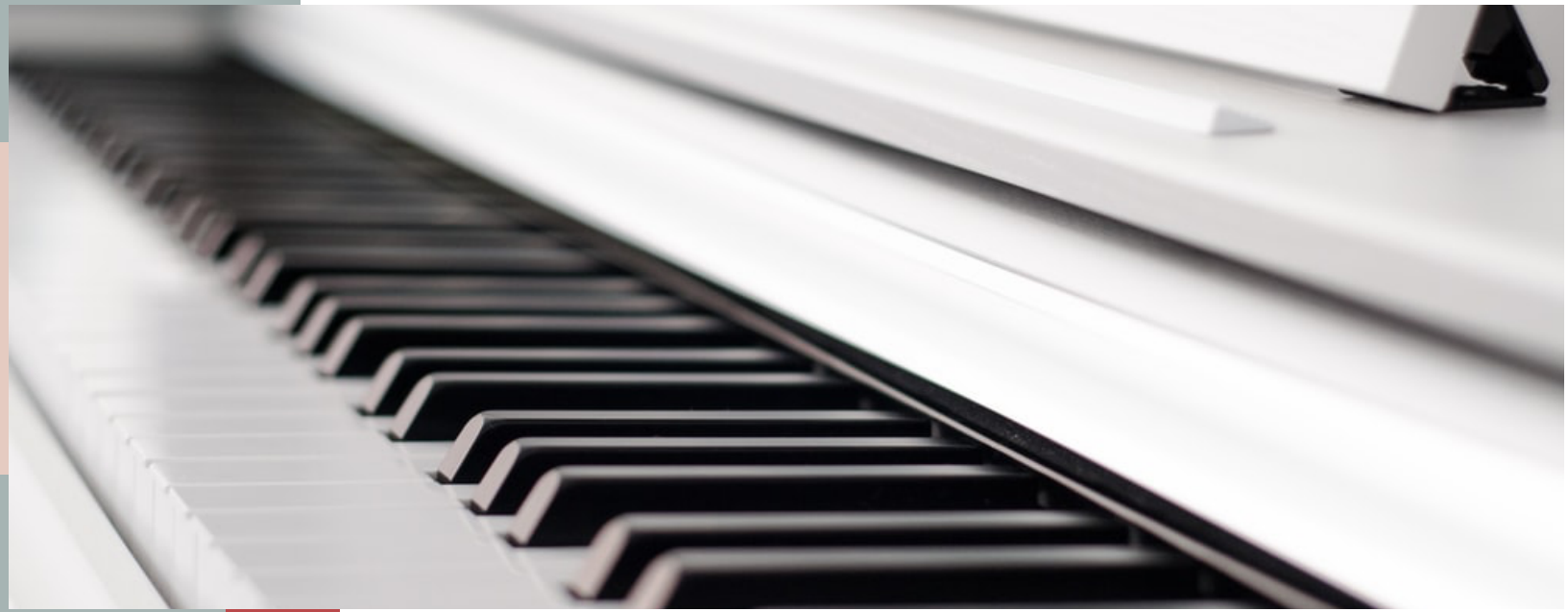


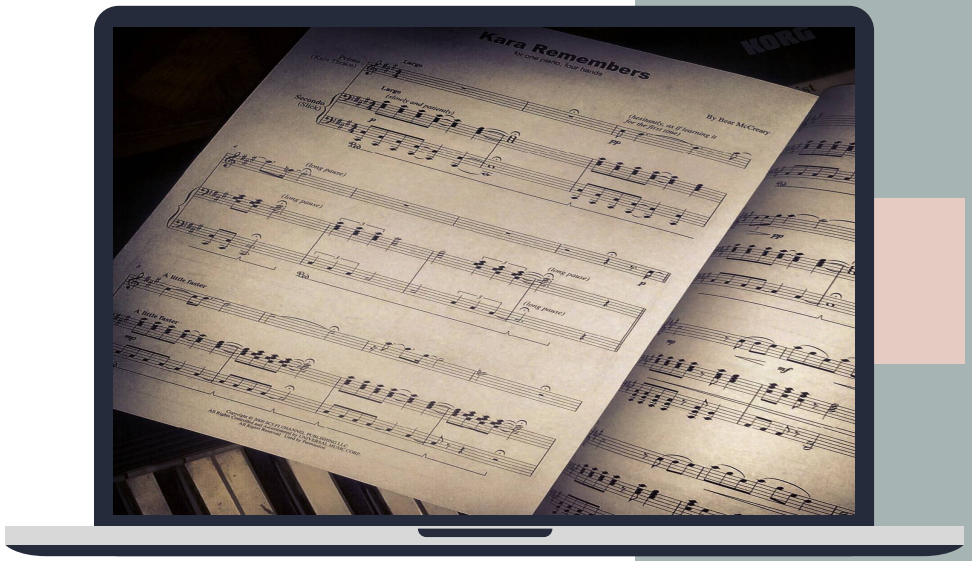
CLASSICAL MUSIC GENERATION
WITH DEEP LEARNING

CYNTHIA WANG

MOTIVATION

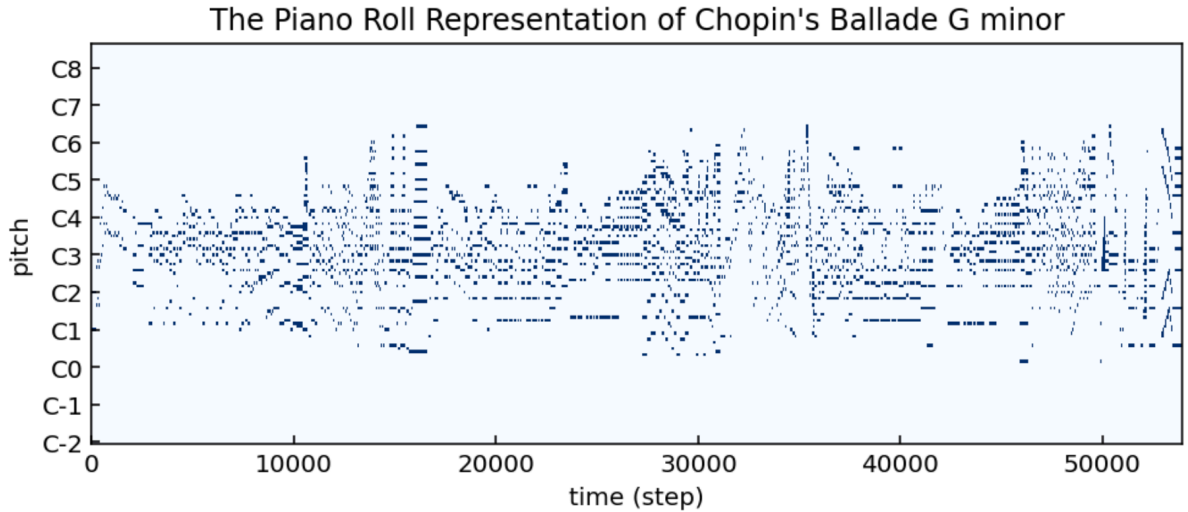
Is data science only about science?

I’m not a musician.
But can I make my own music?



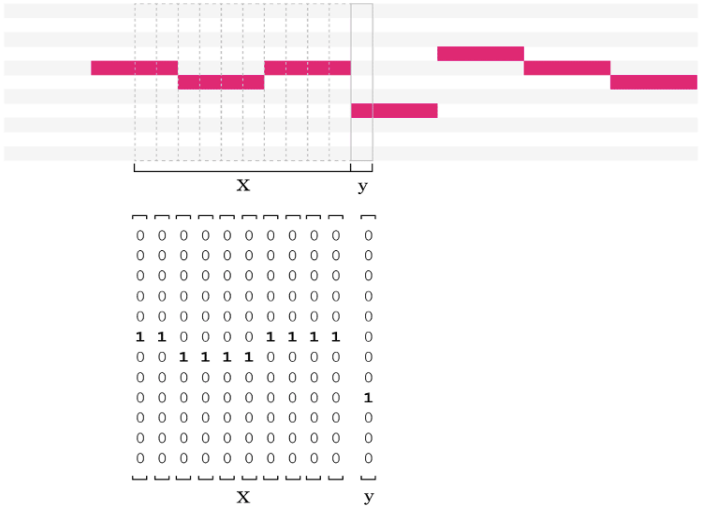
DATA PREPROCESSING

- 316 midi files of classical piano music
- Extract the piano roll of each song with pretty_midi



DATA PREPROCESSING

Training window - X
Target - y



A dictionary of timesteps and notes

```
X  128: '45,49,57,61,73,76,81',
   129: '45,49,57,61,73,76,81',
   130: '45,49,57,61,73,76,81',
   131: '45,49,52,57,61,64,71,73,76,80,81',
   132: '45,49,52,57,61,64,71,73,76,80,81',
   133: '52,64,71,76,80',
   134: '52,64,71,76,80',
y  135: '52,64,71,76,80',
   136: '52,64,71,76,80',
   137: '52,64,71,76,80',
   138: '52,64,71,76,80',
```

➜ Training window size = 50
Total training windows = 1.5 million

gif from: https://brangerbriz.com/blog/using-machine-learning-to-create-new-melodies/

MODEL ARCHITECTURE

Embedding – from a sparse representation to a dense representation

Bidirectional GRU – good with sequential information; mitigate the vanishing gradient problem of RNN and faster than LSTM

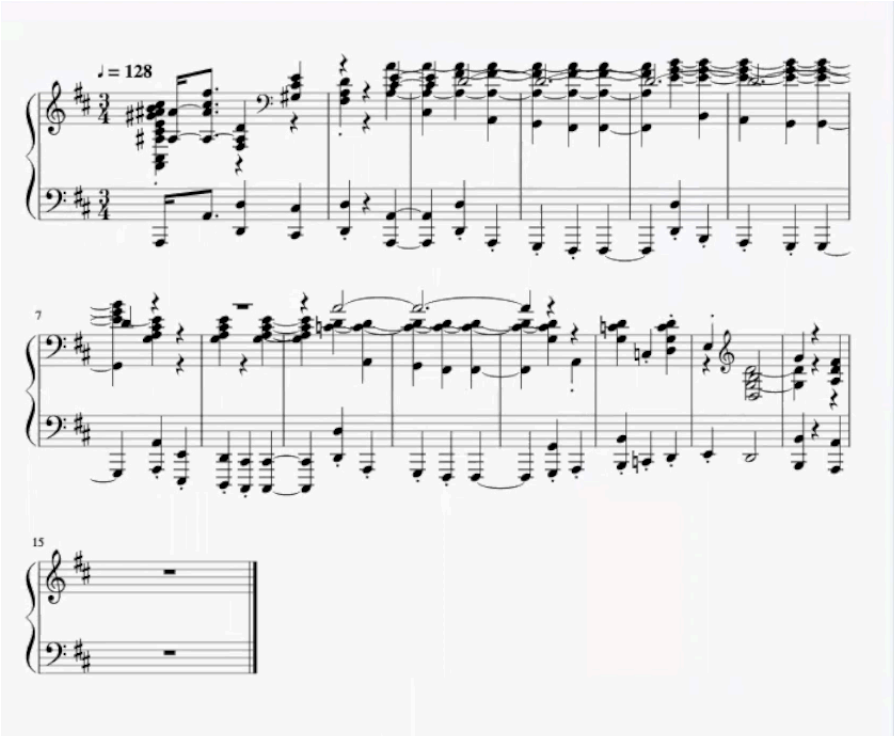
Self-attention – inputs interact with each other and find out who they should pay more attention to

Dropout – prevent overfitting

GENERATION

316 songs

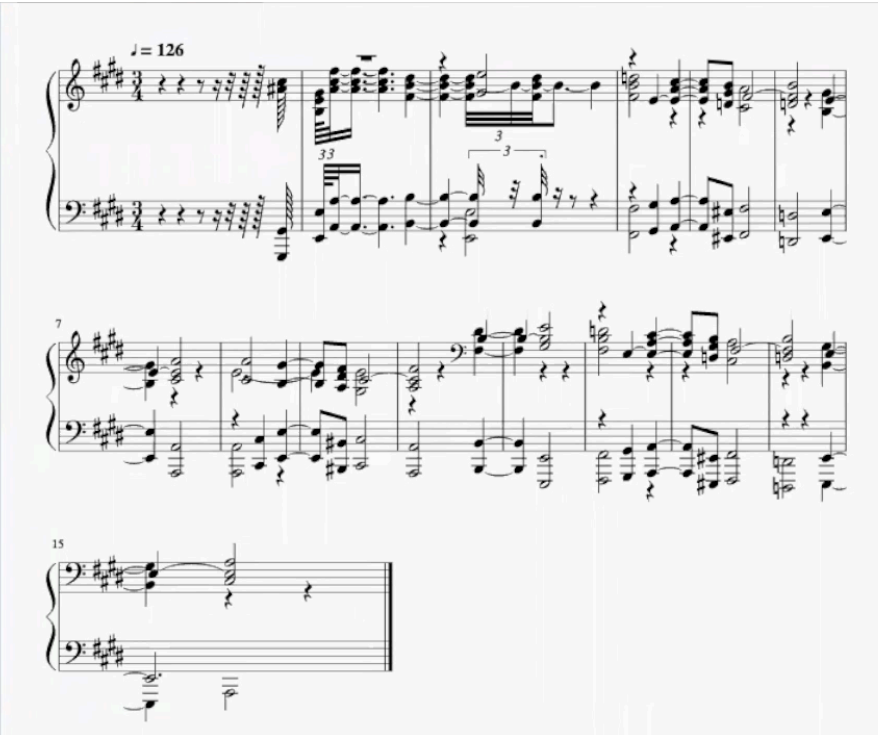
10 epochs



GENERATION

316 songs

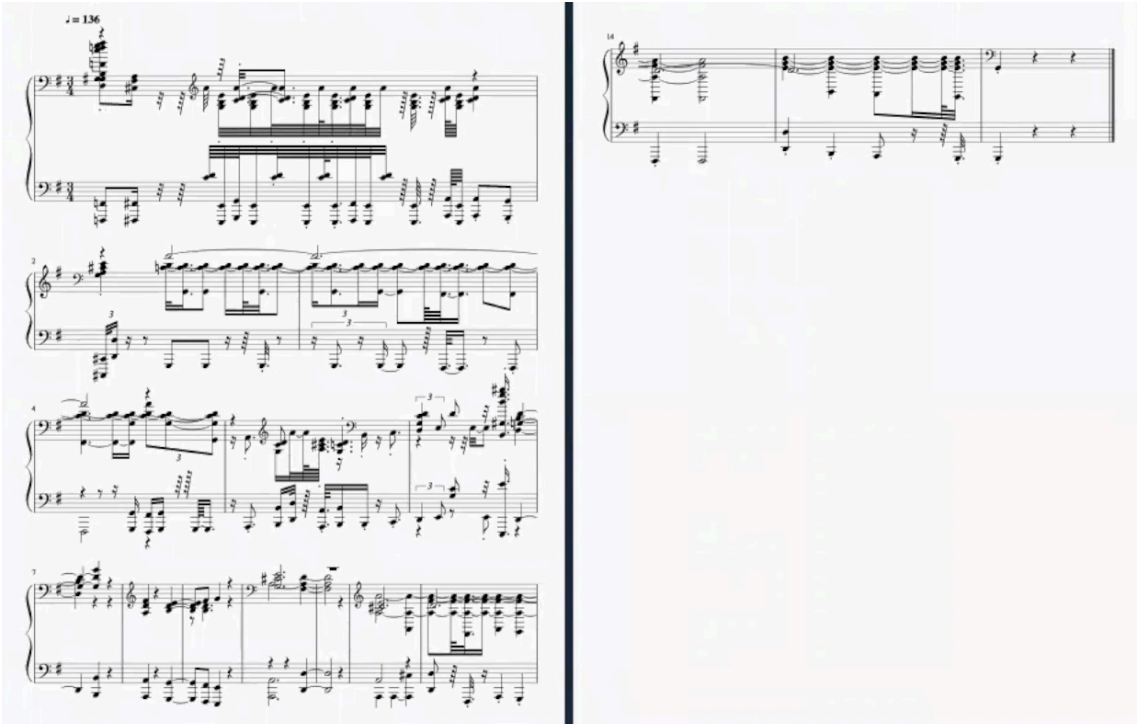
20 epochs



GENERATION

The New Chopin

20 epochs



FUTURE WORK

- Generate songs for other composers
- Explore other instruments
- Explore other model architectures

THANKS!

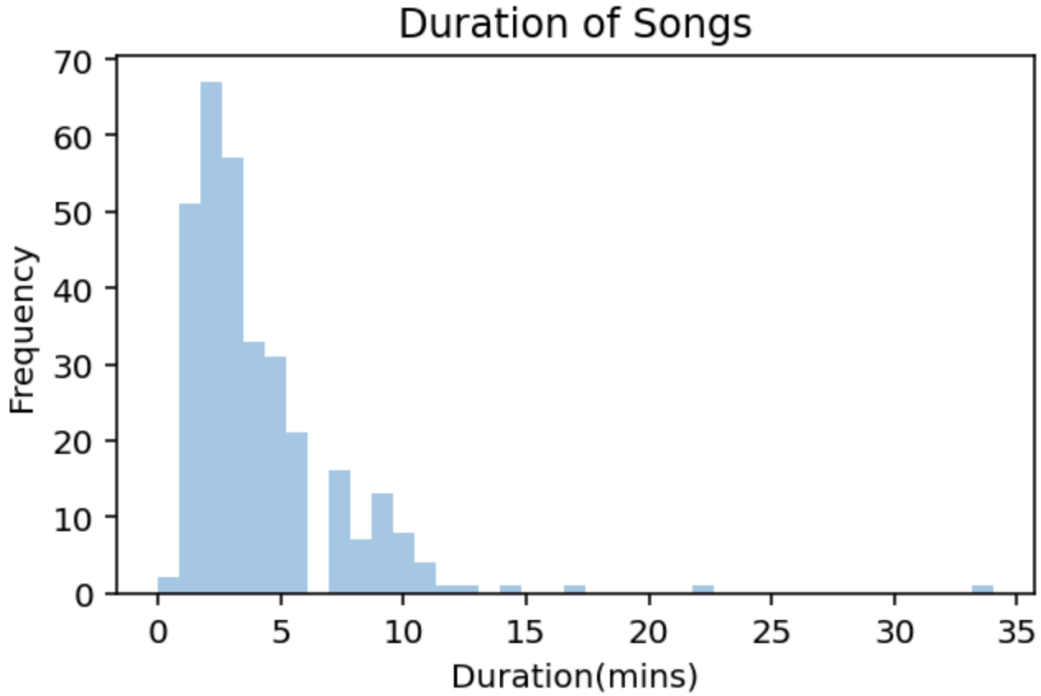
cynthiawang315@gmail.com

linkedin.com/in/cynthiawang181128/

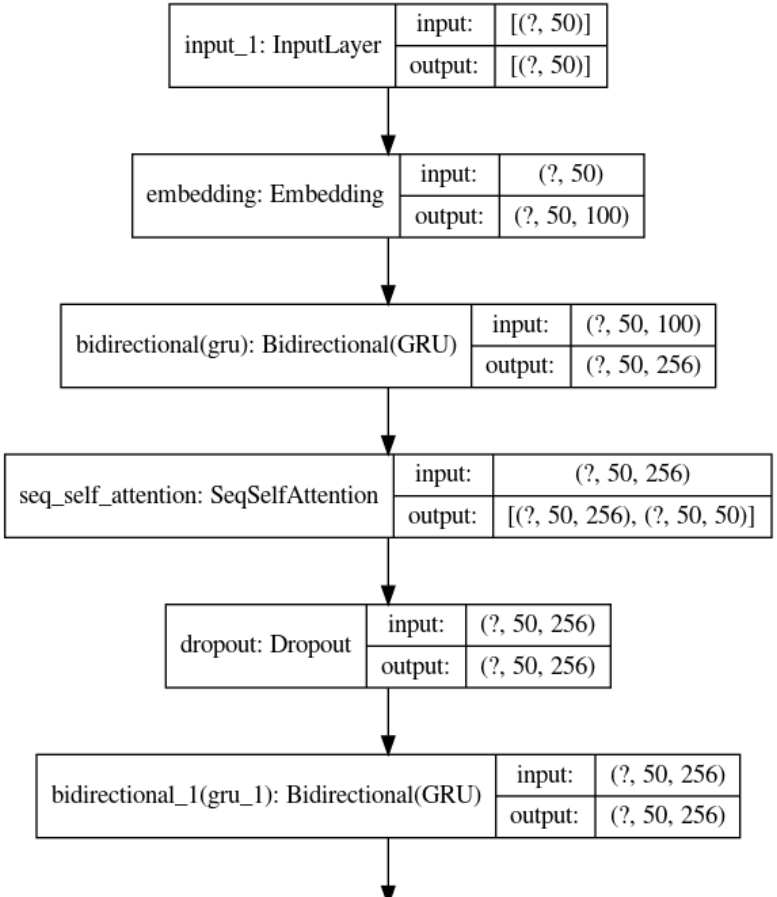
github.com/cynthiawang315

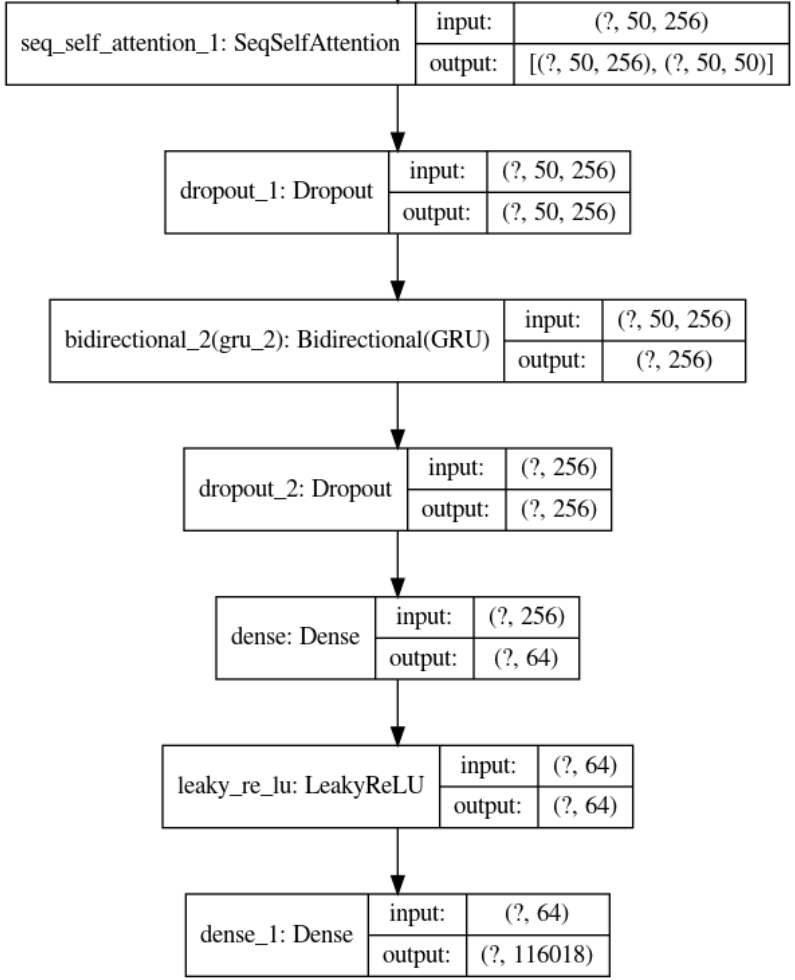
CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

APPENDIX


Duration of Songs
Frequency
70
60
50
40
30
20
10
0
0
5
10
15
20
25
30
35
Duration(mins)

APPENDIX


input_1: InputLayer
input: [(?, 50)]
output: [(?, 50)]
embedding: Embedding
input: (?, 50)
output: (?, 50, 100)
bidirectional(gru): Bidirectional(GRU)
input: (?, 50, 100)
output: (?, 50, 256)
seq_self_attention: SeqSelfAttention
input: (?, 50, 256)
output: [(?, 50, 256), (?, 50, 50)]
dropout: Dropout
input: (?, 50, 256)
output: (?, 50, 256)
bidirectional_1(gru_1): Bidirectional(GRU)
input: (?, 50, 256)
output: (?, 50, 256)


seq_self_attention_1: SeqSelfAttention
input: (?, 50, 256)
output: [(?, 50, 256), (?, 50, 50)]
dropout_1: Dropout
input: (?, 50, 256)
output: (?, 50, 256)
bidirectional_2(gru_2): Bidirectional(GRU)
input: (?, 50, 256)
output: (?, 256)
dropout_2: Dropout
input: (?, 256)
output: (?, 256)
dense: Dense
input: (?, 256)
output: (?, 64)
leaky_re_lu: LeakyReLU
input: (?, 64)
output: (?, 64)
dense_1: Dense
input: (?, 64)
output: (?, 116018)