AI Jukebox

Testing Creativity in AI

Brian McMahon

5 April 2018

Business Case

Exploring the question: can AI be creative?

- Expands scope of artistic applications in the arts, both audio and visual

Audio analysis and implementation essential for:

- Digital assistants
- Speech to text

Application of deep learning: testing the boundaries of AI ability to positively impact the human experience

Tools

Keras

TensorFlow

music21

python

Dataset

Midi files by genre

- Final Fantasy soundtrack:
    - 91 midi files, 51,177 notes and 358 unique notes
    - Successfully utilized in previous successful music generator models*
- Celtic folk tunes:
    - 338 midi files, 159,789 notes and 78 unique notes
    - Distinct, upbeat

**See [blog post](#) by Sigurour Skuli, Towards Data Science.*

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: input, cell/forget, output

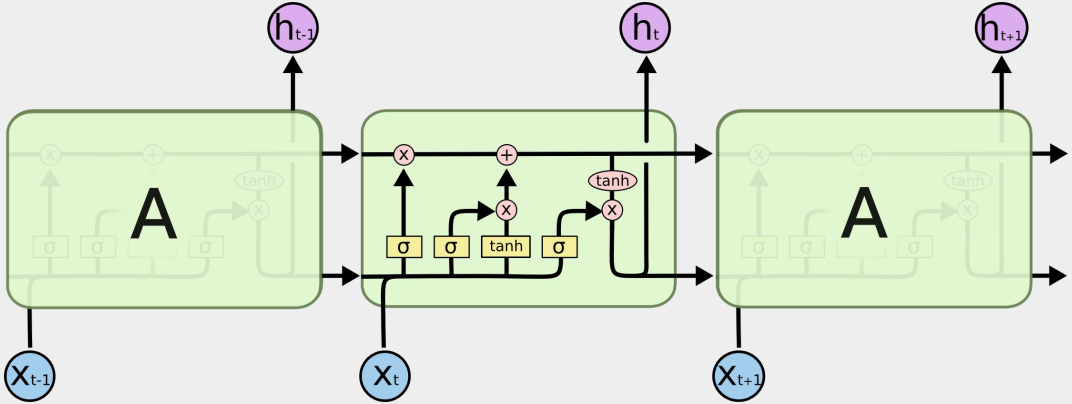
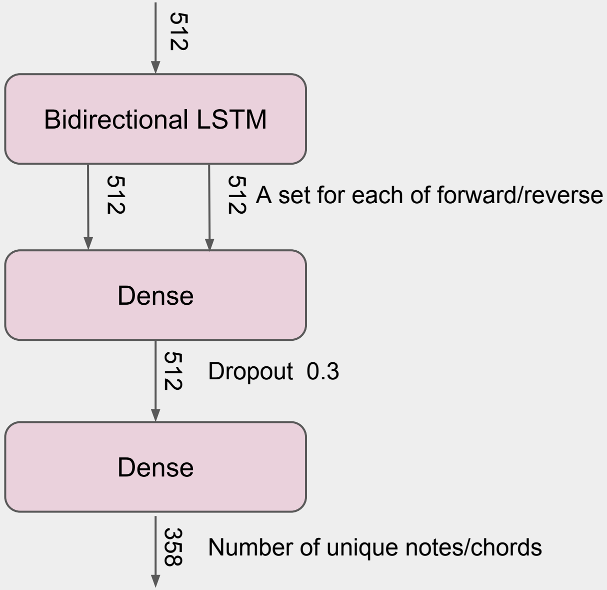


Diagram courtesy of [“colah’s blog”](#).

Architecture

Bidirectional LSTM



- 512 node input layer
- Dropout on dense layer only
- Learning rate 0.001
- Sequence length 200
- Notes generated 500

Generated Notes

Output from the artificial neural network

Robot image courtesy of trendhunter.com.

Results

As the model is generative, the best judges are us!

Looking for:

- Sensible recurring patterns, melodic
- Pleasing to the ear
- Model generates low loss on data, and validation loss on test set

To put into practice:

- Utilize deep learning and neural networks in interpreting audio data
- Generate new content which, perhaps, is removed from human environment
- Supplement artistic work with new rhythms, beats and patterns generated by AI

Next Steps

Continue to refine model performance. Explore:

- Different datasets
- different architectures, such as variational autoencoders
- Different inputs, such as raw audio

Write model into flask app and implement online

- Input a set of midi files, output AI music!

Thank You!

bcm822@gmail.com

Appendix

Lessons Learned

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

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.

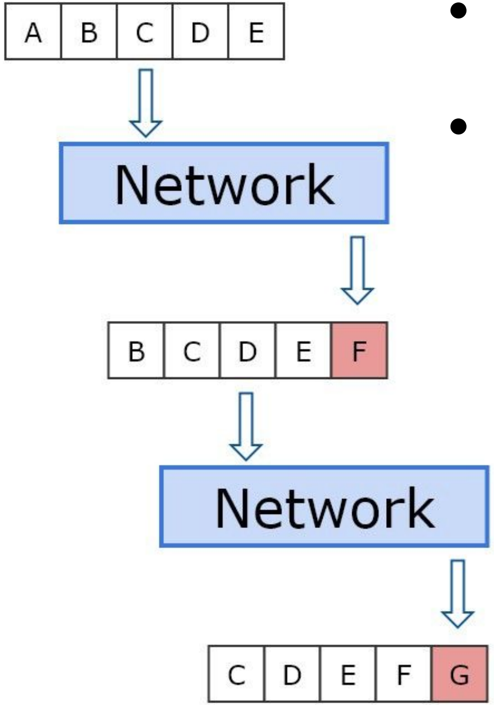
Skúli, Sigurður. "How to Generate Music using a LSTM Neural Network in Keras." www.towardsdatascience.com. December 7, 2017.

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

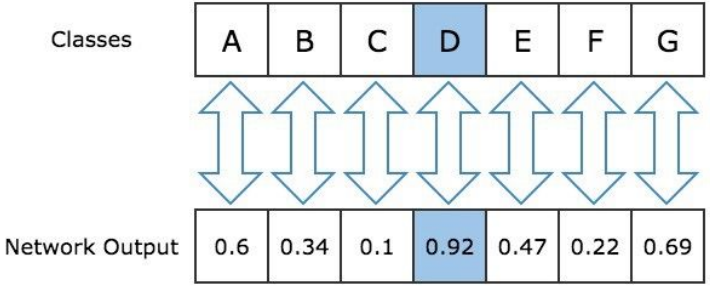
Brownlee, Jason. "Understand the Difference Between Return Sequences and Return States for LSTMs in Keras." https://machinelearningmastery.com. October 24, 2017.

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

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

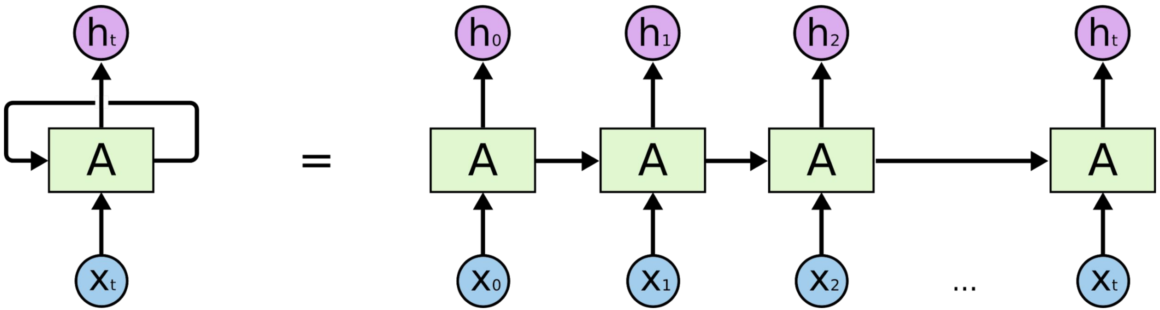


Diagram courtesy of "colah's blog" at https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Diagram (2)

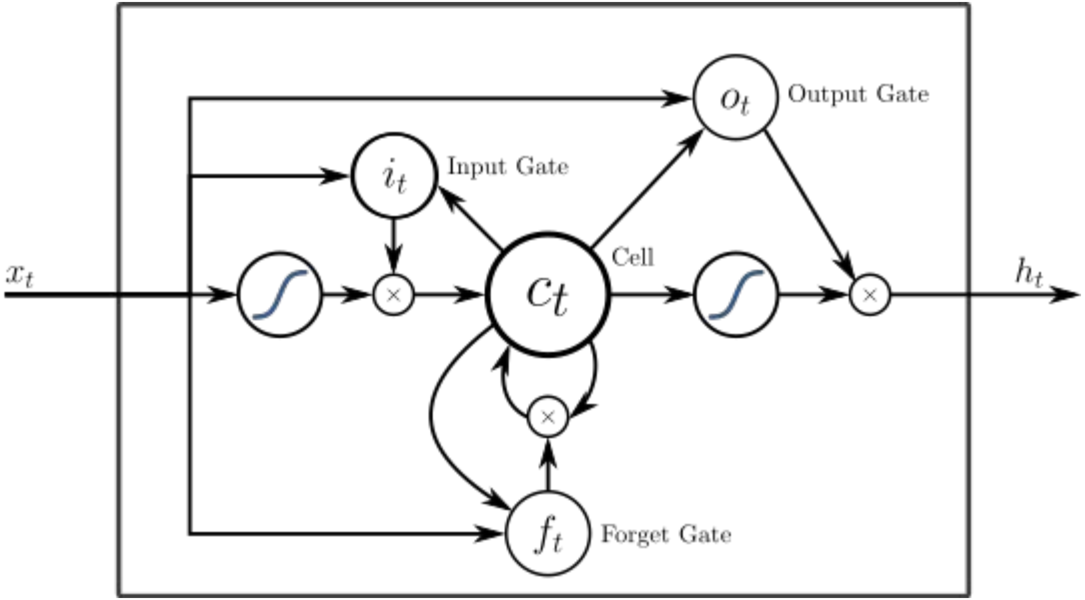


Diagram courtesy of “colah’s blog” at https://colah.github.io/posts/2015-08-Understanding-LSTMs/

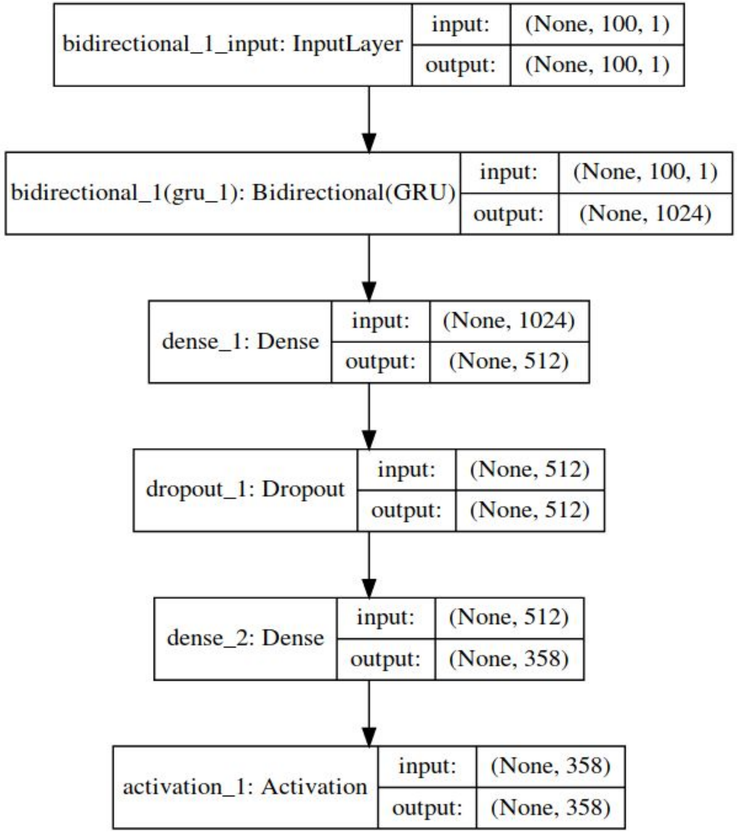
Model (2)

```
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]

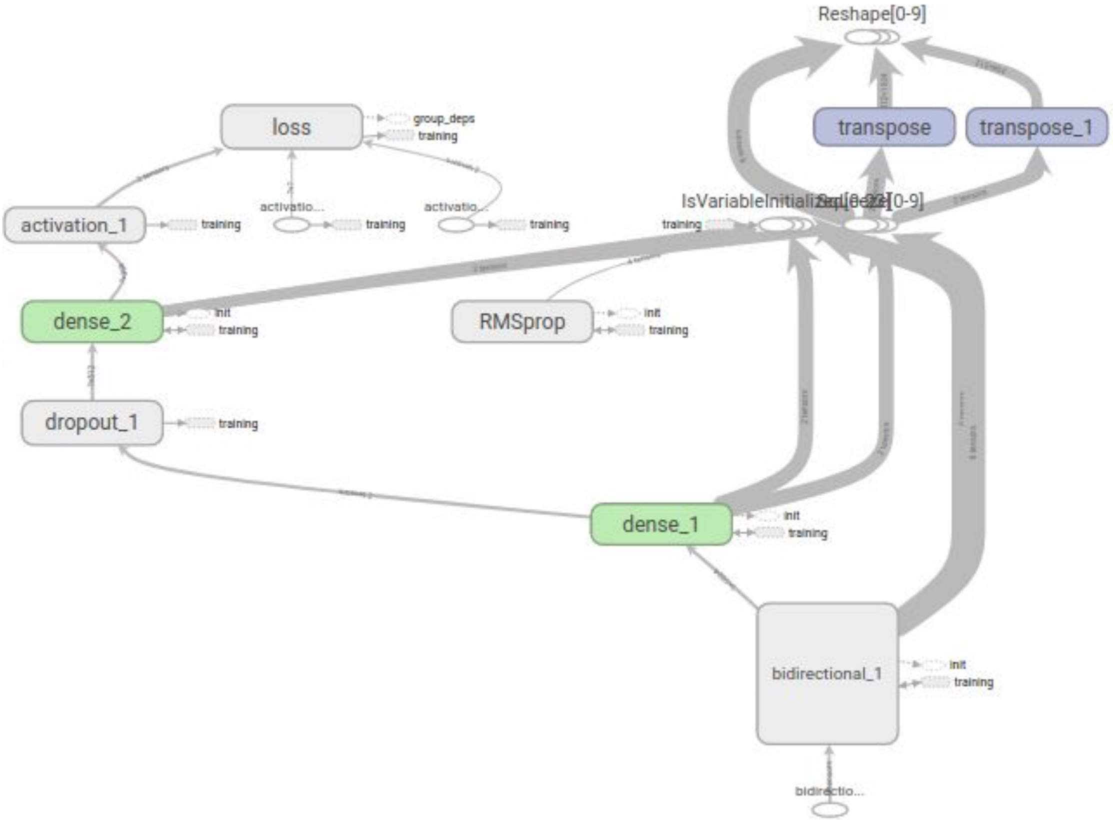


Model (4)

```
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]
Equal[0-9]
training
dense_2
init
training
RMSprop
init
training
dropout_1
training
dense_1
init
training
bidirectional_1
init
training
bidirectio...