โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ	เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี
	Web application for music box song converter
ชื่อนิสิต	นาย ธนาธิป ดอรอมาน 5933630023
	นาย จาริก ศิลปาภินันท์ 5933609023
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2562

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี

นาย ธนาธิป ดอรอมาน
นาย จาริก ศิลปาภินันท์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
สาขาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Web application for music box song converter

Tanatip Doromarn
Jarig Silpapinan

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019

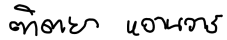
ชื่อโครงงาน (ภาษาไทย)	เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี
ชื่อโครงงาน (ภาษาอังกฤษ)	Web application for music box song converter
ผู้ดำเนินการ	นาย ธนาธิป ดอรอมาน เลขประจำตัวนิสิต 5933630023
	นาย จาริก ศิลปาภินันท์ เลขประจำตัวนิสิต 5933609023
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ทิตยา หวานวารี

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

..... หัวหน้าภาควิชาคณิตศาสตร์

(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) และวิทยาการคอมพิวเตอร์

คณะกรรมการทดสอบโครงงาน

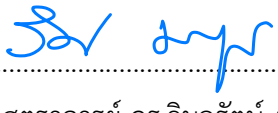


..... อาจารย์ที่ปรึกษาโครงงานหลัก

(ผู้ช่วยศาสตราจารย์ ดร.ทิตยา หวานวารี)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ)



..... กรรมการ

(รองศาสตราจารย์ ดร.วิมลรัตน์ งามอร่ามวรางกูร)

นาย ธนาธิป ดอรอมาน, นาย จาริก ศิลปาภินันท์ : เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี (Web application for music box song converter)
อ.ที่ปรึกษาโครงงาน : ผู้ช่วยศาสตราจารย์ ดร.ทิตยา หวานวารี, 73 หน้า.

โครงการวิจัย เรื่อง “เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี” มีวัตถุประสงค์เพื่อสร้างเว็บแอปพลิเคชันที่สามารถแปลงเพลงเป็นเพลงกล่องดนตรี โดยเว็บแอปพลิเคชันจะใช้ตัวแบบโครงข่ายประสาทเทียมเชิงลึกที่ได้ทำการฝึกสอนมาหาจุดเริ่มต้นและโน้ตเพลงของของเสียง แล้วสังเคราะห์เป็นเสียงเพลงกล่องดนตรีออกมาโดยอัตโนมัติ ซึ่งผู้ใช้สามารถเลือกเพลงที่ต้องการแปลงได้อย่างอิสระ อีกทั้งยังสามารถเลือกฟังเพลงผ่านเว็บแอปพลิเคชันและบันทึกเพลงกล่องดนตรีที่ได้ลงบนหน่วยความจำของคอมพิวเตอร์ ผลที่ได้พบว่าตัวเว็บแอปพลิเคชันช่วยให้ผู้ใช้งานสามารถแปลงเพลงกล่องเพลงได้อย่างสะดวก แต่อย่างไรก็ตามประสิทธิภาพของการแปลงเพลงกล่องดนตรีนั้นขึ้นกับประเภทของเพลงเป็นหลัก

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต ธนาธิป ดอรอมาน.

ลายมือชื่อนิสิต จาริก ศิลปาภินันท์.

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงาน ทิตยา หวานวารี.

ปีการศึกษา 2562.

5933630023, 5933609023 : MAJOR COMPUTER SCIENCE

KEYWORDS: MELOBOX / MUSIC BOX / WEB APPLICATION

Tanatip Doromarn, Jarig Silpapinan: Web application for music box song converter. ADVISOR: Assist. Prof. Dr. Dittaya Wanvarie, 73 pp.

The objective of “Web application for music box song converter” is to create a web application that can convert any music to a music box sound. The web application will use the trained deep neural network model to predict onsets and music notes. Then, the application will automatically synthesize a music box song. Users can freely select any song as they want, and also choose to either listen to the music through the application or save the song on their computers. The result shows that the web application can help users conveniently convert any songs. However, the efficiency of the music box converter heavily depends on the genre of the song.

Department: Mathematics and Computer Science Student’s Signature Tanatip Doromarn

Student’s Signature Jarig Silpapinan

Field of Study: Computer Science Advisor’s Signature Dittaya Wan

Academic Year: 2019

กิตติกรรมประกาศ

โครงงานเว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรีสามารถสำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างดียิ่งของ ผู้ช่วยศาตราจารย์ ดร.ฑิตยา หวานวารี อาจารย์ที่ปรึกษาโครงการ อีกทั้งยังเสียสละเวลาให้ความรู้คำปรึกษา และคอยสนับสนุนให้การดำเนินงานเป็นไปอย่างราบรื่น

ขอขอบพระคุณกรรมการคุมสอบ ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ และ รองศาสตราจารย์ ดร.วิมลรัตน์ งามอร่ามวรางกูร ผู้เป็นกรรมการคุมสอบที่ช่วยแนะแนวทางต่าง ๆ ที่เป็นประโยชน์ต่อโครงงานนี้

สุดท้ายขอขอบคุณทุกท่านที่ไม่ได้กล่าวนามไว้ข้างต้น ที่ให้การสนับสนุนในด้านต่าง ๆ ที่คอยผลักดันให้โครงงานสำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ

สารบัญ

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุผล

ดนตรีมีบทบาทในการดำเนินกิจกรรมของมนุษย์ โดยช่วยชี้นำกำหนดอารมณ์ความรู้สึก ไม่ว่าจะเป็นเพื่อความบันเทิง การพักผ่อนหย่อนใจ หรือการเร้าอารมณ์ตามกิจกรรมต่าง ๆ รูปแบบการใช้ดนตรีของมนุษย์นั้น นอกจากการฟังแล้ว ยังมีการเล่นดนตรีอีกด้วย จึงไม่ใช่เรื่องแปลกหากมีบุคคลต่าง ๆ ที่มีความชื่นชอบดนตรี จะต้องการแกะทำนองหรือโน้ตดนตรีของบทประพันธ์ เพื่อจะทดลองเล่นดูเอง หรือเพื่อเก็บบันทึกไว้

รูปแบบหนึ่งของเสียงดนตรีที่ใช้ในการผ่อนคลายได้ดีคือเสียงของกล่องดนตรี แต่เพลงกล่องดนตรีที่มีในท้องตลาดมีอยู่จำกัด หากต้องการสร้างเพลงกล่องดนตรีเอง จะต้องมีโน้ตดนตรีก่อน จากนั้นจึงนำไปสร้างกล่องดนตรี หรือเล่นด้วยคอมพิวเตอร์เพื่อให้เป็นสียงกล่องดนตรี การแกะโน้ตดนตรีจึงเป็นขั้นตอนที่สำคัญหากต้องการจะสร้างกล่องดนตรี

การแกะโน้ตดนตรีนั้นต้องอาศัยทักษะการจับเสียงตัวโน้ต และความรู้ทางทฤษฎีดนตรีหลายประการ แต่รูปแบบการทำงานนั้นซ้ำ ๆ กัน ไม่ว่าจะเป็นเพลงใดก็ตาม นั่นคือ ต้องจับทำนอง (melody) และจังหวะ (rhythm) ของเพลงให้ได้ก่อน จากนั้นนำมาสร้างเป็นโน้ตเพื่อเล่นด้วยเครื่องดนตรีอื่น ๆ เราจึงสามารถสร้างโปรแกรมคอมพิวเตอร์เพื่อให้จับทำนองหลักและจังหวะของเพลง นำโน้ตที่ได้มาสังเคราะห์เสียงกล่องดนตรีต่อไป โครงงานนี้จะรับข้อมูลเป็นเพลงเอ็มพี 3 (mp3) บนเว็บแอปพลิเคชัน และสร้างแฟ้มข้อมูลเอ็มพี 3 ของเสียงกล่องเพลงด้วย ซอฟต์แวร์ปัญญาประดิษฐ์ โดยจำกัดเฉพาะเพลงแนว พ็อป ร็อก และ อาร์แอนด์บี

1.2 วัตถุประสงค์

พัฒนาโปรแกรมสำหรับตรวจจับโน้ตดนตรีจากแฟ้มข้อมูลเอ็มพี 3 เพื่อสร้างเสียงเพลงแบบกล่องดนตรี และสามารถบันทึกเสียงเก็บในรูปแบบแฟ้มข้อมูลเอ็มพี 3

1.3 ขอบเขตของโครงงาน

- โครงงานนี้ศึกษาศึกษารูปแบบทำนองเพลง จังหวะ และเบส เฉพาะในดนตรีประเภทพ็อป ร็อก และ อาร์แอนด์บี เท่านั้น
- ข้อมูลที่มีการกำกับผลลัพธ์ (labeled data) สำหรับการฝึกสอนตัวแบบเป็นข้อมูลที่มาจากคลัง เพลงของเว็บ MuseScore [5] ซึ่งอยู่ในรูปแบบ MusicXML
- โครงงานนี้เป็นโครงงานพัฒนาเว็บแอปพลิเคชันที่มีส่วนติดต่อผู้ใช้งานสำหรับการรับแฟ้มข้อมูลเสียงประเภทเอ็มพี 3 (.mp3) หรือเวฟ (.wav) แล้วสร้างและเล่นแฟ้มข้อมูลเสียงดนตรีแบบกล่องเพลง

1.4 ขั้นตอนการดำเนินงาน

1. แผนการดำเนินงาน

1. ค้นคว้าหาข้อมูล และ แนวทางการพัฒนาซอฟต์แวร์ปัญญาประดิษฐ์
2. วิเคราะห์และกำหนดขอบเขตของระบบ ศึกษาวิธีการแปลงเพลงด้วยขั้นตอนวิธีการเรียนรู้เชิงลึก สามารถนำมาประยุกต์ใช้ได้
3. ออกแบบระบบ และ พัฒนาระบบ
4. ทดสอบประสิทธิภาพของระบบ และ แก้ไขข้อผิดพลาดที่พบของระบบ
5. สรุปผล และ จัดทำเอกสารประกอบโครงงาน

2. ระยะเวลาการดำเนินงาน

การดำเนินงาน	ปี 2562					ปี 2563		
	เดือน ส.ค.	เดือน ก.ย.	เดือน ต.ค.	เดือน พ.ย.	เดือน ธ.ค.	เดือน ม.ค.	เดือน ก.พ.	เดือน มี.ค.
ค้นคว้าหาข้อมูล								
วิเคราะห์และกำหนดขอบเขตของระบบ								
ออกแบบ และ พัฒนาระบบ								
ทดสอบ และ ปรับปรุงระบบ								
สรุปผล และ จัดทำเอกสารประกอบ โครงงาน								

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ต่อผู้พัฒนา

1. มีความรู้ความเข้าใจในทฤษฎีดนตรีมากขึ้น
2. เพิ่มพูนทักษะการเขียนโปรแกรมและการพัฒนาระบบ
3. เรียนรู้การคิดวิเคราะห์วางแผนการทำงานอย่างเป็นระบบแบบแผน เพื่อให้เกิดประโยชน์สูงสุดตามทรัพยากรที่มีอยู่
4. ฝึกการเรียนรู้ด้วยตนเอง การยอมรับฟังความคิดเห็นของผู้อื่น ความตรงต่อเวลา ตลอดจนมีความรับผิดชอบในหน้าที่
5. เพิ่มพูนทักษะความรู้ความสามารถทางด้านวิทยาการข้อมูล

ประโยชน์ต่อผู้ใช้ระบบ

1. เป็นทางเลือกสำหรับผู้ใช้งานในการช่วยแกะโน้ตดนตรีเพื่อฝึกฝนทักษะการเล่นดนตรีเบื้องต้น
2. เป็นสื่อบันเทิงที่ผู้ใช้สามารถนำบทเพลงที่ชื่นชอบทำแปลงเป็นเสียงแบบกล่องดนตรีและ ยังสามารถบันทึกหรือส่งต่อให้คนอื่นได้เนื่องจากผลลัพธ์อยู่ในรูปแบบแฟ้มข้อมูลเอ็มพี 3

1.6 โครงสร้างของรายงาน

รายงานฉบับนี้ประกอบไปด้วยเนื้อหา 6 บทดังต่อไปนี้

บทที่ 1 จะกล่าวถึง ความเป็นมาและเหตุผล

บทที่ 2 จะกล่าวถึง งานวิจัยและความรู้ที่เกี่ยวข้อง

บทที่ 3 จะกล่าวถึง การจัดเตรียมและรวบรวมข้อมูล

บทที่ 4 จะกล่าวถึง วิธีพัฒนาระบบ

บทที่ 5 จะกล่าวถึง ผลการวิจัย

และบทที่ 6 จะกล่าวถึง ข้อสรุปและข้อเสนอแนะ

บทที่ 2

งานวิจัยและความรู้ที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึง ความรู้และงานวิจัยที่เกี่ยวข้องกับเว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี

2.1 React

React คือ Javascript Library ที่ถูกพัฒนาขึ้นมาโดยทีมงาน Facebook ซึ่งเป็น Open-Source ไว้สำหรับสร้างส่วนต่อประสานผู้ใช้งาน (User interface) บนเว็บไซต์

React ถูกนำมาใช้พัฒนาเว็บไซต์แบบ SPA หรือที่เรียกกันว่า Single Page Application นั่นคือการที่หน้าเว็บไซต์ทำการโหลดไฟล์ html เพียงไฟล์เดียว เพื่อลดการรีโหลดหน้าเว็บโดยไม่จำเป็น และจะทำการรีโหลดเฉพาะส่วนที่ต้องการเท่านั้น โดยใช้ภาษา Javascript ในการดึงข้อมูลมาอัพเดตส่วนต่าง ๆ ในหน้าเว็บ

เนื่องด้วยทางทีมพัฒนาต้องการทำเว็บแอปพลิเคชัน React จึงถูกหยิบมาเพื่อพัฒนาในฝั่งทางหน้าบ้าน (Front end) โดยเป็นการสร้างส่วนต่อประสานผู้ใช้งานบนเว็บไซต์

React นั้น ประกอบไปด้วย 3 ส่วนด้วยกันคือ

1. **Component**

   React จะมองส่วนต่าง ๆ ในเว็บไซต์ไม่ว่าจะเป็น button, searchbox, dropdown เป็น Component ทั้งหมด และ Component แต่ละส่วนสามารถนำมาใช้ใหม่ได้

2. **State**

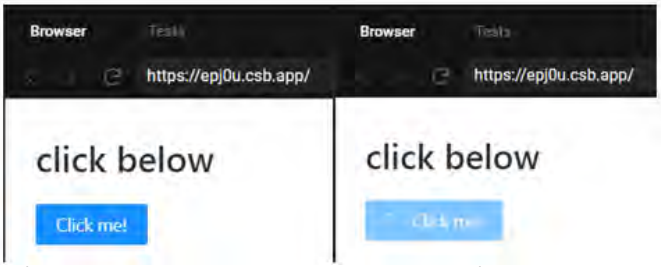
   State เป็นสถานะของข้อมูลทั้งหมดที่อยู่ใน Component โดยจะถูกเก็บไว้ที่อ็อบเจกต์ (object) ภายใน Component

3. **Props**

   Props เป็นคุณสมบัติที่กำหนดขึ้นเพื่อให้ Component สามารถส่งต่อข้อมูลไปยัง Component อื่นได้

ภาพที่ 2.1 ก ตัวอย่างการใช้ State และ Props ใน Clickbox Component

จากภาพที่ 2.1 ก จะเห็นได้ว่า Clickbox Component มีตัวแปร boolean ใน State 1 ตัว คือ loading มีค่าเป็น false (บรรทัดที่ 8-10) และมีฟังก์ชัน enterLoading โดยใช้ eventlistener เพื่อดักจับการคลิก โดยเมื่อทำการคลิก Button แล้วจะมีการเปลี่ยนค่า loading ใน State เป็นค่า true (บรรทัดที่ 12-14) พิจารณาต่อที่ render( ) function (บรรทัดที่ 16-30) มี Button Component ที่ import มาจาก antd design ซึ่งเป็น React UI library (บรรทัดที่ 3 กับ 5) โดยมีการ ส่งผ่าน Props นั่นก็คือ นำค่า loading จาก State และ ฟังก์ชัน enterLoading จาก Clickbox Component ใส่ลงในค่าคุณสมบัติ (Properties) ที่มีชื่อว่า loading และ onClick ของ Button Component โดย onClick เป็นฟังก์ชันที่ดักจับการคลิกของผู้ใช้งาน (บรรทัดที่ 20-24) การทำงานข้างต้นมีการแสดงผลดังภาพ 2.1 ข



ภาพที่ 2.1 ข ตัวอย่างส่วนต่อประสานผู้ใช้งานที่ปรากฏใน browser
(ภาพซ้าย) ก่อนทำการคลิกปุ่ม click me (ภาพขวา) หลังทำการคลิกปุ่ม click me

2.2 Flask

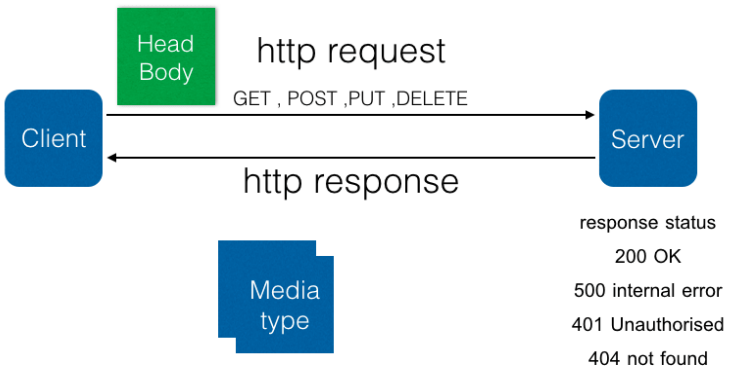
Flask เป็นเว็บเฟรมเวิร์ค (web framework) ที่เขียนด้วยภาษา python โดย Flask ถูกเรียกว่าเป็นเฟรมเวิร์คขนาดจิ๋ว (microframework) เพราะไม่ต้องอาศัยเครื่องหรือคลังโปรแกรมอื่น ๆ ก็สามารถใช้งานได้ อีกทั้งไม่จำเป็นต้องมีฐานข้อมูล และสามารถรองรับส่วนเสริมที่ช่วยเพิ่มความสามารถได้

Flask นั้นสามารถทำงานแบบ REST API ได้นั่นคือการรับ Request จากฝั่ง Client โดยผ่าน HTTP method อย่างเช่น GET POST เป็นต้น แล้วส่ง Response กลับไปหา Client ได้ จึงเหมาะกับการทำเป็นเว็บแอปพลิเคชันโดยมีหน้าที่เป็นฝั่งเซิร์ฟเวอร์ (server)

Flask จึงเหมาะที่จะนำมาทำฝั่งหลังบ้าน (Back end) โดยทำหน้าที่เป็นเซิร์ฟเวอร์สำหรับเว็บแอปพลิเคชัน เพื่อเก็บตัวแบบของโครงข่ายประสาทเทียมแบบลึกที่ได้ทำการฝึกสอนไว้ และสามารถรับไฟล์เสียงได้ทั้ง mp3 และ wav พร้อมส่งกลับให้ทางผู้ใช้งานเว็บไซต์ด้วยไฟล์ wav ได้

2.2.1 REST API

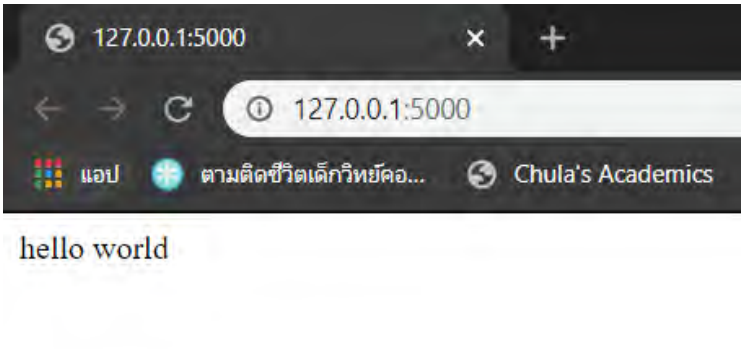
REST หรือ Representational State Transfer เป็นแบบอย่างสถาปัตยกรรมซอฟต์แวร์ (software architectural style) ในการสร้างเว็บเซอร์วิสรูปแบบหนึ่งที่อาศัย HTTP method ได้แก่ GET, POST, PUT, DELETE ในการทำงาน และส่งผลลัพธ์กลับไปในรูปของ JSON, XML เป็นต้น การใช้ REST API ส่งผลให้ทางฝั่ง Client และ ฝั่ง Server สามารถรับส่งข้อมูลข้าม platform ได้อย่างสะดวก โดยตัวข้อมูลนั้นจะถูกเก็บไว้ในส่วนของ Response body (ฝั่ง Client) หรือ Request body (ฝั่ง Server) ดังภาพที่ 2.2.1 ก



ภาพที่ 2.2.1 ก ตัวอย่างลักษณะการทำงานของ REST API

```
MusicBox > backend > venv > application.py > ...
1   from flask import Flask, request
2   app = Flask(__name__)
3
4   @app.route('/' , methods=['GET'])
5
6   def hello_world():
7       if request.method == 'GET':
8           text = 'hello world'
9       return text
10  if __name__ == "__main__":
11      app.run(host= '127.0.0.1',port=5000)
```

ภาพที่ 2.2.1 ข ตัวอย่างการใช้ REST API ด้วย Flask

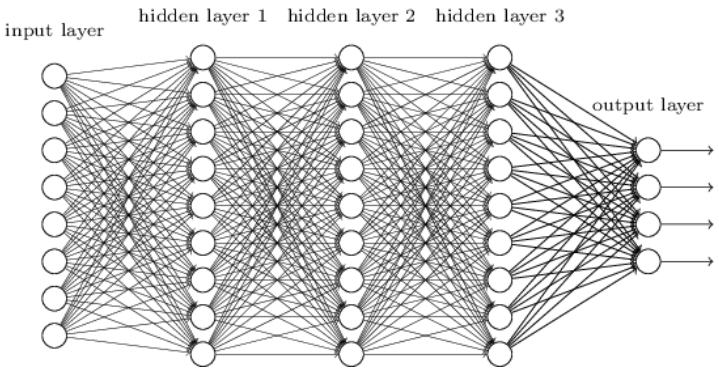


ภาพที่ 2.2.1 ค ตัวอย่าง Response จาก Flask Server เมื่อมีการ Request จากทาง Client

จากภาพที่ 2.2.1 ข เห็นได้ว่ามีการรัน Flask Server โดยกำหนดค่า Local Host เท่ากับ 127.0.0.1 และ Port เท่ากับ 5000 พร้อมทั้งกำหนดเส้นทาง (route) บน URL ไว้ที่ Root ของ Server นั่นคือ 127.0.0.1:5000/ รวมถึงกำหนด GET method ในการ Request (บรรทัดที่ 4) เมื่อทาง Client ทำการ Request มาที่ URL http://127.0.0.1:5000/ ด้วย GET method แล้ว Flask Server จะส่ง Response กลับไปหาฝั่ง Client ด้วยข้อความ hello world (บรรทัดที่ 6-8) ดังภาพที่ 2.2.1 ค

2.3 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึกนับเป็นส่วนหนึ่งของการเรียนรู้ของเครื่อง (machine learning) โดยพื้นฐานของการเรียนรู้เชิงลึกคือ อัลกอริทึมที่พยายามจะสร้างแบบจำลองเพื่อแทนความหมายของข้อมูลในระดับสูงอย่างเช่น ราคาหุ้น ราคาที่ดิน เป็นต้น การเรียนรู้เชิงลึกจะทำการสร้างสถาปัตยกรรมข้อมูลขึ้นมา โดยสถาปัตยกรรมที่ได้ประกอบไปด้วยโครงสร้างย่อย ๆ และแต่ละโครงสร้างที่ได้ มาจากการแปลงที่ไม่เป็นเชิงเส้น (non-linear function)



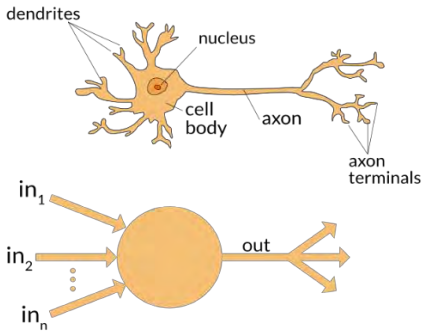
ภาพที่ 2.3 ก ตัวอย่างโครงข่ายประสาทเทียมเชิงลึก

ที่มา: http://neuralnetworksanddeeplearning.com/chap5.html

ลักษณะสถาปัตยกรรมของการเรียนรู้เชิงลึกนั้นเป็นโครงข่ายประสาทเทียมที่มีจำนวนชั้นซ่อนจำนวนหลายชั้น โดยแต่ละชั้นมีการคำนวณที่ซับซ้อน และมีการเรียนรู้ที่แตกต่างกัน นอกจากนี้ยังแบ่งรูปแบบสถาปัตยกรรมของการเรียนรู้เชิงลึกที่เป็นโครงข่ายประสาทเทียมได้อีกหลายแบบ เช่น โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า (Feedforward Neural Networks) โครงข่ายประสาทเทียมแบบสังวัตนาการ (Convolutional Neural Networks) และโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network) ซึ่งมีการนำมาใช้งานอย่างแพร่หลายในทางคอมพิวเตอร์วิทัศน์ การรู้จำเสียงพูด การประมวลผลภาษาธรรมชาติ การรู้จำเสียง และชีวสารสนเทศศาสตร์

2.4 โครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม ถูกสร้างขึ้นเพื่อเลียนแบบการทำงานของโครงข่ายระบบประสาทในสมองของมนุษย์ ที่ประกอบไปด้วยเซลล์ประสาท (neuron) จำนวนมาก โดยแต่ละเซลล์ประสาทประกอบไปด้วยตัวเซลล์ (cell body) โดยมีช่องสัญญาณขาเข้าจากใยประสาทนำเข้า (dendrites) และให้สัญญาณขาออกผ่านแกนประสาทนำออก (axon) ผ่านจุดประสานประสาท (synapse) ออกไปเชื่อมกับใยประสาทนำเข้าของเซลล์ประสาทอื่น ๆ จนเป็นโครงข่ายที่เชื่อมเซลล์ประสาทขนาดใหญ่ (neural network)



ภาพที่ 2.4 ก เซลล์ประสาท (บน) และ เซลล์ประสาทเทียม (ล่าง)

ที่มา: https://appliedgo.net/perceptron/

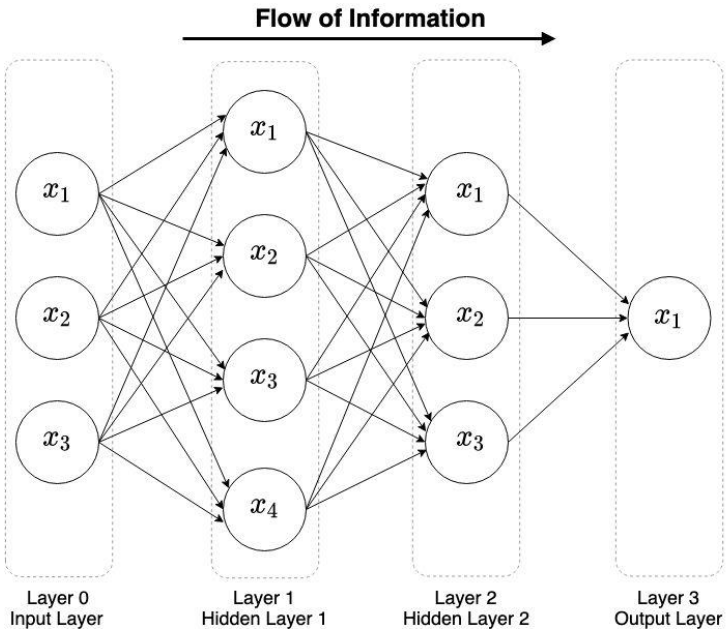
เซลล์ประสาทเทียม (artificial neuron) ในโครงข่ายประสาทเทียม จะถูกเรียกว่า เพอเซปตรอน (perceptron) หรือโหนด (node) โดยที่แต่ละโหนดนั้นจะรับค่าถ่วงน้ำหนัก (weight) จากแต่ละข้อมูลขาเข้านำมารวมกันแล้วเข้าฟังก์ชันกระตุ้น (activation function) ได้ผลลัพธ์ส่งต่อไปยังโหนดอื่น ๆ

พิจารณาที่สมการคำนวณค่าภายในแต่ละโหนดด้านล่าง กำหนดให้ $i \in N$ โดย N เป็นจำนวนนับของลำดับชั้นในโครงข่าย และ i มีค่าอยู่ในช่วง $[1, n]$ เห็นได้ว่าในแต่ละโหนด N_i มีค่า x_i เป็นค่าที่รับเข้ามาจากโหนดก่อนหน้า w_i เป็นค่าน้ำหนักกำกับแต่ละค่า x_i ที่เข้ามา b เป็นค่าอคติ และ f เป็นฟังก์ชันกระตุ้น โดยโหนดแต่ละโหนดสามารถเชื่อมต่อกันเป็นโครงข่ายประสาทเทียมได้ นอกจากนี้เรายังสามารถนำโครงข่ายประสาทเทียมมาทำการแยกชั้นโหนดเป็น 3 กลุ่มหลัก ๆ ด้วยกันนั่นคือ ชั้นข้อมูลขาเข้า (input layer) ชั้นซ่อน (hidden layer) และชั้นข้อมูลขาออก (output layer)

$$\mathrm{N}_i = f(b + \sum_{i=1}^{n} w_i x_i)$$

2.4.1 โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า (Feedforward neural networks)

โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าหรืออีกชื่อคือโครงข่ายประสาทเทียมแบบหลายชั้น (Multi-layered network of neurons) เป็นโครงข่ายประสาทเทียมที่มีการเชื่อมโยงระหว่างโหนดแบบไม่เป็นวังวน (non-cycle) และจะป้อนข้อมูลไปข้างหน้าเริ่มจากโหนดในชั้นขาเข้า (L_{input}) ผ่านชั้นซ่อน (L_{hidden}) จำนวนหลาย ๆ ชั้น จนถึงโหนดชั้นขาออก (L_{output}) โดยไม่มีการวนกลับเข้าไปของข้อมูลขาออก ดังภาพที่ 2.4.1 ก ด้านล่าง



ภาพที่ 2.4.1 ก ลักษณะการผ่านของข้อมูลในโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า

เราจะได้ค่าข้อมูลขาออกของแต่ละโหนดดังสมการในหัวข้อที่ 2.4 โดยให้ฟังก์ชัน f ที่ใช้ในแต่ละโหนดนั้นเป็นฟังก์ชันแบบไม่เชิงเส้น โดยฟังก์ชันเหล่านี้ถูกนำมาใช้เพื่อให้ตัวแบบสามารถเรียนเรียนรู้ความสัมพันธ์แบบไม่เชิงเส้นได้ มีลักษณะกราฟของฟังก์ชันที่กล่าวมาดังภาพที่ 2.4.1 ข โดยฟังก์ชันที่นิยมใช้กันได้แก่

ReLU (Rectified Linear Unit)

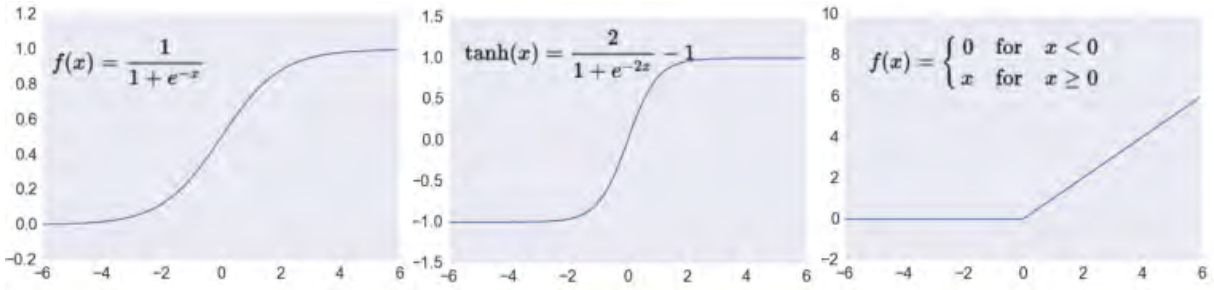
$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases}$$

Sigmoid

$$f(x) = \frac{1}{1 + \ e^{-x}}$$

Tanh

$$f(x) = \frac{2}{1 + \ e^{-2x}} - 1$$



ภาพที่ 2.4.1 ข สมการและกราฟของฟังก์ชันกระตุ้น Sigmoid (ซ้าย) Tanh (กลาง) ReLU (ขวา)

ที่มา : http://www.feeny.org/scaling-data-deep-learning/

การเลือกใช้งานฟังก์ชันกระตุ้นสำหรับโหนดใด ๆ โหนดหนึ่ง จะพิจารณาตามชั้นที่โหนดอยู่เป็นหลัก L_{input} จะไม่มีการใช้งานฟังก์ชันกระตุ้นเนื่องจากเป็นชั้นที่เริ่มแรกสุดในการป้อนข้อมูลไปข้างหน้า

L_{hidden} นั้นมักจะใช้ ReLU เป็นฟังก์ชันกระตุ้นเพื่อป้องกันปัญหาการขาดหายไปของเกรเดียนต์ [8] (gradient vanishing) ในกรณีที่มี L_{hidden} จำนวนมาก และ Tanh นั้นจะถูกนำมาใช้ก็ต่อเมื่อต้องการปรับข้อมูลให้ค่าอยู่ในช่วง [-1,1]

ใน L_{output} ถ้าเป็นปัญหาการถดถอย (Regression) จะนิยมใช้ ReLU หรือ ฟังก์ชันเชิงเส้นปกติ เพื่อให้ข้อมูลขาออกเป็นค่าต่อเนื่อง แต่ถ้าเป็นปัญหาการจำแนกประเภท (classification) จะนิยมใช้ซิกมอยด์ (sigmoid) สำหรับการจำแนกประเภทที่เป็นฐานสอง นั่นคือ 0 หรือ 1

โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้ามักจะถูกนำมาใช้ร่วมกับโครงข่ายประสาทเทียมแบบสังวัตนาการอยู่บ่อยครั้ง โดยนำมาเชื่อมที่ชั้นบนสุด และถูกเรียกว่า ชั้นเครือข่ายแน่น (fully connected network) เพื่อทำการจำแนกประเภท (classifier) ข้อมูลที่ได้มาจากเครือข่ายประสาทเทียมสังวัตนาการ

2.4.2 โครงข่ายประสาทเทียมแบบสังวัตนาการ (Convolutional neural networks)

โครงข่ายประสาทเทียมแบบสังวัตนาการเป็นที่นิยมอย่างมาก และถูกใช้งานอย่างแพร่หลายในปัจจุบัน โดยมักจะนำไปใช้กับปัญหาการรู้จำรูปแบบ (Pattern recognition) จำพวกรูปภาพ เพื่อให้สามารถแยกแยะประเภทที่แตกต่างออกจากกันได้

ลักษณะสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบสังวัตนาการมีชั้นหลัก ๆ อยู่ด้วยกัน 3 ชั้น ได้แก่ ชั้นสังวัตนาการ (convolution layer) ชั้นบ่อรวม (pooling layer) และชั้นโครงข่ายแน่น (fully connected)

ชั้นสังวัตนาการเป็นชั้นที่ใช้ตัวกรอง (filter) ในการทำสังวัตนาการด้วยการกวาดตัวกรองบนข้อมูลขาเข้า โดยกำหนดการเลื่อน (stride) ในการกวาดแต่ละขั้น ซึ่งจะให้ผลลัพธ์เป็นผังลักษณะเด่น (feature map) ดังเช่นสมการสังวัตนาการของภาพสองมิติในแชนเนลใด ๆ ด้านล่าง

$$O[m,n]^K = (I * F)[m,n]^K = \sigma\left(\sum_i \sum_j F[i,j]^K I[m-i,n-j]^K\right)$$

ให้ $K, a, b, c, d, m, n, i, j \in N$ โดยที่

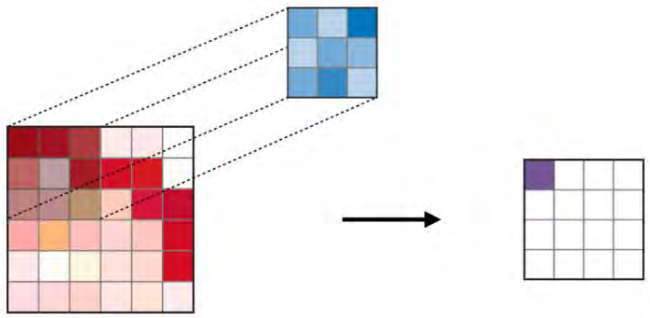
- O คือ ผังลักษณะเด่น โดยระบุตำแหน่งเป็นแถว m หลัก n
- I คือ ตารางข้อมูลขาเข้าขนาดมิติ $c \times d$ โดยระบุตำแหน่งเป็น แถว $m-i$ หลัก $n-j$
- F คือ ตัวกรองขนาดมิติ $a \times b$ โดยระบุตำแหน่งเป็น แถว i หลัก j
- σ คือ ฟังก์ชันกระตุ้นดังภาพ 2.4.1 ข
- K คือ ค่าแชนเนล (channel)

นำตัวกรอง F กวาดบนตารางข้อมูลขาเข้า I ที่แชนเนล K โดยทำการเลื่อน s (stride) ทั้งในแนวแกน x และแกน y และมีการเสริมเต็ม p (padding) เข้าไปใน I เราสามารถคำนวณขนาดมิติในแต่ละมิติของผังลักษณะเด่น O ได้จาก

$$n_{out} = [\frac{n_{in} + 2p - f}{s} + 1]$$

กำหนดให้ ค่าพารามิเตอร์ทั้งหมดในสมการข้างต้นต้องเป็นค่าที่อยู่ในแนวแกนเดียวกัน โดยที่

- n_{out} คือค่ามิติขาออกของตารางข้อมูลในแนวแกนใด ๆ
- n_{in} คือค่ามิติขาเข้าของตารางข้อมูลในแนวแกนใด ๆ
- f คือ ขนาดมิติของตัวกรองในแนวแกนใด ๆ
- p คือ การเสริมเต็ม
- σ คือ ฟังก์ชันกระตุ้นดังใด ๆ ดังภาพ 2.4.1 ข



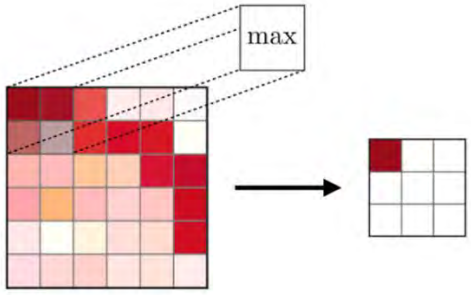
ภาพที่ 2.4.2 ก ตัวอย่างการทำสังวัตนาการของภาพสองมิติที่แชนเนล K

ที่มา : https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

ถัดมาที่ชั้นบ่อรวม ซึ่งเป็นชั้นที่นำผังลักษณะเด่นแชนเนล K ที่ได้จากชั้นสังวัตนาการมาเข้าฟังก์ชันบ่อรวมเช่น การใช้ค่าสูงสุด (max-pooling) หาได้โดย

$$Y[m,n]^K = max_{i,j}[X_{m-i,n-j}]^K$$

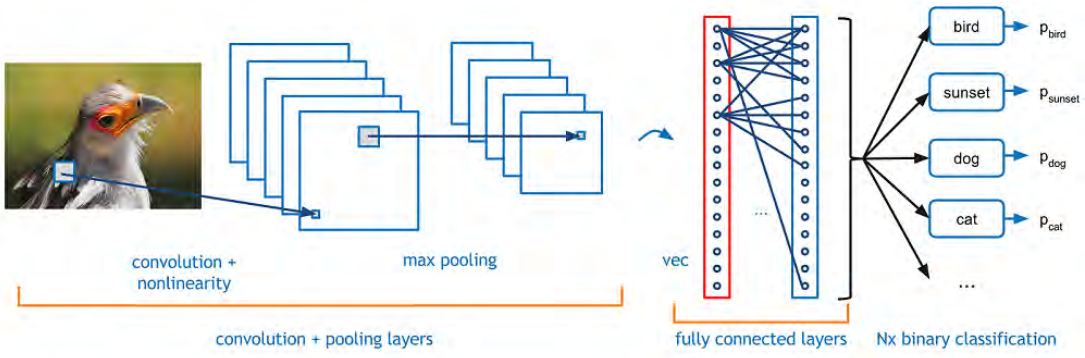
กำหนดให้ X ที่แชนเนล K นั้นมีมิติขนาด $A \times B$ และให้ค่า Y แต่ละตำแหน่งที่แชนเนล K ในตารางผลลัพธ์เป็นค่า $x_{m-i,n-j}$ ที่สูงที่สุด ที่ได้จากการทำ max-pooling โดยกำหนดขนาดหน้าต่าง (window size) เท่ากับ $a \times b$ จะได้ว่า i เท่ากับ $1, \ldots, a$ และ j เท่ากับ $1, \ldots, b$ กำหนดให้มีค่าการเลื่อน s จากนั้นทำการเลื่อนหน้าต่างทั้งแนวแกน x และแกน y ตามในขนาดแต่ละมิติ นั่นคือ a และ b ทำให้ตารางผลลัพธ์ Y ที่ได้นั้นมีมิติขนาด $\frac{A}{s} + 1 \times \frac{B}{s} + 1$



ภาพที่ 2.4.2 ข ตัวอย่างตารางผลลัพธ์แชนเนลที่ K ที่ได้จากชั้นบ่อรวม

ที่มา : https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

นำตารางผลลัพธ์ที่ได้จากชั้นบ่อรวมมาบีบมิติลง (flatten) ให้เหลือเพียงมิติเดียว แล้วนำเข้าสู่ชั้นโครงข่ายแน่น โดยทุก ๆ ค่าจะเชื่อมกับทุกโหนดของชั้นโครงข่ายแน่น และผลลัพธ์ที่ได้จากชั้นโครงข่ายแน่นจะเป็นค่าต่อเนื่องในกรณีที่เป็นปัญหาการถดถอย หรือเป็นค่าคะแนนประเภท (class score) ในปัญหาการจำแนกประเภท ดังภาพที่ 2.4.2 ค



ภาพที่ 2.4.2 ค สถาปัตยกรรมตัวอย่างของโครงข่ายประสาทเทียมแบบสังวัตนาการแบบการจำแนกประเภท

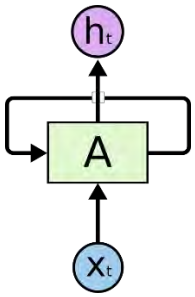
ที่มา : https://www.kdnuggets.com/2016/09/beginners-guide-understanding-convolutional-neural-networks-part-1.html

2.4.3 โครงข่ายประสาทเทียมแบบเวียนซ้ำ (Recurrent neural networks)

โครงข่ายประสาทเทียมแบบวนซ้ำเป็นโครงข่ายประสาทเทียมประเภทที่สามารถประมวลผลข้อมูลขาเข้าที่มีลักษณะเป็นลำดับอย่างเช่น ข้อความที่เป็นลำดับของตัวอักษร เสียงที่เป็นลำดับของคลื่น วิดิโอที่เป็นลำดับของภาพและเสียง เป็นต้น

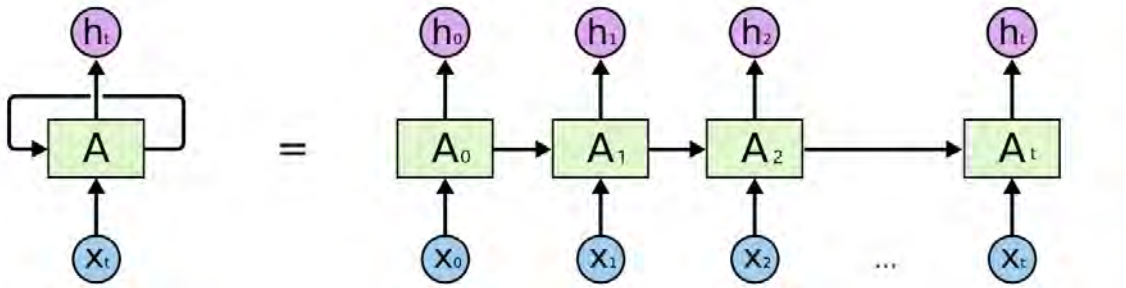
โครงข่ายประสาทเทียมนี้ใช้หลักการนำข้อมูลขาออกของขั้นก่อนหน้ามาเป็นข้อมูลขาเข้าของขั้นปัจจุบันนั่นคือ จะมีสถานะภายใน (internal state) ซึ่งเป็นหน่วยความจำ (memory) ประเภทที่

เอาไว้เก็บสถานะของข้อมูล เพื่อให้ตัวแบบสามารถจดจำลำดับรูปแบบ (pattern) ของข้อมูลขาเข้าที่มีลักษณะเป็นลำดับได้



ภาพที่ 2.4.3 ก ตัวอย่างกลุ่มก้อนใด ๆ ของโครงข่ายประสาทเทียมแบบวนซ้ำ
ที่มา : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

จากแผนภาพที่ 2.4.3 ก กลุ่มก้อนใด ๆ (chunk) ของโครงข่ายประสาทเทียมแบบวนซ้ำจะมีการรับข้อมูลขาเข้า $\mathbf{x_t}$ ณ เวลาใด ๆ ผ่านชั้นซ่อน $\mathbf{A}$ ซึ่งเป็นการวนซ้ำ (loop) โดยส่งผ่านข้อมูลไปยังชั้นของโครงข่ายที่เป็นขั้นถัดไป และได้ข้อมูลขาออก $\mathbf{h_t}$ เราสามารถนำกลุ่มก้อนนี้มาแผ่ออกได้ดังภาพที่ 2.4.3 ข



ภาพที่ 2.4.3 ข ตัวอย่างโครงข่ายภายในกลุ่มก้อนของโครงข่ายประสาทเทียมแบบวนซ้ำ
ที่มา : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

กำหนดให้สมการที่ใช้อธิบายสถานะปัจจุบัน (current state) ในแต่ละชั้น ณ เวลาใด ๆ เป็น

$$A_t = f_h(W_{hp}A_{t-1} + W_{hc}X_t + b_h)$$

โดยที่

- f_h คือ ฟังก์ชันกระตุ้นในชั้นซ่อน
- W_{hp} คือ ค่าน้ำหนักของสถานะก่อนหน้า (previous state)
- W_{hc} คือ ค่าน้ำหนักของข้อมูลขาเข้าในสถานะปัจจุบัน
- b_h คือ ค่าอคติในสถานะปัจจุบัน

นอกจากนี้ยังอธิบายข้อมูลขาออกด้วยสูตรดังต่อไปนี้เช่นกัน

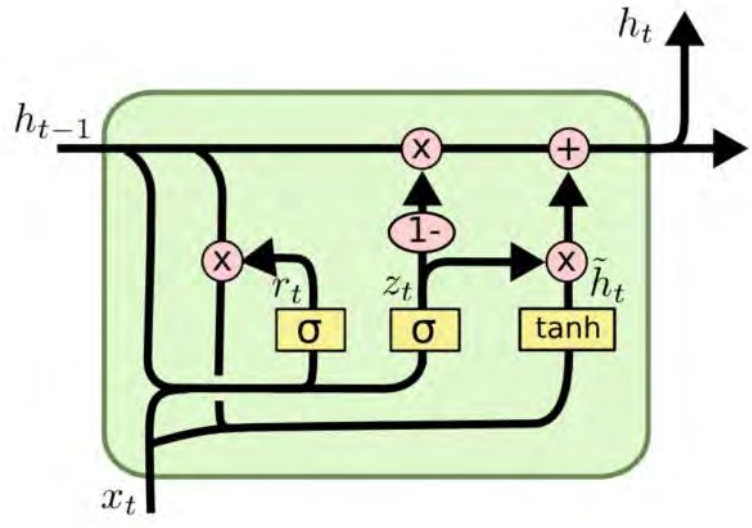
$$y_t = f_y(W_{hy}A_t + b_y)$$

โดยที่

- f_y คือ ฟังก์ชันกระตุ้นในชั้นขาออก
- W_{hy} คือ ค่าน้ำหนักของสถานะขาออก (output state)
- b_y คือ ค่าอคติในสถานะขาออก

อย่างไรก็ตาม สถาปัตยกรรมที่กล่าวมาข้างต้นประสบกับปัญหาการขาดหายไปของเกรเดียนต์ [8] ทำให้ไม่สามารถจดจำลำดับที่มีความเกี่ยวข้องกันในระยะยาวได้ โครงข่ายประสาทเทียมหน่วยความจำระยะสั้นแบบยาว [7] (Long Short-Term Memory Neural Networks – LSTMs) และ โครงข่ายประสาทเทียมหน่วยเวียนซ้ำแบบมีประตูสัญญาณ [8] (Gated Recurrent Unit - GRU) จึงถือกำเนิดขึ้นมาเพื่อแก้ปัญหาดังกล่าว

GRU ถูกคิดค้นโดยในปี 2014 Kyunghyun Cho และคณะพัฒนาต่อยอดจาก LSTM ให้มีโครงสร้างที่ไม่ซับซ้อนโดยการปรับประตูสัญญาณในเซลล์ของ LSTM ที่ประกอบไปด้วย ประตูสัญญาณลืม (Forget Gate), ประตูสัญญาณขาเข้า (Input Gate) และ ประตูสัญญาณขาออก (Output Gate) เป็น ประตูสัญญาณตั้งใหม่ (Reset Gate) และ ประตูสัญญาณปรับ (Update Gate) ซึ่งประตูสัญญาณแบบปรับจะทำหน้าที่พิจารณาว่าควรจะเก็บข้อมูลสถานะก่อนหน้าไว้มากน้อยเพียงใด และ ประตูสัญญาณแบบตั้งใหม่จะทำหน้าที่คำนวณว่าจะนำข้อมูลจากจากสถานะก่อนหน้ามาพิจารณารวมกับข้อมูลขาเข้าปัจจุบันมากน้อยเพียงใด ดังภาพที่ 2.4.3 ค



ภาพที่ 2.4.3 ค สถาปัตยกรรมภายในเซลล์ GRU

ที่มา : https://mc.ai/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control/

เราสามารถอธิบายเซลล์ GRU ด้วยสมการดังต่อไปนี้

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$
$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$
$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

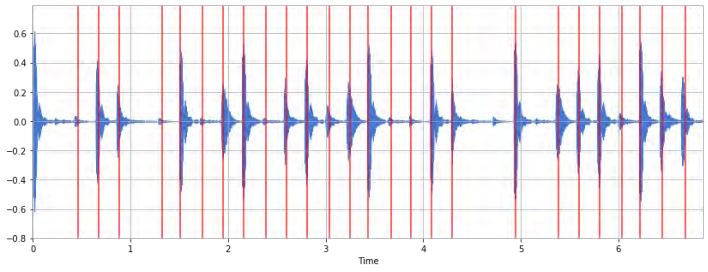
โดยที่

- r_t คือ Reset Gate
- z_t คือ Update Gate
- $\tilde{h}_t$ คือ สถานะซ่อนปัจจุบัน
- h_t คือ สถานะซ่อนขาออก
- W, W_r, W_z คือ เมทริกซ์น้ำหนัก
- h_{t-1} คือ สถานะซ่อนจากเซลล์ก่อนหน้า
- x_t คือ ข้อมูลขาเข้า
- σ คือ ฟังก์ชันกระตุ้นดังภาพ 2.4.1 ข

2.5 เสียงและดนตรี (Audio and Music)

2.5.1 จุดเริ่มต้นของเสียง (Onset)

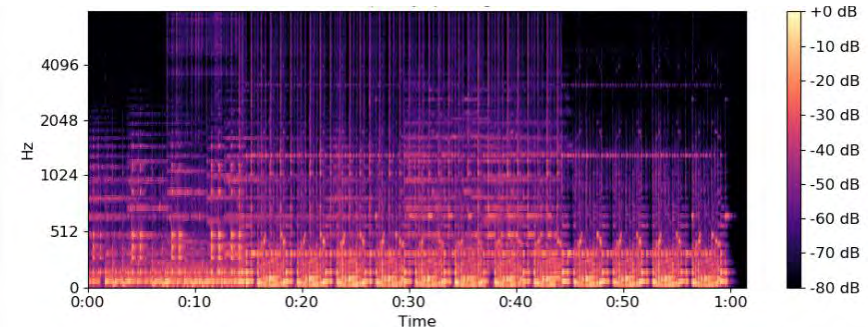
จุดเริ่มต้นของเสียง เป็นคุณสมบัติของเสียงใด ๆ ก็ตามบนโลกใบนี้ รวมถึงเหตุการณ์ที่เกิดขึ้นในดนตรี นั่นคือจุดที่มีการเริ่มกดโน้ตดนตรีแต่ละตัวในเสียงเพลง เราสามารถนำเสียงเพลงมาทำเป็นกราฟคลื่นเสียงได้ โดยให้แกน y เป็นแกนความดัง และแกน x เป็นแกนเวลา (time) เส้นตรงสีแดงขนานแกน y แสดงตำแหน่งจุดเริ่มต้นของเสียง ดังรูปที่ 2.5.1 ก จะเห็นได้ว่าจุดเริ่มต้นของเสียงเป็นจุดที่ค่าแอมพลิจูดมีค่าสูงในระยะเวลาสั้น ๆ



ภาพที่ 2.5.1 ก กราฟคลื่นเสียงที่แสดงตำแหน่งจุดเริ่มต้นของเสียง

ที่มา : https://musicinformationretrieval.com/onset_detection.html

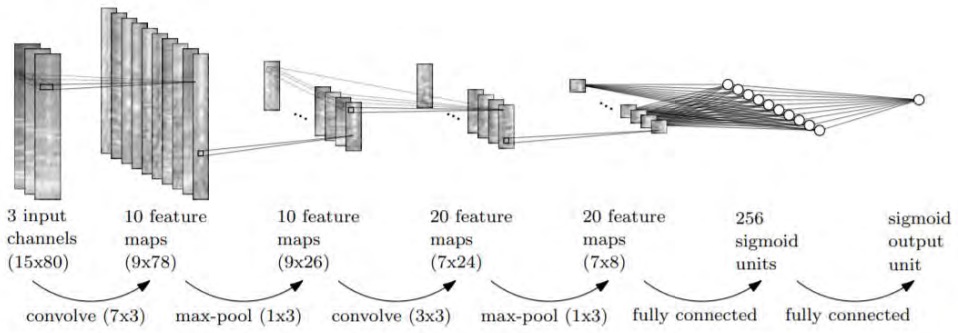
นอกจากนี้แล้วเรายังสามารถนำกราฟคลื่นเสียงมาพลอตกราฟสเปกโทรแกรม (spectrogram) โดยให้แกน y เป็นค่าความถี่ (frequency) แกน x เป็นเวลา (time) และมีความเข้ม (intensity) ที่บอกค่าแอมพลิจูดของแต่ละความถี่ในแต่ละช่วงเวลา



ภาพที่ 2.5.1 ข กราฟสเปกโทรแกรมแบบ mel-frequency

ที่มา : [4] Librosa Documentation

จุดเริ่มต้นของเสียงในสเปกโทรแกรมจะเป็นจุดที่มีการเปลี่ยนของสเปกตรัมต่อเวลา เมื่อลองพิจารณาจากสเปกตรัม จะพบว่า การตรวจจับจุดเริ่มต้นของเสียงคล้ายคลึงกับการหาขอบรูป (edge detection) ในรูปภาพ ด้วยเหตุนี้ การใช้ CNN จึงเป็นตัวเลือกที่เหมาะสมกับการตรวจจับจุดเริ่มต้นของเสียง งานวิจัยของ Jan Schlüter and Sebastian Böck [3] ใช้ชุดเพลงประเภทหลายเสียง (polyphonic) และสร้างตัวแบบที่ใช้ ในการตรวจจับจุดเริ่มต้นของเสียง โดยเริ่มจากข้อมูลเข้าเป็นสเปกโทรแกรมขนาด 15x80 พิกเซลที่สกัดออกมา ส่งเข้าชั้นสังวัตนาการ และชั้นรวมค่าสูงสุด ขนาด 7x3 1x3 3x3 1x3 สลับกันตามลำดับ แล้วปิดท้ายด้วย โครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า ซึ่งใช้ฟังก์ชันกระตุ้นซิกมอยด์ (sigmoid) ขนาด 256 นิวรอน และมี ผลลัพธ์เป็นค่า 0 (ไม่เป็นจุดเริ่มต้น) หรือ 1 (เป็นจุดเริ่มต้นของเสียง)



ภาพที่ 2.5.1 ค สถาปัตยกรรมของตัวแบบหาจุดเริ่มต้นเสียง

ที่มา : [3] Musical Onset Detection with Convolutional Neural Networks

ในการทดลองของ Jan Schlüter and Sebastian Böck จะฝึกสอนตัวแบบ 100 รอบ (epoch) ด้วยวิธีการหาค่าเหมาะที่สุดแบบ SGD และ กำหนดขนาดชุดสำหรับการฝึกสอน (mini-batch) เป็น 256 ตัวอย่าง อัตราการเรียนรู้ (learning rate) ที่ 0.05 เมื่อพิจารณาผลลัพธ์ที่ได้ พบว่าตัวแบบ CNN เอาชนะตัวแบบโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า และ CNN แบบที่ใช้เคอร์เนลเป็นรูปสี่เหลี่ยมมุมฉากให้ประสิทธิภาพที่ดีขึ้นจากการใช้เคอร์เนลเป็นรูปสี่เหลี่ยมจัตุรัส โดยมีค่าความเที่ยงตรง (precision) ที่ 90.5% ค่าการค้นคืน (recall) ที่ 86.6% และ F-measure ที่ 88.5%

2.5.2 ทำนองเพลง (Melody)

ทำนองเพลง [1] เป็นเสียงที่เปล่งออกมาโดยมีความต่อเนื่องกันเป็นระบบ ทำนองเพลงเปรียบเหมือน รูปร่างของบทเพลงที่มีเสียงสูง, ต่ำ, สั้น, ยาว ประกอบกันโดยทั่วไปดนตรีประกอบด้วยทำนองซึ่งเป็น องค์ประกอบที่ง่ายต่อการจำรวมถึงเป็นเอกลักษณ์ของบทเพลงนั้น ๆ ดังนั้นหากต้องการจะแกะบทเพลงใดสักเพลงการเริ่มต้นด้วยการแกะทำนองของเพลงจึงเป็นสิ่งที่ง่ายที่สุด

งานวิจัยของ Li Su [2] ได้สกัดโน้ตจากทำนองของเสียงร้อง หรือ Vocal Melody ด้วยข่ายงาน ประสาทแบบสังวัตนาการ (convolution Neural Network - CNN) ซึ่งเริ่มต้นด้วยการรับแฟ้มข้อมูลเสียงเป็น สัญญาณ 1 มิติ และแปลงเป็นสเปกโทรแกรมเพื่อให้ตรวจจับรูปแบบพฤติกรรมโดยทั่วไปของเสียงได้มีประสิทธิภาพยิ่งขึ้น จากนั้นแปลงเป็นเซปสตรัมทั่วไป (generalized cepstrum - GC) และเซปสตรัมทั่วไปของสเปกตรัม (generalized cepstrum of spectrum - GCoS)

กำหนดให้ความสัมพันธ์ของสเปกโทรแกรม, เซปสตรัมทั่วไป และเซปสตรัมทั่วไปของสเปกตรัม เป็นดังนี้

$$Z_0[k,n] \coloneqq \sigma_0\left(W_f X\right)$$
$$Z_1[q,n] \coloneqq \sigma_1(W_t F^{-1} Z_0)$$
$$Z_2[k,n] \coloneqq \sigma_2\left(W_f F Z_1\right)$$

ให้ Z_0 คือ สเปกโทรแกรม Z_1 คือ เซปสตรัมทั่วไป และ Z_2 คือ เซปสตรัมทั่วไปของสเปกตรัม มีการกำหนดค่าดัชนี (index) k ในสมการ Z_0 , Z_2 เป็นค่าความถี่ (frequency) ในขณะที่ค่าดัชนี q ใน สมการ (2) แสดงถึงค่าคิวเฟรนซี (quefrency) และค่าดัชนี n แสดงถึงเวลา โดยแต่ละสมการจะมีฟังก์ชันกระตุ้น (activation function) เป็น

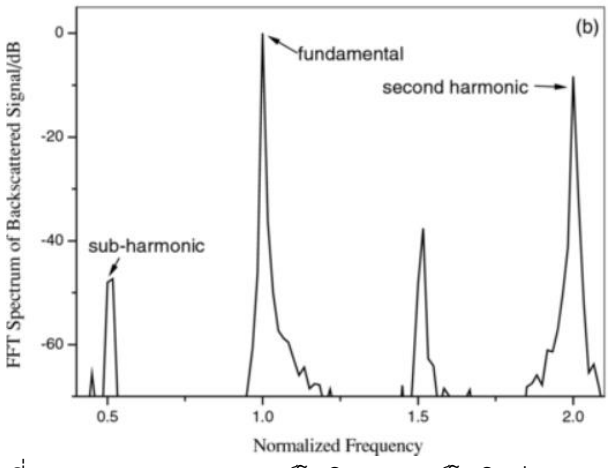
$$\sigma_i(Z) = |ReLU(Z_i)|^{\gamma_i}, \quad \mathrm{i} = 0,1,2$$

สมการข้างต้น มีฟังก์ชัน $\mathbf{ReLU}$ ประกอบกับฟังก์ชันถอดรากเรียงราย (element wise-root function) และกำหนดให้ค่า γ_i อยู่ในช่วง 0 ถึง 1

นอกจากนี้ยังมี ค่า W_f and W_t ซึ่งเป็นตัวกรอง (High-pass filters) มีลักษณะเป็นเมทริกซ์ทแยงมุม (diagonal matrices) ที่ระบุค่าความถี่ตัดและค่าคิวเฟรนซี่ตัด (cutoff frequency and quefrency) นั่นคือค่า k_c และ q_c อยู่ภายในเมทริกซ์ ตามลำดับดังสมการต่อไปนี้

$$W_{f\ or\ t}[l,l] = \begin{cases} 1, & l > \mathrm{k}_c\ or\ \mathrm{q}_c; \\ 0, & otherwise \end{cases}$$

จากนั้นนำ GC และ GCoS นั่นคือสมการ Z_1 และ Z_2 มาใช้ร่วมกันเพื่อวัตถุประสงค์ในการกำจัดเสียงฮาร์โมนิกและฮาร์โมนิกย่อยที่ไม่ต้องการออก (harmonics and sub-harmonics) ดังภาพที่ 2.5.2 ก



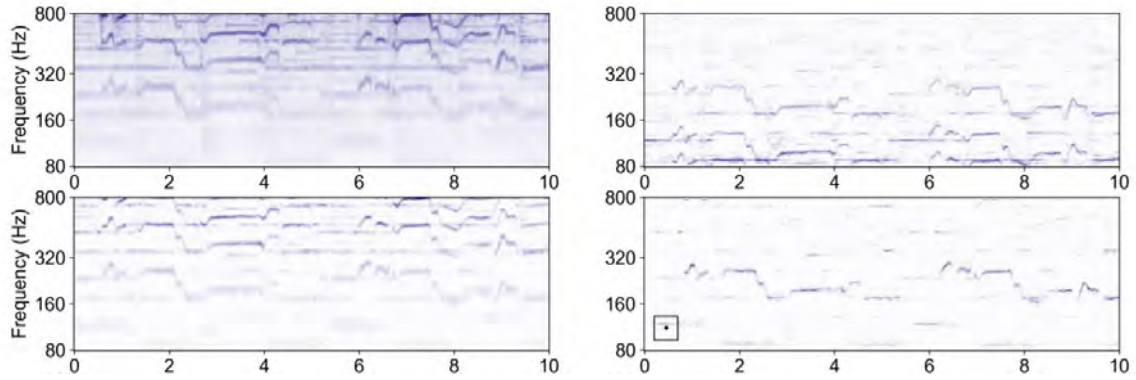
ภาพที่ 2.5.2 ก กราฟแสดงฮาร์โมนิกและฮาร์โมนิกย่อย

ที่มา : A dual-frequency excitation technique for enhancing the sub-harmonic emission from encapsulated microbubbles

ทำการเปลี่ยนโดเมนคิวเฟรนซี่ของสมการ Z_1 ให้เป็นโดเมนความถี่ จากนั้นนำโดเมนความถี่ของทั้ง Z_1 และ Z_2 มาทำ log-frequency scale ได้เป็นค่าดัชนี p และนำโดเมนความถี่ที่ได้และ โดเมนของเวลา ของทั้ง GC และ GCoS มาผ่านตัวกรอง W_f and W_t แล้วนำมารวมกันเป็นสมการดังนี้

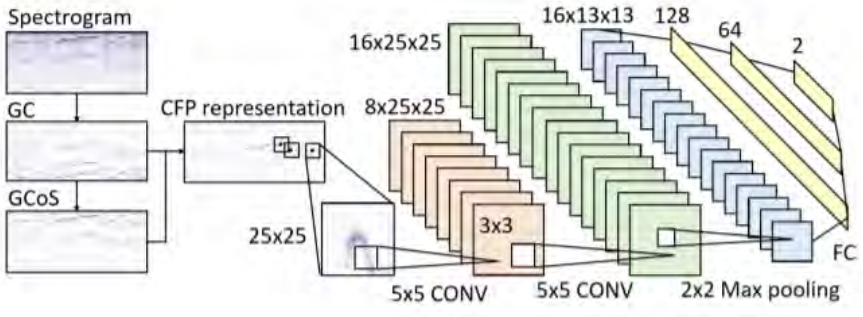
$$Y[p,n] = \tilde{Z}_1[p,n]\tilde{Z}_2[p,n]$$

สิ่งที่ได้จากการใช้ GC และ GCoS ร่วมกัน เรียกว่ากราฟของคาบและความถี่ (Combined Frequency and Periodicity - CFP) ดังภาพที่ 2.5.2 ข



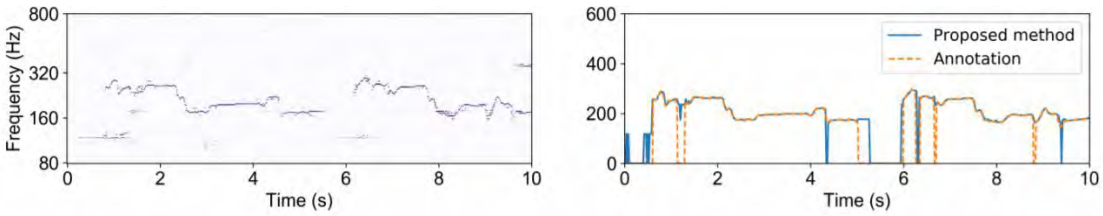
ภาพที่ 2.5.2 ข กราฟแสดง สเปกโทรแกรม (ซ้ายบน) GC (ขวาบน) GCoS (ซ้ายล่าง) CFP (ขวาล่าง)
ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

CFP เป็นกราฟที่สามารถระบุตำแหน่งระดับเสียงในโดเมนความถี่ได้ การจำแนกว่าจุดใดบนระนาบเวลา-ความถี่ เป็นเส้นรอบขอบ (contour) ของทำนองจากเสียงร้องนั้น จะตัดแบ่ง CFP เป็นชิ้นย่อย ๆ ขนาด 25x25 พิกเซล โดยกำหนดให้แต่ละชิ้นมีจุดศูนย์กลางตรงกับจุดสูงสุดของเส้นความถี่ที่ปรากฏบน CFP แล้วจึงส่งชิ้นย่อยเข้าไปยังตัวแบบ CNN ซึ่งมีชั้นสังวัตนาการ 2 ชั้น ชั้นแรกประกอบด้วยเคอร์เนลขนาด 5x5 จำนวน 8 เคอร์เนล และ ชั้นถัดไปประกอบด้วยเคอร์เนลขนาด 3x3 จำนวน 16 เคอร์เนล ถัดไปเป็นชั้นบ่อรวมค่าสูงสุด (max pooling) 1 ชั้น ขนาด 13x13 จำนวน 16 เคอร์เนล และชั้นสุดท้ายเป็นชั้นโครงข่ายแน่น ซึ่งประกอบด้วยนิวรอนจำนวน 128, 64 และ 2 นิวรอน ตามลำดับดังภาพที่ 2.5.2 ค ผลลัพธ์ที่ได้ของตัวแบบ CNN คือเวกเตอร์ ขนาด 2x1 ซึ่งแสดงถึงความน่าจะเป็นของการเป็นทำนองเพลงจากเสียงร้อง โดยกำหนดฟังก์ชันค่าเสียหาย (loss function) เป็น Cross-Entropy และใช้ขั้นตอนวิธี Adam ในการปรับค่าน้ำหนัก



ภาพที่ 2.5.2 ค สถาปัตยกรรมของตัวแบบการสกัดทำนองเพลง
ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

ผลการทดลองพบว่า การเลือกเอาดัชนความถี่ผลลัพธที่มีความน่าจะเป็นของทำนองเพลงจากเสียงร้องสูงที่สุด หรือ CNN Max-Out มีค่าความแม่นยำ (accuracy) ที่ 83.5% บนชุดข้อมูล MIREX2005 ซึ่งมีค่าความแม่นยำมากสุดเมื่อเทียบกับวิธีอื่น ๆ



ภาพที่ 2.5.2 ง กราฟแสดง CNN outputs (ซ้าย) ผลลัพธ์ CNN-MaxOut (ขวา)

ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

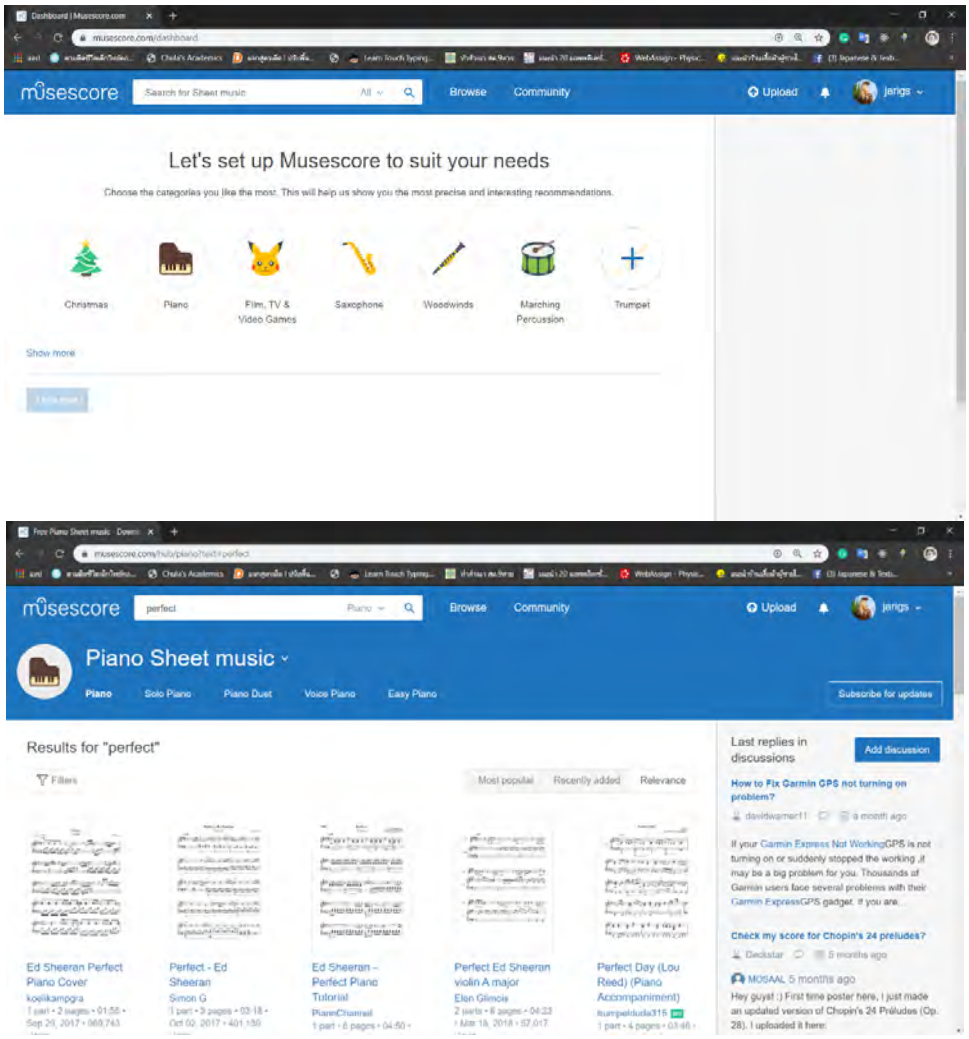
บทที่ 3

การจัดเตรียมและรวบรวมข้อมูล

ในบทนี้จะกล่าวถึง วิธีการที่นำไฟล์ MusicXML จากเว็บไซต์ musescore.com และการเตรียมข้อมูลเพลงก่อนนำไปใช้โดยข้อมูลเพลงที่จะนำไปให้เครื่องเรียนรู้จะถูกแปลงเป็นภาพสเปกโทรแกรมเพื่อให้ตัวแบบสามารถหาจุดเริ่มต้นของเสียงและทำนองหลักของเพลงได้อย่างเหมาะสม

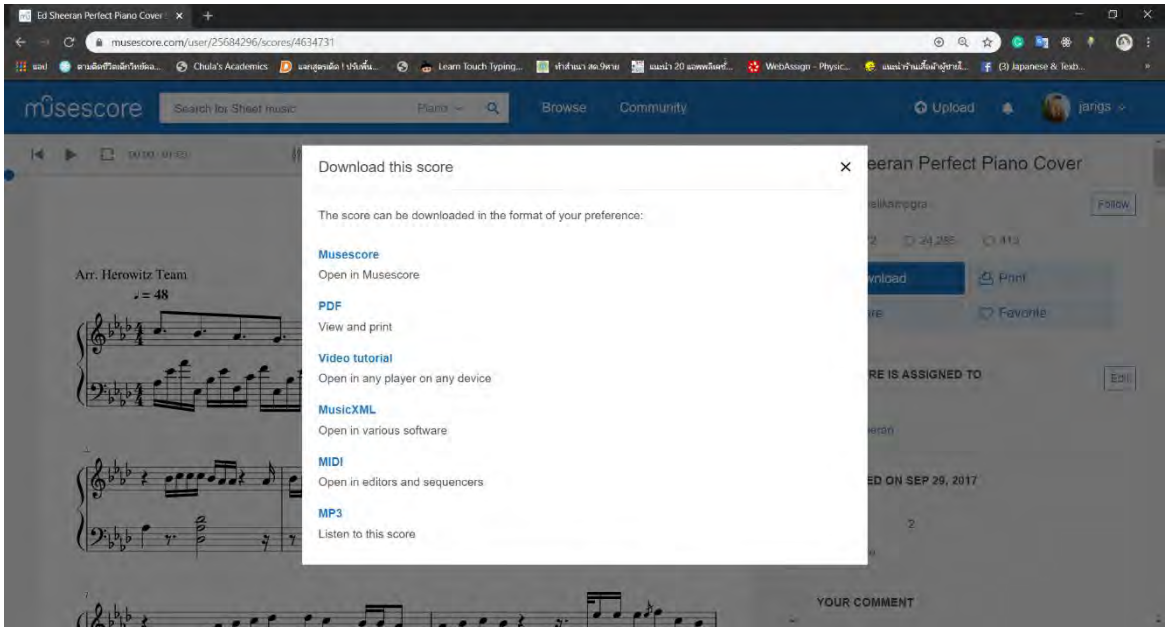
3.1 ขั้นตอนการเก็บข้อมูล

1. เข้าเว็บไซต์ musescore.com จากนั้นทำการ login เข้าสู่ระบบ และ ค้นหาเพลงที่ต้องการดาวน์โหลด โดยเลือก tag ค้นหาเป็นเพลงเปียโน



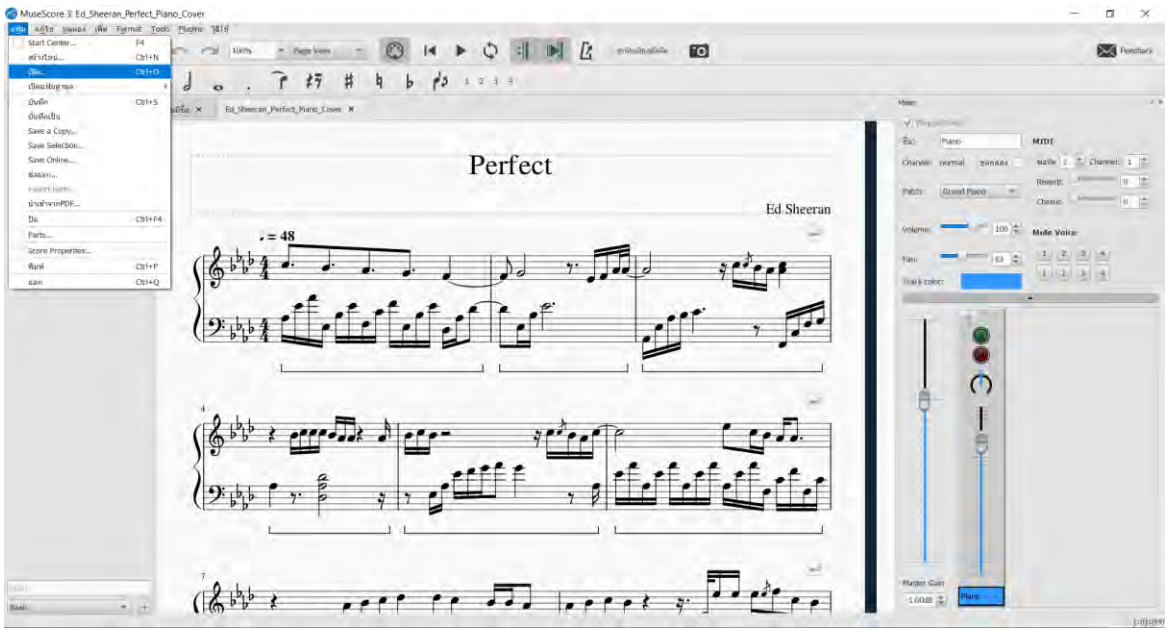
ภาพที่ 3.1 ก หน้า homepage หลังจาก login (บน) ผลลัพธ์ที่ได้จากการค้นหาเพลงเปียโน (ล่าง)

2. คลิกเลือกเพลงที่ต้องการ กดปุ่ม Download ฝั่งขวามือ เลือก MusicXML



ภาพที่ 3.1 ข ขั้นตอนการดาวน์โหลดไฟล์ MusicXML

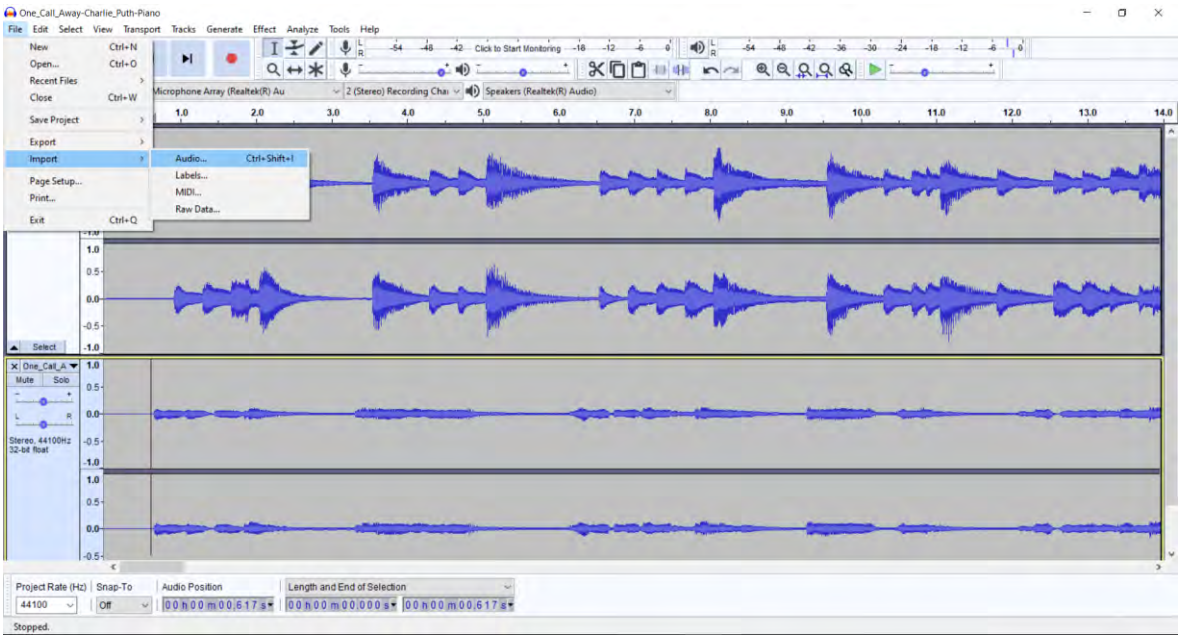
3. เปิดไฟล์ MusicXML ที่ดาวน์โหลดมาด้วยโปรแกรม Musescore ที่สามารถดาว์โหลดจากเว็บไซต์ musescore.com ดังนี้ แฟ้ม > เปิด > เลือกไฟล์ MusicXMLที่ต้องการเปิด



ภาพที่ 3.1 ค ขั้นตอนการเปิดไฟล์ MusicXML

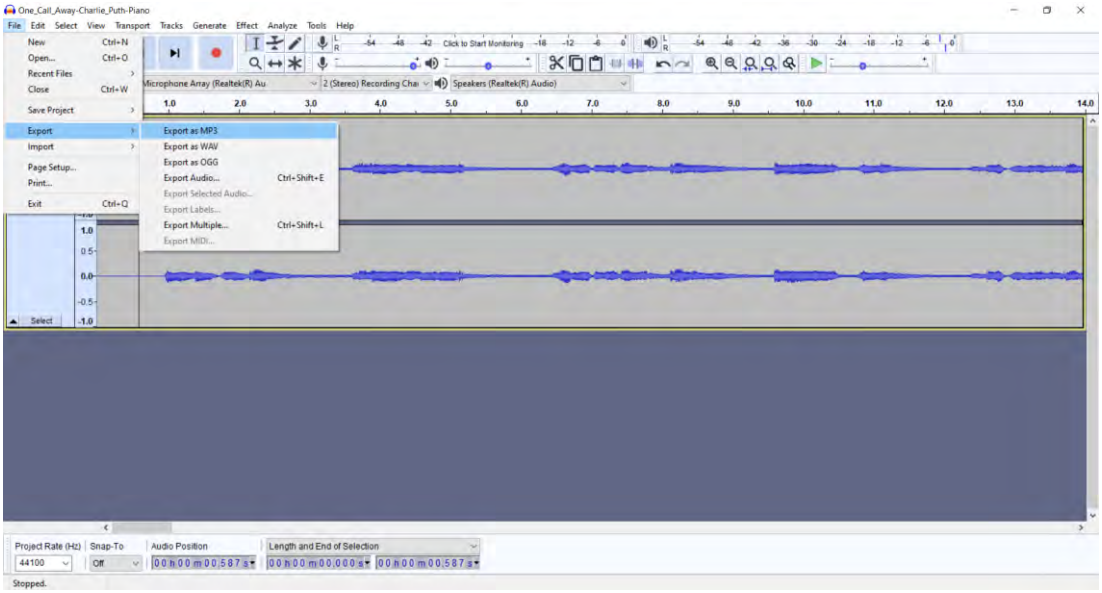
4. ทำการส่งออกไฟล์ mp3 และ MusicXML แบบ Uncompressed File ดังนี้
   แฟ้ม > ส่งออก > เลือก type ที่ต้องการในช่อง Save as type เป็น mp3 แล้วส่งออก และทำข้อ 4 นี้อีกครั้งหนึ่งโดยส่งออกไฟล์เป็น MusicXML แบบ Uncompressed File

5. เปิดโปรแกรม Audacity เลือกไฟล์ mp3 ที่ได้จาก Musecore และเพลงจริงที่มีเสียงนักร้องและเสียงดนตรีอื่น ๆ (vocal and other) ดังนี้ เลือก File > Import > Audio จากนั้นเลือกเพลงเปียโนและทำซ้ำอีกครั้งกลับเพลงจริง



ภาพที่ 3.1 ง ขั้นตอนการนำเข้าไฟล์ mp3 ของเพลงเปียโน (บน) และเพลงจริง (ล่าง) บนโปรแกรม Audacity

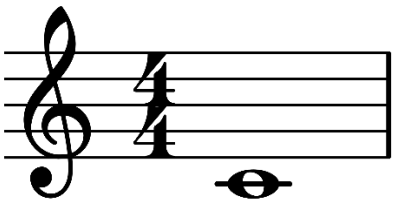
6. ทำการปรับจังหวะ (tempo) และ ยืดหดคลื่นของทั้งเพลงเปียโนและเพลงจริงให้ตรงกันมากที่สุดเพื่อให้สามารถกำหนดป้ายกำกับ (label) ของจุดเริ่มต้นเสียงกับทำนองเพลงได้
7. ลบแถบเพลงเปียโนหลังจากทำการปรับลูกคลื่นของทั้งสองเพลงให้ตรงกัน จากนั้นทำการส่งออกแถบเพลงจริงเป็นไฟล์ mp3 ดังนี้ เลือก File > Export > Export as MP3



ภาพที่ 3.1 จ ขั้นตอนการส่งออกไฟล์ mp3 ของเพลงจริง (ล่าง) บนโปรแกรม Audacity

3.2 MusicXML

MusicXML ย่อยมาจาก Music Extensible Markup Language คือภาษาหนึ่งที่ใช้ในการแสดงผลข้อมูลโน้ตดนตรี ซึ่งภาษาที่ใช้กำหนดรูปแบบของคำสั่งภาษา HTML หรือที่เรียกว่า Meta Data จะใช้สำหรับกำหนดรูปแบบของคำสั่ง Markup ต่าง ๆ หากเปรียบเทียบกับภาษา HTML จะแตกต่างกันที่ HTML ถูกออกแบบมาเพื่อการแสดงผลอย่างเดียวเท่านั้น อาทิเช่น การแสดงผลตัวเล็ก ตัวหนา ตัวเอียง ที่ปรากฏบนเว็บเพจทั่วไป ในขณะที่ภาษา MusicXML นั้นถูกออกแบบมาเพื่อเก็บข้อมูลและโครงสร้างของข้อมูลนั้น ๆ ไว้ด้วยกัน โดยจะมีโครงสร้างที่ประกอบด้วยแท็กเปิด และแท็กปิด เช่นเดียวกับภาษา HTML และ XML



ภาพที่ 3.2 ก สัญลักษณ์กำกับทางดนตรีบนโน้ตบรรทัดห้าเส้น

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
   "-//Recordare//DTD MusicXML 3.1 Partwise//EN"
   "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
 <part-list>
  <score-part id="P1">
   <part-name>Music</part-name>
  </score-part>
 </part-list>
 <part id="P1">
  <measure number="1">
   <attributes>
    <divisions>1</divisions>
    <key>
     <fifths>0</fifths>
    </key>
    <time>
     <beats>4</beats>
     <beat-type>4</beat-type>
    </time>
    <clef>
     <sign>G</sign>
     <line>2</line>
    </clef>
   </attributes>
   <note>
    <pitch>
     <step>C</step>
     <octave>4</octave>
    </pitch>
    <duration>4</duration>
    <type>whole</type>
   </note>
  </measure>
 </part>
</score-partwise>
```

ภาพที่ 3.2 ข MusicXML ของโน้ตบรรทัดห้าเส้นในภาพ 3.2 ก

บทที่ 4

วิธีพัฒนาระบบ

ในบทนี้จะกล่าวถึงขั้นตอนการดำเนินการพัฒนาเว็บแอปพลิเคชัน ซึ่งประกอบไปด้วยการพัฒนาระบบของเครื่องบริการหรือเซิร์ฟเวอร์ การพัฒนาส่วนติดต่อผู้ใช้ และ การพัฒนาส่วนการประมวลผลเสียงดนตรี

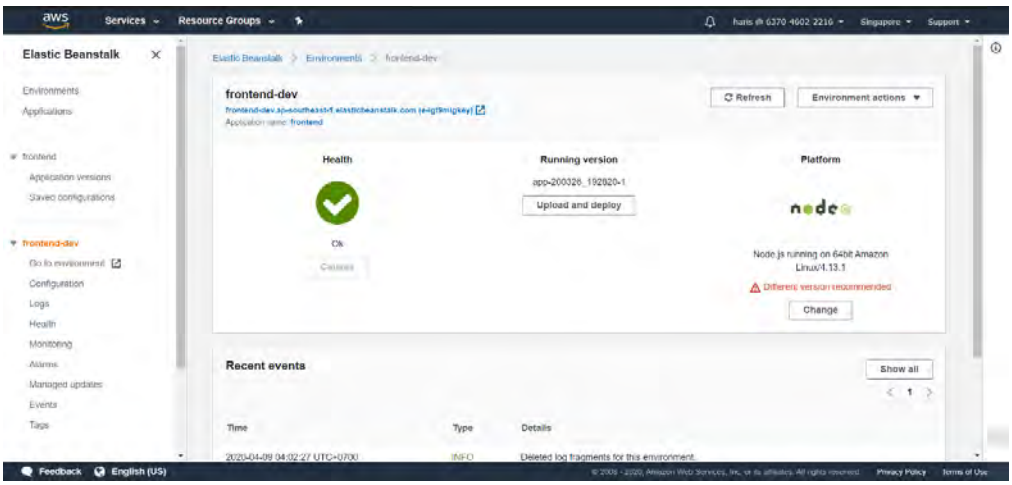
4.1 การพัฒนาระบบเครื่องบริการหรือเซิร์ฟเวอร์

Amazon web service เป็นแพลตฟอร์มคลาวด์ (Cloud Platform) ขนาดใหญ่ มีบริการที่หลากหลายไม่ว่าจะเป็นบริการเกี่ยวกับ ฐานข้อมูล (Database) การจัดเก็บข้อมูล (Storage) การเรียนรู้ของเครื่องและปัญญาประดิษฐ์ (Machine learning and AI) รวมไปถึงสิ่งที่จำเป็นสำหรับการพัฒนาเซิร์ฟเวอร์ทั้งในฝั่งหน้าบ้านและหลังบ้านของทีมพัฒนาเว็บแอปพลิเคชัน โดยสามารถแจกแจงบริการที่ทางผู้พัฒนาได้นำมาใช้ดังนี้

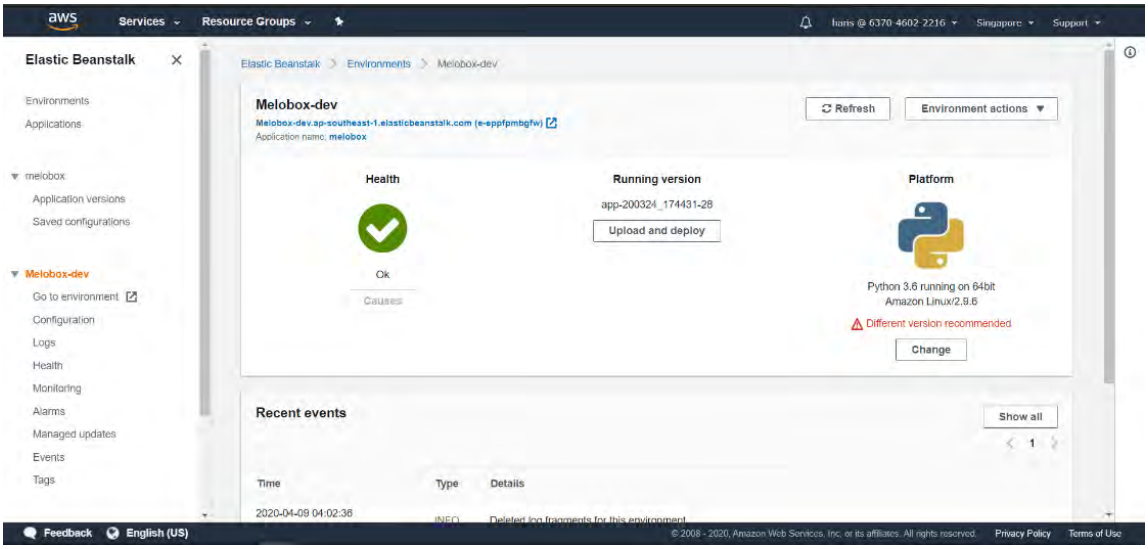
4.1.1 AWS Elastic Beanstalk

Elastic Beanstalk เป็นบริการสร้างเซิร์ฟเวอร์ที่ตั้งค่าเซิร์ฟเวอร์แบบอัตโนมัติ โดยให้เหมาะสมกับแอปพลิเคชันที่พัฒนาในแต่ละภาษาเช่น Python, PHP, Node.js เป็นต้น และยังมีการรองรับเว็บเซิร์ฟเวอร์ (Web server) อย่าง Apache, Nginx, Passenger และ IIS อีกด้วย

เนื่องด้วย React ถูกพัฒนาด้วย Node.js ที่เป็น JavaScript runtime และ Flask ถูกพัฒนาด้วยภาษา Python ทางทีมพัฒนาได้นำแอปพลิเคชันทั้งฝั่งหน้าบ้านและหลังบ้านอัปโหลดขึ้นบนสภาพแวดล้อมของ Elastic Beanstalk



ภาพที่ 4.1.1 ก เซิร์ฟเวอร์หน้าบ้านของ React บน Elastic Beanstalk



ภาพที่ 4.1.1 ข เซิร์ฟเวอร์หลังบ้านของ Flask บน Elastic Beanstalk

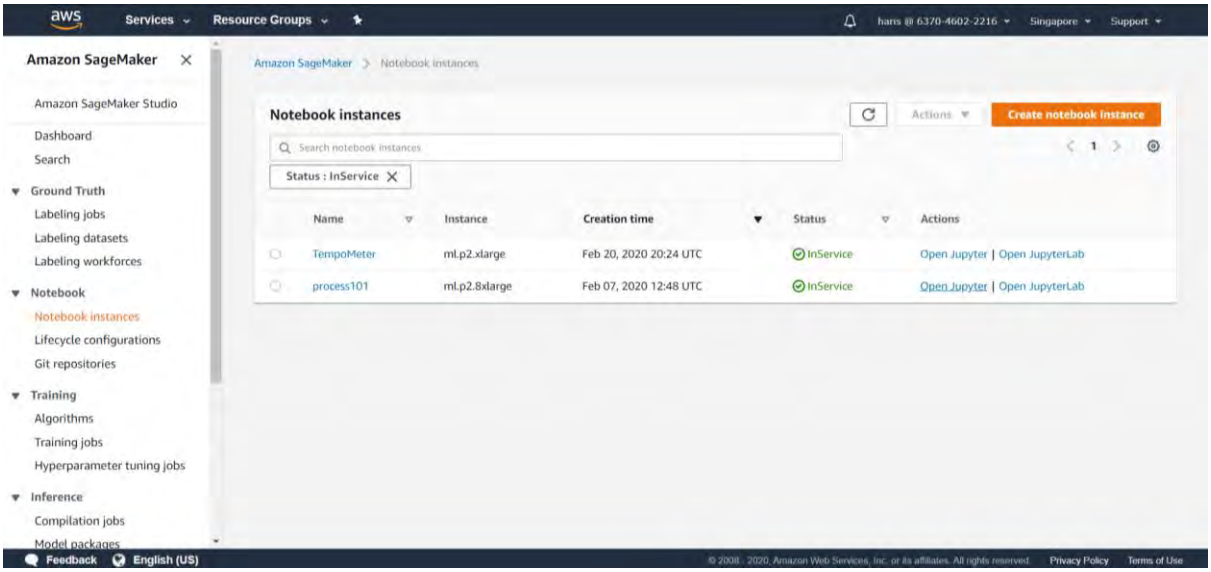
4.1.2 Amazon SageMaker

ทีมพัฒนาเลือกใช้ SageMaker เป็นสภาพแวดล้อมไว้สำหรับทำการเรียนรู้ของเครื่อง โดยใช้บริการผ่านตัวเครื่องโน้ตบุ๊ก (Notebook instances) ซึ่งรองรับเฟรมเวิร์คต่าง ๆ ที่ใช้กับการเรียนรู้ของเครื่องเช่น Tensorflow, Keras, Pytorch เป็นต้น และกำหนดตัวเครื่องโน้ตบุ๊ก เป็นประเภท ml.p2.8xlarge เพื่อให้เพียงพอต่อการฝึกสอนโครงข่ายประสาทเทียมเชิงลึกสำหรับการแปลงเพลงเป็นเพลงกล่องดนตรี

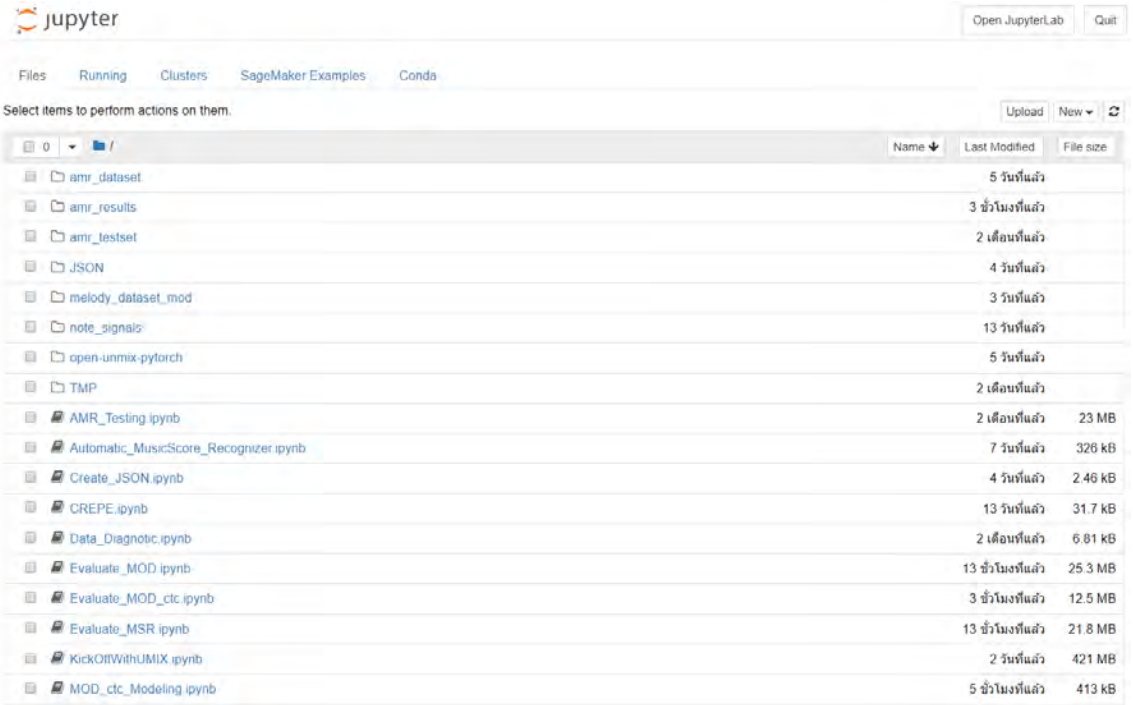
Amazon SageMaker Overview Features Pricing FAQs Developer Resources Customers

Accelerated Computing – Current Generation	vCPU	GPU	Mem (GiB)	GPU Mem (GiB)	Network Performance
ml.p3.2xlarge	8	1xV100	61	16	Up to 10 Gbps
ml.p3.8xlarge	32	4xV100	244	64	10 Gigabit
ml.p3.16xlarge	64	8xV100	488	128	25 Gigabit
ml.p3dn.24xlarge	96	8xV100	768	256	100 Gigabit
ml.p2.xlarge	4	1xK80	61	12	High
ml.p2.8xlarge	32	8xK80	488	96	10 Gigabit

ภาพที่ 4.1.2 ก แสดงรายละเอียดของตัวเครื่องโน้ตบุ๊กแบบ ml.p2.8xlarge (แถวล่างสุด)



ภาพที่ 4.1.2 ข แสดงรายชื่อตัวเครื่องโน้ตบุ๊กบน SageMaker



ภาพที่ 4.1.2 ค ตัวอย่างสภาพแวดล้อมหลังจากทำการเชื่อมต่อเครื่องโน้ตบุ๊ก

4.2 การพัฒนาส่วนติดต่อผู้ใช้งาน

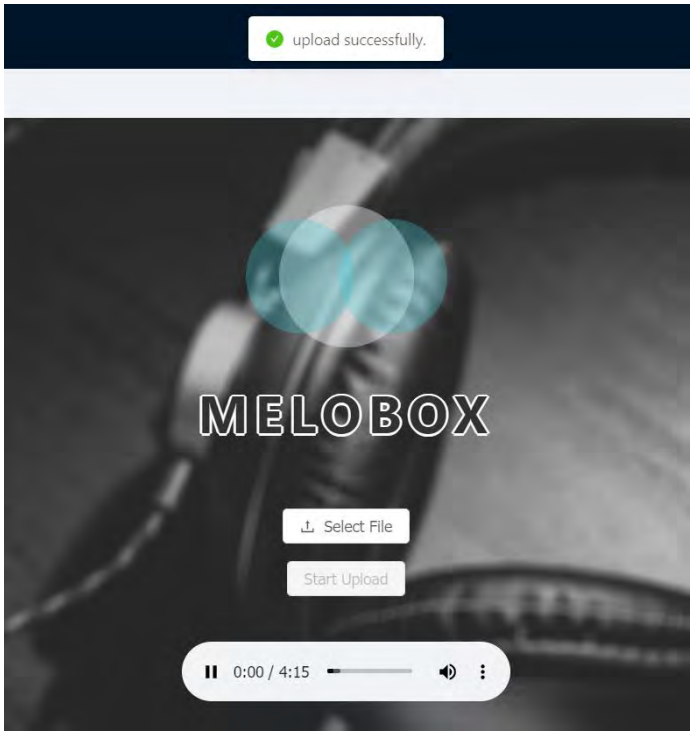
ภาพที่ 4.2 ก ตัวอย่างหน้าเว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี

เว็บแอปพลิเคชันของทางทีมพัฒนาได้ทำขึ้นเพื่อให้ผู้ใช้งานสามารถใช้บริการแปลงเพลงเป็นเพลงกล่องดนตรีได้อย่างสะดวกยิ่งขึ้นดังภาพที่ 4.2 ก โดยสามารถใช้งานได้ดังนี้

1. กดปุ่ม Select File เพื่อเลือกไฟล์เพลงนามสกุล mp3 หรือ wav ที่ต้องการ โดยสามารถทำการเปลี่ยนแปลงไฟล์ที่เลือกด้วยเลือกไฟล์ใหม่หรือกดปุ่มลบไฟล์ที่แนบไว้ได้ดังภาพที่ 4.2 ข

ภาพที่ 4.2 ข ตัวอย่างส่วนต่อประสานผู้ใช้ในการจัดการไฟล์ที่อัปโหลดบนเว็บไซต์

2. กดปุ่ม Start Upload เพื่ออัปโหลดไฟล์
3. เมื่ออัปโหลดสำเร็จ และทำการแปลงไฟล์เพลงเป็นเพลงกล่องดนตรีเสร็จสิ้นจะขึ้นกล่องแจ้งเตือนว่าอัปโหลดสำเร็จ และสามารถกดเล่นเพลงรวมถึงสามารถดาวน์โหลดไฟล์เพลงได้ที่ปุ่มสามจุดทางขวาสุดของแถบเล่นเพลง ดังภาพที่ 4.2 ค

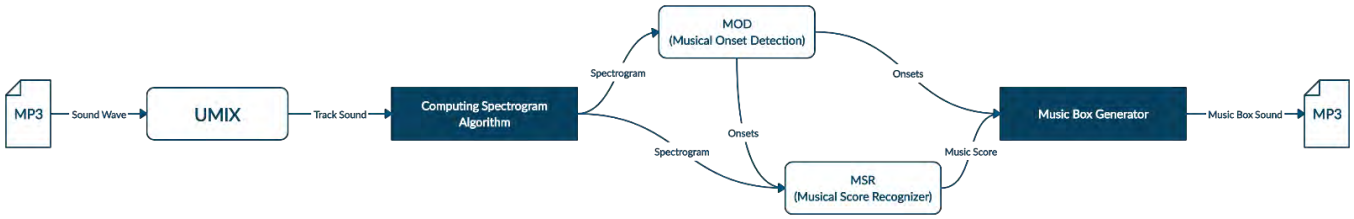


ภาพที่ 4.2 ค ตัวอย่างส่วนต่อประสานผู้ใช้เมื่อทำการอัปโหลดไฟล์เสร็จสิ้น

4.3 การพัฒนาส่วนการประมวลผลเสียงดนตรี

4.3.1 ภาพรวมการทำงาน

ส่วนการประมวลผลเสียงดนตรี เป็นส่วนหลักในการตรวจจับจุดเริ่มต้นเสียงและโน้ตดนตรีด้วยขั้นตอนวิธีการเรียนรู้ด้วยเครื่องเชิงลึก โดยมีภาพรวมการทำงานดังภาพต่อไปนี้



ภาพที่ 4.3.1 แผนภาพการทำงานของระบบที่พัฒนา

ส่วนการประมวลผลเสียงดนตรีเป็นการเรียกภาพรวมการทำงานทั้งหมดที่เกี่ยวข้องกับการตรวจจับจุดเริ่มต้นเสียงและโน้ตดนตรีเพื่อใช้เป็นข้อมูลในการประกอบเสียงกล่องดนตรี ซึ่งมีหลักการทำงานคร่าว ๆ คือ รับเสียงเพลงในรูปคลื่นเสียงแล้วส่งให้ UNMIX แยกเสียงตามประเภทออกเป็นแทร็กต่าง ๆ ได้แก่ เสียงร้อง, เสียงเบส, เสียงกลอง และ เสียงอื่น ๆ เพื่อเลือกรวมเสียงดังกล่าวตามความถี่เสียงที่เหมาะสมกับการตรวจจับโน้ตดนตรีแยกประเภทตามกุญแจประจำหลัก เช่น สเปกโทรแกรมของกุญแจประจำหลัก G จะเกิดจากการรวมคลื่นเสียงของแทร็กเสียงร้องและเสียงอื่น ๆ ในขณะที่สเปกโทรแกรมของกุญแจประจำหลัก F จะเกิดจากการรวมคลื่นเสียงของแทร็กเสียงกลอง เสียงเบส และ เสียงอื่น ๆ

เมื่อได้สเปกโทรแกรมตามจุดประสงค์ในการตรวจจับโน้ตดนตรีตามกุญแจประจำหลักแล้วจะส่งสเปกโทรแกรมดังกล่าวให้ตัวแบบที่แยกประเภทตามการทำงานเฉพาะอย่างโดยมีรายละเอียดการทำงานดังนี้

- MOD - Musical Onset Detector เป็นการบูรณาการการทำงานระหว่างโครงข่ายประสาทเทียมแบบเวียนซ้ำและโครงข่ายประสาทเทียมแบบสังวัตนาการ มีหน้าที่ในการตรวจจับจุดเริ่มต้นของเสียง เพื่อเป็นส่วนประกอบในการสร้างเสียงกล่องดนตรีด้วยตัวสร้างกล่องดนตรี (Music Box Generator)
- MSR – Musical Score Recognizer เป็นโครงข่ายประสาทเทียบแบบสังวัตนาการ มีหน้าที่ในการตรวจจับโน้ตดนตรี โดยรับจุดเริ่มต้นของเสียงจาก MOD เป็นข้อมูลรับเข้าอีกข้อมูลหนึ่งมีจุดประสงค์เพื่อการตวรจจับโน้ตดนตรีที่ปรากฏบนจุดเริ่มต้นของเสียง เมื่อได้โน้ตดนตรีในแต่ละจุดเริ่มต้นของเสียงแล้วจะถูกใช้เป็นข้อมูลรับเข้าของตัวสร้างกล่องดนตรีเพื่อใช้สร้างเสียงกล่องดนตรีในลำดับถัดไป

4.3.2 Open-Unmix

เนื่องจาก Open-Unmix เป็นตัวแบบโครงข่ายประสาทเทียมเชิงลึกที่สามารถทำการแยกเสียงแหล่งกำเนิดเสียงได้ [9] (Source Separation) เพื่อให้ตัวแบบ MOD และ MSR สามารถสกัดจุดเริ่มต้นของเสียงและโน้ตเพลงตามแยกแต่ละแทร็ค (track) ของเครื่องดนตรีที่ได้จากเพลงขาเข้า ทางผู้พัฒนาจึงนำโมเดลนี้มาใช้โดยสกัดเป็นเสียงคนร้อง (vocal) และเสียงอื่น ๆ (Bass, drum and other)

4.3.3 การรวบรวมข้อมูล

เริ่มต้นผู้พัฒนาได้กำหนดรายชื่อเพลงที่จะใช้ฝึกสอนตัวแบบไว้ทั้งหมด 533 เพลง แต่ปรากฏมีเพลงใน MueseScore ให้เก็บอยู่แค่ 103 เพลง จากนั้นจึงการตรวจสอบความเหมาะสมของเพลงทั้ง 103 เพลงปรากฏว่ามีเพลงที่สามารถใช้ฝึกสอนตัวแบบได้แค่ 75 จาก 533 เพลง เนื่องด้วยขั้นตอนวิธีที่ออกแบบมาสำหรับการถอดสัญลักษ์ต่าง ๆ ในไฟล์ MusicXML เกิดความผิดพาดในการถอดข้อมูลโดยไม่ทราบสาเหตุ และ บางเพลงมี tempo ในแต่ละห้องดนตรีแตกต่างกันทำให้เกิดความซับซ้อนและเกินความจำเป็นในการถอดข้อมูลมาเพียงเพื่อฝึกสอนตัวแบบการตรวจจับโน้ตดนตรี Music Box

ดังนั้นในการพัฒนานี้จะมีข้อมูลสำหรับฝึกสอนตัวแบบทั้งหมด 75 เพลง ใช้ทดสอบตัวแบบจำนวน 3 เพลง ซึ่งข้อมูลเหล่านี้จะถูกถอดข้อมูลจาก MusicXML ไปจัดเก็บไว้ในไฟล์ JSON ดังรูป (ใส่ด้วย 1) โดยภายในไฟล์ JSON จะมีข้อมูลที่ถูกเก็บในป้ายกำกับต่าง ๆ ดังนี้

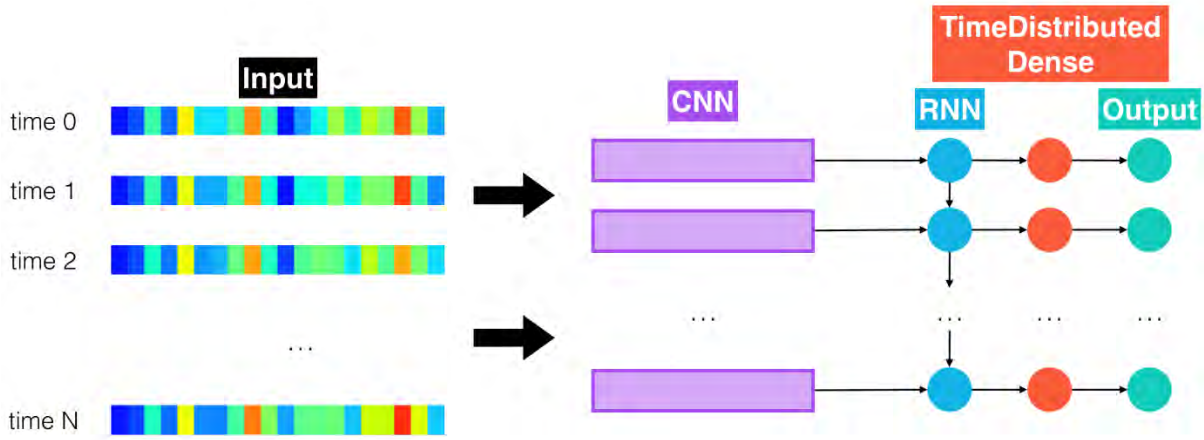
1. path จะเก็บ directory ของไฟล์เสียงที่จะใช้ฝึกสอนตัวแบบ
2. sample เอาไว้เก็บหมายเลขตัวอย่างเพื่อใช้ในการตัดไฟล์เสียงเป็นคลิปเสียงสั้น ๆ คลิปเสียงละ 20 วินาที ยกตัวอย่างเช่น sample = 0 คือตัดตั้งแต่วินาทีที่ 0-20, sample = 1 คือตัดตั้งแต่วินาทีที่ 21-40 เป็นต้น ครับ
3. total duration เก็บความยาวของไฟล์เสียงนั้น ๆ ทั้งหมดในหน่วย วินาที
4. g_clef_label เก็บ onsets และ โน้ตดนตรี ณ onset นั้น ๆ ของโน้ตดนตรีในกุญแจประจำหลัก G เพื่อใช้เป็น Ground Truth หรือ label data ในการฝึกสอนตัวแบบ
5. f_clef_label เก็บ onsets และ โน้ตดนตรี ณ onset นั้น ๆ ของโน้ตดนตรีในกุญแจประจำหลัก F เพื่อใช้เป็น Ground Truth หรือ label data ในการฝึกสอนตัวแบบ

{"path": "amr_dataset/142/Too_Good_at_Goodbyes-Sam_Smith", "sample": 0, "total_duration": 223.3469387755102, "duration": 20, "g_clef_label": "0.0-F3/A3/D4~2.61-F3/A3/D4~5.22-G3/A3/C4~6.52-G3/C4/E4~7.5-F4~7.83-F3/Bb3/D4~11.41-F4/C5~11.74-F4/C5~12.07-F4/C5~12.39-F4/C5~12.72-F4/D5~13.04-F4/C5~13.7-G4~15.65-F4/A4~16.96-C4/G4~17.28-A4~17.61-Bb4~17.93-A4~18.26-G4~18.59-F4~19.08-C4/F4~19.24-G4", "f_clef_label": "0.0-D2/A2/D3~2.61-D2/A2/D3~5.22-F2/C3/F3~7.83-C2/G2/C3~10.43-G2/D3/G3~13.04-D2/A2/D3~15.65-F2/C3/F3~18.26-C2/G2/C3"}

ภาพที่ 4.3.3 ลักษณะการเก็บข้อมูลในไฟล์ JSON

4.3.4 โครงสร้าง Musical Onset Detector (MOD)

ทางผู้จัดทำได้ออกแบบโครงข่ายประสาทเทียมเชิงลึก ซึ่งเป็นสถาปัตยกรรมร่วมกันระหว่างโครงข่ายประสาทเทียมแบบเวียนซ้ำ และ โครงข่ายประสาทเทียมแบบสังวัตนาการ โดยมีการทดลองปรับเปลี่ยนโครงสร้างย่อยภายในสถาปัตยกรรมดังกล่าวเพื่อจุดประสงค์ในการแสวงหาโครงสร้างที่มีความเหมาะสมกับการจับจุดเริ่มต้นเสียงที่มีประสิทธิภาพ ดังต่อไปนี้



ภาพที่ 4.3.4 ก แผนภาพสถาปัตยกรรมของตัวแบบ MOD

Layer (type)	Output Shape	Param #
the_input (InputLayer)	(None, None, 161)	0
conv1d (Conv1D)	(None, None, 900)	23329800
bn_conv_1d (BatchNormalizati	(None, None, 900)	3600
rnn (GRU)	(None, None, 200)	660600
bn_rnn_1d (BatchNormalizatio	(None, None, 200)	800
time_distributed_1 (TimeDist	(None, None, 1)	201
sigmoid (Activation)	(None, None, 1)	0

Total params: 23,995,001
Trainable params: 23,992,801
Non-trainable params: 2,200

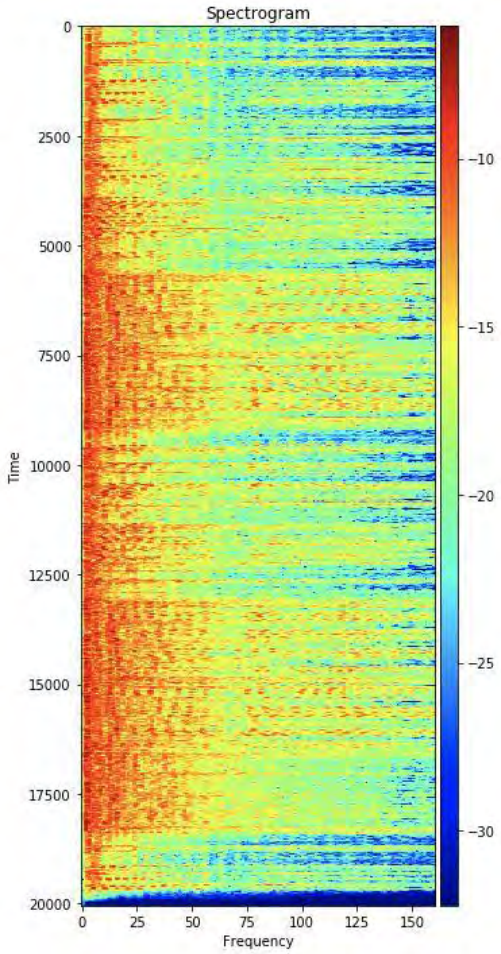
ภาพที่ 4.3.4 ข ชั้นของตัวแบบ MOD

1. CNN ตัวกรอง (kernel function) 200 แบบ ขนาด 1x80 – GRU จำนวน 200 หน่วย
2. CNN ตัวกรอง (kernel function) 200 แบบ ขนาด 1x161 – GRU จำนวน 200 หน่วย
3. CNN ตัวกรอง (kernel function) 900 แบบ ขนาด 1x80 – GRU จำนวน 900 หน่วย
4. CNN ตัวกรอง (kernel function) 900 แบบ ขนาด 1x161 – GRU จำนวน 900 หน่วย

หมายเหตุ เครื่องหมาย – แทนการแบ่งชั้นของชั้นซ้อนตัว (hidden layer) แบบต่าง ๆ
CNN คือ Convolution Neural Network
GRU คือ Gate Recurrent Unit

4.3.5 การเตรียมชุดฝึกสอน MOD

การสร้างข้อมูลนำเข้าของ MOD เริ่มต้นด้วยการนำไฟล์เสียงในข้อมูลชุดฝึกสอนทั้งหมดมาตัดแบ่งเป็นคลิปเสียงเท่า ๆ กันขนาด 20 วินาที ซึ่งกำหนดให้อัตราการสุ่ม หรือ sampling rate ของแต่ละคลิปเสียงเท่ากับ 22,400 Hz และ กำหนด window size เท่ากับ 20 ms ในการคำณวนขนาดของ FFT window ที่ใช้คำณวนสเปกโทรแกรม แล้วจึงนำคลิปเสียงทั้งหมดมาสร้างสเปกโทรแกรมที่มีความถี่เสียงและเวลาเป็นโดเมนและเรนจ์ตามลำดับ ซึ่งสเปกโทรแกรมของแต่ละคลิปเสียงจะถูกเก็บใน NumPy Array เพื่อเตรียมสำหรับการฝึกสอนตัวแบบ MOD ในลำดับถัดไป



ภาพที่ 4.3.5 สเปกโทรแกรมของคลิปเสียงที่ตัดออกมา

การสร้างข้อมูลตัวกำกับ หรือ ข้อมูลตัวเฉลย จะสอดคล้องกับมิติของข้อมูลส่งออกของตัวแบบ MOD ที่จะพิจารณาเป็นลำดับในแต่ละกรอบเวลา (time frame) ของสเปกโทรแกรม คือ ในแต่ละคลิปเสียงจะเก็บข้อมูลจุดเริ่มต้นเสียงด้วยจำนวนเต็ม 0 และ 1 ในตำแหน่งจุดเวลาที่เป็นจุดเริ่มต้นเสียง และ ตำแหน่งจุดเวลาที่ไม่เป็นจุดเริ่มต้นเสียง ตามลำดับ ซึ่งในแต่ละตำแหน่งเวลานี้จะถูกเก็บด้วย NumPy array ขนาด 1 มิติ โดยกำกับให้ตำแหน่งเวลาหนึ่ง ๆ แทนด้วยดัชนีของ NumPy array เช่น บนดัชนีที่ 3 ของ NumPy array จะเทียบเท่ากับเวลา 0.03 วินาที ในเวลาของคลิปเสียง

4.3.6 วิธีวัดผลตัวแบบ MOD

ประเมิลผลด้วยค่า Recall โดยกำหนดในจุดเริ่มต้นของเสียงที่ตัวแบบทำนายได้มีความคลาดเคลื่อนไม่เกิน 30 ms โดยมีสมการดังนี้

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

โดยที่

- $True\ Positive$ คือ จำนวนจุดเวลาที่ทำนายถูกว่าเป็นจุดเริ่มต้นเสียงโดยสามารถคลาดเคลื่นได้ไม่เกิน $\pm 30\ ms$
- $False\ Negative$ คือ จำนวนจุดเวลาที่ทำนายผิดว่าไม่เป็นจุดเริ่มต้นเสียง

4.3.7 โครงสร้าง Musical Score Recognizer (MSR)

ทางผู้จัดทำได้ออกแบบโครงข่ายประสาทเทียมเชิงลึก ซึ่งเป็นสถาปัตยกรรมโครงข่ายประสาทเทียมแบบสังวัตนาการ โดยมีโครงสร้างที่ถอดแบบมาจาก VGGNet ซึ่งถูกนำเสนอเป็นครั้งแรกโดย Simonyan and Zisserman ในปี ค.ศ. 2014 [10] โดยมีการทดลองปรับเปลี่ยนโครงสร้างย่อยภายในสถาปัตกรรมดังกล่าวเพื่อจุดประสงค์ในการแสวงหาโครงสร้างที่มีความเหมาะสมกับการจับโน้ตดนตรีที่มีประสิทธิภาพ ดังต่อไปนี้

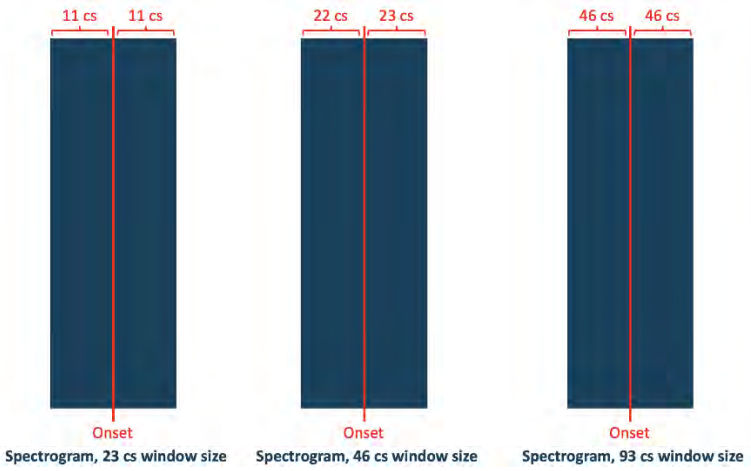
Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 161, 10, 32)	2048
activation_8 (Activation)	(None, 161, 10, 32)	0
batch_normalization_7 (Batch	(None, 161, 10, 32)	128
max_pooling2d_4 (MaxPooling2	(None, 53, 10, 32)	0
dropout_5 (Dropout)	(None, 53, 10, 32)	0
conv2d_7 (Conv2D)	(None, 53, 10, 64)	43072
activation_9 (Activation)	(None, 53, 10, 64)	0
batch_normalization_8 (Batch	(None, 53, 10, 64)	256
conv2d_8 (Conv2D)	(None, 53, 10, 64)	86080
activation_10 (Activation)	(None, 53, 10, 64)	0
batch_normalization_9 (Batch	(None, 53, 10, 64)	256
max_pooling2d_5 (MaxPooling2	(None, 26, 10, 64)	0
dropout_6 (Dropout)	(None, 26, 10, 64)	0
conv2d_9 (Conv2D)	(None, 26, 10, 128)	172160
activation_11 (Activation)	(None, 26, 10, 128)	0
batch_normalization_10 (Batc	(None, 26, 10, 128)	512
conv2d_10 (Conv2D)	(None, 26, 10, 128)	344192
activation_12 (Activation)	(None, 26, 10, 128)	0
batch_normalization_11 (Batc	(None, 26, 10, 128)	512
max_pooling2d_6 (MaxPooling2	(None, 13, 10, 128)	0
dropout_7 (Dropout)	(None, 13, 10, 128)	0
flatten_2 (Flatten)	(None, 16640)	0
dense_3 (Dense)	(None, 1024)	17040384
activation_13 (Activation)	(None, 1024)	0
batch_normalization_12 (Batc	(None, 1024)	4096
dropout_8 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 56)	57400
activation_14 (Activation)	(None, 56)	0

Total params: 17,751,096
Trainable params: 17,748,216
Non-trainable params: 2,880

ภาพที่ 4.3.7 ชั้นของตัวแบบ MSR

4.3.8 การเตรียมชุดฝึกสอน MSR

การสร้างข้อมูลนำเข้าของ MSR จะสร้างสเปกโทรแกรมจากจุดเริ่มต้นของเสียงที่ได้จากข้อมูลตัวกำกับโดยกำหนดให้จุดเริ่มต้นของเสียงเป็นจุดกึ่งกลางของสเปกโทรแกรม แล้วทำการสร้างสเปกโทรแกรม 3 แบบ แยกตามขนาดหน้าต่าง (window size) ได้แก่ 230 ms, 460 ms และ 930 ms ดังภาพที่ 4.3.8 เมื่อได้สเปกโทรแกรมที่มีขนาดหน้าต่างที่ต่างกัน 3 สเปกโทรแกรมจึงรวมสเปกโทรแกรมทั้งสามเข้าด้วยกันเป็นสเปกโทรแกรม 3 แชนเนล (3 channel spectrogram) ใช้สำหรับฝึกสอนตัวแบบในลำดับถัดไป



ภาพที่ 4.3.8 การตัดแบ่งภาพสเปกโทรแกรมตามขนาดต่าง ๆ

การสร้างข้อมูลตัวกำกับ หรือ ข้อมูลตัวเฉลย จะสอดคล้องกับมิติของข้อมูลส่งออกของตัวแบบ MSR คือ ในแต่ละคลื่นเสียงจะเก็บข้อมูลโน้ตดนตรีด้วยเวกเตอร์ฐานสอง (binary vector) ในแต่ละตำแหน่งเวลาด้วย NumPy array ขนาด 2 มิติ โดยกำกับให้ตำแหน่งเวลาหนึ่ง ๆ แทนด้วยดัชนีของ NumPy array เช่น บนดัชนีที่ 3 ของ NumPy array จะเทียบเท่ากับเวลา 0.03 วินาที ในเวลาของคลิปเสียง

บทที่ 5

ผลการวิจัย

ใบบทนี้จะกล่าวถึง ผลของการดำเนินการวิจัยของตัวแบบการเรียนรู้เชิงลึกทั้ง 2 แบบ คือ Musical Onset Detector และ Musical Score Recognizer และ สรุปผลการวิจัยทั้งหมดดังนี้

5.1 การทดสอบ Musical Onset Detector (MOD)

หมายเหตุ 1.) ค่า Recall ของการประเมิลผล MOD โดยส่วนใหญ่จะมีค่าค่อนข้างสูง เนื่องจากจำนวนจุดเวลาที่ไม่ใช่จุดเริ่มต้นเสียงมีปริมาณมากกว่าจุดเวลาของจุดเริ่มต้นเสียงค่อนข้างมาก ดังนั้นจึงทำให้การทำนายจุดเวลาที่ไม่ใช่จุดเริ่มต้นเสียงของตัวแบบในปริมาณมาก ๆ จะทำให้ค่า Recall สูงตามไปด้วย

2.) MOD Type 1 คือ CNN ตัวกรอง (kernel function) 200 แบบ ขนาด 1x80 – GRU จำนวน 200 หน่วย

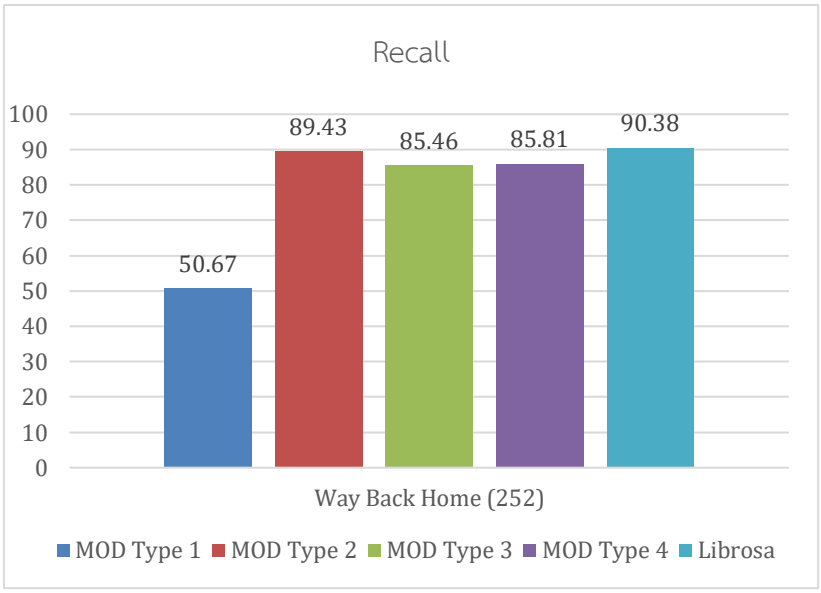
3.) MOD Type 2 คือ CNN ตัวกรอง (kernel function) 200 แบบ ขนาด 1x161 – GRU จำนวน 200 หน่วย

4.) MOD Type 3 คือ CNN ตัวกรอง (kernel function) 900 แบบ ขนาด 1x80 – GRU จำนวน 900 หน่วย

5.) MOD Type 4 คือ CNN ตัวกรอง (kernel function) 900 แบบ ขนาด 1x161 – GRU จำนวน 900 หน่วย

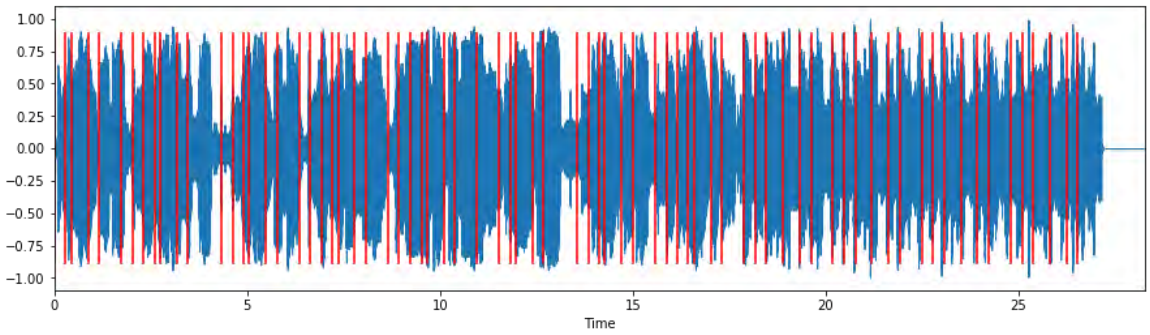
6.) คลังโปรแกรม Librosa [4] เป็นคลังโปรแกรมเกี่ยวกับการวิเคราะห์เสียงและดนตรี โดยใช้ Librosa ในการหาจุดเริ่มต้นของเสียงในข้อมูลชุดทดสอบ และนำจุดเริ่มต้นของเสียงที่ได้มาทำการประเมินผลร่วมกับตัวแบบ MOD ทั้ง 4 แบบ

5.1.1 ผลการทดสอบด้วยเพลง Way Back Home (252)

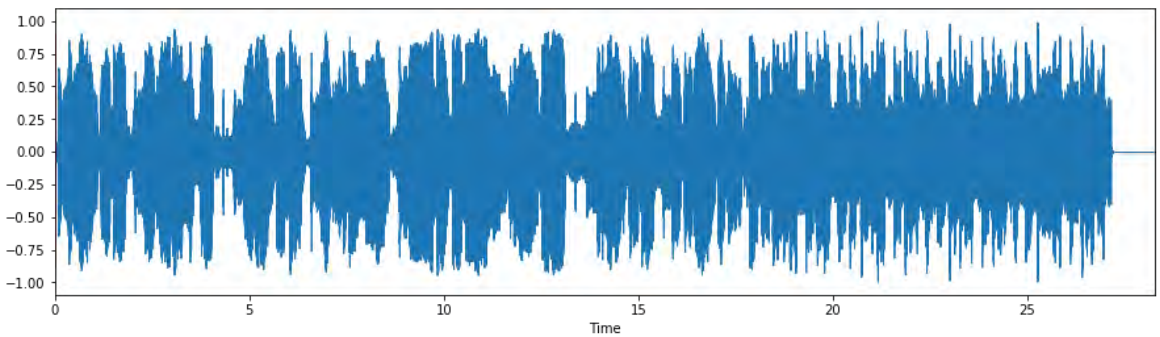


ภาพที่ 5.1.1 ก กราฟแสดงค่า Recall ของแต่ละตัวแบบ MOD บนเพลง Way Back Home

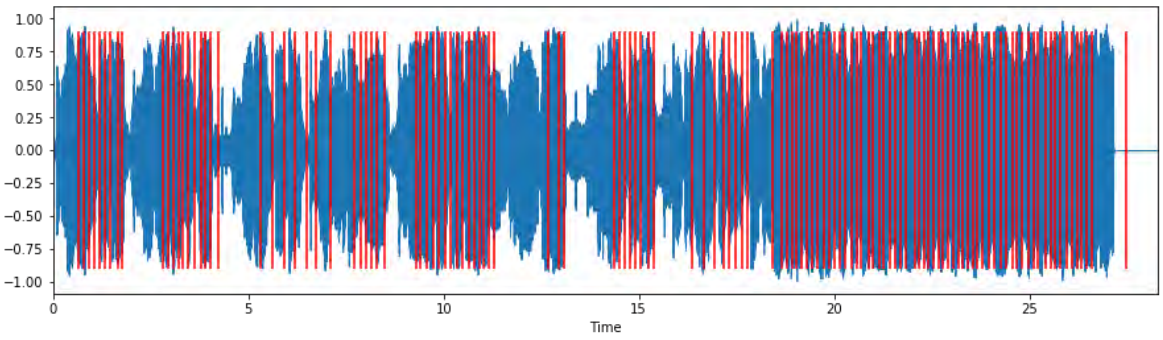
เพื่อให้การประเมินจากค่า Recall มีความชัดเจนมากยิ่งขึ้น ดังนั้นผู้ทดสอบจึงทำการวาดกราฟคลื่นเสียงและจุดเริ่มต้นของเสียงโดยกำหนดให้จุดเริ่มต้นเสียง ณ จุดเวลาต่าง ๆ แทนด้วยขีดแนวตั้งสีแดง ดังตาราง 5.1.1 ซึ่งเป็นกราฟที่อ้างอิงข้อมูลจากข้อมูลชุดทดสอบและกราฟจากตัวแบบในประเภทต่าง ๆ ดังภาพ 5.1.1 ข, 5.1.1 ค, 5.1.1 ง, 5.1.1 จ, 5.1.1 ฉ และ 5.1.1 ช



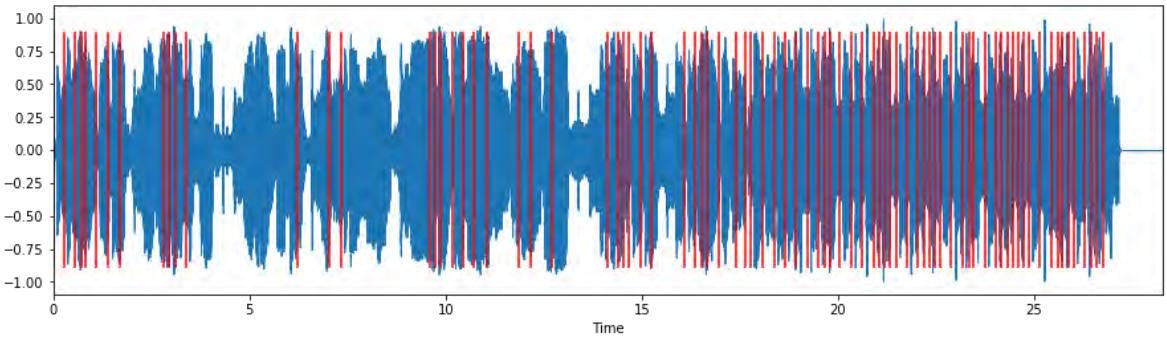
ภาพที่ 5.1.1 ข จุดเริ่มต้นเสียงของข้อมูลชุดทดสอบ



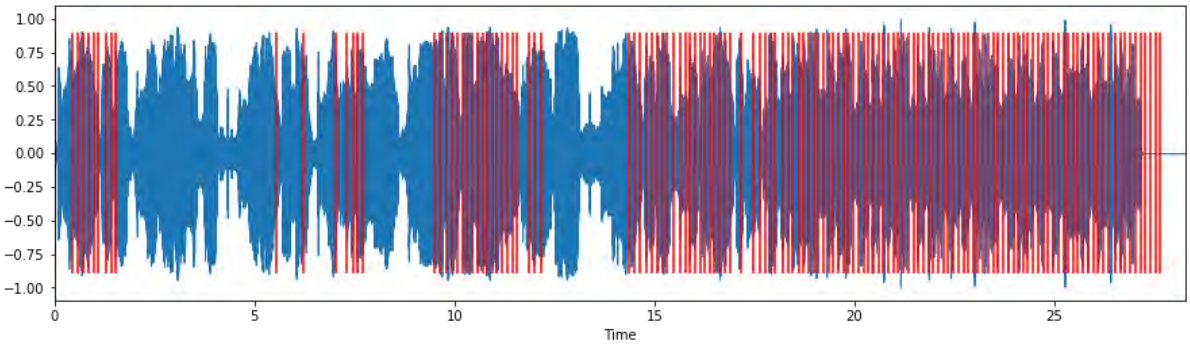
ภาพที่ 5.1.1 ค จุดเริ่มต้นเสียงของ MOD Type 1



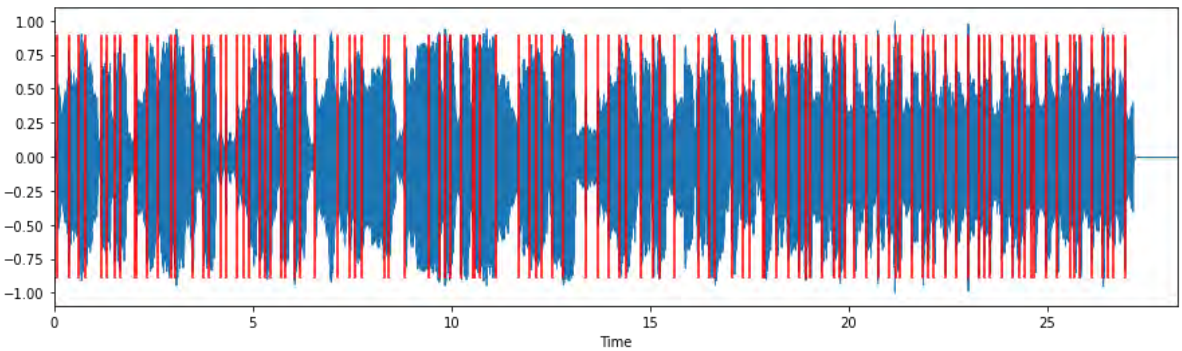
ภาพที่ 5.1.1 ง จุดเริ่มต้นเสียงของ MOD Type 2



ภาพที่ 5.1.1 จ จุดเริ่มต้นเสียงของ MOD Type 3

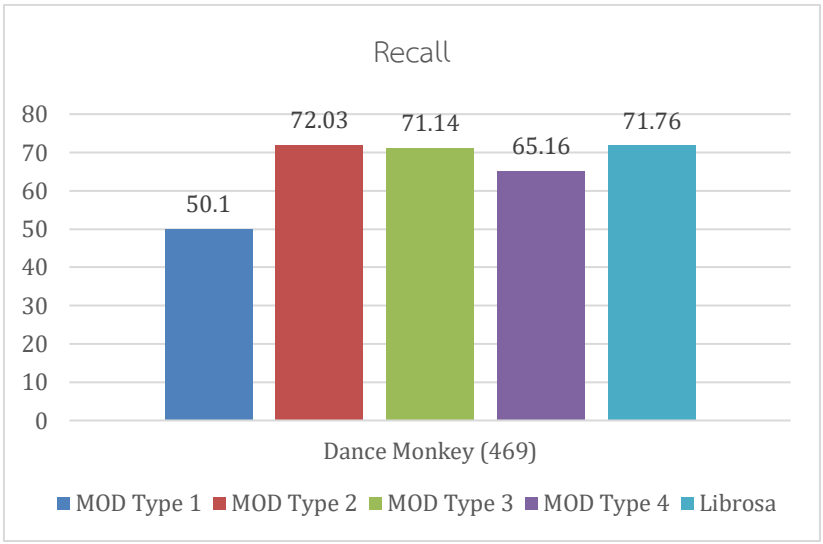


ภาพที่ 5.1.1 ฉ จุดเริ่มต้นเสียงของ MOD Type 4



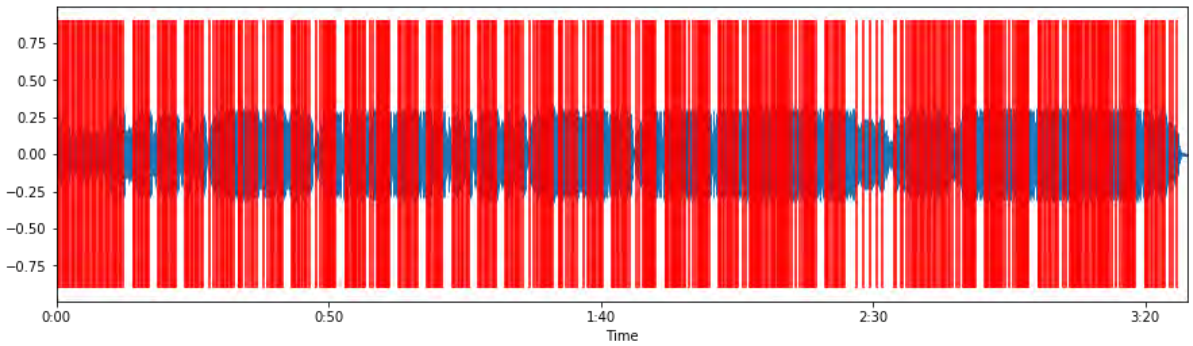
ภาพที่ 5.1.1 ช จุดเริ่มต้นเสียงของ Librosa

5.1.2 ผลการทดสอบด้วยเพลง Dance Monkey (469)

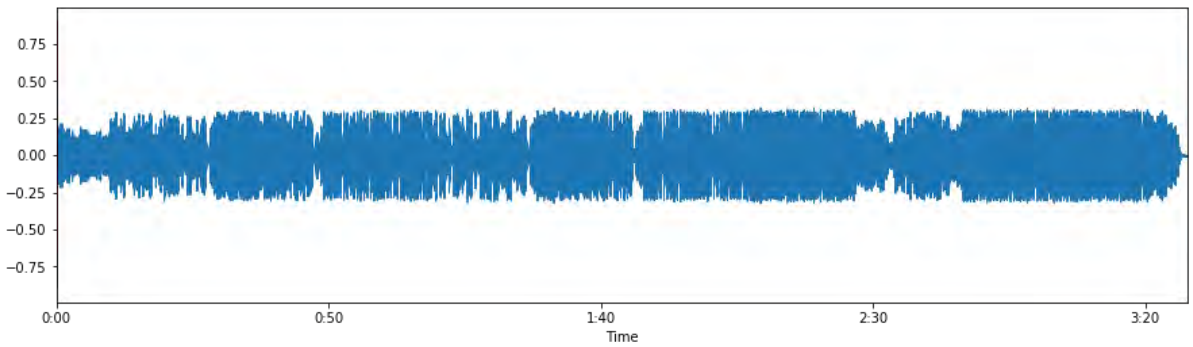


ภาพที่ 5.1.2 ก กราฟแสดงค่า Recall ของแต่ละตัวแบบ MOD บนเพลง Dance Monkey

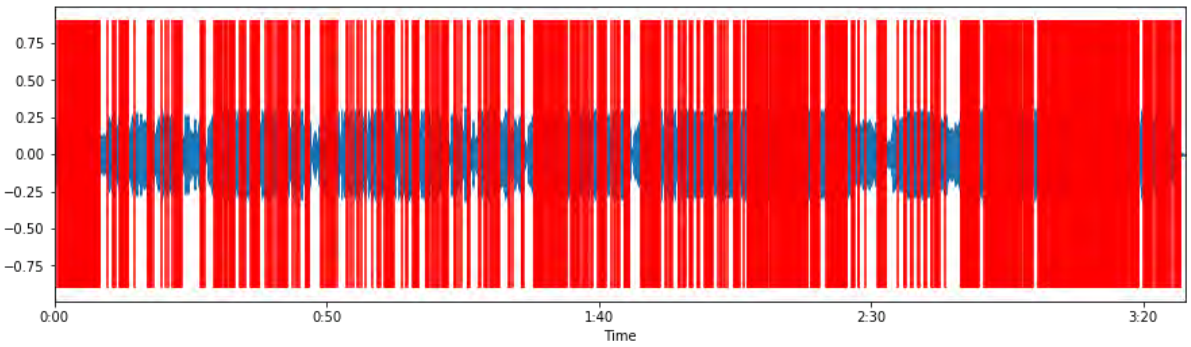
เพื่อให้การประเมินจากค่า Recall มีความชัดเจนมากยิ่งขึ้น ดังนั้นผู้ทดสอบจึงทำการวาดกราฟคลื่นเสียงและจุดเริ่มต้นของเสียงโดยกำหนดให้จุดเริ่มต้นเสียง ณ จุดเวลาต่าง ๆ แทนด้วยขีดแนวตั้งสีแดง ดังตาราง 5.1.2 ซึ่งเป็นกราฟที่อ้างอิงข้อมูลจากข้อมูลชุดทดสอบและกราฟจากตัวแบบในประเภทต่าง ๆ ดังภาพ 5.1.1 ข, 5.1.1 ค, 5.1.1 ง, 5.1.1 จ, 5.1.1 ฉ และ 5.1.1 ช



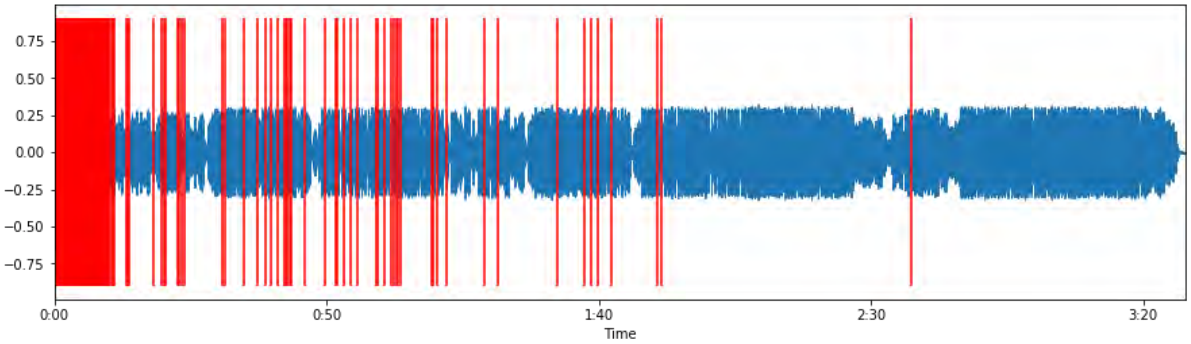
ภาพที่ 5.1.2 ข จุดเริ่มต้นเสียงของข้อมูลชุดทดสอบ



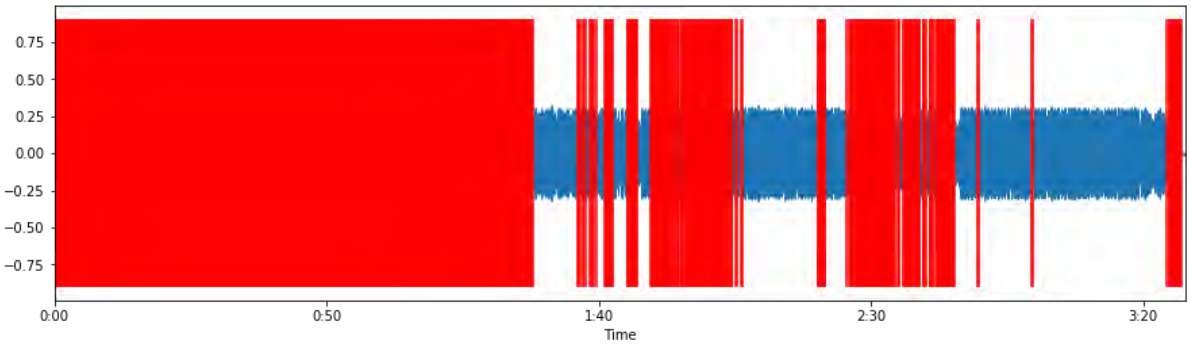
ภาพที่ 5.1.2 ค จุดเริ่มต้นเสียงของ MOD Type 1



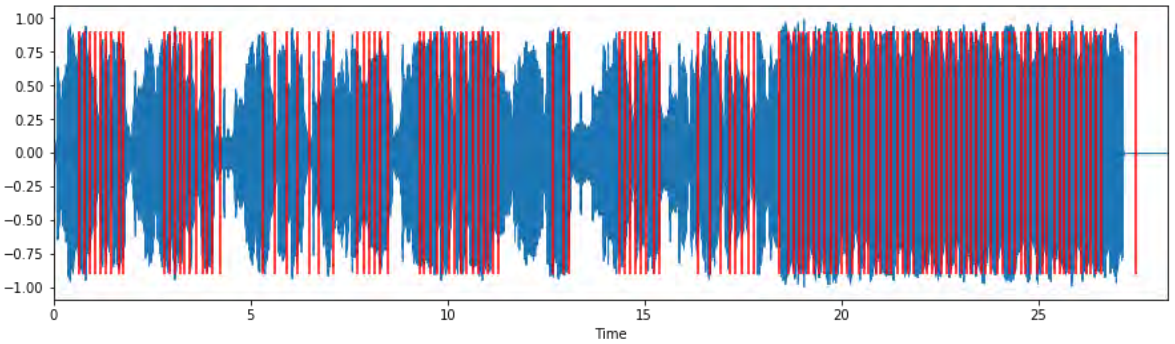
ภาพที่ 5.1.2 ง จุดเริ่มต้นเสียงของ MOD Type 2



ภาพที่ 5.1.2 จ จุดเริ่มต้นเสียงของ MOD Type 3

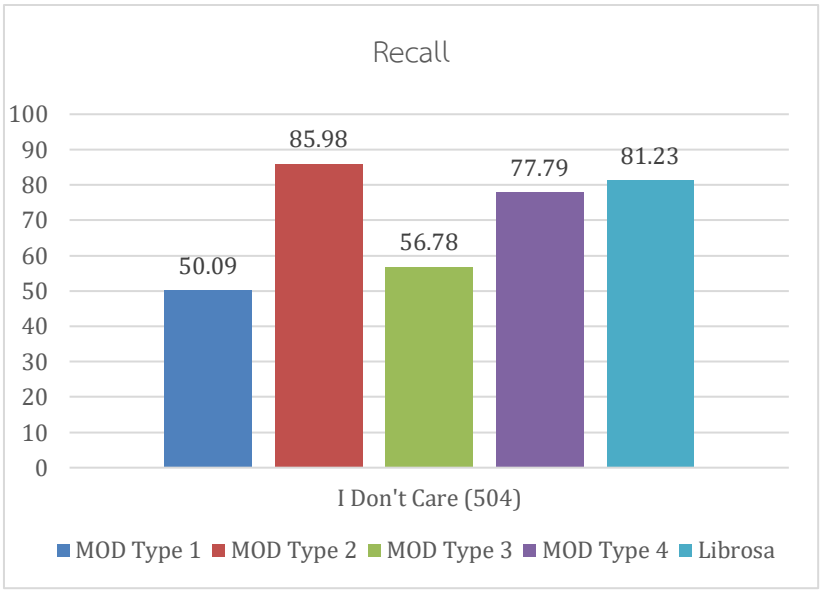


ภาพที่ 5.1.2 ฉ จุดเริ่มต้นเสียงของ MOD Type 4



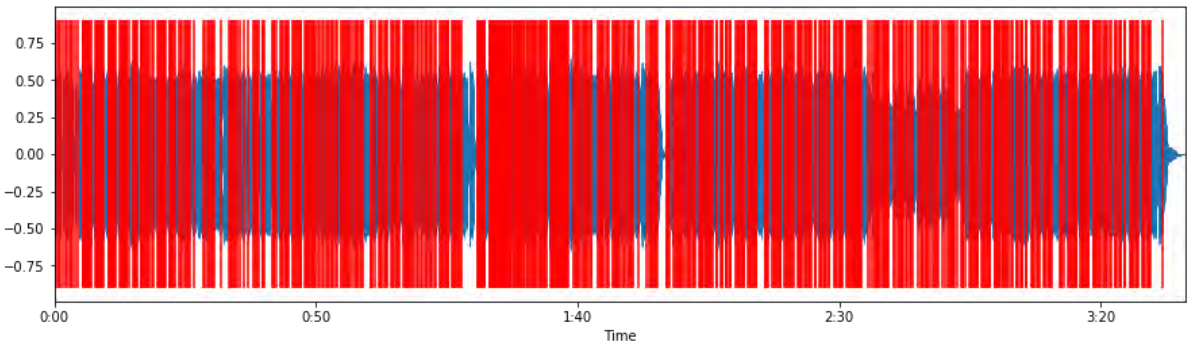
ภาพที่ 5.1.2 ช จุดเริ่มต้นเสียงของ Librosa

5.1.3 ผลการทดสอบด้วยเพลง I don’t care (504)

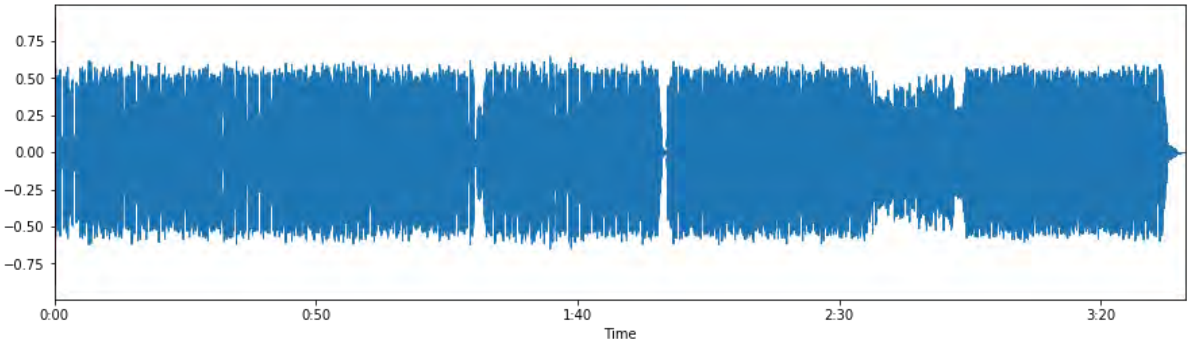


ภาพที่ 5.1.3 ก กราฟแสดงค่า Recall ของแต่ละตัวแบบ MOD บนเพลง I Don’t Care

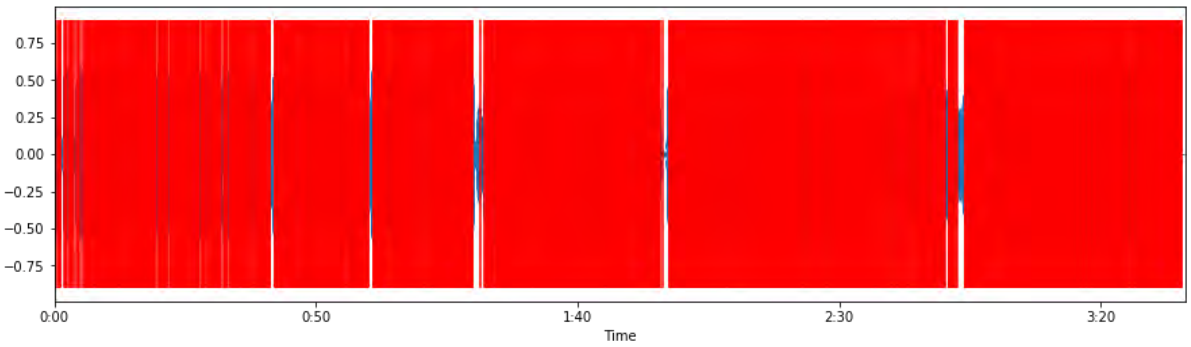
เพื่อให้การประเมินจากค่า Recall มีความชัดเจนมากยิ่งขึ้น ดังนั้นผู้ทดสอบจึงทำการวาดกราฟคลื่นเสียงและจุดเริ่มต้นของเสียงโดยกำหนดให้จุดเริ่มต้นเสียง ณ จุดเวลาต่าง ๆ แทนด้วยขีดแนวตั้งสีแดง ดังตาราง 5.1.3 ซึ่งเป็นกราฟที่อ้างอิงข้อมูลจากข้อมูลชุดทดสอบและกราฟจากตัวแบบในประเภทต่าง ๆ ดังภาพ 5.1.1 ข, 5.1.1 ค, 5.1.1 ง, 5.1.1 จ, 5.1.1 ฉ และ 5.1.1 ซ



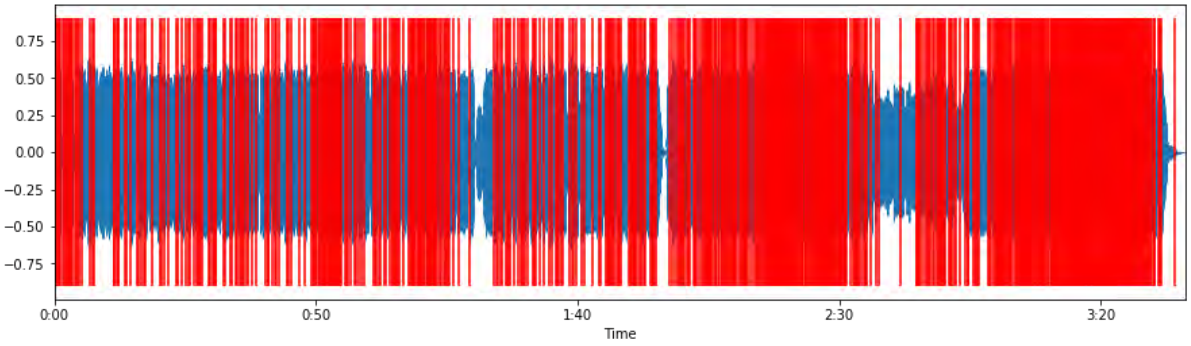
ภาพที่ 5.1.3 ข จุดเริ่มต้นเสียงของข้อมูลชุดทดสอบ



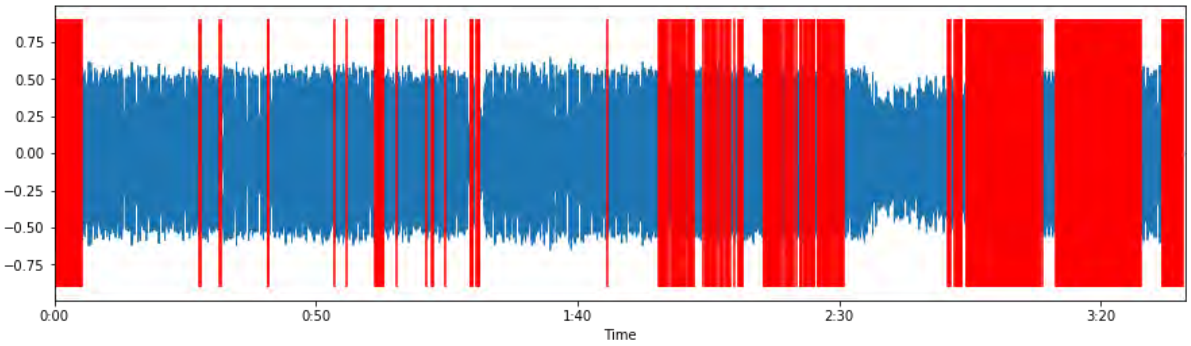
ภาพที่ 5.1.3 ค จุดเริ่มต้นเสียงของ MOD Type 1



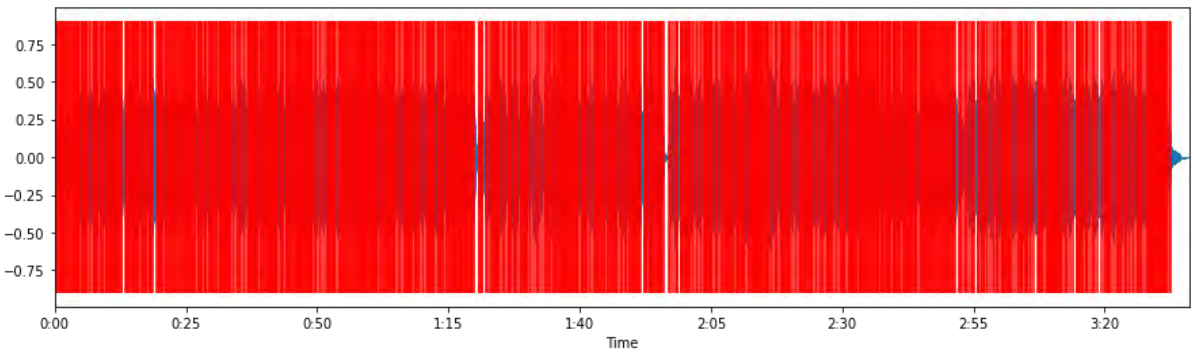
ภาพที่ 5.1.3 ง จุดเริ่มต้นเสียงของ MOD Type 2



ภาพที่ 5.1.3 จ จุดเริ่มต้นเสียงของ MOD Type 3



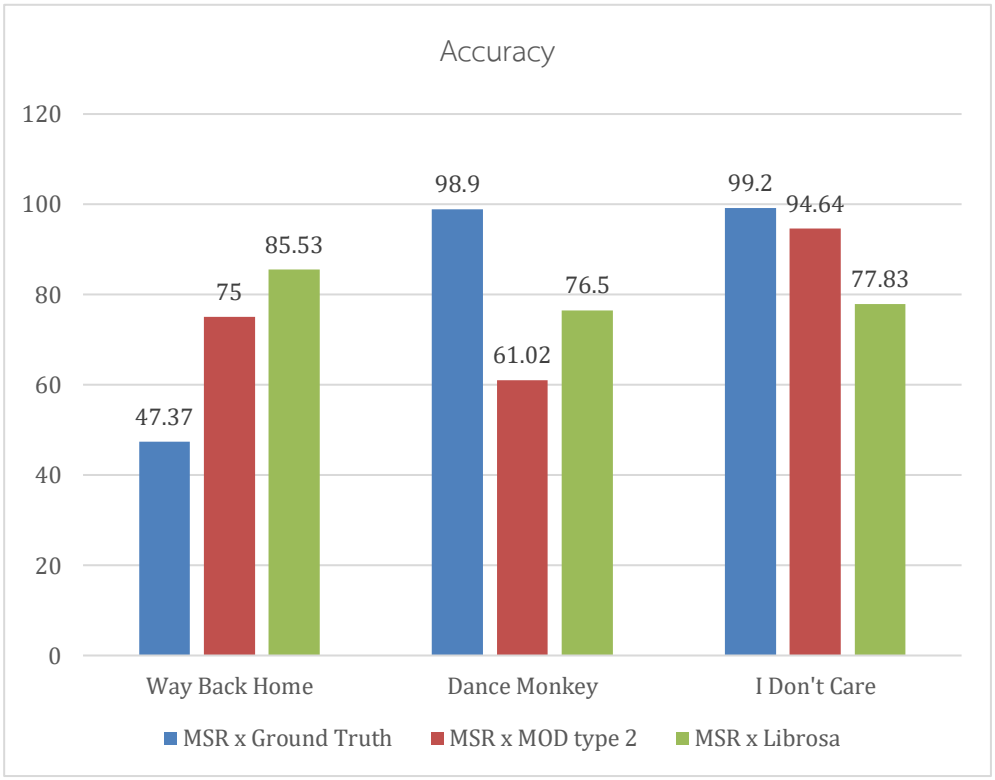
ภาพที่ 5.1.3 ฉ จุดเริ่มต้นเสียงของ MOD Type 4



ภาพที่ 5.1.3 ช จุดเริ่มต้นเสียงของ Librosa

5.2 การทดสอบ Musical Score Recognition (MSR)

เป็นการประเมินผลด้วยค่าความแม่นยำ (Accuracy) ของการตรวจจับโน้ตดนตรี ด้วยค่าความแม่นยำที่เกิดจากการทำนายโน้ตดนตรีถูกประเภท ณ ผู้ทดสอบจึงทดสอบด้วยจุดเริ่มต้นเสียงจากแหล่งต่าง ๆ ดังกราฟต่อไปนี้



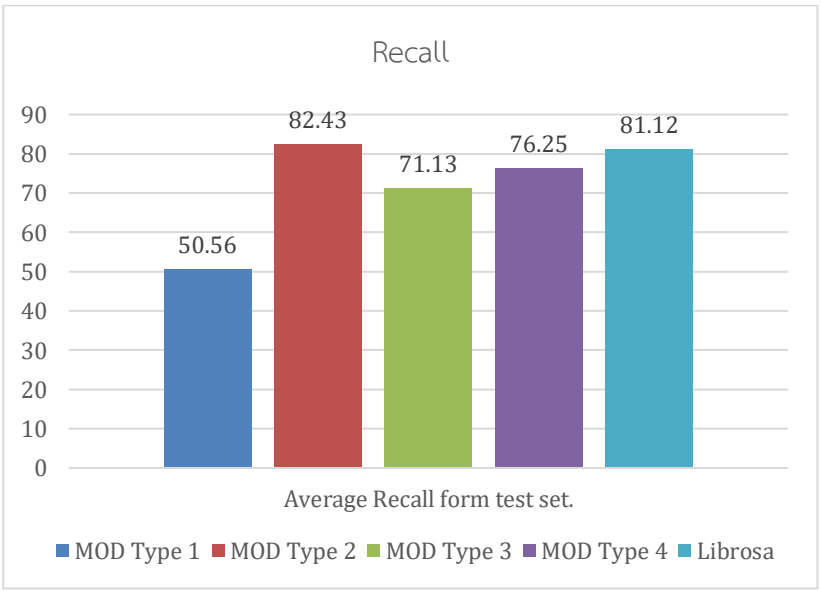
ภาพที่ 5.2 กราฟแสดงค่าความแม่นยำของตัวแบบ MSR บนข้อมูลชุดทดสอบ

บทที่ 6

ข้อสรุปและข้อเสนอแนะ

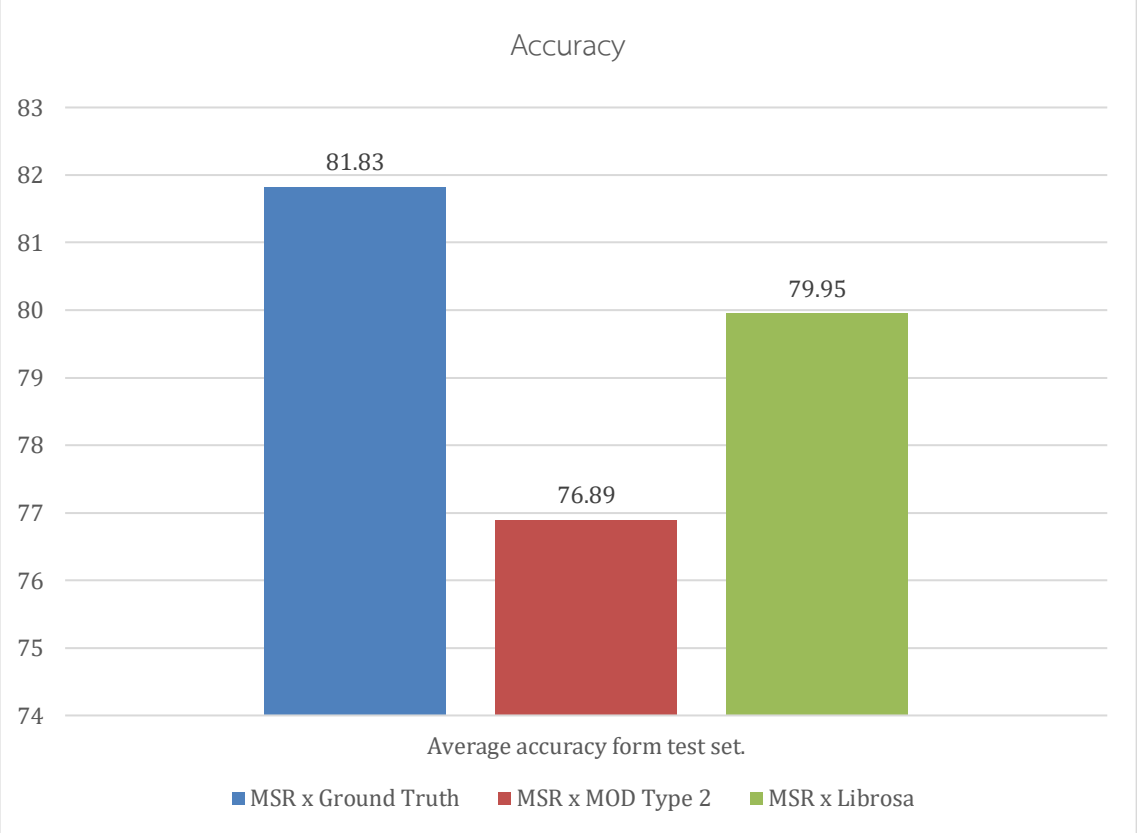
6.1 สรุปผลการดำเนินงาน

จากการทดลองตรวจจับจุดเริ่มต้นเสียงด้วยการแบ่งประเภทตัวแบบ MOD เป็นประเภทต่าง ๆ ตามการปรับแต่งสถาปัตยกรรมภายใน ได้ออกเป็น 4 ประเภท ตามที่ได้กล่าวในบท 4 และ บท 5 ไปแล้วนั้น สามารถสรุปได้ว่า MOD ประเภทที่ 2 ให้ผลลัพธ์ที่ดีเมื่อเทียบอีกกับ 3 ประเภทมากที่สุด รองลงมาคือ ประเภทที่ 4, 3 และ 1 ตามลำดับ การผลการทดลองบ่งชี้ว่าการเพิ่มจำนวนฟังก์ชันเคอร์เนล และ หน่วยของ GRU ไม่ได้ส่งผลต่อประสิทธิภาพโดยรวมของตัวแบบอย่างมีนัยสำคัญ อย่างไรก็ตามถึงแม้ว่าตัวแบบ MOD ประเภทที่ 2 ให้ผลลัพธ์ที่ดีที่สุดในการทดลองแต่ก็ไม่ได้ให้ผลลัพธ์ที่เที่ยงตรงเมื่อเทียบกับชุดทดสอบอาจด้วยสาเหตุหลายประการ ซึ่งประการที่สำคัญประการหนึ่งที่ผู้ทดลองตระหนักถึงคือการที่ชุดแบบฝึกสอนของตัวแบบไม่มีความสมดุลของข้อมูลโดยมีจำนวนข้อมูลที่ไม่ได้เป็นจุดเริ่มต้นเสียงเยอะกว่าจุดเริ่มต้นเสียง ซึ่งจำนวนข้อมูลที่ไม่ได้เป็นจุดเริ่มต้นเสียงสามารถประเมินเป็นตัวเลขได้มากกว่าร้อยละ 90 ของข้อมูลทั้งหมดที่มีในชุดฝึกสอน



ภาพที่ 6.1 ก กราฟสรุปผลค่า Recall เฉลี่ยของจุดเริ่มต้นของเสียงจากแต่ละตัวแบบ MOD และ Librosa

จากการทดลองตรวจจับโน้ตดนตรีด้วยตัวแบบ MSR เมื่อนำค่าความแม่นยำจากการประเมินตัวแบบประเภทต่าง ๆ ด้วยเพลงในชุดทดสอบมาหาค่าเฉลี่ยได้ผลดังกราฟด้านล่าง จากกราฟดังกล่าวหากไม่ประเมินผลตัวแบบ MSR ด้วยจุดเริ่มต้นจากข้อมูลกำกับจะพบว่า ตัวแบบ MSR ด้วยจุดเริ่มต้นเสียงจากคลังโปรแกรม Librosa ให้ค่าความแม่นยำมากที่สุด



ภาพที่ 6.1 ข กราฟสรุปผลความแม่นยำเฉลี่ยของโน้ตดนตรีจากแต่ละตัวแบบ MSR

6.2 ปัญหาของงานวิจัยและวิธีการแก้ไข

ปัญหาที่ 1 : จำนวนข้อมูลที่ใช้ฝึกสอนมีจำนวนจำกัด ทำให้ตัวแบบไม่สามารถเรียนรู้ทักษะการแต่งเพลงที่หลากหลายได้ ส่งผลให้เพลงกล่องดนตรีที่ได้ออกมานั้นมีรูปแบบที่จำเพาะเจาะจงกับแนวเพลงที่อยู่ในข้อมูลที่ใช้ฝึกสอน

ปัญหาที่ 2 : รูปแบบของข้อมูลที่ใช้ฝึกสอนซึ่งอยู่ในรูปแบบ musicXML มีป้ายกำกับของแต่ละสัญลักษณ์ทางดนตรีที่หลากหลายยากต่อการพัฒนาขั้นตอนวิธีในการถอดความได้ทั้งหมด ซึ่งส่งผลให้ผู้พัฒนาต้องเพิ่มความสามารถของขั้นตอนวิธีในการถอดความเป็นประจำตลอดช่วงพัฒนาเมื่อพบป้ายกำกับสัญลักษณ์ในรูปแบบใหม่

ปัญหาที่ 3 : การเก็บรวบรวมข้อมูลเป็นไปด้วยความยากลำบาก ด้วยข้อจำกัดหลาย ๆ ประการ ดังต่อไปนี้

1. เว็บไซต์ที่ใช้เก็บรวบความข้อมูลมีการกำหนดขอบเขตการดาวน์โหลดข้อมูล
2. กระบวนการเตรียมข้อมูลใช้เวลามาก เนื่องจากเมื่อผู้พัฒนาดาวน์โหลดข้อมูลในรูปแบบ musicXML มาแล้วจะต้องทำการเทียบเพลงที่สร้างจาก musicXML กับ เพลงจริงว่ามีจังหวะ และ ทำนองหลักสอดคล้องกันหรือไม่ในทุก ๆ เพลงที่ดาวน์โหลดมา แล้วจึงทำการตรวจสอบการถอดความจากป้ายกำกับในไฟล์ musicXML แต่ละเพลงว่ามีความถูกต้องหรือไม่ หากพบว่าไม่ถูกต้องจะต้องหาสาเหตุของการถอดความที่ผิดพลาดก่อนที่จะเริ่มดำเนินการแก้ไขขั้นตอนวิธีในการถอดความได้
3. เมื่อผู้พัฒนาหมดหนทางในการจัดการปัญหาการถอดความข้อมูลในไฟล์ musicXML หรือการจัดการปัญหามีความยุ่งยากซับซ้อนและไม่คุ้มค่าต่อเวลาในการแก้ไขที่มีอยู่อย่างจำกัด ผู้พัฒนาจึงจำเป็นต้องตัดข้อมูลเหล่านั้นออกไป ซึ่งส่งผลกระทบต่อจำนวนข้อมูลที่ใช้ฝึกสอนตัวแบบอย่างหลีกเลี่ยงไม่ได้

ปัญหาที่ 4 : ไฟล์ musicXML มีความผันผวนไม่แน่นอนเพราะสามารถเปลี่ยนแปลงข้อมูลได้เองในบางครั้ง ทำให้ผู้พัฒนาต้องหมั่นตรวจสอบและแก้ไขอยู่ตลอดเวลา

ปัญหาที่ 5 : แหล่งที่ใช้ในการรวบรวมข้อมูลฝึกสอนนั้นมีข้อมูลให้ดาวน์โหลดจำกัด อีกทั้งประเภทของเพลงยังไม่หลากหลายเช่น ไม่มีเพลงไทย หรือ ไม่มีเพลงสากลสมัยหลังปี ค.ศ. 2000 เป็นต้น

เอกสารอ้างอิง

[1] Daniel T, Politoske. Music. 1992 p. 4

[2] Li su "Melody extraction (vocal) using Pitch-base CNN (2018)."
Available from: https://arxiv.org/pdf/1804.09202.pdf

[3] Jan Schlüter and Sebastian Böck "Musical Onset Detection with Convolutional Neural Networks." Available from:
http://www.ofai.at/~jan.schlueter/pubs/2013_mml.pdf

[4] Librosa Documentation. Available from: https://librosa.github.io/librosa/

[5] MuseScore.org Available from https://musescore.org/

[6] Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015 Available from: http://neuralnetworksanddeeplearning.com/chap5.html

[7] Klaus Greff, Rupesh K. Srivastava, Jan Koutn´ık, Bas R. Steunebrink, Jurgen Schmidhuber "LSTM: A Search Space Odyssey" Available from:
https://arxiv.org/pdf/1503.04069.pdf?fbclid=IwAR0OZaAqpcsYgsFRFUVZtca91gCO_MHcpTfd5A4AjiJLy_52uaYQSYyEIUY

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling" Available from:https://arxiv.org/pdf/1412.3555.pdf?fbclid=IwAR2iUpRJkEf9ihMczLsZAMQZhZ2Dg4F37_LKchkj2Yu2BL7OC7BQPikjFT4

[9] Open-Unmix - A Reference Implementation for Music Source Separation
Available from: https://sigsep.github.io/open-unmix/#paper

[10] Karen Simonyan, Andrew Zisserman "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITIN"
Available from: https://arxiv.org/pdf/1409.1556.pdf

แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal

ปีการศึกษา 2562

ชื่อโครงงาน (ภาษาไทย)	เว็บแอปพลิเคชันแปลงเพลงเป็นเพลงกล่องดนตรี
ชื่อโครงงาน (ภาษาอังกฤษ)	Web application for music box song converter
อาจารย์ที่ปรึกษา	ผศ. ดร.ฑิตยา หวานวารี
ผู้ดำเนินการ	นายจาริก ศิลปาภินันท์ เลขประจำตัวนิสิต 5933609023
	นายธนาธิป ดอรอมาน เลขประจำตัวนิสิต 5933630023
	สาขาวิชา วิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หลักการและเหตุผล

ดนตรีมีบทบาทในการดำเนินกิจกรรมของมนุษย์ ช่วยชี้นำกำหนดอารมณ์ความรู้สึก ไม่ว่าจะเป็นเพื่อความบันเทิง การพักผ่อนหย่อนใจ หรือการเร้าอารมณ์ตามกิจกรรมต่าง ๆ รูปแบบการใช้ดนตรีของมนุษย์นั้นนอกจากการฟังแล้ว ยังมีการเล่นดนตรีอีกด้วย จึงไม่ใช่เรื่องแปลกหากมีบุคคลต่าง ๆ ที่มีความชื่นชอบดนตรีจะต้องการแกะทำนองหรือโน้ตดนตรีของบทประพันธ์ เพื่อจะทดลองเล่นดูเอง หรือเพื่อเก็บบันทึกไว้

รูปแบบหนึ่งของเสียงดนตรีที่ใช้ในการผ่อนคลายได้ดีคือเสียงของกล่องดนตรี แต่เพลงกล่องดนตรีที่มีในท้องตลาดมีอยู่จำกัด หากต้องการสร้างเพลงกล่องดนตรีเอง จะต้องมีโน้ตดนตรีก่อน จากนั้นจึงนำไปสร้างกล่องดนตรี หรือเล่นด้วยคอมพิวเตอร์เพื่อให้เป็นสียงกล่องดนตรี การแกะโน้ตดนตรีจึงเป็นขั้นตอนที่สำคัญหากต้องการจะสร้างกล่องดนตรี

การแกะโน้ตดนตรีนั้นต้องอาศัยทักษะการจับเสียงตัวโน้ต และความรู้ทางทฤษฎีดนตรีหลายประการ แต่รูปแบบการทำงานนั้นซ้ำ ๆ กัน ไม่ว่าจะเป็นเพลงใด ๆ นั่นคือ ต้องจับทำนอง (melody) และจังหวะ (rhythm) ของ เพลงให้ได้ก่อน จากนั้นนำมาสร้างเป็นโน้ตเพื่อเล่นด้วยเครื่องดนตรีอื่น ๆ ดังนั้น เราจึงสามารถสร้างโปรแกรม คอมพิวเตอร์เพื่อให้จับทำนองหลักและจังหวะของเพลง จากนั้นจึงนำโน้ตที่ได้มาสังเคราะห์เสียงกล่องดนตรี ต่อไป โครงงานนี้จะรับข้อมูลเป็นเพลงเอ็มพี 3 (mp3) บนเว็บแอปพลิเคชัน และสร้างแฟ้มข้อมูลเอ็มพี 3 ของเสียงกล่องเพลงด้วย ซอฟต์แวร์ปัญญาประดิษฐ์ โดยจำกัดเฉพาะเพลงแนว พ็อป ร็อก และ อาร์แอนด์บี

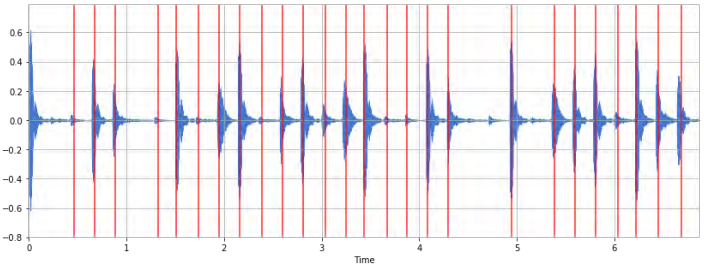
วัตถุประสงค์

พัฒนาโปรแกรมสำหรับตรวจจับโน้ตดนตรีจากแฟ้มข้อมูลเอ็มพี 3 เพื่อสร้างเสียงเพลงแบบกล่องดนตรี และสามารถบันทึกเสียงเก็บในรูปแบบแฟ้มข้อมูลเอ็มพี 3

ความรู้ที่เกี่ยวข้อง

1. จุดเริ่มต้นของเสียง (Onset)

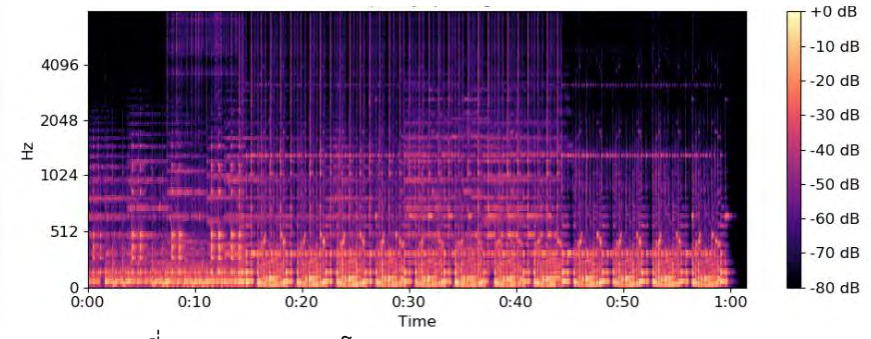
จุดเริ่มต้นของเสียง เป็นคุณสมบัติของเสียงใด ๆ ก็ตามบนโลกใบนี้ รวมถึงเหตุการณ์ที่เกิดขึ้นในดนตรี นั่นคือจุดที่มีการเริ่มกดโน้ตดนตรีแต่ละตัวในเสียงเพลง เราสามารถนำเสียงเพลงมาทำเป็นกราฟคลื่นเสียงได้ โดยให้แกน y เป็นแกนความดัง และแกน x เป็นแกนเวลา (time) เส้นตรงสีแดงขนานแกน y แสดงตำแหน่งจุดเริ่มต้นของเสียง ดังรูปที่ 2.5.1 ก จะเห็นได้ว่าจุดเริ่มต้นของเสียงเป็นจุดที่ค่าแอมพลิจูดมีค่าสูงในระยะเวลาสั้น ๆ



ภาพที่ 1 ก กราฟคลื่นเสียงที่แสดงตำแหน่งจุดเริ่มต้นของเสียง

ที่มา : https://musicinformationretrieval.com/onset_detection.html

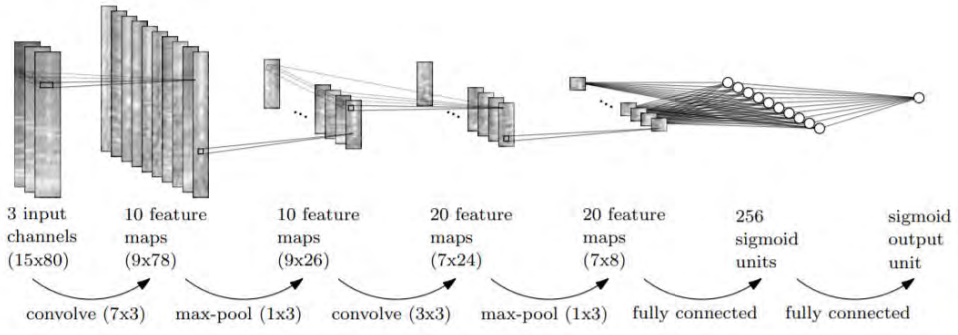
นอกจากนี้แล้วเรายังสามารถนำกราฟคลื่นเสียงมาพลอตกราฟสเปกโทรแกรม (spectrogram) โดยให้แกน y เป็นค่าความถี่ (frequency) แกน x เป็นเวลา (time) และมีความเข้ม (intensity) ที่บอกค่าแอมพลิจูดของแต่ละความถี่ในแต่ละช่วงเวลา



ภาพที่ 1 ข กราฟสเปกโทรแกรมแบบ mel-frequency

ที่มา : [4] Librosa Documentation

จุดเริ่มต้นของเสียงสเปกโทรแกรมจะเป็นจุดที่มีการเปลี่ยนของสเปกตรัมต่อเวลา เมื่อลองพิจารณาจากสเปกตรัม จะพบว่า การตรวจจับจุดเริ่มต้นของเสียงนั้นคล้ายคลึงกับการหาขอบรูป (edge detection) ในรูปภาพ ด้วยเหตุนี้ การใช้ CNN จึงเป็นตัวเลือกที่เหมาะสมกับการตรวจจับจุดเริ่มต้นของเสียง งานวิจัยของ Jan Schlüter and Sebastian Böck [3] ใช้ชุดเพลงประเภทหลายเสียง (polyphonic) และสร้างตัวแบบที่ใช้ ในการตรวจจับจุดเริ่มต้นของเสียง โดยเริ่มจากข้อมูลเข้าเป็นสเปกโทรแกรมขนาด 15x80 พิกเซลที่สกัดออกมา ส่งเข้าชั้นสังวัตนาการ และชั้นรวมค่าสูงสุดขนาด 7x3 1x3 3x3 1x3 สลับกันตามลำดับ แล้วปิดท้ายด้วยโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้าซึ่งใช้ฟังก์ชันกระตุ้นซิกมอยด์ (sigmoid) ขนาด 256 นิวรอน และมี ผลลัพธ์เป็นค่า 0 (ไม่เป็นจุดเริ่มต้น) หรือ 1 (เป็นจุดเริ่มต้นของเสียง)



ภาพที่ 1 ค สถาปัตยกรรมของตัวแบบหาจุดเริ่มต้นเสียง

ที่มา : [3] Musical Onset Detection with Convolutional Neural Networks

ในการทดลองจะฝึกสอนตัวแบบ 100 รอบ (epoch) ด้วยวิธีการหาค่าเหมาะที่สุดแบบ SGD และ กำหนดขนาดชุดสำหรับการฝึกสอน (mini-batch) เป็น 256 ตัวอย่าง อัตราการเรียนรู้ (learning rate) ที่ 0.05 เมื่อพิจารณาผลลัพธ์ที่ได้ พบว่าตัวแบบ CNN เอาชนะตัวแบบโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า และ CNN แบบที่ใช้เคอร์เนลเป็นรูปเหลี่ยมมุมฉากให้ประสิทธิภาพที่ดีขึ้นจากการใช้เคอร์เนลเป็นรูปสี่เหลี่ยมจัตุรัส โดยมีค่าความเที่ยงตรง (precision) ที่ 90.5% ค่าการค้นคืน (recall) ที่ 86.6% และ F-measure ที่ 88.5%

2. ทำนองเพลง (Melody)

ทำนองเพลง [1] เป็นเสียงที่เปล่งออกมาโดยมีความต่อเนื่องกันเป็นระบบ ทำนองเพลงเปรียบเหมือน รูปร่างของบทเพลงที่มีเสียงสูง, ต่ำ, สั้น, ยาว ประกอบกันโดยทั่วไปดนตรีประกอบด้วยทำนองซึ่งเป็น องค์ประกอบที่ง่ายต่อการจำรวมถึงเป็นเอกลักษณ์ของบทเพลงนั้น ๆ ดังนั้นหากต้องการจะแกะบทเพลงใดสักเพลงการเริ่มต้นด้วยการแกะทำนองของเพลงจึงเป็นสิ่งที่ง่ายที่สุด

งานวิจัยของ Li Su [2] ได้สกัดโน้ตจากทำนองของเสียงร้อง หรือ Vocal Melody ด้วยข่ายงาน ประสาทแบบสังวัตนาการ (convolution Neural Network - CNN) ซึ่งเริ่มต้นด้วยการรับ

แฟ้มข้อมูลเสียงเป็น สัญญาณ 1 มิติ และแปลงเป็นสเปกโทรแกรมเพื่อให้ตรวจจับรูปแบบพฤติกรรมโดยทั่วไปของเสียงได้มีประสิทธิภาพยิ่งขึ้น จากนั้นแปลงเป็นเซปสตรัมทั่วไป (generalized cepstrum - GC) และเซปสตรัมทั่วไปของสเปกตรัม (generalized cepstrum of spectrum - GCoS)

กำหนดให้ความสัมพันธ์ของสเปกโทรแกรม, เซปสตรัมทั่วไป และเซปสตรัมทั่วไปของสเปกตรัม เป็นดังนี้

$$Z_0[k,n] \coloneqq \sigma_0\left(W_f X\right)$$
$$Z_1[q,n] \coloneqq \sigma_1(W_t F^{-1} Z_0)$$
$$Z_2[k,n] \coloneqq \sigma_2\left(W_f F Z_1\right)$$

ให้ Z_0 คือ สเปกโทรแกรม Z_1 คือ เซปสตรัมทั่วไป และ Z_2 คือ เซปสตรัมทั่วไปของสเปกตรัม มีค่าดัชนี (index) k ในสมการ Z_0 , Z_2 เป็นค่าความถี่ (frequency) ขณะที่ค่าดัชนี q ใน สมการ (2) แสดงถึงค่าคิวเฟรนซี (quefrency) และค่าดัชนี n แสดงถึงเวลา โดยแต่ละสมการจะมีฟังก์ชันกระตุ้น (activation function) เป็น

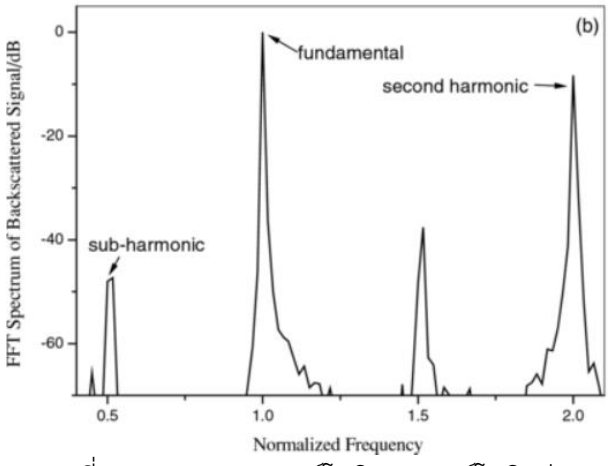
$$\sigma_i(Z) = |ReLU(Z_i)|^{\gamma_i}, \quad \mathrm{i} = 0,1,2$$

สมการข้างต้น มีฟังก์ชัน ReLU ประกอบกับฟังก์ชันถอดรากเรียงราย (element wise-root function) และกำหนดให้ค่า γ_i อยู่ในช่วง 0 ถึง 1

นอกจากนี้ยังมี ค่า W_f and W_t ซึ่งเป็นตัวกรอง (High-pass filters) มีลักษณะเป็นเมทริกซ์ทแยงมุม (diagonal matrices) ที่ระบุค่าความถี่ตัดและค่าคิวเฟรนซี่ตัด (cutoff frequency and quefrency) นั่นคือค่า k_c และ q_c อยู่ภายในเมทริกซ์ ตามลำดับดังสมการต่อไปนี้

$$W_{f\ or\ t}[l,l] = \begin{cases} 1, & l > \mathrm{k}_c\ or\ \mathrm{q}_c; \\ 0, & otherwise \end{cases}$$

จากนั้นนำ GC และ GCoS นั่นคือสมการ Z_1 และ Z_2 มาใช้ร่วมกันเพื่อวัตถุประสงค์ในการกำจัดเสียงฮาร์โมนิกและฮาร์โมนิกย่อยที่ไม่ต้องการออก (harmonics and sub-harmonics)



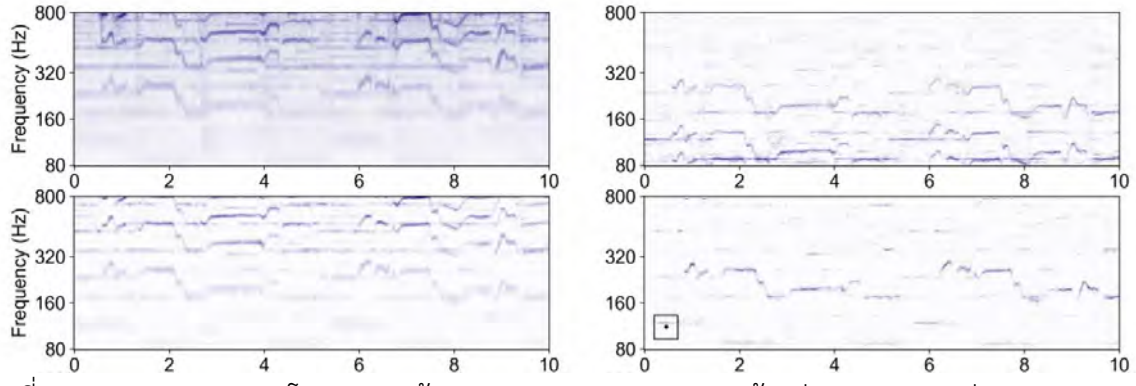
ภาพที่ 2 ก กราฟแสดงฮาร์โมนิกและฮาร์โมนิกย่อย

ที่มา : [8] A dual-frequency excitation technique for enhancing the sub-harmonic emission from encapsulated microbubbles

ทำการเปลี่ยนโดเมนคิวเฟรนซี่ของสมการ Z_1 ให้เป็นโดเมนความถี่ จากนั้นนำโดเมนความถี่ของทั้ง Z_1 และ Z_2 มาทำ log-frequency scale ได้เป็นค่าดัชนี p และนำโดเมนความถี่ที่ได้และ โดเมนของเวลา ของทั้ง GC และ GCoS มาผ่านตัวกรอง W_f and W_t แล้วนำมารวมกันเป็นสมการดังนี้

$$Y[p,n] = \tilde{Z}_1[p,n]\tilde{Z}_2[p,n]$$

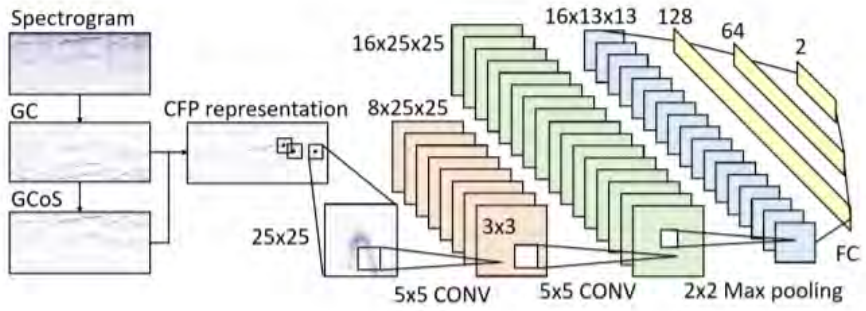
สิ่งที่ได้จากการใช้ GC และ GCoS ร่วมกัน เรียกว่ากราฟของคาบและความถี่ (Combined Frequency and Periodicity - CFP) ดังภาพที่ 2 ข



ภาพที่ 2 ข กราฟแสดง สเปกโทรแกรม (ซ้ายบน) GC (ขวาบน) GCoS (ซ้ายล่าง) CFP (ขวาล่าง)

ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

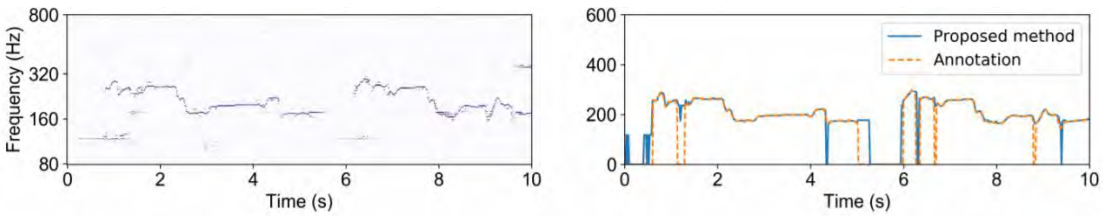
CFP เป็นกราฟที่สามารถระบุตำแหน่งระดับเสียงในโดเมนความถี่ได้ การจำแนกว่าจุดใดบนระนาบเวลา-ความถี่ เป็นเส้นรอบขอบ (contour) ของทำนองจากเสียงร้องนั้น จะตัดแบ่ง CFP เป็นชิ้นย่อย ๆ ขนาด 25x25 พิกเซล โดยกำหนดให้แต่ละชิ้นมีจุดศูนย์กลางตรงกับจุดสูงสุดของเส้นความถี่ที่ปรากฏบน CFP แล้วจึงส่งชิ้นย่อยเข้าไปยังตัวแบบ CNN ซึ่งมีชั้นสังวัตนาการ 2 ชั้น ชั้นแรกประกอบด้วยเคอร์เนลขนาด 5x5 จำนวน 8 เคอร์เนล และ ชั้นถัดไปประกอบด้วยเคอร์เนลขนาด 3x3 จำนวน 16 เคอร์เนล ถัดไปเป็นชั้นรวมค่าสูงสุด (max pooling) 1 ชั้น ขนาด 13x13 จำนวน 16 เคอร์เนล และชั้นสุดท้ายเป็นชั้นโครงข่ายแน่น ซึ่งประกอบด้วยนิวรอนจำนวน 128, 64 และ 2 นิวรอน ตามลำดับ ดังภาพที่ 2 ค ผลลัพธ์ที่ได้ของตัวแบบ CNN คือเวกเตอร์ ขนาด 2x1 ซึ่งแสดงถึงความน่าจะเป็นของการเป็นทำนองเพลงจากเสียงร้อง โดยกำหนดฟังก์ชันค่าเสียหาย (loss function) เป็น Cross-Entropy และใช้ขั้นตอนวิธี Adam ในการปรับค่าน้ำหนัก



ภาพที่ 2 ค สถาปัตยกรรมของตัวแบบการสกัดทำนองเพลง

ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

ผลการทดลองพบว่า การเลือกเอาดัชนความถี่ผลลัพธ์ที่มีความน่าจะเป็นของทำนองเพลงจากเสียงร้องสูงที่สุด หรือ CNN Max-Out มีค่าความแมนยำ (accuracy) ที่ 83.5% บนชุดข้อมูล MIREX2005 ซึ่งมีค่าความแม่นยำมากสุดเมื่อเทียบกับวิธีอื่น ๆ

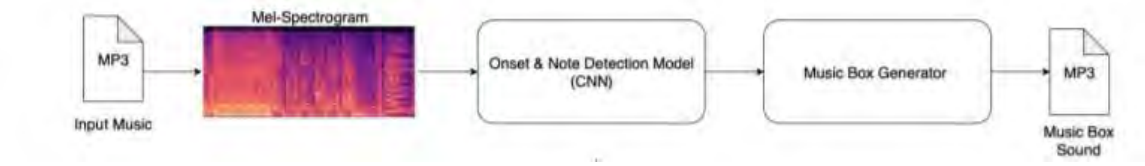


ภาพที่ 2 ง กราฟแสดง CNN outputs (ซ้าย) ผลลัพธ์ CNN-MaxOut (ขวา)

ที่มา : [2] Melody extraction (vocal) using Pitch-base CNN (2018)

ภาพรวมของระบบ

เมื่อรับข้อมูลเป็นแฟ้มข้อมูลเอ็มพี 3 มาแล้ว จะแปลงให้เป็นสเปกโทรแกรมก่อนโดยใช้คลังโปรแกรม Librosa [5] จากนั้นจะตรวจจับความถี่ของทำนองหลัก และตรวจจับจุดเริ่มต้นของโน้ตด้วยตัวแบบโครงข่ายประสาทเทียมแบบสังวัตนาการ แล้วสร้างโน้ตเพลงที่ต้องเล่นโดยพิจารณาจากความถี่และตำแหน่งจุดเริ่มต้นของความถี่ จากนั้นจะสังเคราะห์เป็นเสียงกล่องดนตรีในรูปแบบแฟ้มข้อมูลเอ็มพี 3 ภาพรวมของระบบเป็นดังแสดงในรูปที่ 1



รูปที่ 1

ในการสร้างตัวแบบเพื่อสกัดทำนองหลัก และตรวจจับจุดเริ่มต้นของโน้ตนั้น จะใช้ข้อมูลเพลงจริงจากแฟ้มข้อมูลเอ็มพี 3 มาเปรียบเทียบกับโน้ตดนตรีจากแฟ้มข้อมูล musicXML ที่นำมาจากเว็บ Musesore [7] โดยอนุมานให้โน้ตดนตรีนี้เป็นโน้ตที่ถูกต้องของเพลง เพื่อใช้เป็นตัวอย่างในการฝึกสอนตัวแบบ

ในการประเมินผลความถูกต้องของตัวแบบจะใช้หลักเกณฑ์ดังต่อไปนี้

1. ประเมินความถูกต้องในการตรวจจับจุดเริ่มต้นของโน้ตด้วย F-Score โดยจะพิจารณาว่าถูกต้องเมื่อได้ตำแหน่งเริ่มต้นของเสียงที่ความคลาดเคลื่อนไม่เกิน 25 ms จากตำแหน่งจริงใน โน้ตเพลง ส่วนตำแหน่งซึ่งตรวจพบแต่ไม่มีในโน้ตเพลง และตำแหน่งที่มีในโน้ตเพลงแต่สกัด ออกมาไม่ได้ จะถือว่าเป็นผลบวกลวง (false positive) และผลลบลวง (false negative) ตามลำดับ
2. ประเมินความถูกต้องในการตรวจจับโน้ตเพลงด้วยค่าความแม่นยำ (accuracy) โดยจะ พิจารณาว่าโน้ตถูกต้องเมื่อมีการระบุชื่อและคู่แปด (octave) ของโน้ต ณ ตำแหน่งเวลาที่ ถูกต้อง ซึ่งตำแหน่งเวลาจะได้จากการตรวจจับจุดเริ่มต้นของโน้ตในขั้นตอนก่อนหน้า เช่น หากตรวจจับได้โน้ต C4 หมายถึง โน้ต C ที่คู่แปดลำดับ 4 แต่ในโน้ตเพลงระบุโน้ต C5 จะไม่ นับว่าถูกต้อง และหากระบุตำแหน่งเวลาผิด ก็จะไม่นับว่าถูกต้องเช่นเดียวกัน

ขอบเขตของโครงงาน

- โครงงานนี้ศึกษาศึกษารูปแบบทำนองเพลง จังหวะ และเบส เฉพาะในดนตรีประเภทพ็อป ร็อก และอาร์แอนด์บี เท่านั้น
- ข้อมูลที่มีการกำกับผลลัพธ์ (labeled data) สำหรับการฝึกสอนตัวแบบเป็นข้อมูลที่มาจากคลัง เพลงของเว็บ MuseScore [7] ซึ่งอยู่ในรูปแบบ MusicXML
- โครงงานนี้เป็นโครงงานพัฒนาเว็บแอปพลิเคชันที่มีส่วนติดต่อผู้ใช้งานสำหรับการรับแฟ้มข้อมูลเสียงประเภทเอ็มพี 3 (.mp3) หรือเวฟ (.wav) แล้วสร้างและเล่นแฟ้มข้อมูลเสียงดนตรีแบบกล่องเพลง

วิธีการดำเนินงาน

2. แผนการดำเนินงาน
    1. ค้นคว้าหาข้อมูล และ แนวทางการพัฒนาซอฟต์แวร์ปัญญาประดิษฐ์
    2. วิเคราะห์และกำหนดขอบเขตของระบบ ศึกษาวิธีการแปลงเพลงด้วยขั้นตอนวิธีการเรียนรู้เชิงลึกที่สามารถนำมาประยุกต์ใช้ได้
    3. ออกแบบระบบ และ พัฒนาระบบ
    4. ทดสอบประสิทธิภาพของระบบ และ แก้ไขข้อผิดพลาดที่พบของระบบ
    5. สรุปผล และ จัดทำเอกสารประกอบโครงงาน

2. ระยะเวลาการดำเนินงาน

การดำเนินงาน	ปี 2562					ปี 2563		
	เดือน ส.ค.	เดือน ก.ย.	เดือน ต.ค.	เดือน พ.ย.	เดือน ธ.ค.	เดือน ม.ค.	เดือน ก.พ.	เดือน มี.ค.
ค้นคว้าหาข้อมูล								
วิเคราะห์และกำหนดขอบเขตของระบบ								
ออกแบบ และ พัฒนาระบบ								

ทดสอบ และ ปรับปรุงระบบ								
สรุปผล และ จัดทำเอกสารประกอบ โครงงาน								

ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ต่อผู้พัฒนา

1. มีความรู้ความเข้าใจในทฤษฎีดนตรีมากขึ้น
2. เพิ่มพูนทักษะการเขียนโปรแกรมและการพัฒนาระบบ
3. เรียนรู้การคิดวิเคราะห์วางแผนการทำงานอย่างเป็นระบบแบบแผน เพื่อให้เกิดประโยชน์สูงสุดตามทรัพยากรที่มีอยู่
4. ฝึกการเรียนรู้ด้วยตนเอง การยอมรับฟังความคิดเห็นของผู้อื่น ความตรงต่อเวลา ตลอดจนมีความรับผิดชอบในหน้าที่
5. เพิ่มพูนทักษะความรู้ความสามารถทางด้านวิทยาการข้อมูล

ประโยชน์ต่อผู้ใช้ระบบ

1. เป็นทางเลือกสำหรับผู้ใช้งานในการช่วยแกะโน้ตดนตรีเพื่อฝึกฝนทักษะการเล่นดนตรีเบื้องต้น
2. เป็นสื่อบันเทิงที่ผู้ใช้สามารถนำบทเพลงที่ชื่นชอบทำแปลงเป็นเสียงแบบกล่องดนตรีและ ยังสามารถบันทึกหรือส่งต่อให้คนอื่นได้เนื่องจากผลลัพธ์อยู่ในรูปแบบแฟ้มข้อมูลเอ็มพี 3
3.

อุปกรณ์และเครื่องมือที่ใช้

1. ฮาร์ดแวร์
    1. MacBook Pro (13in, July 2019) with 1.4GHz quad-core 8th-gen i5 Memory 8.00 GB
    2. Dell Inpiron14 3000series Intel Core i7-5500U CPU @2.40GHz Memory 8.00 GB
    3. ASUS ROG Strix G531GV-AL072T Intel Core i5-9300H @2.40GHz Memory 8.00 GB Nvidia Geforce RTX 2060
    4. Cloud Platform Service: AWS Amazon (VM)
2. ซอฟต์แวร์

1. Microsoft Office
2. JetBrain PyCharm
3. Jupyter Notebook
4. Visual Studio code
5. MuseScore

งบประมาณ

1. MIDI Keyboard	ราคา	5,000	บาท
2. ค่าลงทะเบียนเรียนทฤษฎีดนตรี	ราคา	1,800	บาท
3. ค่าบริการ Cloud Service Platform	ราคา	1,500	บาท
4. ค่าสมาชิก MuseScore	ราคา	1,534.37	บาท
รวมงบประมาณที่เสนอขอ		9,834.37	บาท

<u>หมายเหตุ</u> ให้ถัวจ่ายได้ทุกรายการ

ประวัติผู้จัดทำ

Mr. Tanatip Doromarn

นาย ธนาธิป ดรอมาน

เกิด 17 มิถุนายน 1997

ชั้นปีที่ 4 คณะวิทยาศาสตร์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

อีเมล: lelouch.zero735@gmail.com

Mr. Jarig Silpapinun

นาย จาริก ศิลปาภินันท์

ชาตะ 12 กรกฎาคม 1996

โรงเรียน เทพศิรินทร์

ชั้นปีที่ 4 คณะวิทยาศาสตร์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

สาขาคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

อีเมล: knight44085@icloud.com