


**딥 러닝을 활용한 음악  
Beat Note  
자동생성 서비스가  
있는 VR 리듬게임  
Beat Make & Crush**

Team VRR

박영준

VR rhythm game with Beat Note automatic generation service  
using deep learning



# 01 필요한 기술

Requirement

# 무엇이 필요할까?

---



input - 음원  
output - 게임에 필요한 Beat Note

## 딥러닝

The background is a dark blue, almost black, space filled with abstract, glowing geometric shapes. On the left, a large, translucent red cube is partially visible. In the center, a teal-colored, angular shape resembling a stylized 'V' or a folded piece of paper is prominent. To the right, a dark blue, faceted shape is visible. Numerous thin, white, light-ray-like lines radiate from various points, creating a sense of depth and movement. The overall aesthetic is futuristic and high-tech.

# 02 딥러닝 라이브러리

Deep Learning Library - Keras

# 케라스(Keras)란?

Keras: The Python Deep Learning library



# Keras

**케라스는 파이썬으로 구현된 쉽고 간결한 딥러닝 라이브러리**

내부적으로는 텐서플로우(TensorFlow), 티아노(Theano), CNTK 등의 딥러닝 전용 엔진이 구동되지만 케라스 사용자는 복잡한 내부 엔진을 알 필요는 없습니다. 직관적인 API로 쉽게 다층퍼셉트론 모델, 컨볼루션 신경망 모델, 순환 신경망 모델 또는 이를 조합한 모델은 물론 다중 입력 또는 다중 출력 등 다양한 구성을 할 수 있습니다.

# 케라스 기본 개념



## 케라스의 핵심적인 데이터 구조는 **모델**

케라스에서 제공하는 시퀀스 모델로 원하는 레이어를 쉽게 순차적으로 쌓을 수 있습니다. 다중 출력이 필요한 등 좀 더 복잡한 모델을 구성하려면 케라스 함수 API를 사용하면 됩니다

데이터셋  
생성하기

모델  
구성하기

모델  
학습과정  
설정하기

모델  
학습  
시키기

학습과정  
살펴보기

모델  
평가하기

모델  
사용하기

# 케라스 기본 개념



케라스에서 제공하는 시퀀스 모델  
필요하는 등 좀 더 복잡한 모델을

데이터셋  
생성하기

모델  
구성하기

모델  
학습  
설정

In [1]:

# 0. 사용할 패키지 불러오기

```
from keras.utils import np_utils
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Activation
```

# 1. 데이터셋 생성하기

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(60000, 784).astype('float32') / 255.0
x_test = x_test.reshape(10000, 784).astype('float32') / 255.0
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
```

# 2. 모델 구성하기

```
model = Sequential()
model.add(Dense(units=64, input_dim=28*28, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
```

# 3. 모델 학습과정 설정하기

```
model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])
```

# 4. 모델 학습시키기

```
hist = model.fit(x_train, y_train, epochs=5, batch_size=32)
```

# 5. 학습과정 살펴보기

```
print('## training loss and acc ##')
print(hist.history['loss'])
```

The background is a dark, abstract composition featuring several large, semi-transparent geometric shapes. On the left, a large red shape with a white line runs diagonally. In the center, a teal shape is prominent. To the right, a dark blue shape is visible. Numerous thin, white light rays or lines radiate from various points, creating a sense of depth and movement. The overall color palette is dominated by dark blues, teals, and reds, with white highlights from the text and light rays.

# 03 데이터 셋

Data set

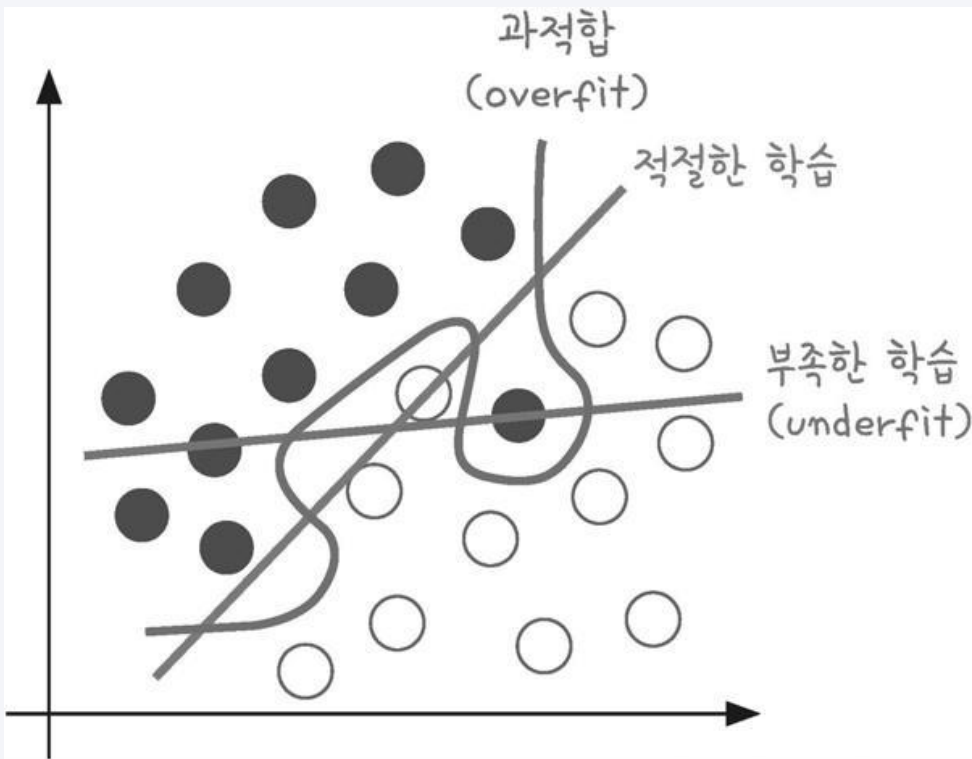


자료의 집합 – 모델을 학습시키기 위한 자료들

## 학습 셋 vs 테스트 셋

## 과적합(Over Fitting)

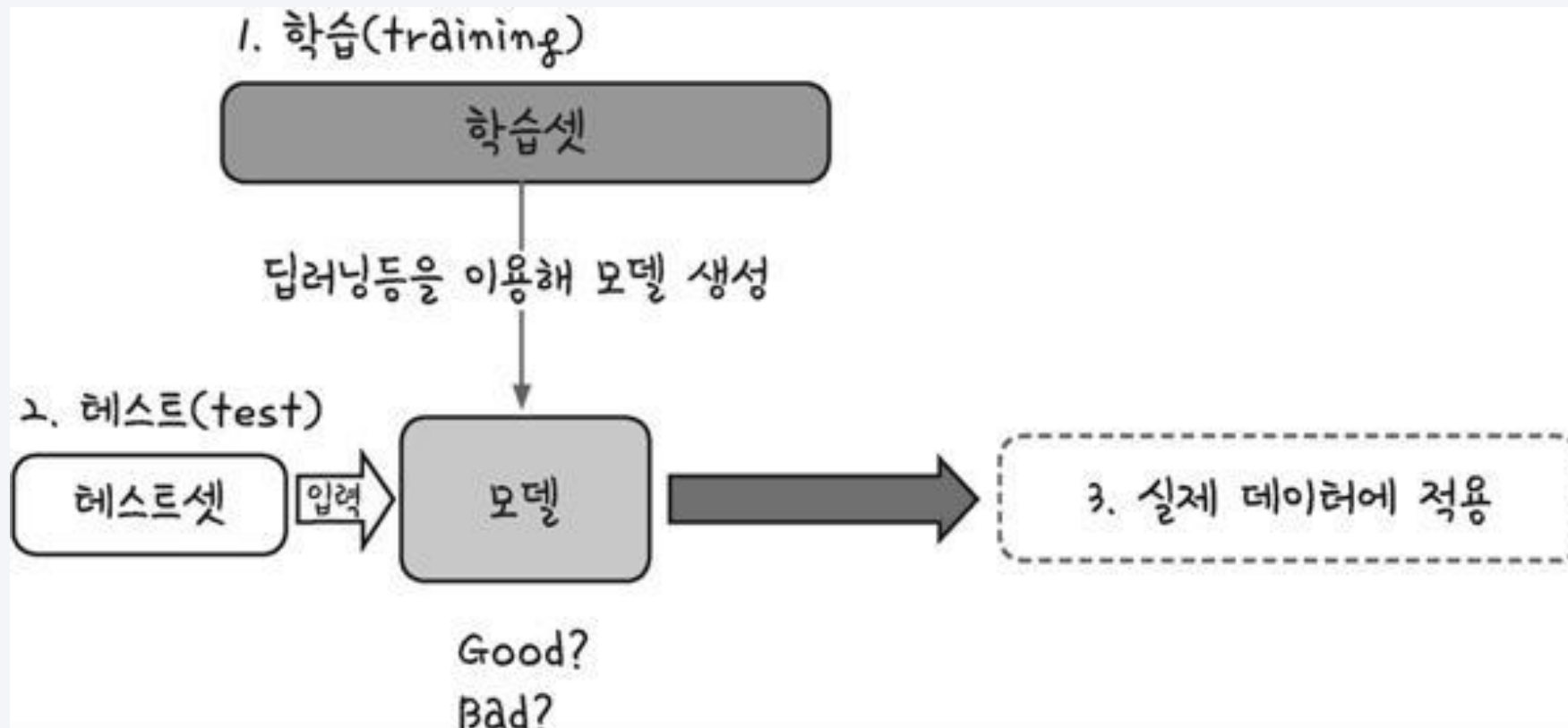
모델이 학습 데이터셋 안에서는 일정 수준 이상의 예측 정확도를 보이지만 새로운 데이터에 적용하면 잘 맞지 않는 것



두 그룹  
정확히 나누기 불가능!

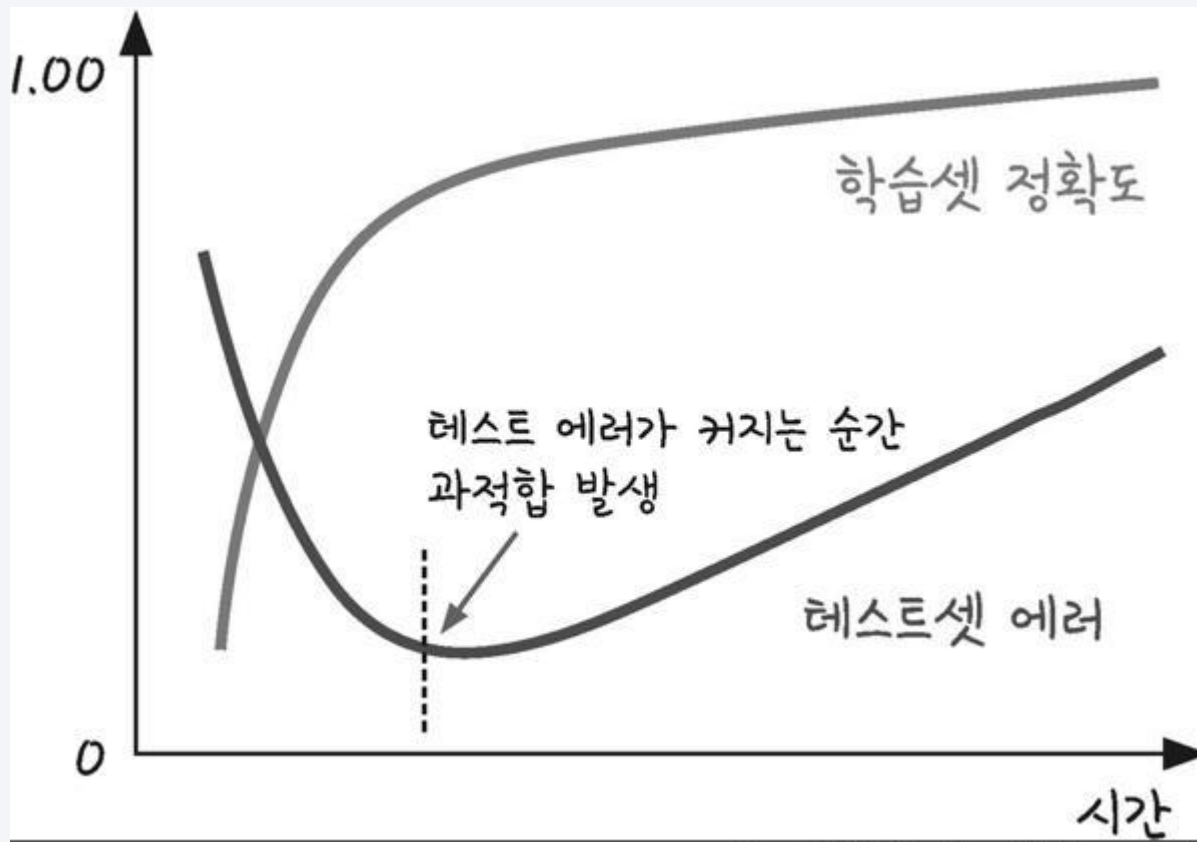
# 데이터 셋

데이터 셋을 학습셋과 테스트셋으로 분리

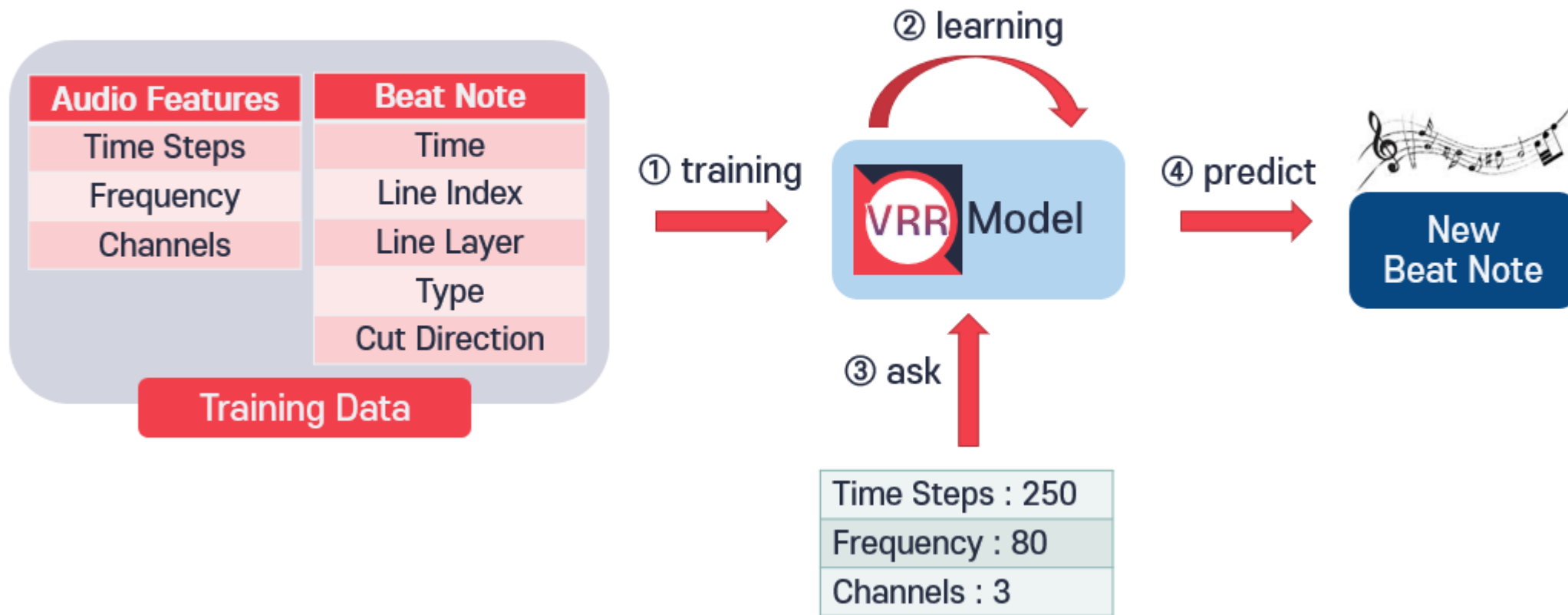


## 데이터 셋

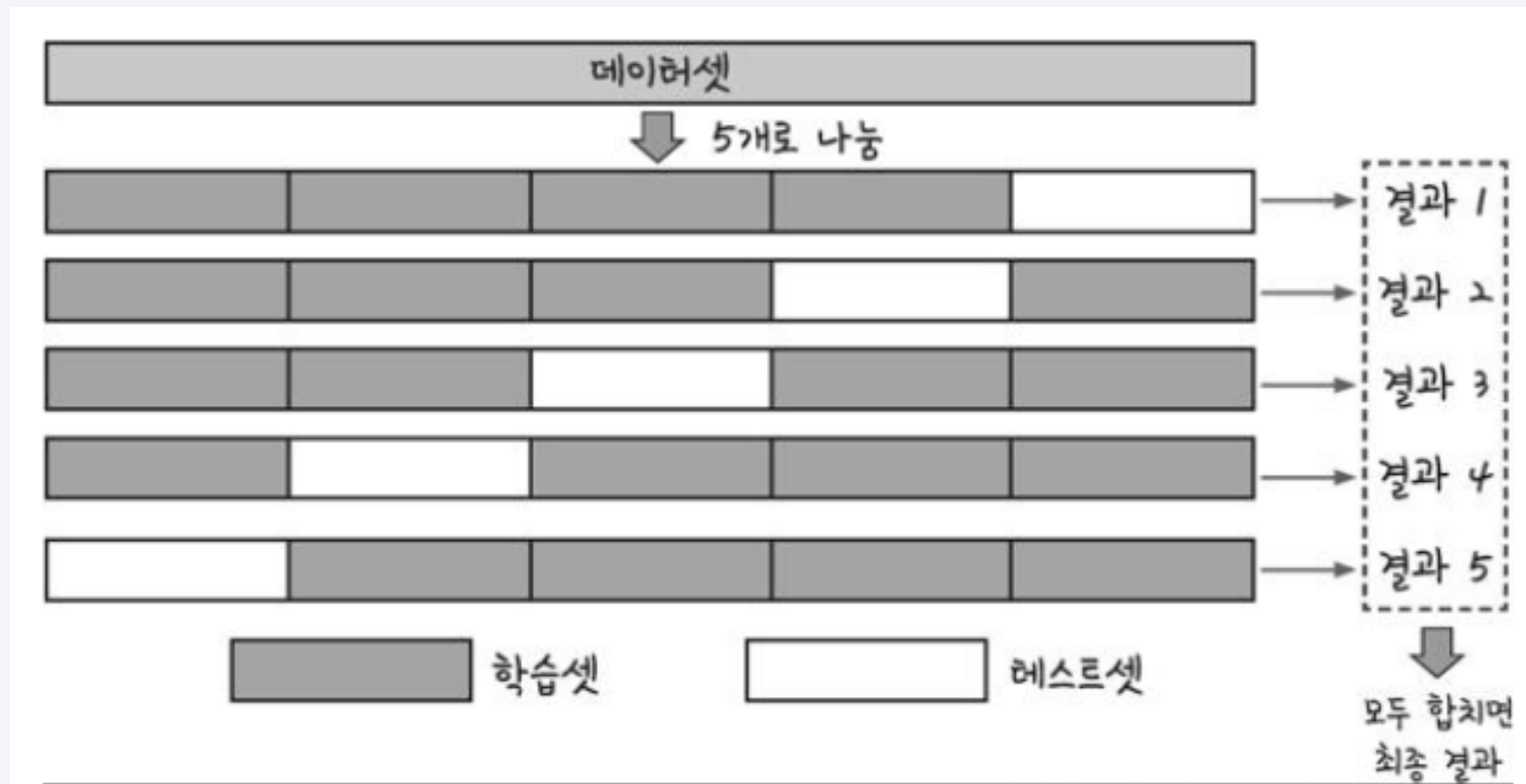
학습 데이터 셋만으로 평가한 예측 성공률이 높아  
져도 테스트 셋에서 효과가 없음 -> 과적합



## VRR 지도 학습(Supervised Learning) Flow Chart



# 데이터 셋





## Directory structure

- | -Cover.jpg
- | -info.dat
- | -Normal.dat
- | -Hard.dat
- | -Expert.dat
- | -song.egg

Hard.dat - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
{ "_version": "2.0.0", "_BPMChanges": [], "_events": [ { "_time": 0, "_type": 4, "_value": 3 }, { "_time": 1, "_type": 4, "_value": 3 }, { "_time": 28, "_type": 4, "_value": 3 }, { "_time": 28.75, "_type": 1, "_value": 7 }, { "_time": 29.5, "_type": 1, "_value": 3 }, { "_time": 60.75, "_type": 0, "_value": 3 }, { "_time": 61.5, "_type": 0, "_value": 3 }, { "_time": 61.5, "_type": 1, "_value": 7 }, { "_time": 76, "_type": 1, "_value": 7 }, { "_time": 76, "_type": 9, "_value": 0 }, { "_time": 94, "_type": 1, "_value": 7 }, { "_time": 94.75, "_type": 1, "_value": 7 }, { "_time": 95.5, "_type": 1, "_value": 7 }, { "_time": 116, "_type": 9, "_value": 0 }, { "_time": 116, "_type": 8, "_value": 0 }, { "_time": 120, "_type": 3, "_value": 2 }, { "_time": 138.25, "_type": 2, "_value": 2 }, { "_time": 138.25, "_type": 3, "_value": 5 }, { "_time": 156, "_type": 0, "_value": 1 }, { "_time": 156, "_type": 1, "_value": 1 }, { "_time": 176, "_type": 3, "_value": 0 }, { "_time": 176, "_type": 0, "_value": 0 }, { "_time": 176, "_type": 9, "_value": 0 }, { "_time": 192, "_type": 9, "_value": 0 }, { "_time": 192, "_type": 0, "_value": 1 }, { "_time": 192, "_type": 1, "_value": 3 }, { "_time": 210, "_type": 0, "_value": 7 }, { "_time": 210, "_type": 1, "_value": 7 }, { "_time": 210, "_type": 1, "_value": 7 } ] }
```

Normal.dat - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
{ "_version": "2.0.0", "_BPMChanges": [], "_events": [ { "_time": 0, "_type": 4, "_value": 3 }, { "_time": 1, "_type": 4, "_value": 3 }, { "_time": 28, "_type": 4, "_value": 3 }, { "_time": 28.75, "_type": 1, "_value": 7 }, { "_time": 29.5, "_type": 1, "_value": 3 }, { "_time": 60.75, "_type": 0, "_value": 3 }, { "_time": 61.5, "_type": 0, "_value": 3 }, { "_time": 61.5, "_type": 1, "_value": 7 }, { "_time": 76, "_type": 1, "_value": 7 }, { "_time": 76, "_type": 8, "_value": 0 }, { "_time": 94, "_type": 1, "_value": 7 }, { "_time": 94.75, "_type": 1, "_value": 7 }, { "_time": 95.5, "_type": 1, "_value": 7 }, { "_time": 116, "_type": 9, "_value": 0 }, { "_time": 116, "_type": 1, "_value": 7 }, { "_time": 120, "_type": 3, "_value": 2 }, { "_time": 138.25, "_type": 1, "_value": 6 }, { "_time": 138.25, "_type": 0, "_value": 1 }, { "_time": 156, "_type": 3, "_value": 5 }, { "_time": 156, "_type": 2, "_value": 5 }, { "_time": 176, "_type": 2, "_value": 0 }, { "_time": 176, "_type": 1, "_value": 2 }, { "_time": 176, "_type": 1, "_value": 2 } ] }
```

# 데이터 셋

## Beat Saber note 분석

강남스타일 expart.dat 분석

```
{
  "_version": "2.0.0",

  /*배경 효과 이벤트*/
  "_events": [{"_time": 0, "_type": 0, "_value": 1}....]
time      시간
type      배경변화 유형
value     배경 값
```

```
/*배치된 비트 노트에 관한 데이터*/
"_notes": [{"_time": 5, "_lineIndex": 2, "_lineLayer": 0, "_type": 1, "_cutDirection": 1}.....]
time      시간
lineindex 좌우위치 (왼쪽부터 0123)
lineLayer 상하위치 (밑에서부터 012)
type      노트 색   (빨 : 0, 파 : 1)
cutdirection 자르는 방향
(아래 : 0 위 : 1 오른쪽 : 2 왼쪽 : 3 대각선 : 4, 5, 6, 7)
```

/\*장애물에 관한 데이터\*/

```
"_obstacles": [{"_time": 39.75, "_lineIndex": 2, "_type": 0,
  "_duration": 1.5, "_width": 2}....]
time      시간
lineindex 시작위치
type      장애물 유형
duration  지속시간
width     장애물 넓이
```



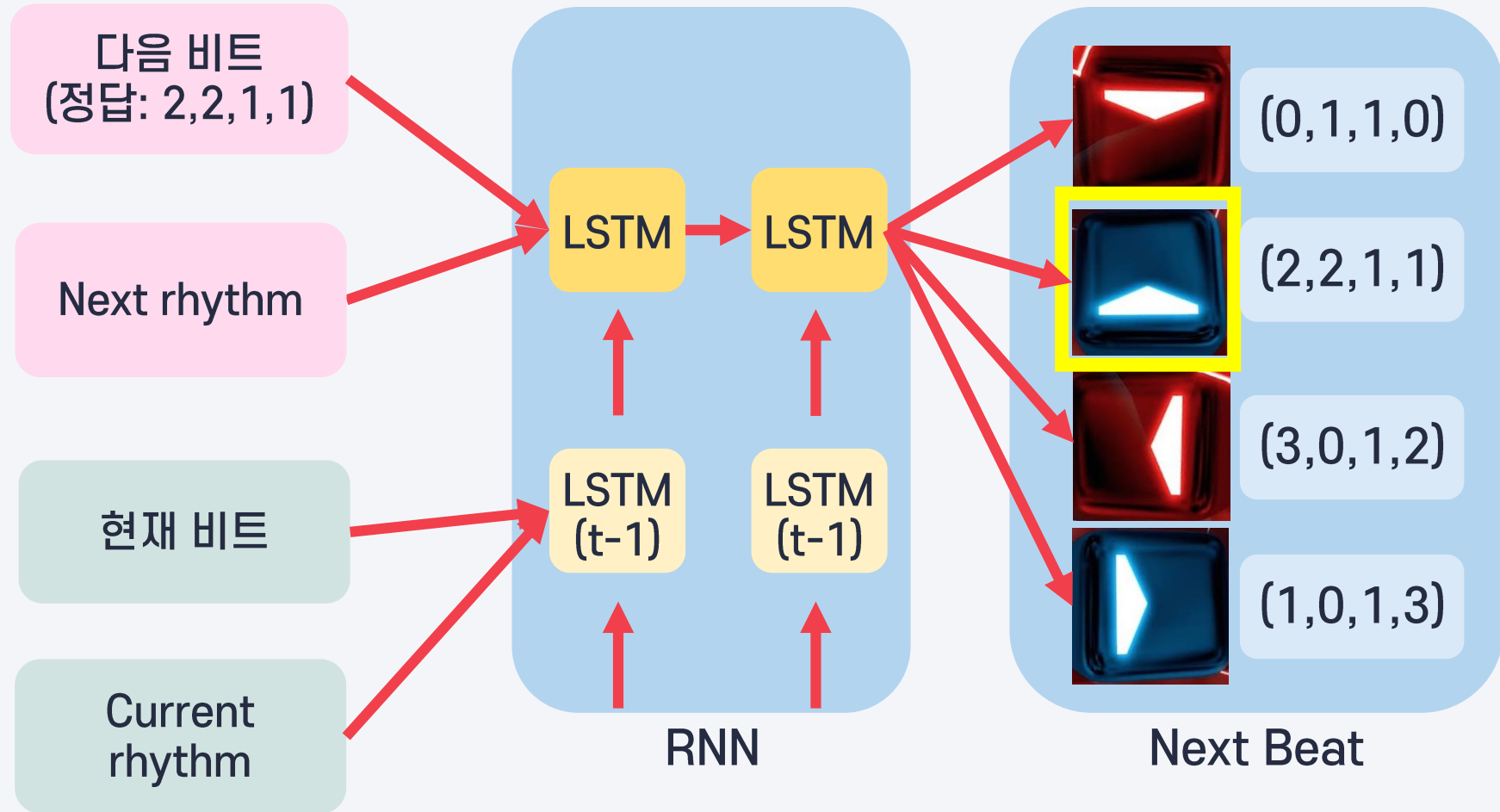
## 데이터 셋

### Next beat Prediction (다음 비트 예측)

[ (문제) :

(Current Rhythm,  
현재 비트, Next Rhythm)  
그렇다면, 다음 비트 값은?

(정답): 다음 비트 ]



# Q & A

VR Rhythm game beat maker using deep learning