



# DeepFire



Generating Rap Lyrics and Synthesizing Audio  
Using Deep Learning

Justin Dieter ([jdieter@stanford.edu](mailto:jdieter@stanford.edu))

Alp Ozturk ([aoturk@stanford.edu](mailto:aoturk@stanford.edu))

Jonathan Zwiebel ([jzwiebel@stanford.edu](mailto:jzwiebel@stanford.edu))

# Goals

Understanding that the production of hip hop music requires both a linguistic and musical sense of creativity, we aim to:

- Develop an application to synthesize the lyrics, rhythm, and audio of a rap track represents a challenging machine learning problem.
- Develop techniques that are transferable to more widely applicable areas of both musical and linguistic analysis and synthesis.
- Tackle problems that require extensive creativity, traditionally thought to be strictly human endeavors.

# Data

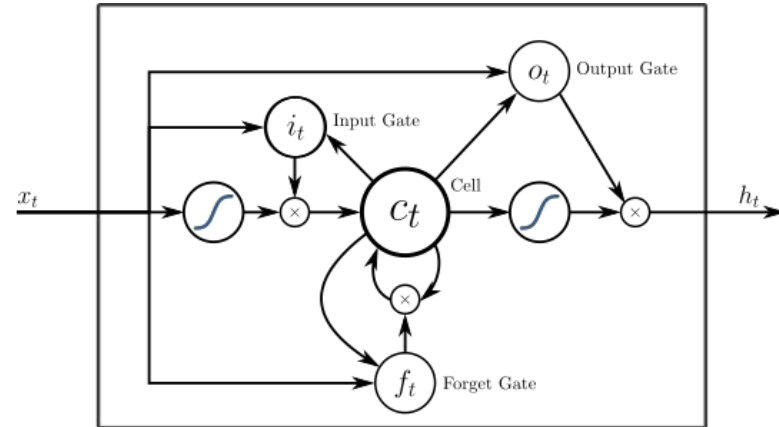
- Rap Lyrics obtained from a variety of sites that compile them.
  - Billboard Top 20 “Hot Rap Songs” - Week of December 9
  - Selection of songs compiled from “Greatest of All Time” Lists
- The vocal track of a single song “Fast Life” by Kool G Rap and Nas was used to learn the rhythmic structure of rap.

# Project Outline

1. Lyric Generation - Basic
2. Lyric Generation - Deep Learning
3. Extracting Beats from Instrumental
4. Synthesizing Rhythm to Overlay Instrumental
5. Incorporating Lyrics to Beat
6. Synthesizing voice and stretching audio to correspond to previously generated rhythms. (Currently using Amazon Polly for Voice Synthesis)

# LSTM (Long short-term memory) Architecture

- Both our rhythmic and lyrical generation used an LSTM architecture.
- LSTM is a type of recurrent neural network that features a forget gate to allow the network to be able to learn what is important from previous data.
- In the case of lyrics this allows the model to have context rather than generating based solely on the previous word. In the case of rhythm this allows the model to understand a rhythmic pattern.



[https://upload.wikimedia.org/wikipedia/commons/thumb/5/53/Peephole\\_Long\\_Short-Term\\_Memory.svg/542px-Peephole\\_Long\\_Short-Term\\_Memory.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/5/53/Peephole_Long_Short-Term_Memory.svg/542px-Peephole_Long_Short-Term_Memory.svg.png)

# Rhythmic Generation

- Considered a 'vocabulary' of possible rhythms that could occur within a beat.
- Rounded to 16th notes as this is generally sufficient for rhythms that occur in rap
- Model consists of a standard generative LSTM architecture
- Used Python library librosa to extract the timestamps of musical onsets from the song used as training data.
- Features consist of one-hot vector for the previous beat. Output of LSTM is a softmax for the next possible beat.

# Baseline Lyric Generation

- Compiled corpus of 10-20 rap songs as corpus
- Processed text by making lower case, removing punctuation.
- Constructed a markov chain text generator.
- Whole words used, not characters.
- Line ending count as words.
- Generated songs about 300 words, using previous 1, 2, 3 words to determine the next one.
- Using 3 words led to verbatim repetition of existing lyrics, but 1 & 2 lead to coherent, if odd, lyrics.

# Example Markov Chain Verses:

- “You can fight but my own lingo I like were on my bank account yeah. They copyin and I be dead her name. Shiftin gears shift on it before. That was legendary.”
- “I’m moonwalking but you’ll find me. Devil in the roof show up top up back it up man them backstreets.”
- “Surprised when you really wanna be alive. Take a sunken place to gun you made up. It up in the court jump.”



# Results

- No numerical way to measure quality of rap.
- However, our results are significantly lacking from what a human could produce despite coming close to mimicking the form.
- The rhythm is sporadic and not as consistent or well constructed as it is actual rap.
- The lyrics are rather nonsensical and do not follow a consistent story as an actual rap song would.
- The voice contains little inflection and is very monotone.
- However, for an attempt at an extremely challenging problem our model produces a better than expected quality of results.

# Future Work

- Better analysis of correlations between rhythm and lyrics rather than analyzing the two independently as we are currently.
- Incorporation of wider variety of rap
  - More songs
  - Songs by decade
  - Songs by artist
- Numerical evaluation
  - Objective measurements.
    - Rhyme density
  - Subjective measurements.
    - Blind Rating
    - Test people on generated lyrics vs real ones - Rapper Turing Test

# References

- Malmi, E., Takala, P., Toivonen, H., Raiko, T., & Gionis, A. (2016). DopeLearning. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16.  
doi:10.1145/2939672.2939679
- ArXiv, E. T. (2015, May 20). Machine-Learning Algorithm Mines Rap Lyrics, Then Writes Its Own. Retrieved November 21, 2017, from <https://www.technologyreview.com/s/537716/machine-learning-algorithm-mines-rap-lyrics-then-writes-its-own/>
- Cooper, M. (2017). Encore.ai [Computer software]. Retrieved from <https://github.com/dyelax/encore.ai>
- LSTMetallica: Generation drum tracks by learning the drum tracks of 60 Metallica songs. (2016, July 16). Retrieved November 21, 2017, from <https://keunwoochoi.wordpress.com/2016/02/23/lstmetallica/>
- LibROSA[Computer Software]. (2017). Retrieved from <https://librosa.github.io/librosa/index.html>
- Rap Music: Top Rap Songs Chart. (n.d.). Retrieved December 12, 2017, from <https://www.billboard.com/charts/rap-song/2017-12-09>
- The Best Rap Songs of All Time. (n.d.). Retrieved December 12, 2017, from <https://www.ranker.com/list/best-rap-songs-of-all-time/ranker-hip-hop>