SPICED

# Melody Starter

Can Çağıncan

# AI Music Generation

## some applications:

- Composition of music
- Music mastering
- Music recomendation
- Hit song prediction and acquisition for record companies
- Making of new sounds
- Smart instruments

# Melody Starter

## the Project

Music generation. Trained with classical piano pieces

## the Aim

Compose melody from given notes, 1st stage: only sequence of notes!

## the Method

CNN with WaveNet architecture

# Generating music
## some challenges for AI:

Creative and musical challenges:
creativity,
musical context,
understanding musical structure and style,
incorporating improvisation.

Technical challenges:
Data availability,
Computational and model complexity,
Interpretability.

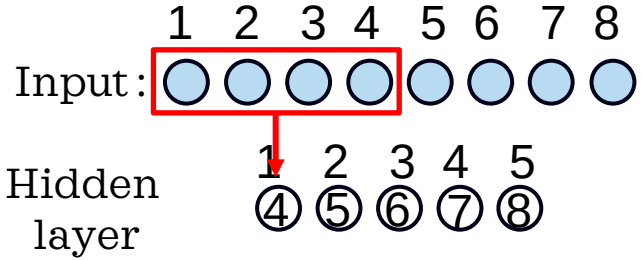
# Generating music

## WaveNet:

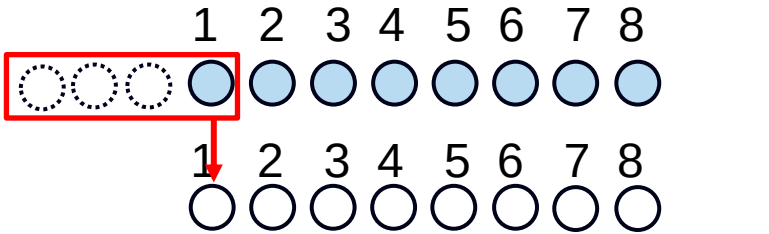
based on **Dilated** 1D **Causal** Convolution layers

It is a timeseries problem! Causal Convolution:

1D Convolution Layer




1D Causal Convolution Layer

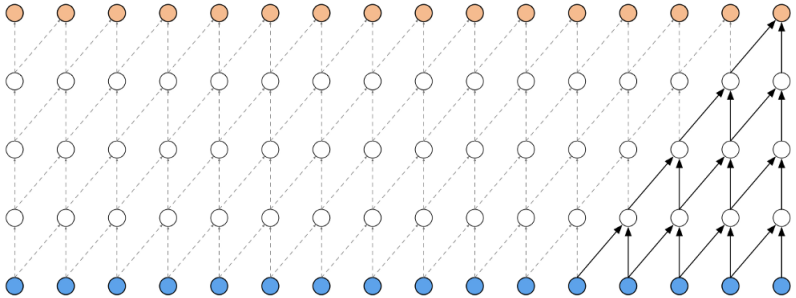


# Generating music

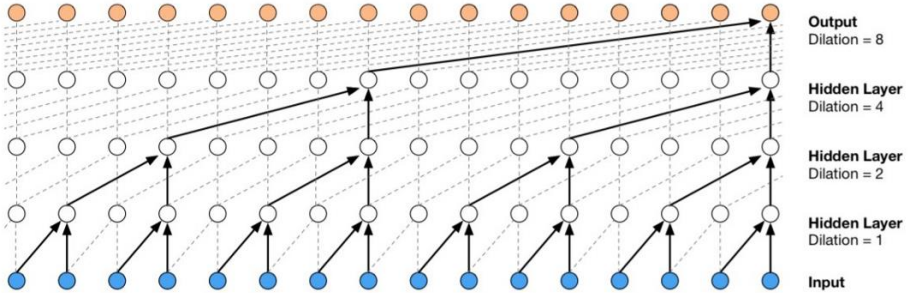
## WaveNet:

Handicap: narrow receptive field

1D Causal Convolution

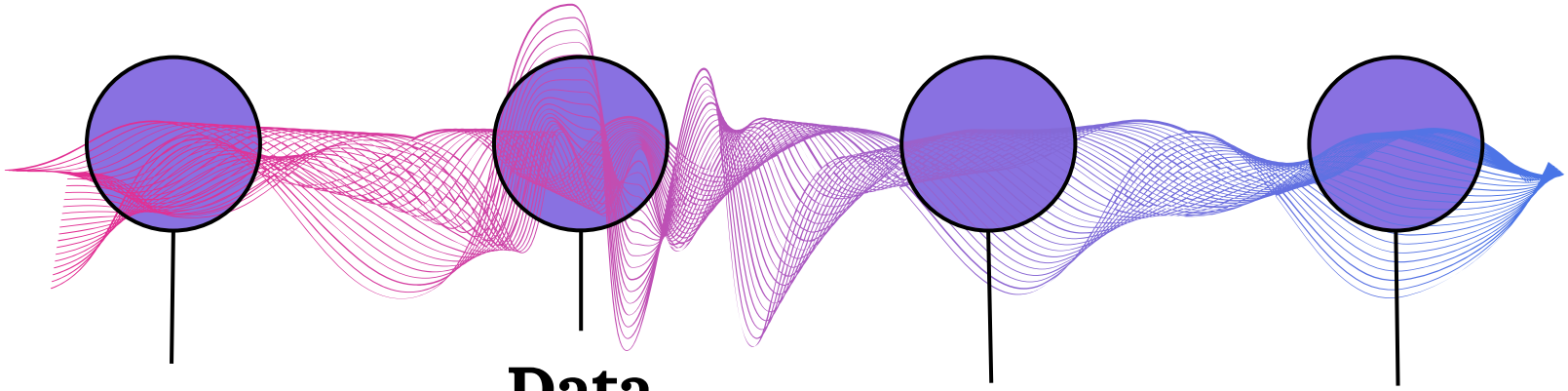


Dilated 1D Causal Convolution



SPICED

# Code



## Extract Musics

From MIDI files by **Music21** and **Mido** packages as sequence of notes

## Data Preparation

Cut song in **pieces** and combine all songs

## Train model

Train model in **wavenet** architecture

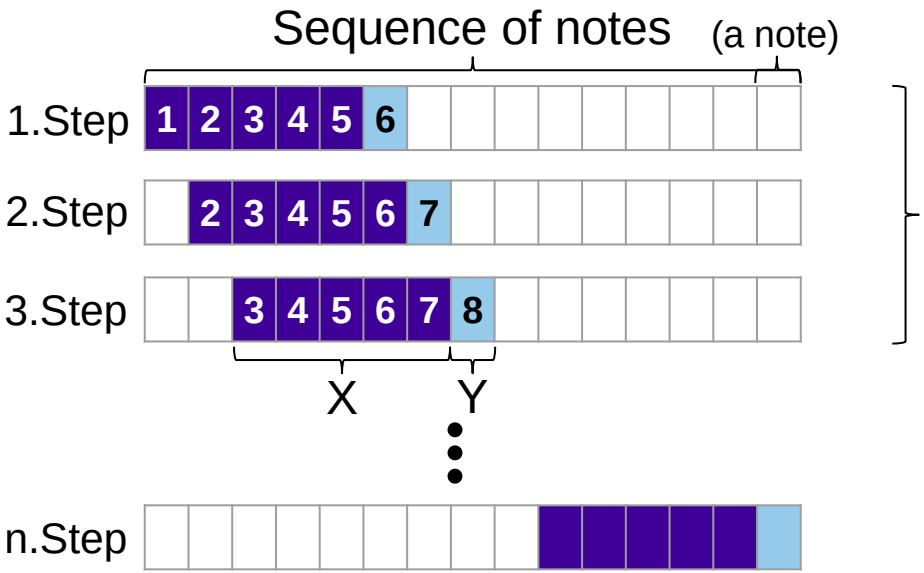
## Start a melody

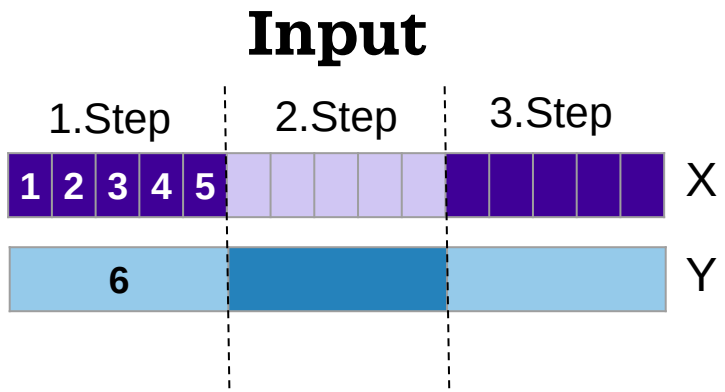
**Give** a **melody** to generate melody

# Data Preparation

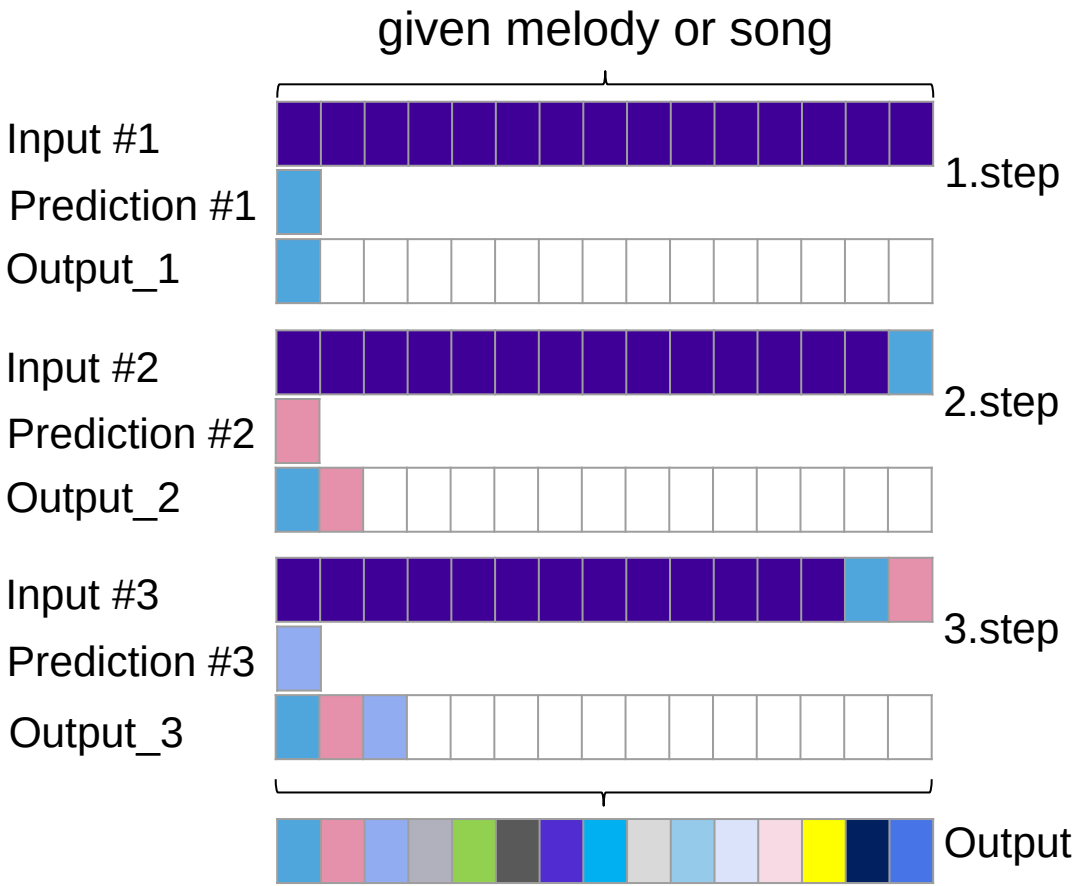
**1.Extract :** into sequence of notes

**2.Cut in pieces :** in a way to predict subsequent note from previous notes.



# Prediction Processing



# Results

**Model 1**
Music21

**Model 2**
Mido

in progress ...

**Next steps:**

- Finish the second model
- Add note duration, delay and amplitude
- Train with more data
- Try another models

# Thanks!

Do you have any questions?

can.cagincan@fu-berlin.de

# Model

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv1d (Conv1D) | (None, 16, 64) | 256 |
| dropout (Dropout) | (None, 16, 64) | 0 |
| max_pooling1d (MaxPooling1D) | (None, 16, 64) | 0 |
| dense (Dense) | (None, 16, 256) | 16640 |
| conv1d_1 (Conv1D) | (None, 16, 64) | 49216 |
| dropout_1 (Dropout) | (None, 16, 64) | 0 |
| max_pooling1d_1 (MaxPooling1D) | (None, 8, 64) | 0 |
| conv1d_2 (Conv1D) | (None, 4, 256) | 82176 |
| global_max_pooling1d (GlobalMaxPooling1D) | (None, 256) | 0 |
| dense_1 (Dense) | (None, 256) | 65792 |
| dense_2 (Dense) | (None, 108) | 27756 |

Total params: 241,836
Trainable params: 241,836
Non-trainable params: 0