**목표**

1. 신호 전처리
   1. 신호의 가공 형태
      1. MFCC
      2. Mel-spectrogram
      3. Spectrogram
   2. 음악 신호를 자르는 단위
      1. Per frame (20~40ms)
      2. Per piano 박자
   3. 가공된 신호의 사용 방법
      1. 전체주파수 대역
      2. 타겟주파수 대역 (피아노가 사용하는 주파수 대역)
2. 딥러닝 모델
   1. 음성, 음악과 같은 동적인 모델의 딥러닝 방법
   2. 신경망 모델
      1. denseNet
3. 딥러닝 출력
   1. 회귀(regression)인가 분류(classification)인가?
      1. 회귀(regression)
      2. 분류(classification)
4. 기타
   1. 음악 신호의 이해
      1. 피아노 신호의 특성
      2. 음악 신호의 특성
   2. 구글 코랩 사용방법 -
   3. 입력 음악파일 확보
5. **신호 전처리**
   1. **신호의 가공 형태**
      1. **MFCC** MFCC란 어쩌구 저쩌구
      2. **Mel-spectrogram** MFCC란 어쩌구 저쩌구
      3. **Spectrogram** mel-spectrogram은 spectrogram을 mel-scale로 변환하여 처리한 것이다. 주로 음성데이터를 처리할 때 많이 사용한다. 그런데 피아노는 음성데이터가 아닌데, mel-scale로 처리하는 것이 적절한지에 대한 의문이 존재한다. 따라서 spectrogram에 대한 조사도 추가하였다.  
          한가지 고려하여야 할 점은, 피아노의 배음도 결국 인간이 듣기 좋은 형태로 형성되었다는 점이다. 따라서 mel-spectrogram을 이용하는 것이 적절할 수도 있다.  
          기본적으로 spectrogram과 mel-spectrogram의 형태가 비슷하기 때문에, 같은 딥러닝에 두 데이터를 모두 활용 가능할 수도 있다.
   2. **음악 신호를 자르는 단위**
      1. **Per frame (20~40ms)**어쩌구 저쩌구
      2. **Per piano 박자**어쩌구 저쩌구
   3. **가공된 신호의 사용 방법**
      1. **전체 주파수 대역** 전체 주파수 대역을 이용하는 것이, 딥러닝을 구현하는 것이 더 간편하다. 또한 우리가 피아노 신호의 특성을 잘 모르더라도, 이를 딥러닝이 해결하도록 넘길 수 있다는 장점도 존재한다. 그러나 실질적으로 딥러닝이 피아노가 이용하는 음역대만 잘 캐치하도록 학습될지는 모를 일이다.
      2. **타겟주파수 대역 (피아노가 사용하는 주파수 대역)**어쩌구 저쩌구
6. **딥러닝 모델**
   1. **음성, 음악과 같은 동적인 모델의 딥러닝 방법**
      1. **어쩌구 이론** MFCC란 어쩌구 저쩌구
   2. **신경망 모델**
      1. **denseNet**어쩌구 저쩌구
7. **딥러닝 출력**
   1. **회귀(regression)인가 분류(classification)인가?**
      1. **회귀(regression)** 회귀의 경우 MFCC와 같이, 입력과 같은 데이터 형식으로 출력하는 것을 말한다. 이때 정답 label은 유저가 임의로 생성한 것을 사용한다.   
          위와 같은 방식은, 출력 결과를 다시 정답 label과 비교하는 딥러닝을 수행해야 하는 단점이 있다. 즉 두 단계로 이어진 딥러닝을 수행해야 한다는 것이다. 그리고 인간의 오차(건반을 기계와 똑같이 칠 수는 없다)로 인해 오히려 부족한 결과를 낼 수도 있다.
      2. **분류(classification)** 출력의 형태는 피아노의 음, 정확하게 말하면 타이밍(타건 시점). 지속시간(음의 지속), 음의 높이가 출력될 것이다.  
          타이밍의 경우, 입력을 잘라서 사용하기 때문에 특정하기가 쉽다.  
          지속시간의 경우, 피아노 음의 대한 이해가 필요할 것이다.  
          음의 높이의 경우도 피아노에 음에 대한 이해가 필요하다.
   2. **신경망 모델**
      1. **denseNet**어쩌구 저쩌구
8. **기타**
   1. **음악 신호의 이해**
      1. **피아노 신호의 특성**(https://m.blog.naver.com/ylefor/220054727055)   
          배음: 기본 주파수 이외의 주파수가 기준 주파수의 N배의 비율로 발생하는 것을 말한다. 배음은 악기의 울림통 구조에 따라 달라지며, 따라서 같은 피아노라도 다른 음색을 가질 수 있다.   
          피아노의 대역: 피아노의 88건반은 최저음이 27.5hz, 최고음이 4186hz이다. 소리로는 그 4배음까지 들을 수 있다?(이건 무슨 소리? -> 가청주파수는 20khz까지라는 소리인듯. 실질적으로 구분 가능한 소리는 4000hz정도 이므로, 피아노는 음역을 더 확장해도 별 의미가 없다.)  
         (https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=shinbygirl82&logNo=220211818842)  
          배음: 피아노 현의 진동은 특정 주파수의 한해서가 아니라, 여러 무수한 부분진동을 포함하고 있다. 여기서 1진동을 기본 진동 혹은 기음이라고 하며, 다른 정수배의 진동들을 부분진동 혹은 부분음이라 한다. 이러한 부분진동들을 바로 배음이라고 하는 것이다.   
          인하모니시티(inharmonicity): 이론적으로 화음은 배음들이 수학적이고 복합적으로 결합하여 만들어진다. 따라서 이론적으로는 모든 현의 진동(다시 말해 배음)을 수학적으로 나타낼 수 있을 것이다. 그러나 실제 진동들은 이러한 수학적계산과 일치하지 않는다. 현의 경직성(굵기와 장력에 의해 좌우)에 따라 실제로는 다르게 발생한다. 따라서 실제로는 기본음의 정수배가 아닌, 배음 배수의 제곱에 정비례하여 나타난다. 인하모니시티로 인해 음악은 화려하거나 탁하거나 거슬리는등 다양하게 들리게 된다.   
          배음과 맥놀이, 자연배음열, 음계등등의 내용이 있으나 필요한 내용인지는 확실히 모르겠다. 추가바람
      2. **음악 신호의 특성** 출력의 형태는 피아노의 음, 정확하게 말하면 타이밍(타건 시점). 지속시간(음의
   2. **구글 코랩 사용방법** 출력의 형태는 피아노의 음, 정확하게 말하면 타이밍(타건 시점). 지속시간(음의
   3. **입력 음악파일 확보** 출력의 형태는 피아노의 음, 정확하게 말하면 타이밍(타건 시점). 지속시간(음의
   4. **멜로다인**
   5. **단순한 음원에서 복잡한 음원의 사용** 음악과 관련된 딥러닝은 자료가 부족하고, 우리의 시도가 어느정도의 난이도를 가지는지 측정하기 어렵다. 따라서 보컬이 없고 악기의 개수가 적은 음원부터 학습해보고, 성능이 괜찮다면 복잡한 음원에 대해서도 시도해 볼 수 있을 것이다.  
       단순한 음원에서 복잡한 음원으로의 전환할 때, 딥러닝을 새로 시작할 수도 있지만, 기존의 학습된 가중치를 가지고 시도할 수도 있다. 보다 깨끗한 데이터에서 얻은 초기값을 통하여 더 나은 결과를 도출할 수 있다고 본다. (즉 ㄱ
   6. **Spleeter**<https://github.com/deezer/spleeter/blob/master/README.md>  
      음원을 자동으로 분류해주며(텐서플로우 이용), 피아노도 포함해서 분류가 가능하다. 해당 라이브러리를 뜯어봐서 도움을 얻거나, 피아노를 분류시킨 데이터를 이용해서 학습시키는 것도 가능할 듯 하다.
   7. **미분류**자동 채보라 검색하면 검색 결과 이것저것 나온다.   
      <http://www.aitimes.com/news/articleView.html?idxno=140852>   
      https://www.mapianist.com/community/35109