تر هستند. اما در این مدل نیز همانند سایر مدل‌ها، دقت در آکوردهای پیچیده‌تر نسبت به آکوردهای ساده کمی پایین‌تر است. بنابراین، برای بهبود بیش‌تر عملکرد مدل، پیشنهاد می‌شود از مجموعه‌داده‌هایی با تنوع آکورد متوازن‌تر استفاده کرد و مدل را با انواع موسیقی مختلف آموزش داد تا دقت مدل در آکوردهای پیچیده هم‌تراز با آکوردهای ساده‌تر شود.

منابع

[1] J. C. Brown, "Calculation of a Constant Q Spectral Transform," *The Journal of the Acoustical Society of America,* vol. 89, no. 1, p. 425–434, 1991.

[2] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," arXiv preprint, -, 2017.

[3] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data,* vol. 6, no. , p. , 2019.

[4] N. S. N. P. J. U. L. J. A. N. G. Ł. K. a. I. P. Ashish Vaswani, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017.

[5] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[6] M.-T. Luong, H. Pham and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015.

[7] J. Cheng, L. Dong and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, TX, USA, 2016.

[8] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017.

[9] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Minneapolis, MN, USA, 2019.

[10] Y. Liu and M. Lapatas, "Text Summarization with Pretrained Encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China, 2019.

[11] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," OpenAI / arXiv preprint, 2018.

[12] A. V. J. U. N. S. I. S. C. H. A. M. D. M. D. H. M. D. a. D. E. Cheng-Zhi Anna Huang, "Music Transformer: Generating Music with Long-Term Structure," arXiv preprint arXiv:1809.04281, , 2018.

[13] E. J. H. a. J. P. Bello, "Rethinking Automatic Chord Recognition with Convolutional Neural Networks," in *11th International Conference on Machine Learning and Applications (ICMLA)*, Boca Raton, FL, USA, 2012.

[14] N. B.-L. a. S. D. Siddharth Sigtia, "Audio Chord Recognition with a Hybrid Recurrent Neural Network," in *16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, 2015.

[15] F. K. a. G. Widmer, "A Fully Convolutional Deep Auditory Model for Musical Chord Recognition," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Salerno, Italy, 2016.

[16] X. F. a. W. L. Yiming Wu, "Automatic Audio Chord Recognition with MIDI-trained Deep Feature and BLSTM-CRF Sequence Decoding Model," in *Music Information Retrieval Evaluation eXchange (MIREX)*, Suzhou, China, 2017.

واژه‌نامه

English Term	معادل فارسی
Augmented Triad	آکورد افزوده
Bidirectional Transformer	ترنسفورمر دوسویه
Chord	آکورد
Chroma Features	ویژگی‌های کرومای طیفی
Conditional Random Field (CRF)	میدان تصادفی شرطی
Constant-Q Transform (CQT)	تبدیل Q ثابت
Convolutional Neural Network (CNN)	شبکه عصبی کانولوشنی
Data Augmentation	داده‌افزایی (افزایش داده)
Deep Neural Network (DNN)	شبکه عصبی عمیق
Diminished Triad	آکورد کاسته
Fully Connected Layer	لایه تمام‌متصل
Gaussian Mixture Model (GMM)	مدل آمیخته گوسی
Harmonic Structure	ساختار هارمونیک
Harmony	هارمونی
Hidden Markov Model (HMM)	مدل مارکوف مخفی
Hybrid RNN	شبکه بازگشتی ترکیبی
Inversion (Chord Inversion)	وارونگی آکورد
Long Short-Term Memory (LSTM)	حافظه کوتاه‌مدت–بلندمدت
Major Triad	آکورد ماژور
Minor Triad	آکورد مینور
Multi-Head Attention	توجه چندسری
Music Information Retrieval (MIR)	بازیابی اطلاعات موسیقی
Non-negative Matrix Factorization (NMF)	فاکتورگیری ماتریس غیرمنفی

Octave	اکتاو
Overlap Ratio (OR)	نسبت همپوشانی
Pitch	زیر و بمی صدا
Pitch Shifting	تغییر زیر و بمی (شیفت فرکانسی)
Recurrent Neural Network (RNN)	شبکه عصبی بازگشتی
Rectified Linear Unit (ReLU)	واحد خطی اصلاح‌شده
Residual Connection	اتصال باقی‌مانده
Root Note	نت ریشه
Self-Attention	مکانیزم توجه به خود
Structured RNN	شبکه بازگشتی ساختاریافته
Temporal Dependency	وابستگی زمانی
Tetrad Chord	آکورد چهارصدایی
Third (Chord Interval)	نت سوم
Transformer	ترنسفورمر
Triad Chord	آکورد سه‌صدایی
Weighted Average Overlap Ratio (WAOR)	نسبت همپوشانی وزنی میانگین

پیوست

لینک کدهای پروژه:

گوگل درایو:

https://drive.google.com/drive/folders/1KNNLCeFOtCY1ln5LWcjdpCeO18iEAE18?usp=sharing

گیت‌هاب:

https://github.com/rocelload/Automatic-Chord-Recognition-Using-Deep-Learning-Techniques

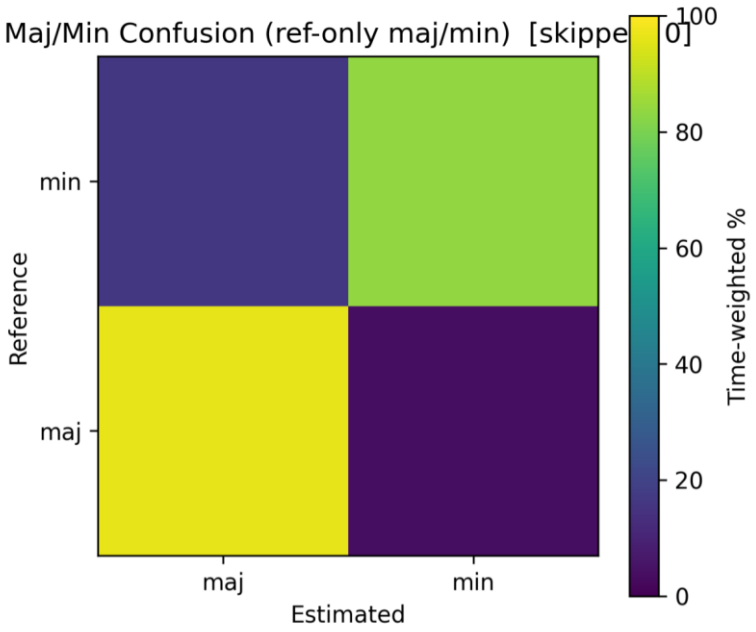
لینک مجموعه‌داده پروژه:

Isophonics Dataset

Robbie Williams Dataset

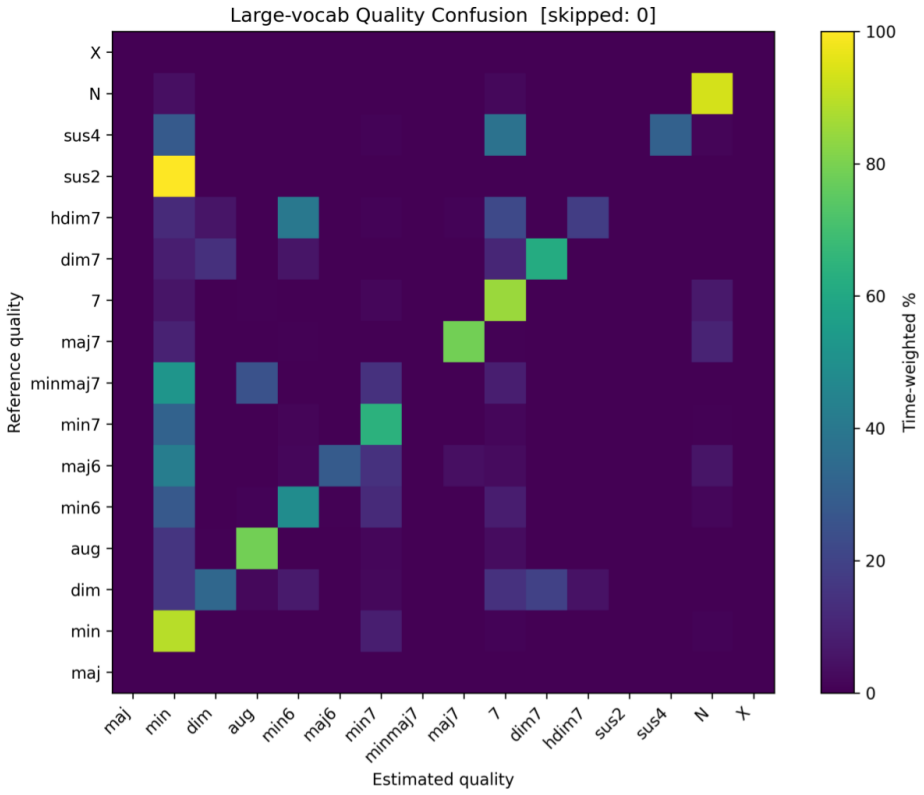
USPop / USPop2002-Chords

نمودارهای عملکرد مدل:



شکل ۲۱ - confusion matrix برای حالت ساده‌ی واژگان کوچک (Maj/Min)

این نمودار یک confusion matrix برای حالت ساده‌ی واژگان کوچک (Maj/Min) است و نشان می‌دهد مدل تا چه اندازه در تشخیص درست آکوردهای ماژور و مینور موفق بوده است. محور عمودی آکوردهای مرجع (واقعی) و محور افقی آکوردهای تخمینی مدل را نمایش می‌دهد. خانه‌ی پایین‌چپ به رنگ زرد پررنگ است که بیانگر آن است که بیشتر آکوردهای ماژور به درستی شناسایی شده‌اند. خانه‌ی بالا-راست نیز با رنگ سبز روشن دیده می‌شود که نشان‌دهنده‌ی عملکرد مناسب مدل در تشخیص آکوردهای مینور است. بخش‌های بالا-چپ و پایین-راست (یعنی خطاها) رنگ تیره‌تری دارند که نشان می‌دهد میزان اشتباهات (مثل پیش‌بینی مینور به جای ماژور یا بالعکس) نسبتاً پایین است.



شکل ۲۲ - confusion matrix مربوط به حالت واژگان بزرگ (Large Vocabulary)

این نمودار confusion matrix مربوط به حالت واژگان بزرگ (Large Vocabulary) است که شامل انواع متنوع‌تری از آکوردها (حدود ۱۷۰ کلاس) می‌باشد. محور عمودی آکوردهای مرجع (واقعی) و محور افقی آکوردهای تخمینی مدل را نشان می‌دهد. رنگ‌های روشن‌تر (سبز و زرد) بیانگر درصد بالاتر پیش‌بینی صحیح یا اشتباه برای بازه‌های زمانی قطعه هستند.

همان‌طور که دیده می‌شود، بیشترین تمرکز رنگ در قطر اصلی ماتریس قرار دارد (مثلاً برای آکوردهای maj، min، dim، maj7 و ۷)، که نشان می‌دهد مدل در بسیاری از کلاس‌ها توانسته است پیش‌بینی‌های درستی داشته باشد. با این حال، نسبت به حالت Maj/Min، پراکندگی رنگ‌ها در ستون‌ها و ردیف‌های مجاور بیشتر است؛ یعنی مدل گاهی آکوردهای پیچیده‌تر (مثل maj7 یا min7) را با هم اشتباه می‌گیرد. همچنین در برخی کلاس‌های کمتر رایج (مانند sus2 یا sus4) دقت پایین‌تر بوده و مدل تمایل دارد آن‌ها را به نزدیک‌ترین آکوردهای پایه‌ای (maj/min) نگاشت کند.

Automatic Chord Recognition Using Deep Learning Techniques

Abstract

Automatic chord recognition in music is a topic that has recently attracted the attention of researchers in the field of music processing. This is because this field is one of the fundamental subjects in music and is the basis for the harmony formation of harmony and musical structure. So, advances in this field are not only beneficial for automatic music generation, but also to various fields, such as automatic notation and structural analysis of music. In recent years, significant progress has been made in this subject using deep learning.

Early methods were based on manual feature extraction such as classical chroma and statistical models like HMM. Although these approaches were pioneer, but they had serious limitations in terms of accuracy and generalizability. Later, Convolutional Neural Networks (CNNs) were able to reduce the dependency on expert domain knowledge in feature extraction by learning features directly from audio data. The combination of CNNs with recurrent models (RNN/LSTMs)

improved the modeling of temporal dependencies, and methods like the understanding of temporal dependencies, and methods such as Hybrid RNN and CNN+CRF offered better accuracy in identifying chord continuity. However, these methods are still weak in analyzing temporal dependencies between chords, resulting in lower chord recognition accuracy. Therefore, implementing a model that can consider both short-term and long-term dependencies of chords would significantly contribute to progress in music-related deep learning research.

In this study, a model based on Bidirectional Transformer (BTC) Architecture is processed which simultaneously uses past and future information to account for both short-term and long-term dependencies between chords. As a result, after implementation and evaluation that the model achieves higher accuracy compared to its predecessors.

Shahid Beheshti University
Faculty of Computer Science and Engineering

Automatic Chord Recognition Using Deep Learning Techniques

By:
Rojin Ansari Dakhel

A THESIS SUBMITTED
FOR THE DEGREE OF
BACHELOR OF SCIENCE

Supervisor
Dr. Yasser Shekofteh

September 2025