AI Jukebox

An Exploration in Generative Models

Brian McMahon

5 April 2018

"What I cannot create, I do not understand."

-Richard Feynman

Generative Model

Powerful approach to un/semi-supervised learning - *no labels required*

Discover hidden structure within data

Generate new, unique data from internal, latent structure

Potential creations of generative models:

- Images
- Audio
- Text
- Code
- Design
- Blueprints
- Physical structures

LSTM Network

- Have “memory”, allowing information to persist, including *long-term dependencies*
- At each timestep, previous state is passed in along with new input
- “Gate” functionality managing “cell” state: forget, input, output

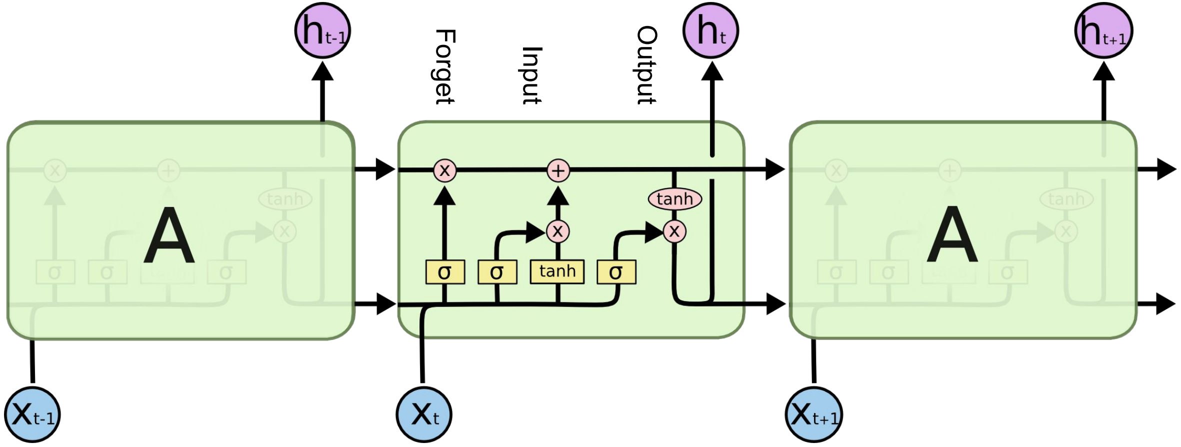


Diagram courtesy of [“colah’s blog”](colah’s blog).

AI Jukebox

Generative model

Latent space of music mapped by model “memory”

Exploration of creativity in AI

Tools

Keras

TensorFlow

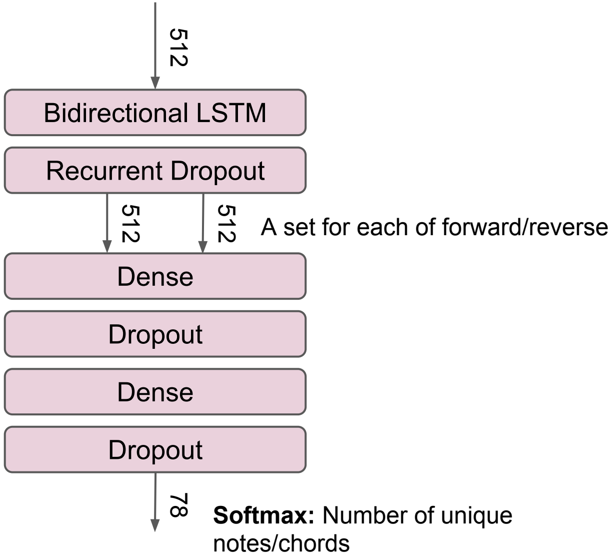
python™

music21

musescore

Architecture

Bidirectional LSTM



- **Dataset:** collection of midi files
- 512 node input layer, softmax for each unique note/chord in collection
- Bidirectional (forward/reverse) dual layers
- Dropout 0.5 on all layers
- Learning rate 0.001
- Sequence length 200
- Notes generated 500

Dance Music

Generated output from training on dance music

Piano

Drums

Robot image courtesy of trendhunter.com.

Key Takeaways

Explored one way a model can generate unique, new content

Evocative beat patterns - but perhaps not in the running for awards just yet

Model just “scratches the surface” of generative modelling in music - more work to be done!

Next Steps

Continue to refine model performance. Explore a variety of:

- Datasets - collections of music by genre, artist, style
- Architectures - GAN, variational autoencoders, attention RNN
- Inputs - raw audio, text

Write model into flask app and implement online

- Input a collection of music, output AI generated content!

Thank You!

bcm822@gmail.com

Appendix

Resources

Dorsey, Brannon. “Using Machine Learning to Create New Melodies.” https://brangerbriz.com/. 10 May 2017.

Nayebi, Aran. "GRUV: Algorithmic Music Generation using Recurrent Neural Networks." Stanford University. 2015.

Skúli, Sigurður. "How to Generate Music using a LSTM Neural Network in Keras." www.towardsdatascience.com. December 7, 2017.

Brownlee, Jason. "Stacked LSTM Networks." https://machinelearningmastery.com. August 18, 2017.

Brownlee, Jason. "Understand the Difference Between Return Sequences and Return States for LSTMs in Keras." https://machinelearningmastery.com. October 24, 2017.

“Understanding LSTM Networks.” Colah’s Blog. https://colah.github.io. 27 August 2015.

Goodfellow, Ian. “Deep Learning.” MIT Press. http://www.deeplearningbook.org/. 2016.

“Magenta.” Tensorflow. Magenta.tensorflow.org.

A Model that Remembers

Recurrent (esp. LSTM) model an essential component of:

- Sound and speech recognition
- Time series prediction: traffic, recommender systems, stock movement
- Natural Language Processing (NLP): machine translation, chatbots
- Digital assistants

Creativity in AI

A long disputed and contentious question: can AI be creative?

- "Remixing" precedent with a dose of stochasticity
- Potential to generate new thoughts and ideas unbounded by the human experience

Datasets

Scraped by genre from various websites

Genre	# Midi	# Notes	# Unique Notes	Source
Celtic	338	159,789	78	Tadpole Tunes
Dance	200	309,967	663	MidiWorld
Game	91	51,177	358	Final Fantasy soundtracks*
Classical				MidiWorld

**See blog post by Sigurour Skuli, Towards Data Science.*

Evaluation

As the model is generative (as opposed to discriminative), the best judges are us

Testing whether LSTM can successfully capture:

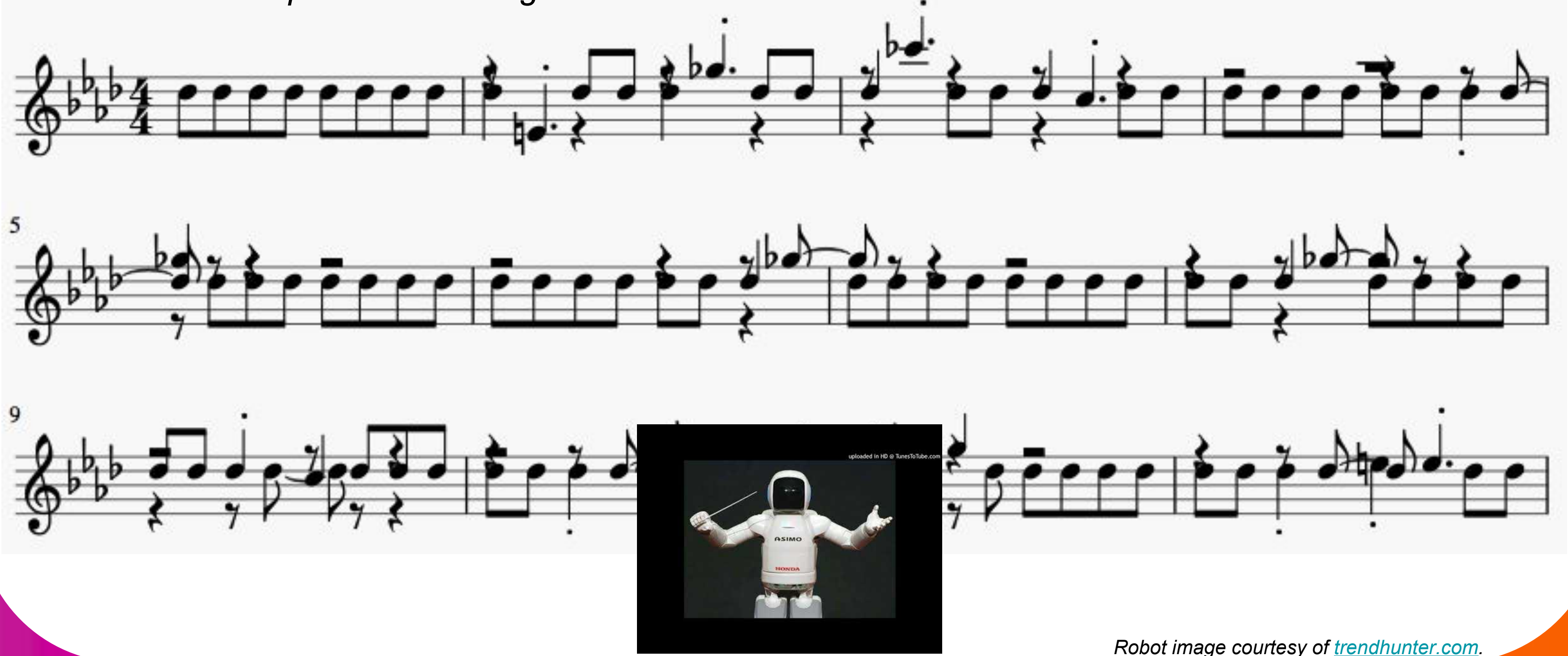
- Repeating long term structure, strong temporal constraints
- Low train and validation loss
- Most importantly, pleasing to the ear

[replace with final]

Celtic Music

[to narrow down to two best performing]

Generated output from training on Celtic music

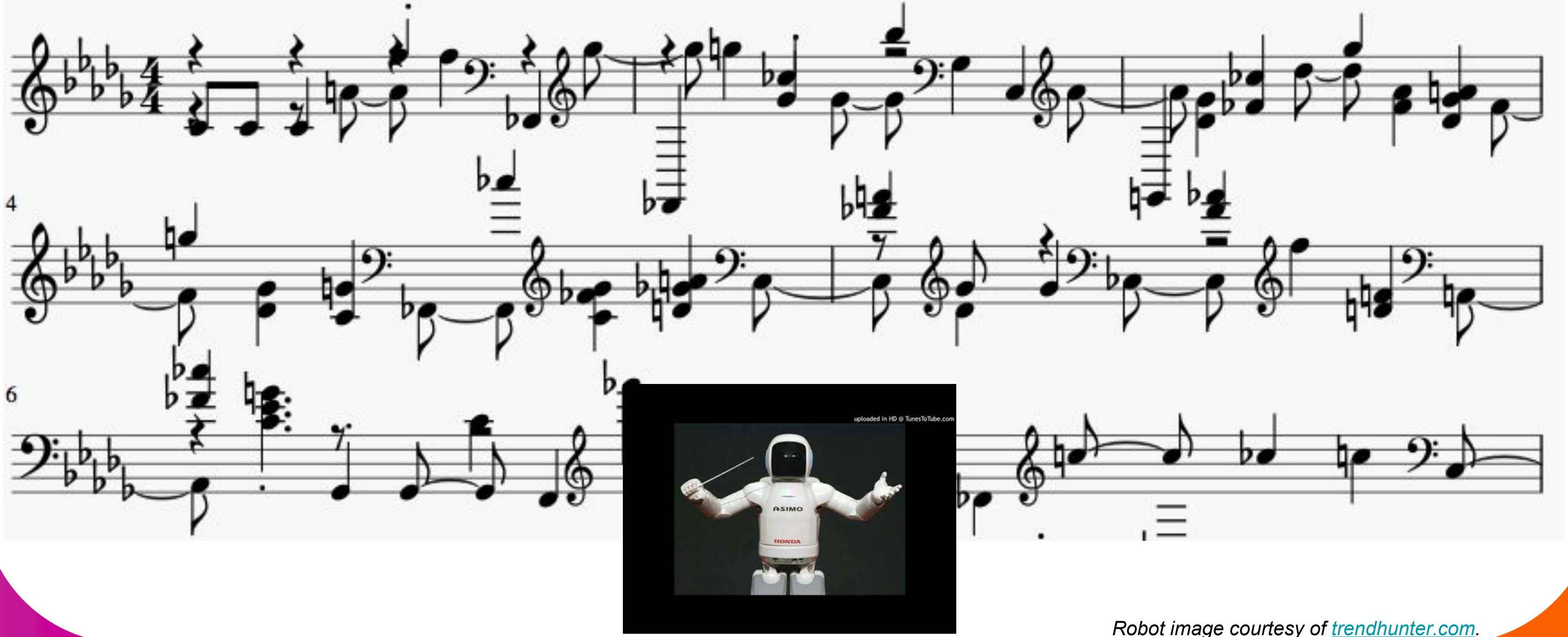


Robot image courtesy of trendhunter.com.

Game Music

[to narrow down to two best performing]

Generated output from training on Final Fantasy soundtracks



Robot image courtesy of trendhunter.com.

Classical Music

[to train]

[to narrow down to two best performing]

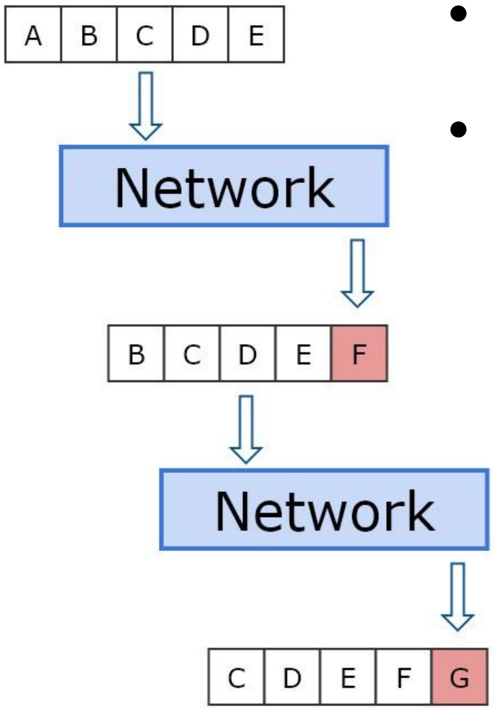
Generated output from training on Classical music

Robot image courtesy of trendhunter.com.

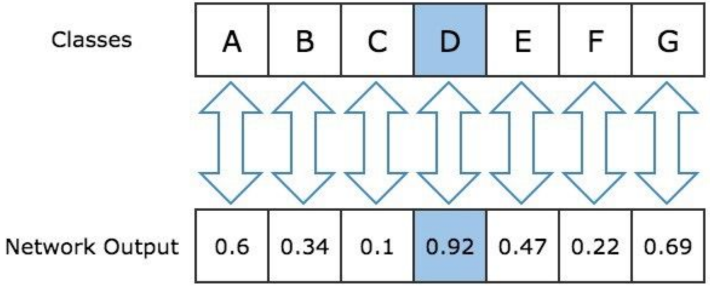
Lessons Learned

- Successfully implemented a functional AI music generator
- Tested the audio and generative capabilities of neural networks
- Utilized various audio format preprocessing

Sequence Generation



- Model generates each note/chord by looking at the previous 100 and taking the highest probability next note/chord
- This shifts the considered set by 1 each time



Diagrams courtesy of Sigurour Skuli, Towards Data Science.

Recurrent Network

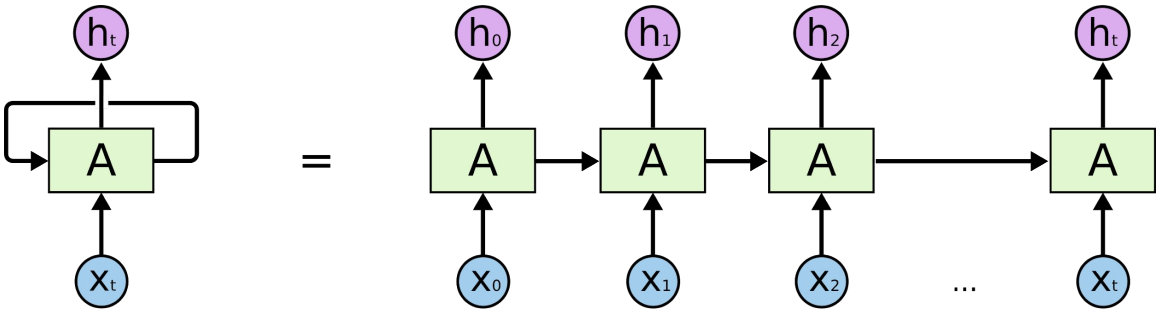


Diagram courtesy of "colah's blog" at https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Diagram (2)

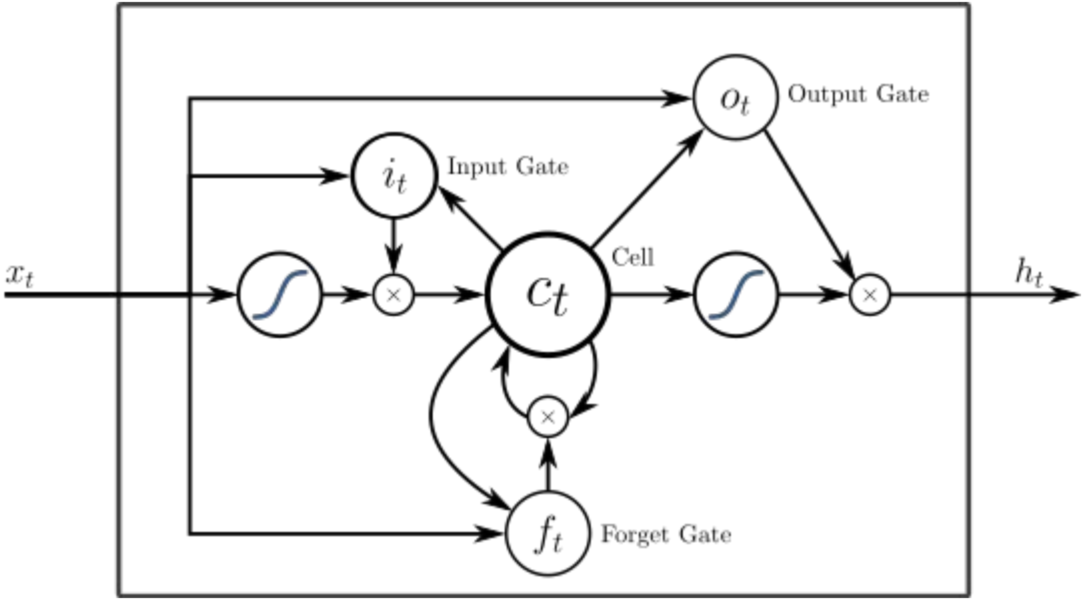


Diagram courtesy of "colah's blog" at https://colah.github.io/posts/2015-08-Understanding-LSTMs/

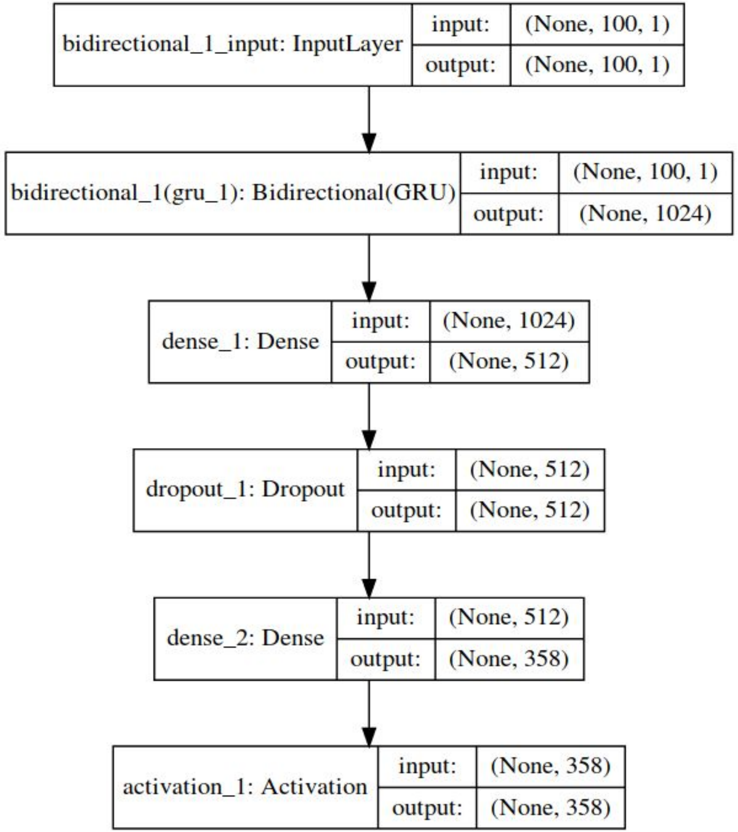
Model (2)

In [5]:
```
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
bidirectional_1 (Bidirection (None, 1024)              1579008   
_________________________________________________________________
dense_1 (Dense)              (None, 512)               524800    
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 358)               183654    
_________________________________________________________________
activation_1 (Activation)    (None, 358)               0         
=================================================================
Total params: 2,287,462
Trainable params: 2,287,462
Non-trainable params: 0
_________________________________________________________________
```

Model (3)

[note this is GRU]

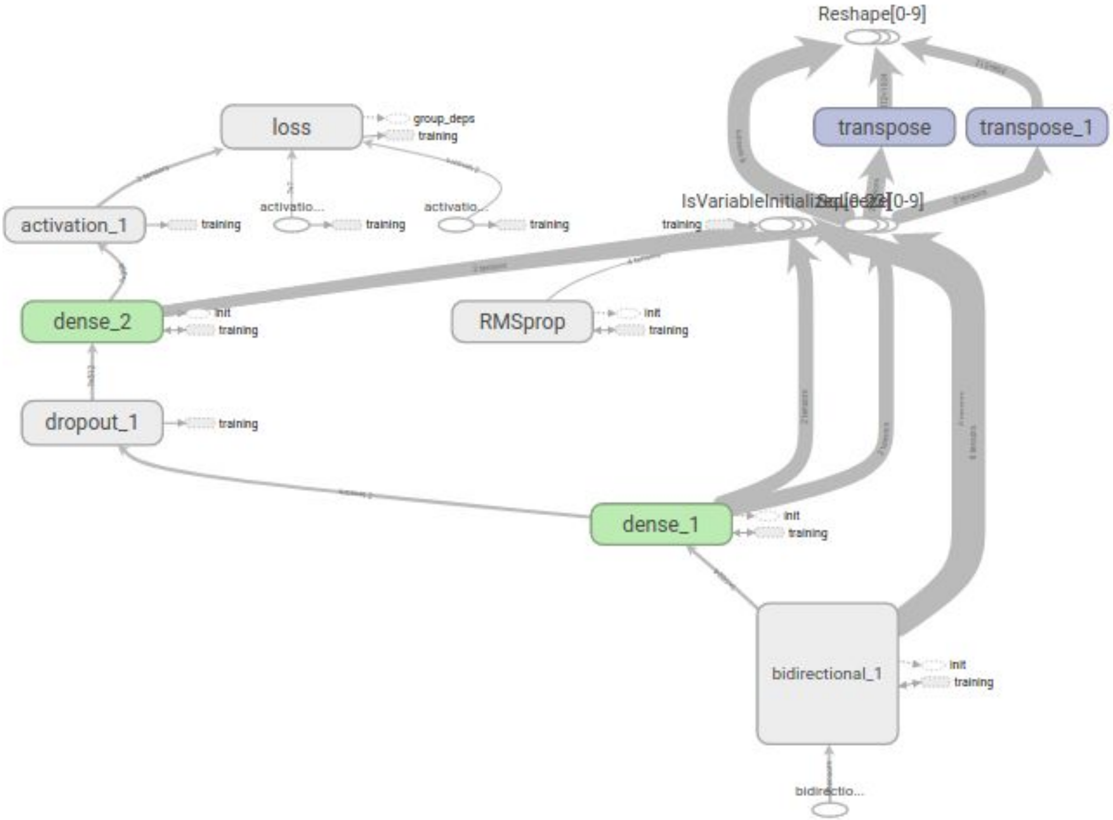


Model (4)

```
model = Sequential()
model.add(Bidirectional(LSTM(first_layer), input_shape=(timesteps, data_dim)))
model.add(Dense(first_layer))
model.add(Dropout(drop))
model.add(Dense(n_vocab)) # based on number of unique notes
model.add(Activation('softmax'))

rms = optimizers.RMSprop(lr=0.001, rho=0.9, epsilon=None, decay = 0.0)
model.compile(loss='categorical_crossentropy',optimizer=rms)
```

Model (5)



Reshape[0-9]
transpose
transpose_1
loss
group_deps
training
activation_1
training
activatio...
training
activatio...
training
IsVariableInitialized[0-22]
training
dense_2
init
training
RMSprop
init
training
dropout_1
training
dense_1
init
training
bidirectional_1
init
training
bidirectio...