

Deep Learning Techniques for Music Generation Control (8)

Jean-Pierre Briot

`Jean-Pierre.Briot@lip6.fr`

Laboratoire d'Informatique de Paris 6 (LIP6)
Sorbonne Université – CNRS



Programa de Pós-Graduação em Informática (PPGI)
UNIRIO



Control

Deep Learning for Music Generation – Technical Challenges

1. *Ex Nihilo* Generation

» vs Accompaniment (Need for Input)

2. Length Variability

» vs Fixed Length

3. Content Variability

» vs Determinism

4. Control

» ex: Tonality conformance, Maximum number of repeated notes...

5. Structure

6. Originality

» vs Conformance

7. Incrementality

» vs Single-step or Iterative Generation

8. Interactivity

» vs (Autistic) Automation

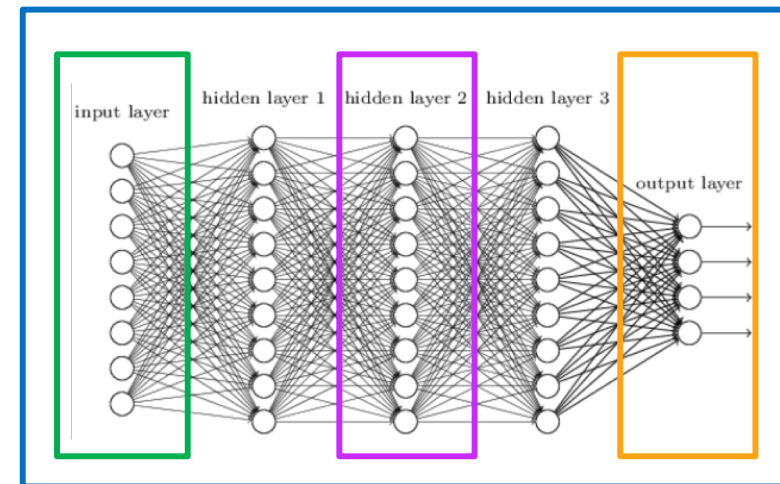
9. Explainability

#4 Limitation – Control

- No Control on Generation
 - Tonality Conformance
 - Ending with the same note at the first one
 - Avoiding too many repetitions of the same note
 - Not too many notes
 - etc.

Control

- Strategies:
 - Sampling
 - Conditionning (Parametrization)
 - Input Manipulation
 - Reinforcement
 - Unit Selection
 - Bottom up (Low-level adjustment)
 - » Ex: Sampling
 - Top down (Structure imposition)
 - » Ex: Unit and Selection
- Entry points (Hooks)
 - Input
 - Hidden
 - Output
 - Encapsulation/Reformulation



#4 Limitation – Control – #1 Solution: Sampling

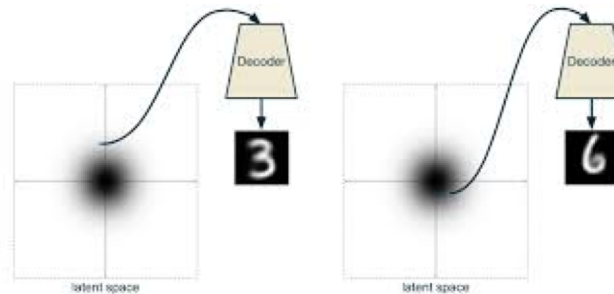
- Sampling Control

- Content Variability

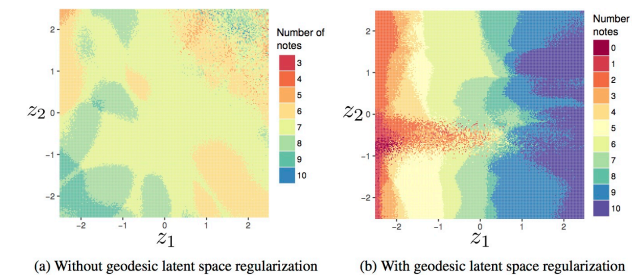
- » Threshold (to avoid notes too unlikely)

$$p_{threshold}(s_t | s_{<t}) := \begin{cases} 0 & \text{if } p(s_t | s_{<t}) / \max_{s_t} p(s_t | s_{<t}) < threshold, \\ p(s_t | s_{<t}) / z & \text{otherwise.} \end{cases}$$

- Variational Autoencoder

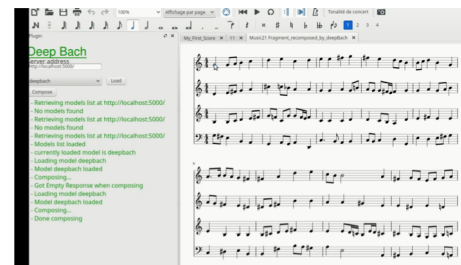


- Geodesic Latent Space Regularization [Hadjeres & Nielsen, 2017]



- Incremental Generation

- » DeepBach [Hadjeres et al., 2017]



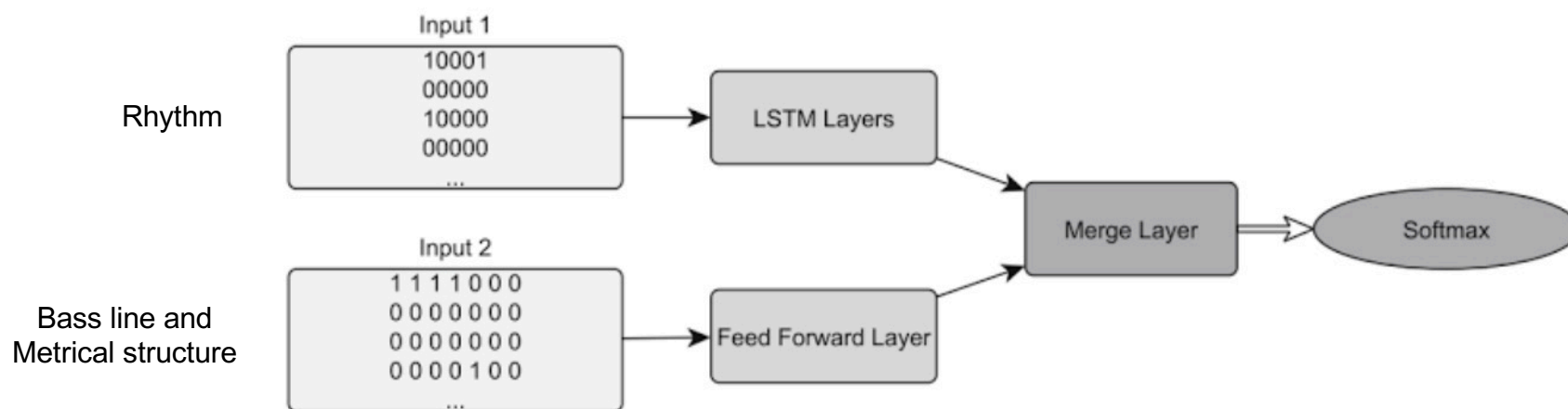
#4 Limitation – Control – #2 Solution: Conditioning

- Condition the architecture on some information
- To control/parameterize the architecture and the generation
- Ex:
 - Bass line
 - Beat structure
 - Chord progression
 - Previously generated note
 - Positional constraints on notes
 - Musical genre or style
 - Instrument

#4 Limitation – Control – #2 Solution: Conditioning

#1 Ex: Rhythm Generation [Makris et al., 2017]

- Generates Drum lines
- Recurrent architecture – Iterative feedforward generation
- Trained on a dataset of Drum and bass patterns



- Pop music dataset
- Example of generation
- Example of generation
- with a complex bass line conditioning

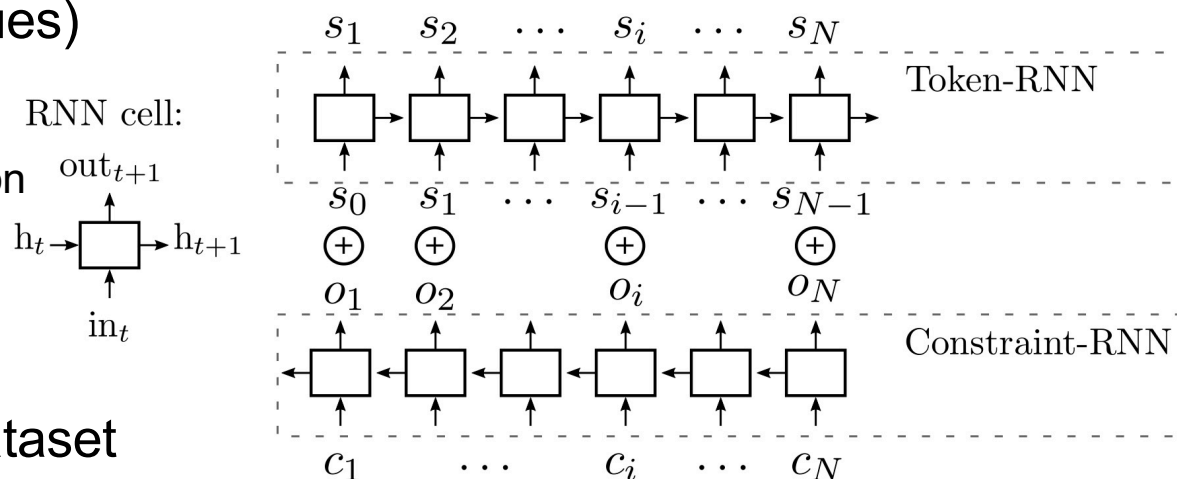


#4 Limitation – Control – #2 Solution: Conditioning

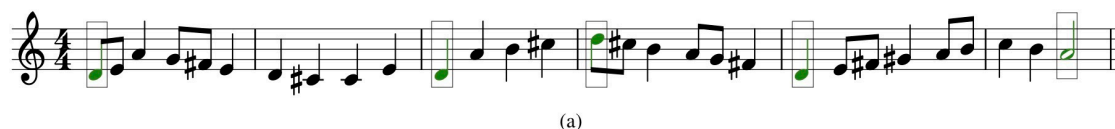
#2 Ex: Anticipation-RNN [Hadjeres et al., 2017]

- Generates Melodies
- Recurrent architecture – Iterative feedforward generation
- Positional Constraints (Note values)
- Backward Constraint-RNN

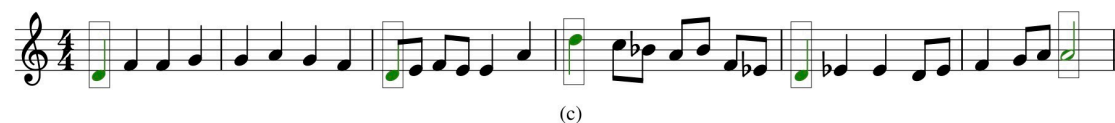
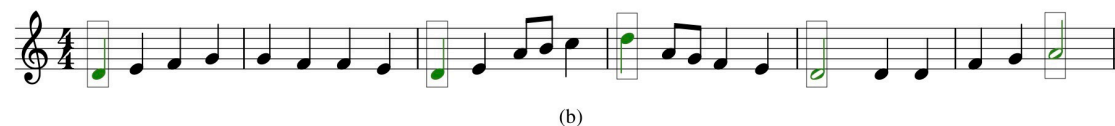
- Summarizes constraints_[i,N] information
- Used as Conditioning input_{i-1}



- Bach melodies (from chorale) dataset



- Examples of generation



#4 Limitation – Control – #2 Solution: Conditioning

#3 Ex: MidiNet [Yang et al., 2017]

- Conditioning information

- Previous measure
- Chord sequence

- Scope:

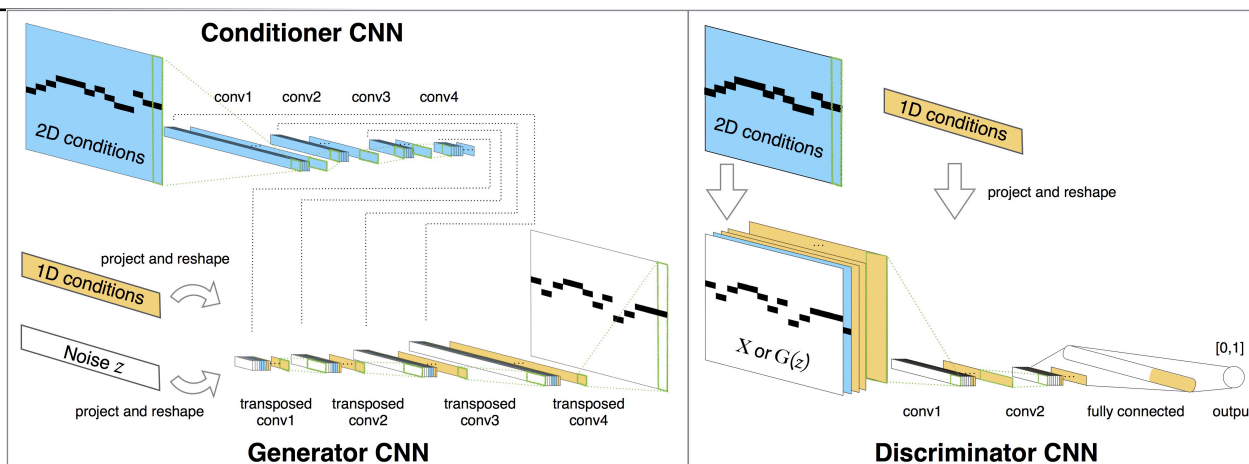
- Previous measure (1D conditions)
- Various previous measures (2D conditions)

- Fine control:

- Conditioning on previous measure 1D/2D and on chord sequence 1D/2D for one/all convolutional layers
- Ex: previous measure 1D and on chord sequence 2D for all convolutional layers
 - » Follows more chord sequence



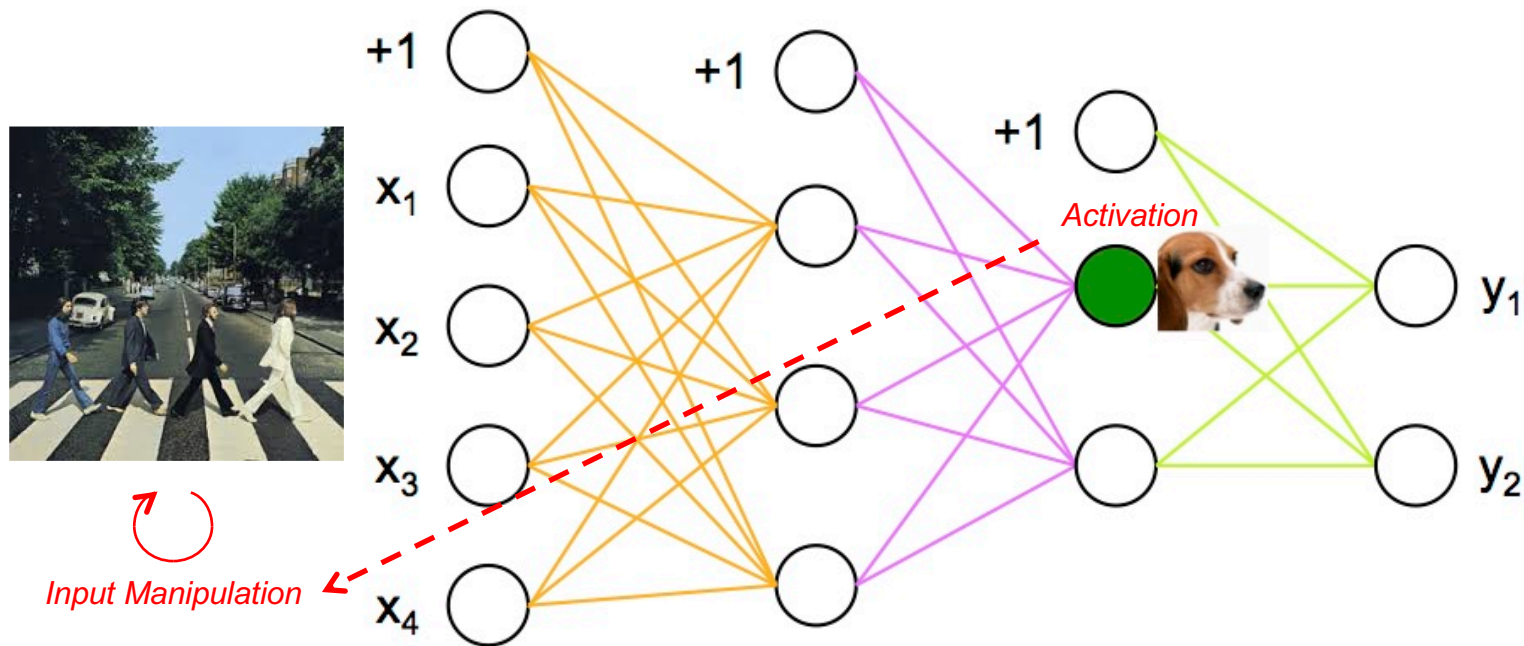
- Pop music dataset <https://soundcloud.com/vgtsv6jf5fwq/model3>



#4 Limitation – Control – #3 Solution: Input Manipulation

- The input of the network is incrementally manipulated, guided by **Gradient Descent Machinery**, in order to match a target property
- Input:
 - Initial Input content
 - or brand New/Empty (randomly initialized)
- Target properties (Ex.):
 - Maximizing the activation of a specific unit, to exaggerate some visual element correlated (detection) to this unit
 - » Deep Dream [Mordvintsev et al., 2015]
 - Maximizing some similarity to a given target, to create a consonant melody
 - » DeepHear [Sun, 201X]
 - Maximizing both similarity to style and similarity to content
 - » Style transfer [Gatys et al., 2015]
 - Imposing higher level structure (form, tonality, meter)
 - » C-RBM [Lattner et al., 2016]

Deep Dream [Mordvintsev et al. 2015]



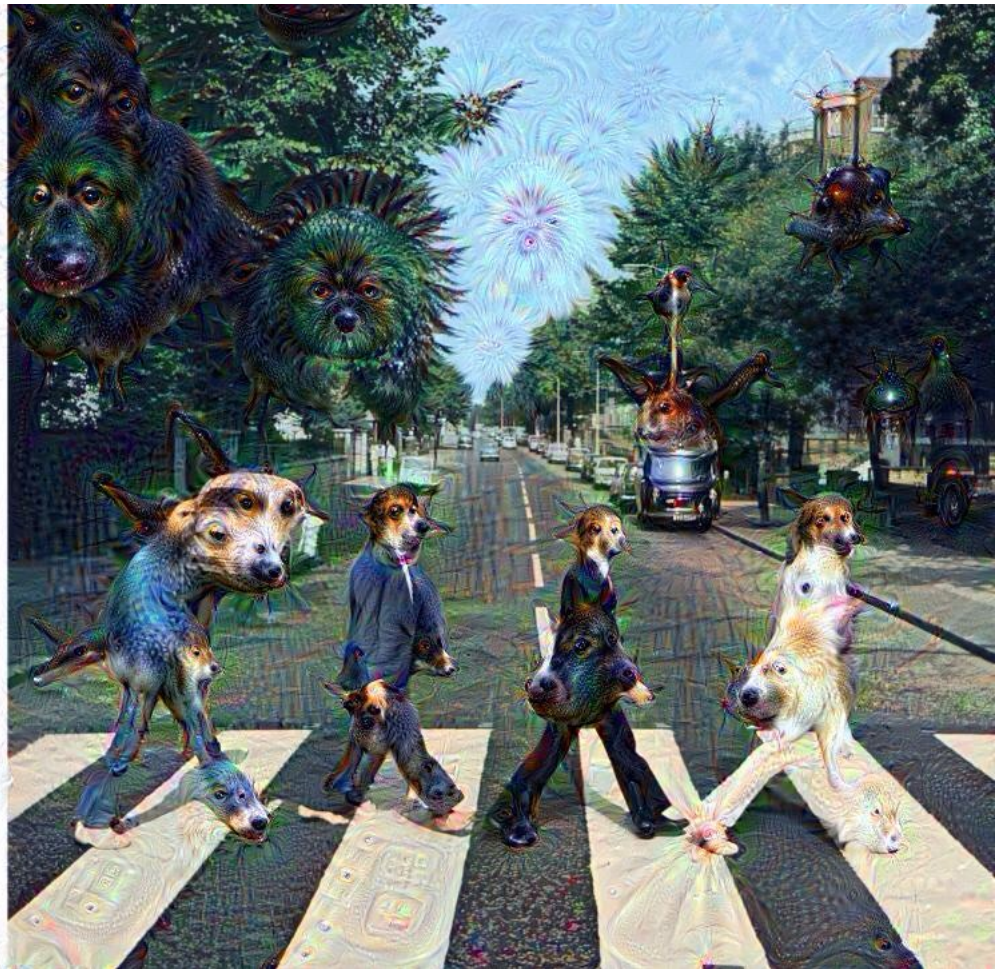
- Network is (or has been) trained on a large images dataset
- The objective is to ~~minimize the cost~~ to maximize the activation of a specific unit (neuron) which activates for a specific pattern(s), ex. a dog face 
- An initial image is iteratively slightly altered (ex: jitter, under gradient ascent control) to maximize that specific activation
- This will favor emergence of occurrences of that specific pattern in the image

Deep Dream

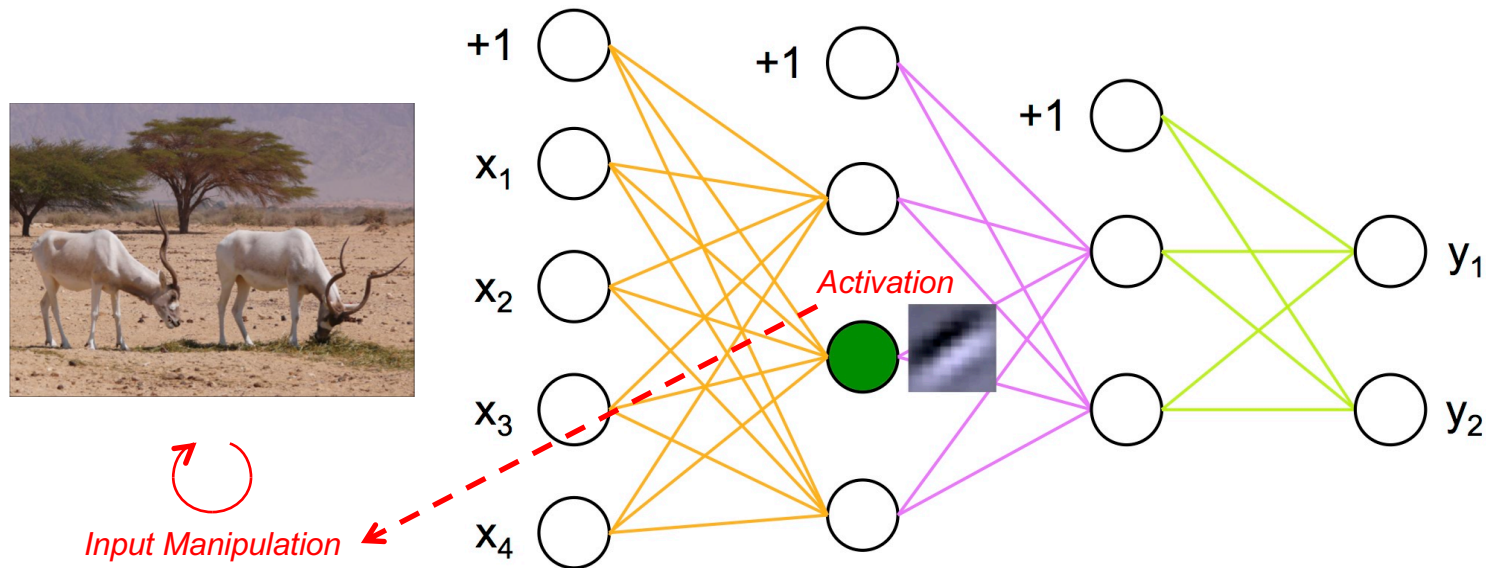
Initial Image



Deep Dream Image

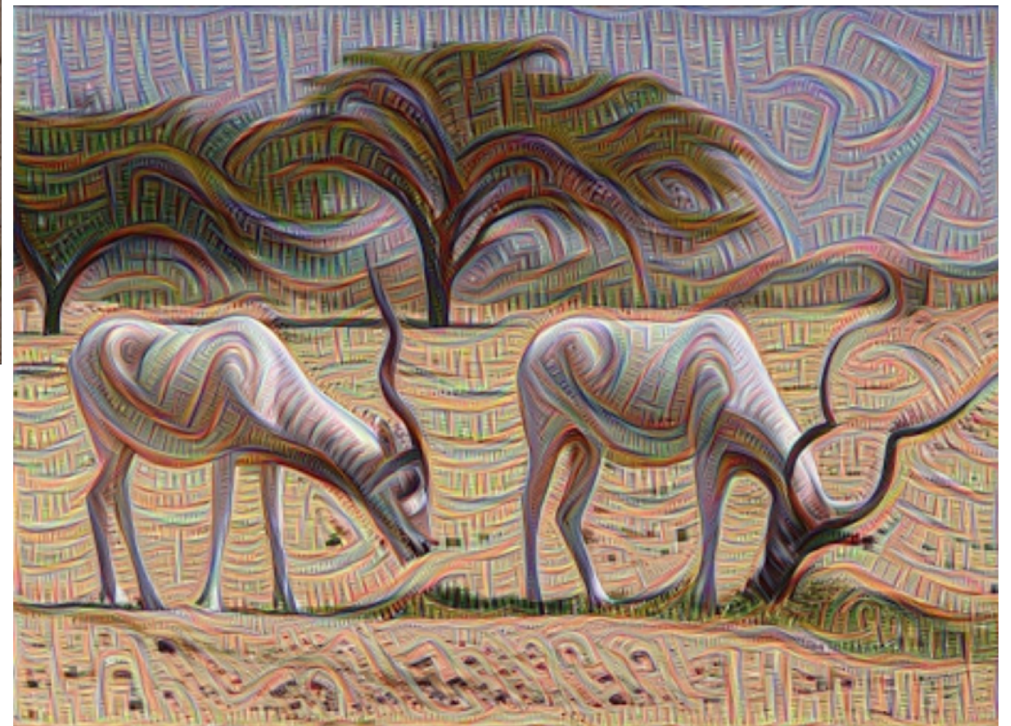


Deep Dream



- Case of a lower-level unit (neuron) pattern 
- Results in texture insertion

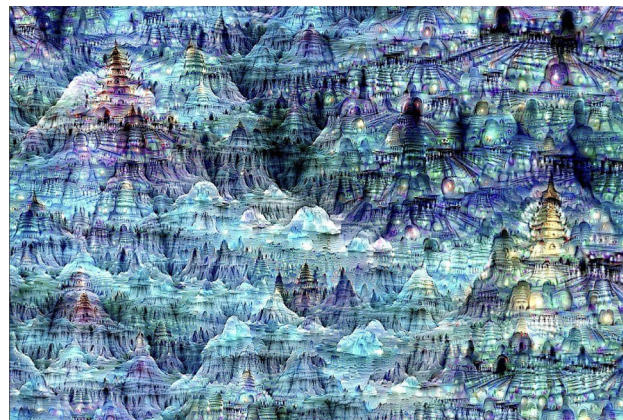
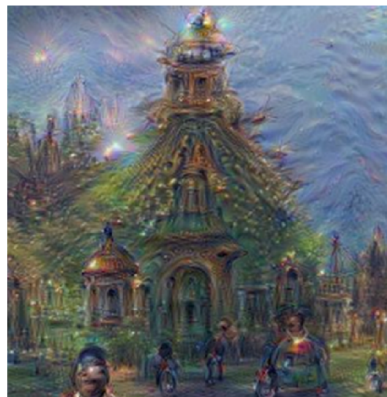
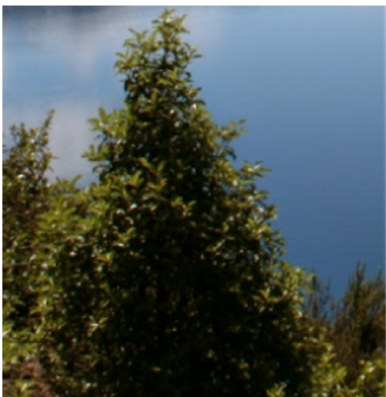
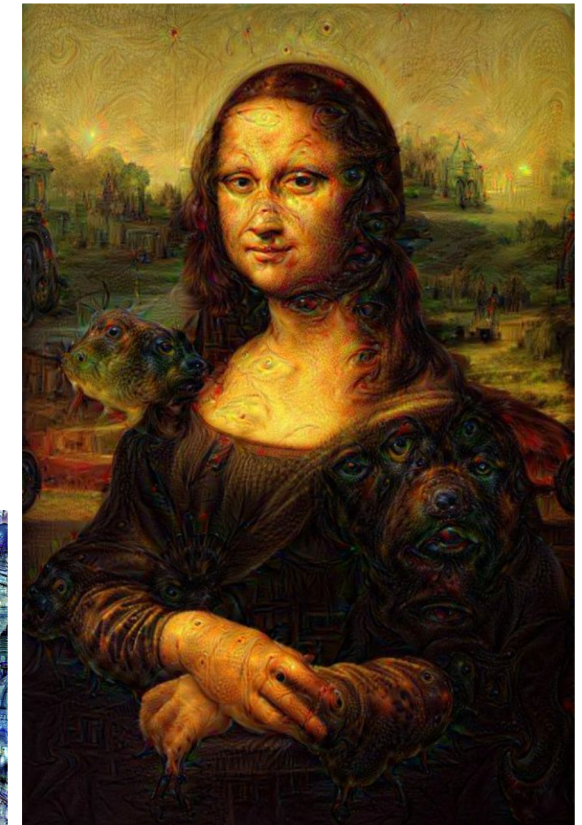
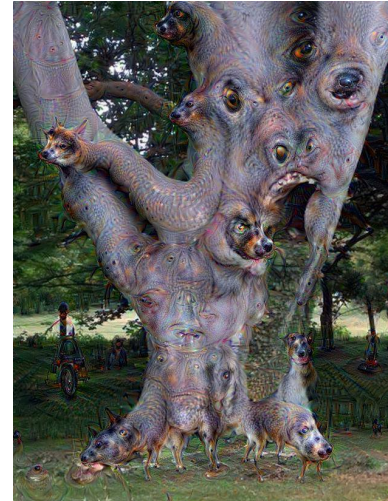
Deep Dream



Lower Layer Unit Maximization Resulting image
Deep Learning – Music Generation – 2019

Deep Dream

- One may use other maximization strategies (joint, similarity measures, etc.), alterations, etc., resulting in various transformation effects



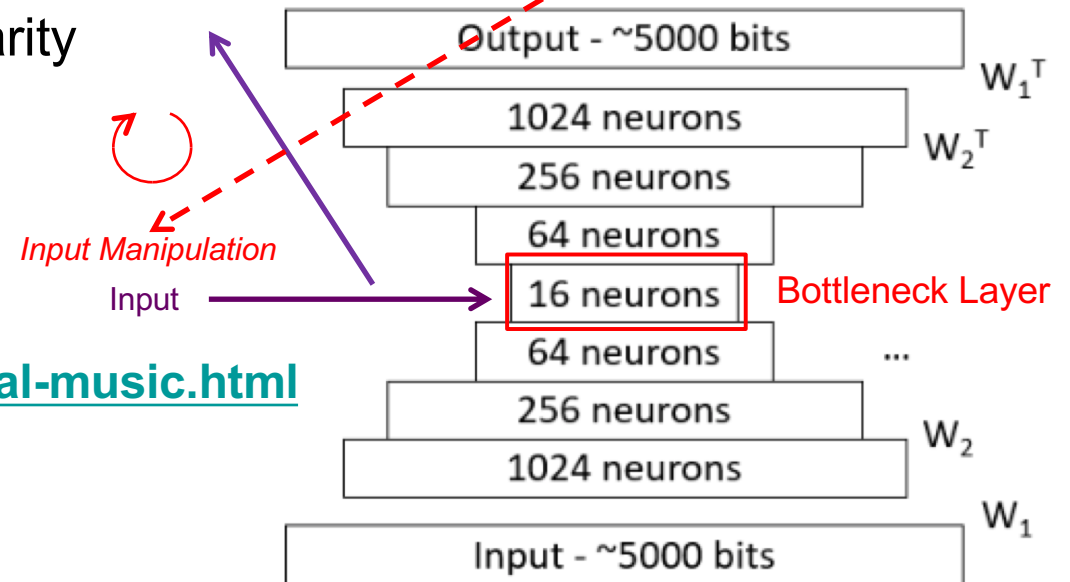
DeepHear [Sun, 2016]

Generation of Consonant Melody as a naive Accompaniment/Counterpoint to an Existing Reference Melody

- Input Random Data into 16 Neurons Bottleneck Layer
- Generated Melody: Output of the Higher Layer Decoder
- Alter the Input Data in order to Maximize Similarity between Generated Melody and Reference Melody

$err(S) = \sum_{\text{pitches } i, \text{ timesteps } t} ((i, t) \text{ in melody}) \cdot (1 - S[i, t])^2$
Similarity
 Penalize Missing Note when Melody has Note

- Controlled by Gradients of Similarity



<https://fephsun.github.io/2015/09/01/neural-music.html>

Style Transfer

Style Transfer [Gatys et al., 2015]

1. Train Network with Images

2. Content Capture of 1st (Content) Image

- Feed Forward 1st Image into Deep Network
- Capture Content information
 - » Activations of filters/neurons for each layer



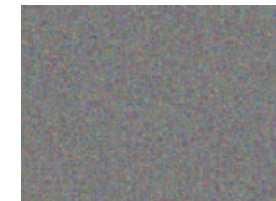
3. Style Capture of 2nd (Style) Image

- Feed Forward 2nd Image into Deep Network
- Capture Style information
 - » Feature space(s) : Correlations between feature filters/neurons for each layer

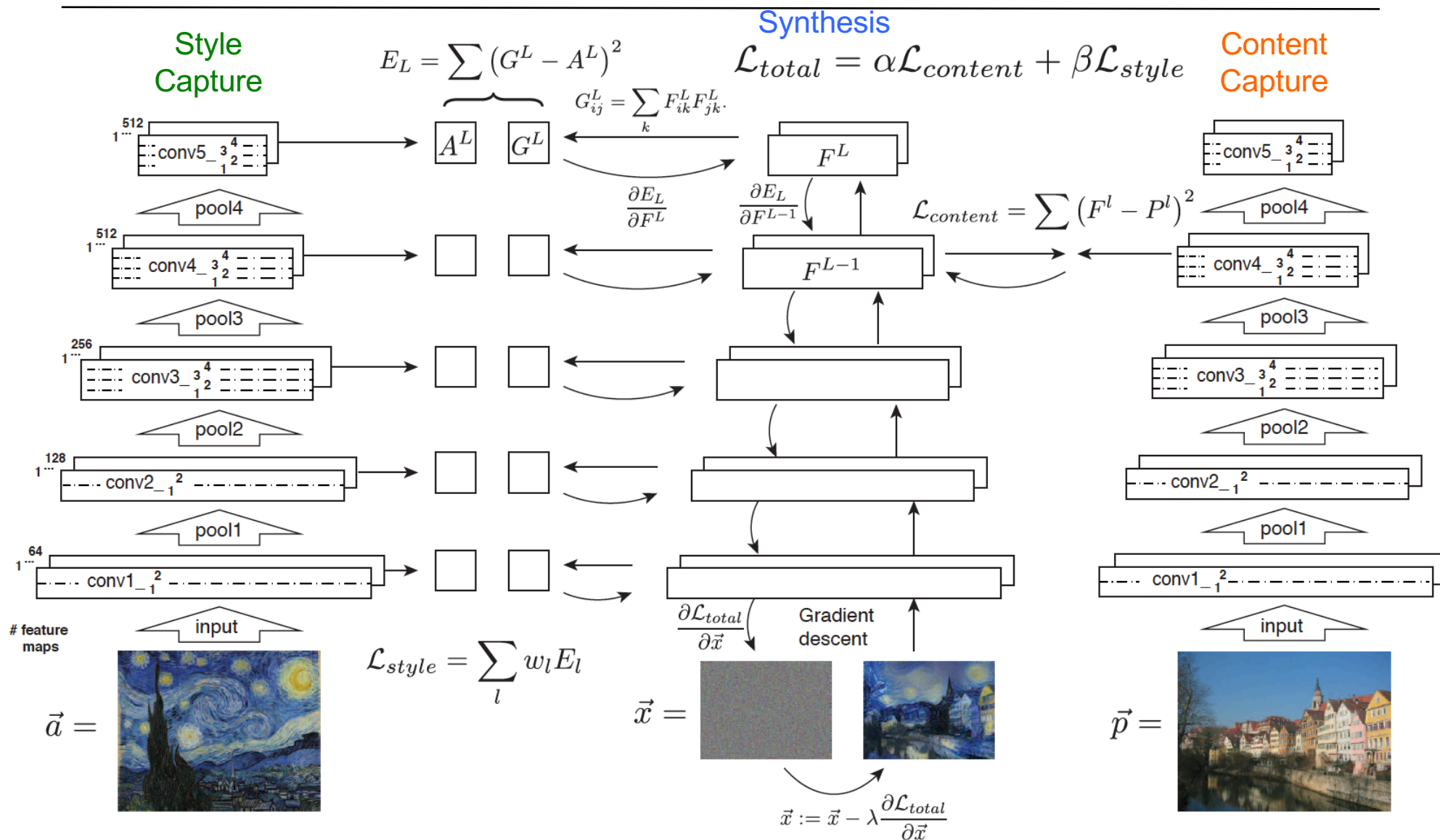


4. Search/Synthesize Hybrid Image

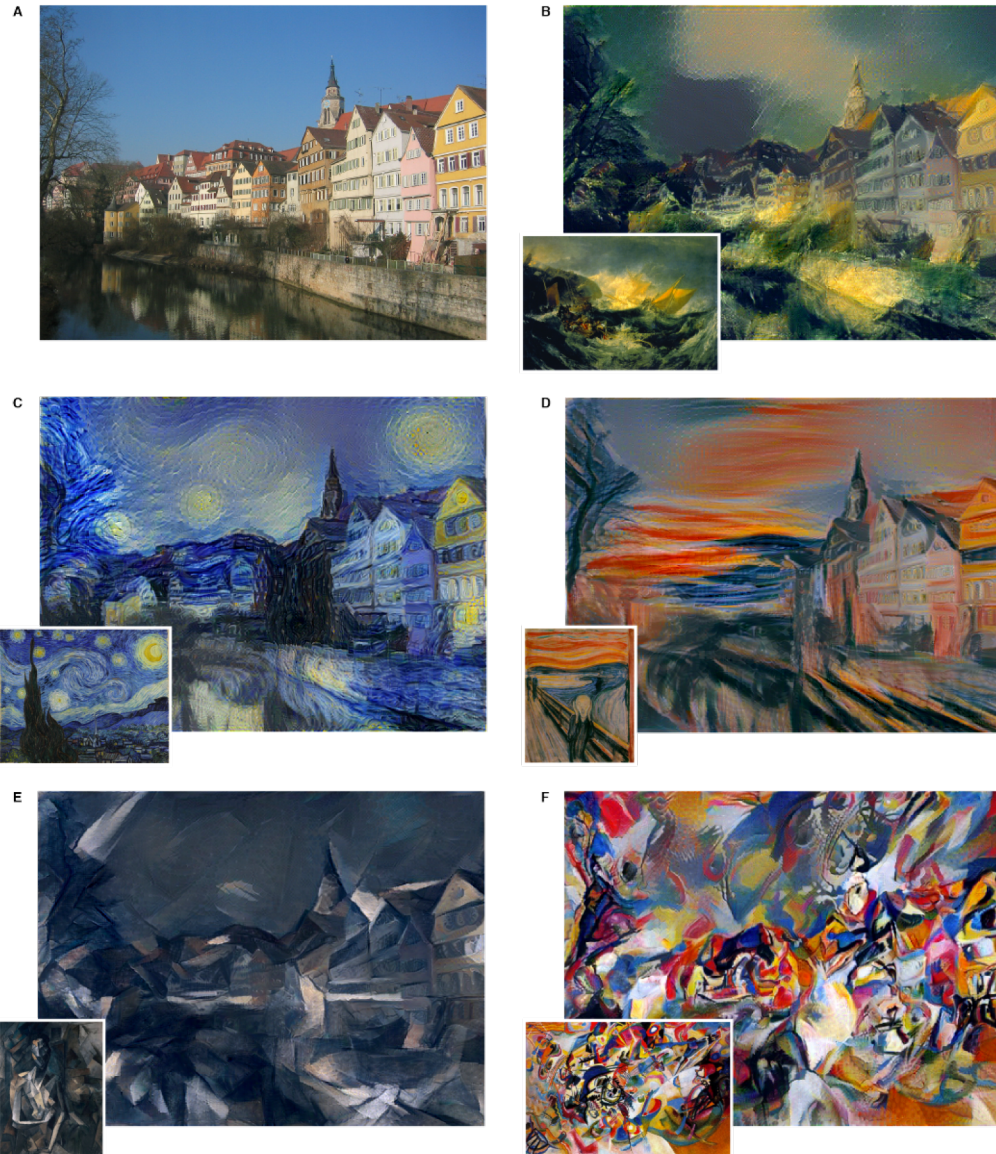
- Feed Forward Random Image into Deep Network
- Compute its Style and Contents features
- Compute the Style and Contents Loss/Differences (with Content Image and Style Image)
- Compute the Gradients (Standard Back-Propagation)
- Update the Image with λ *Gradients
 - » λ : update rate
- Iterate until reaching Target Losses



Style Transfer

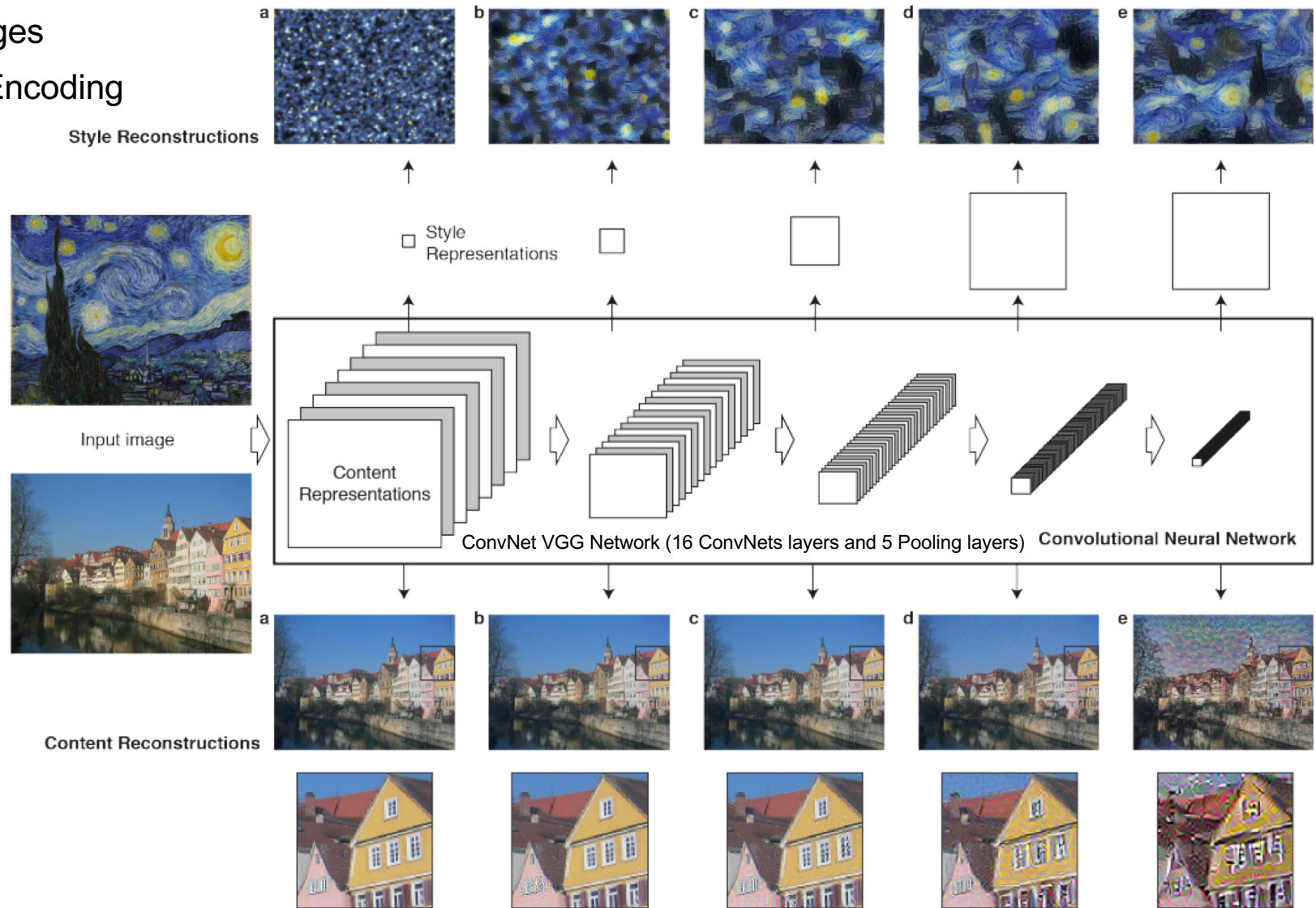


Style Transfer (Guess the Artist/Painting :)



Style Transfer

Reconstructing Images from the Network Encoding



Style Transfer

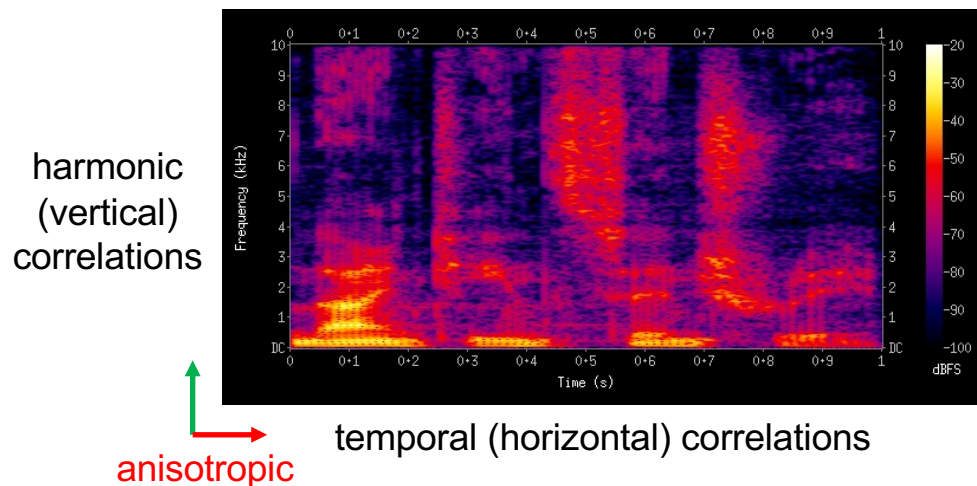


Style Transfer

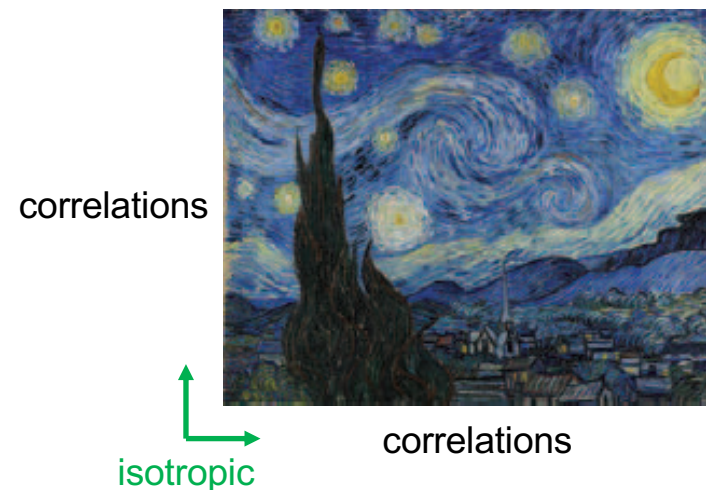
- Has been applied to music
 - Ex: [Ulyanov & Lebedev, 2016]
 - Audio (Spectrogram)

<https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/>

- Generally more difficult
 - Music = Temporal Data (Images)
 - Temporal (horizontal) correlations $\neq$ Harmonic (vertical) correlations (anisotropy)
 - as opposed to horizontal $\approx$ vertical correlations for images (isotropy)



Jean-Pierre Briot



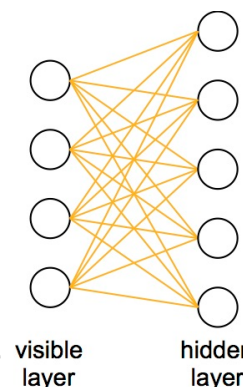
Deep Learning – Music Generation – 2019

Audio texture synthesis and style transfer [Ulyanov & Lebedev, 2016]



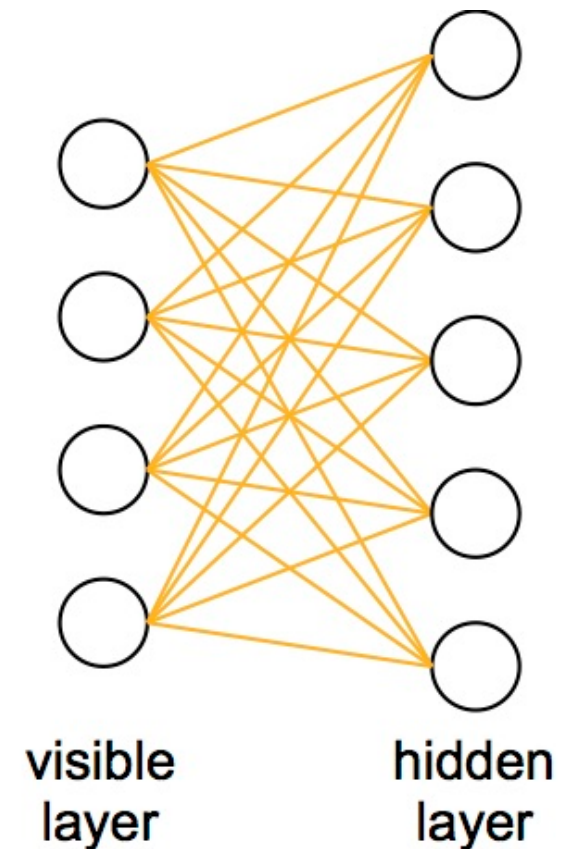
#4 Limitation – Control – #3 Solution: Input Manipulation & #1 Solution: Sampling

- Constrained sampling, C-RBM [Lattner et al., 2016]
- Convolutional Restricted Boltzmann Machine (RBM)
- Combination of:
 - **Input Manipulation** guided by **Gradient Descent** of current sample
 - » to impose Higher-Level Structure/Constraints:
 - Structure
 - Tonality
 - Meter
 - **Sampling Control**, by **Selective Gibbs sampling (SGS)**
 - » at a Selected Low-Level (subset of variables)
 - » to realign selectively the sample to the learnt distribution



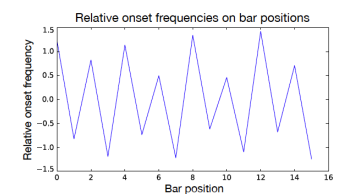
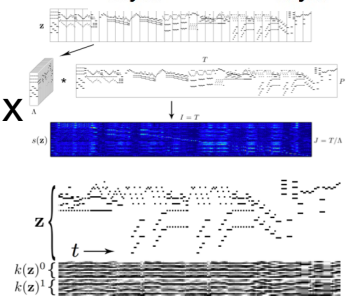
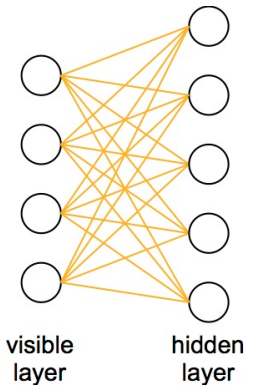
Restricted Boltzmann Machine (RBM) [Hinton & Sejnowski, 1986]

- Generative Stochastic Neural network
- Can learn a Probability Distribution over its set of inputs
- Was used for Pre-Training [Hinton & Salakhutdinov, 2006]
- Analog to Autoencoder
 - No Output
 - » Input Layer also acts as an Output
 - Stochastic
 - Boolean, or Multinomial/Discrete or Continuous (Values)
- Can learn efficiently with few examples
- Generation of a sample from the model learnt:
 - Random initialization of the visible layer vector v
 - » Following a standard uniform distribution
 - Run Gibbs sampling until convergence



Input Manipulation & Constrained sampling

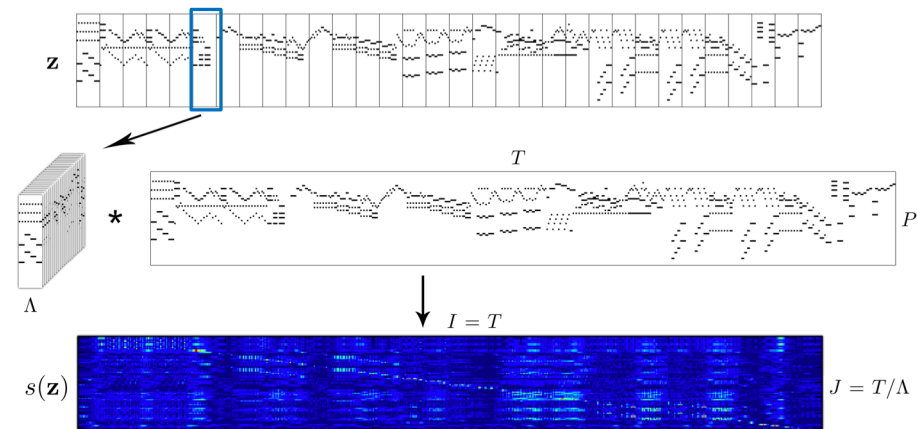
- Constrained sampling, C-RBM [Lattner et al., 2016]
- Convolutional Restricted Boltzmann Machine (RBM)
- Combination of:
 - **Input Manipulation** guided by **Gradient Descent** of current sample
 - » to impose Higher-Level Structure/Constraints:
 - Structure (Structure Repetition, Ex: AABA), via Self-Similarity Matrix
 - Tonality, via Similarity of Distribution of Pitch-Classes
 - Meter (Rhythm Pattern/Signature and Beat Accent)
 - **Sampling Control**, by **Selective Gibbs sampling (SGS)**
 - » at a Selected Low-Level (subset of variables)
 - » to realign selectively the sample to the learnt distribution
 - Alternate **IP/GD** and **SGS**, controlled by **Simulated Annealing**
 - But not exact as, e.g., **Markov Constraints** [Pachet & Roy, 2011]



Input Manipulation & Constrained sampling

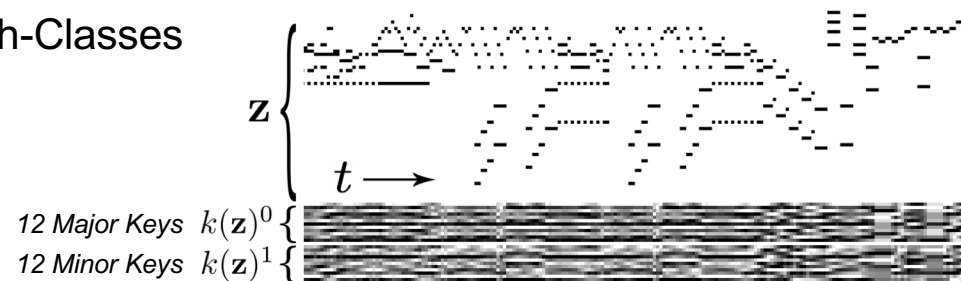
– Structure (Repetition Structure, Ex: AABA)

- » Self-Similarity Matrix
- » For each Music Slice



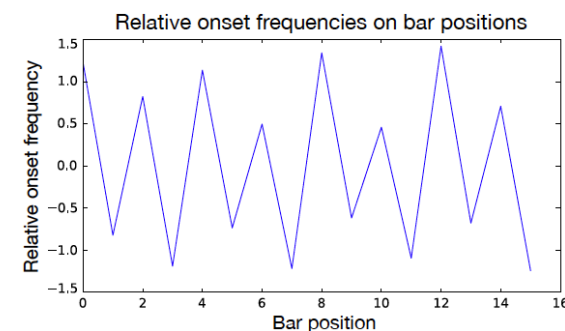
– Tonality, via Similarity of Distribution of Pitch-Classes

- » Key Estimation Vectors over Time

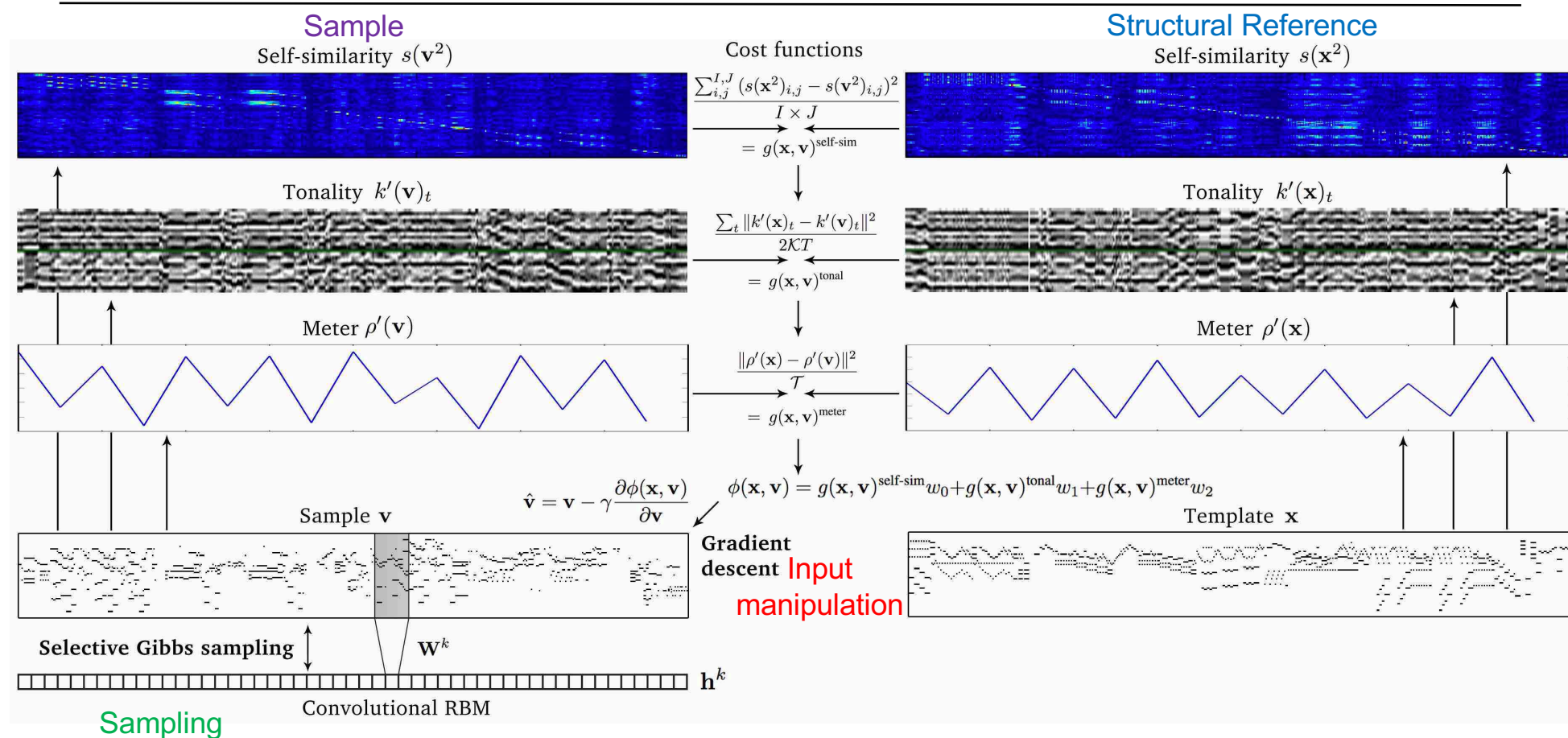


– Meter

- » Duration and Accent Patterns (ex: on 1st and 3rd Beats)
- » Via Relative Occurrence of Note Onsets



C-RBM [Lattner et al., 2016]



Both *Manipulation* and *Sampling* of Input
because RBM's "Output" is its Input

<https://soundcloud.com/pmgrbm>

C-RBM Examples

- RNN-RBM Sample



- Unconstrained Sample



- Template Piece

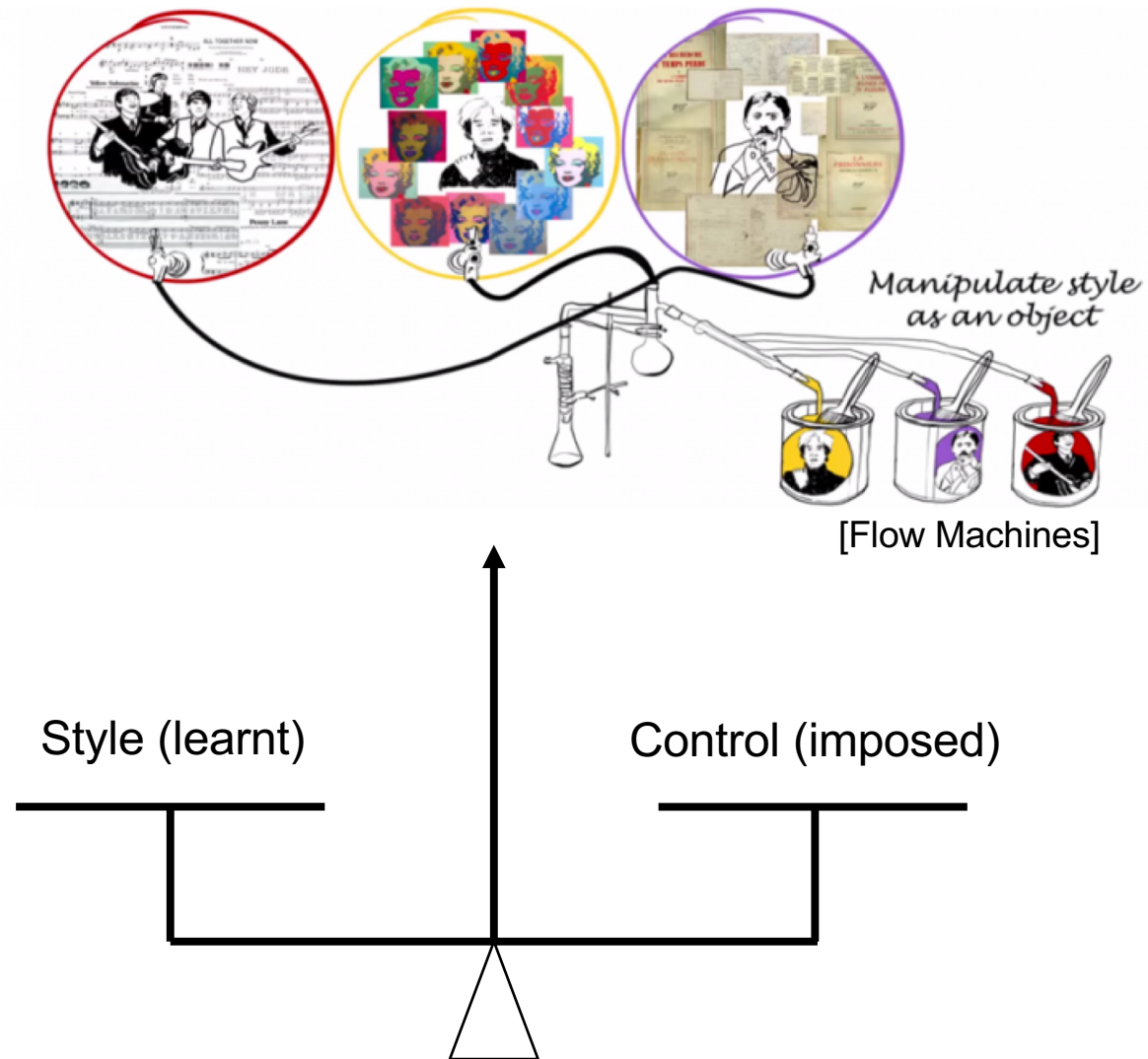


- Constrained Sample



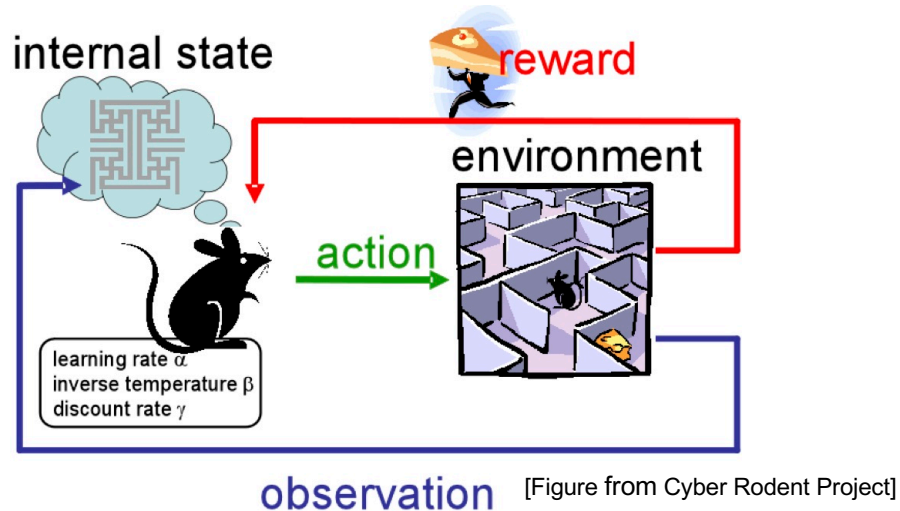
<https://soundcloud.com/pmgrbm>

Style vs/and Control



#4 Limitation – Control – #4 Solution: Reinforcement Learning

- Reformulation of Melody Generation as a Reinforcement Learning Problem

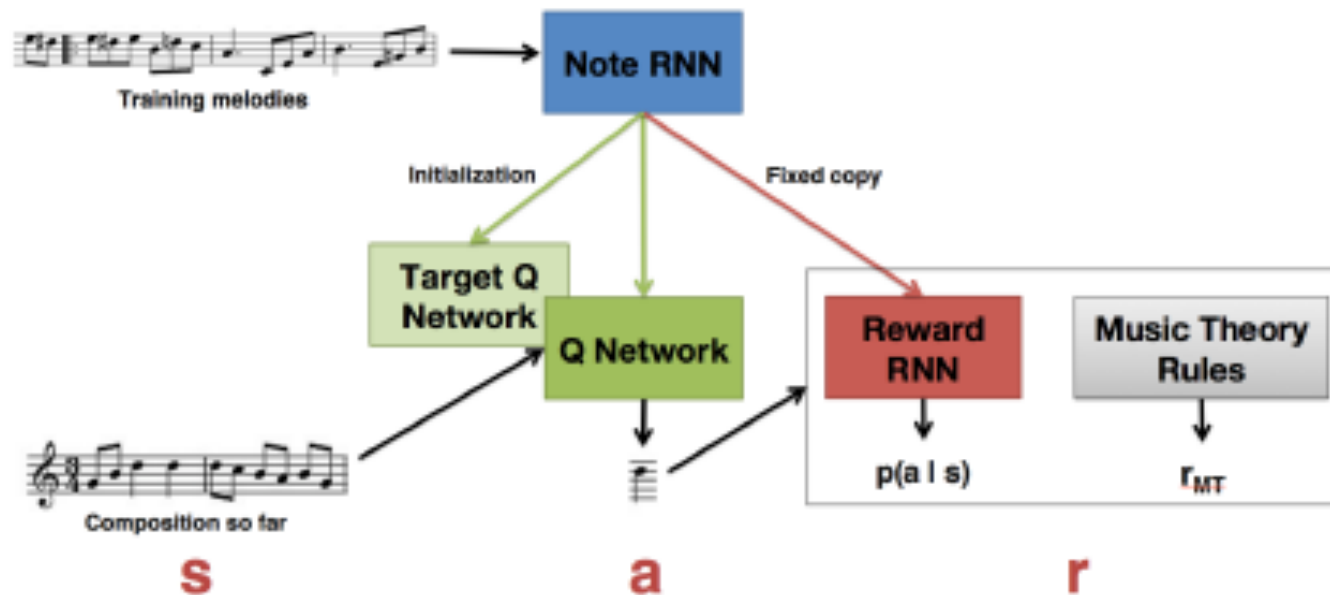


- Sequential decision/action problem
 - Objective: Learn via past decisions/actions a near optimal policy (state $\rightarrow$ action) for maximizing the gain (sum of rewards)
- State: Melody generated so far (Succession of notes)

Composition so far
 - Action: Generation of next note 

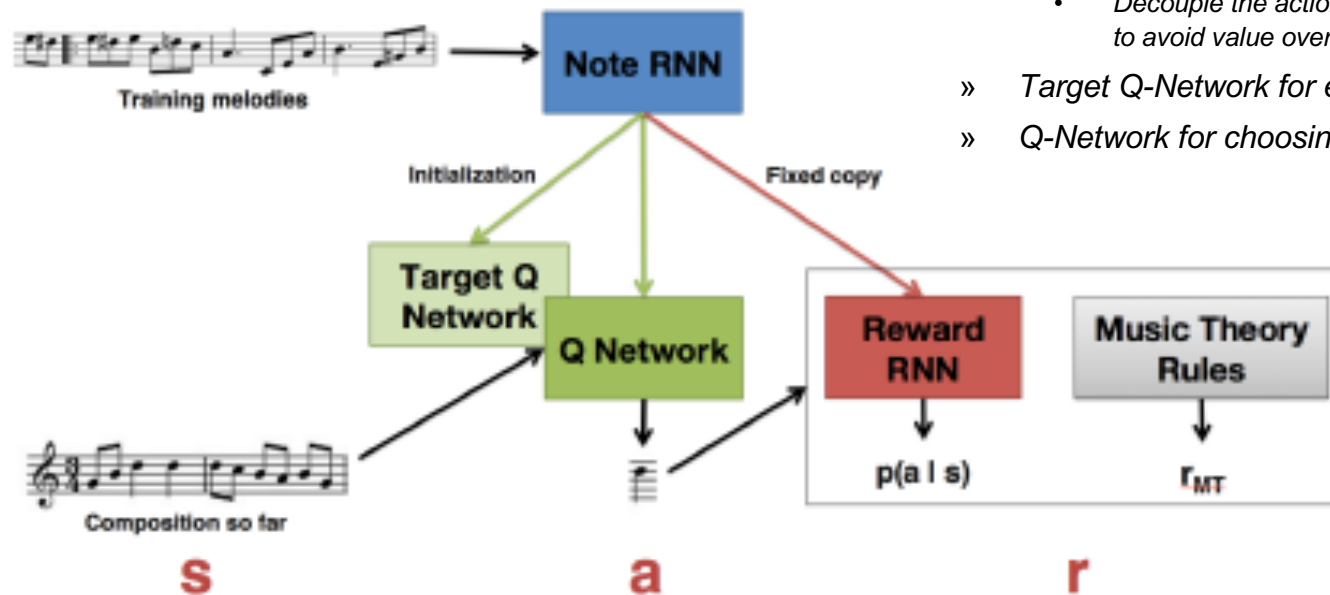
Deep Reinforcement Learning (RL) – RL-Tuner [Jaques et al., 2016]

- *RNN (Recurrent Neural Network) Trained on the Corpus*
- **Reward** = Combination of:
 - Adherence to **What has been Learnt** by the RNN
 - » Similarity to the Prediction by the RNN
 - Adherence to **Music Theory Rules**
 - » Tonality, Avoid excessive repetitions...
 - » Open: Arbitrary **User-defined Targets**



Deep Reinforcement Learning (RL) – RL-Tuner [Jaques et al., 2016]

- *RNN (Recurrent Neural Network) Trained on the Corpus*
- **Reward** = Combination of:
 - Adherence to **What has been Learnt** by the RNN
 - » Similarity to the Prediction by the RNN
 - Adherence to **Music Theory Rules**
 - » Tonality, Avoid excessive repetitions...
 - » Open: Arbitrary **User-defined Targets**
- Reinforcement Learning implemented via *Deep Learning ! (Deep Reinforcement learning)*
- *Double Deep Reinforcement Learning*
 - » Double Q-Learning [Hasselt et al., 2015]
 - Decouple the action selection from the evaluation, to avoid value overestimation
 - » Target Q-Network for estimating Gain (Q)
 - » Q-Network for choosing next Action (next Note)



RL-Tuner – Results

without

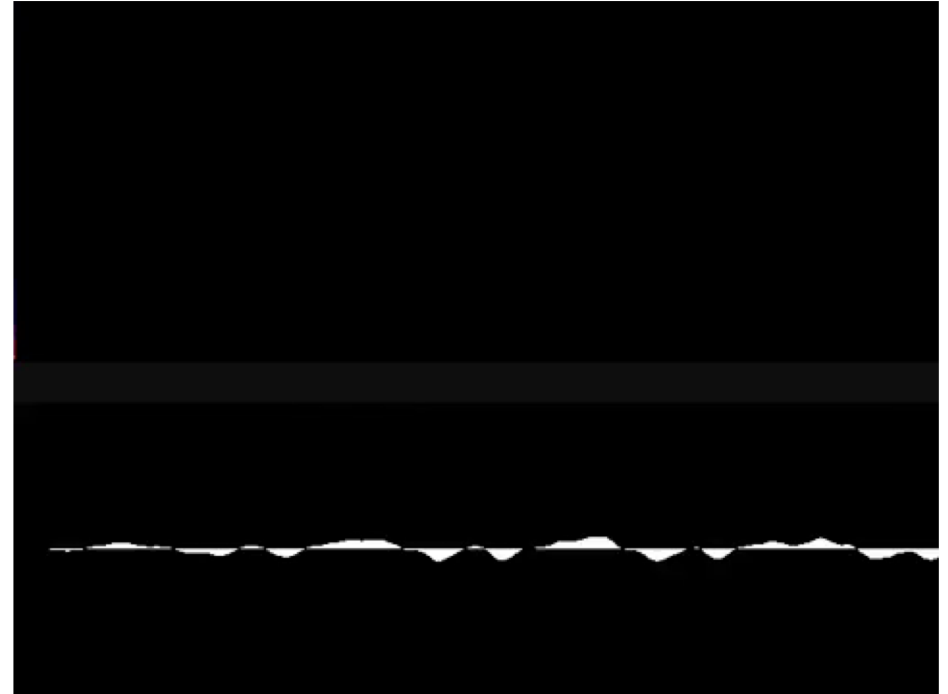
with

Behavior	Note RNN	Q-Learning	Psi	G
Notes excessively repeated	63.3%	0.0%	0.02%	0.03%
Notes not in key	0.1%	1.0%	0.6%	28.7%
Mean autocorrelation (lag 1,2,3)	-.16, .14, -.13	-.11, .03, .03	-.10, -.01, .01	.55, .31, .17
Leaps resolved	77.2%	91.1%	90.0%	52.2%
Compositions starting with tonic	0.9%	28.8%	28.7%	0.0%
Compositions with unique max note	64.7%	56.4%	59.4%	37.1%
Compositions with unique min note	49.4%	51.9%	58.3%	56.5%
Notes in motif	5.9%	75.7%	73.8%	69.3%
Notes in repeated motif	0.007%	0.11%	0.09%	0.01%

RL-Tuner – Demo



LSTM

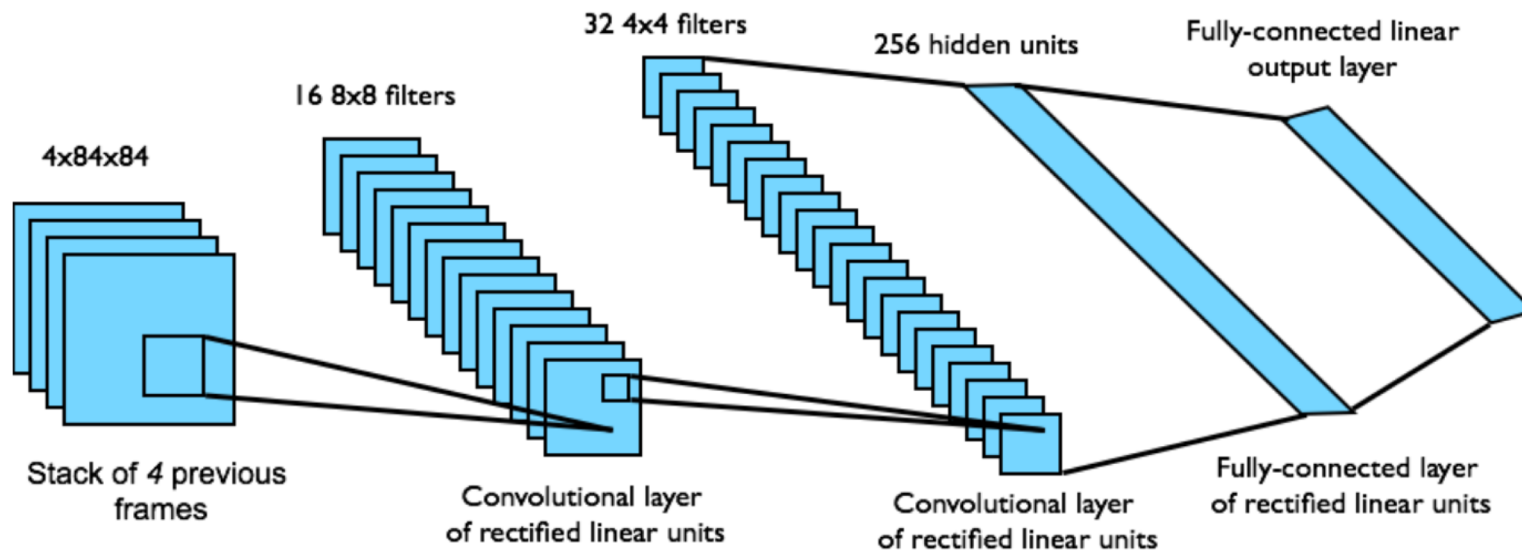
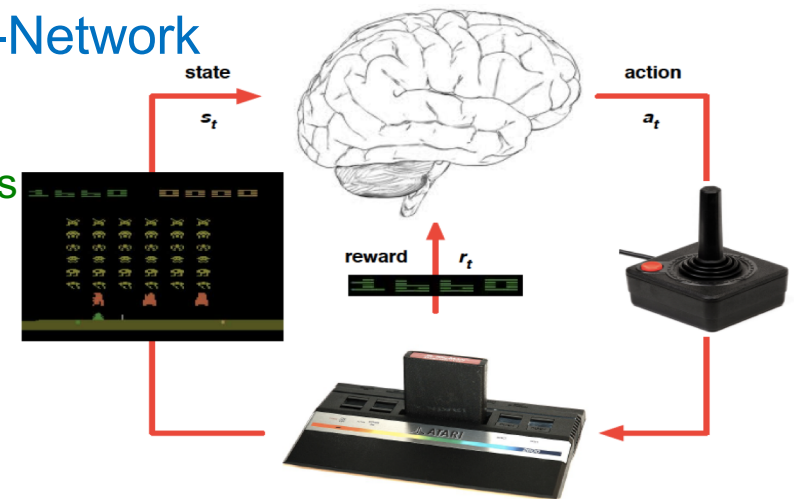


RL-Tuner

<https://magenta.tensorflow.org/2016/11/09/tuning-recurrent-networks-with-reinforcement-learning>

Deep Q-Learning [Minh et al. 2013]

- Atari Games Playing (DeepMind Technologies)
- Represent/Estimate Q Value function
(Gain for a <state, action> pair) as a Deep Q-Network
- Training of Q Value function Network
 - Inputs: Game screen raw pixels & joystick/button positions
 - Outputs: Q-values (captured from game play)
 - » Reward $\in \{-1, 0, 1\}$
- On-Line Training through Game Play
- Works with ANY (ATARI) Game !



Also:

AlphaGo
[Silver et al., 2016]

AlphaGo Zero
[Silver et al., 2017]

#4 Limitation – Control –

#5 Solution: Unit Selection [Bretan et al., 2016]

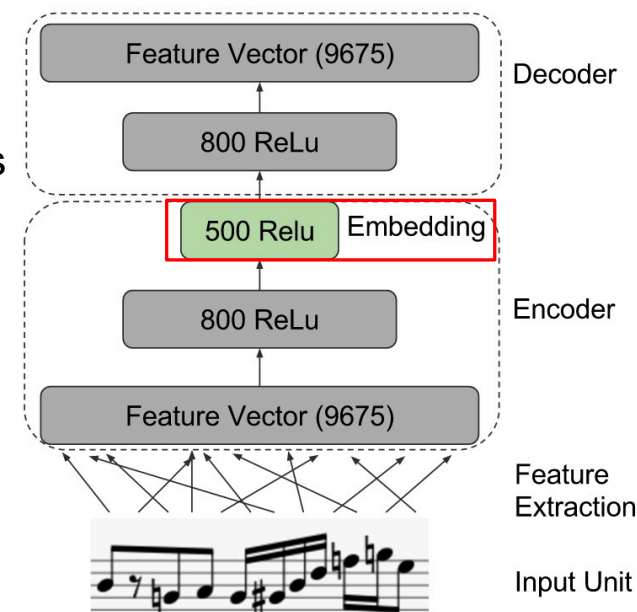
- Concatenation of Music Units Queried from a Database
- Key Process: Unit Selection, based on:
 - Successor Semantic Relevance
 - Concatenation Cost
 - Inspired by Text-to-Speech Generation Systems

- Feature Extraction

- By Stacked Autoencoders
- To Create Features Vector (Embedding) Representation of Units
- Used to Compute Distances between Units
- To Control Unit Selection

- Top Down Approach

- Structure (Next Unit Features)
- Fill (Select Best Fit Unit)



Unit Selection

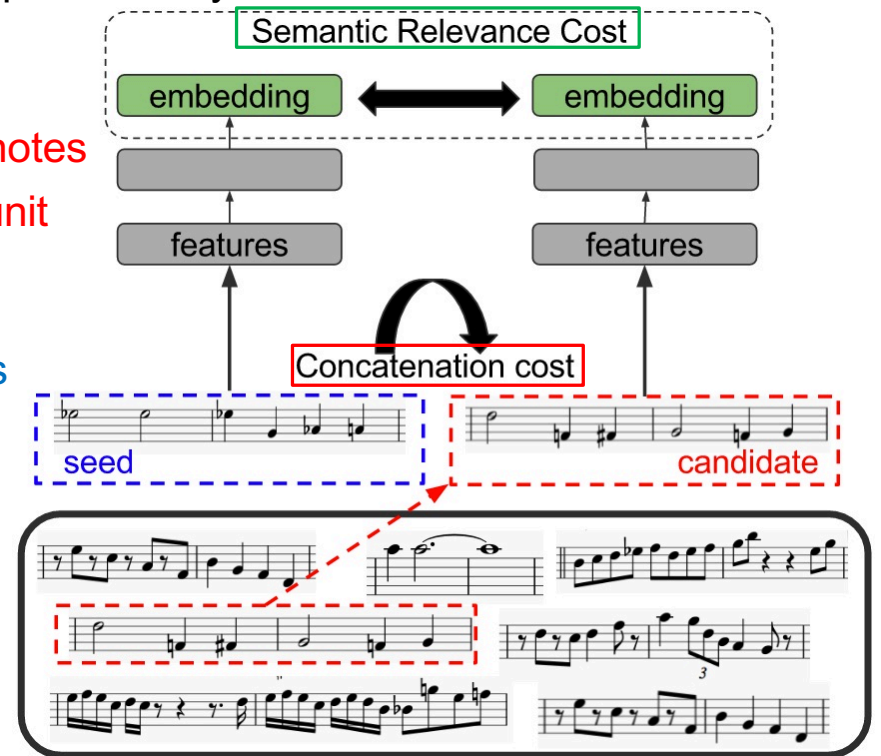
Unit Selection, based on:

- Successor Semantic Relevance
 - » Based on a model of transition between units
 - » As learnt by a 1st LSTM Recurrent Network
 - » Relevance = distance to the (ideal) next unit as predicted by the model

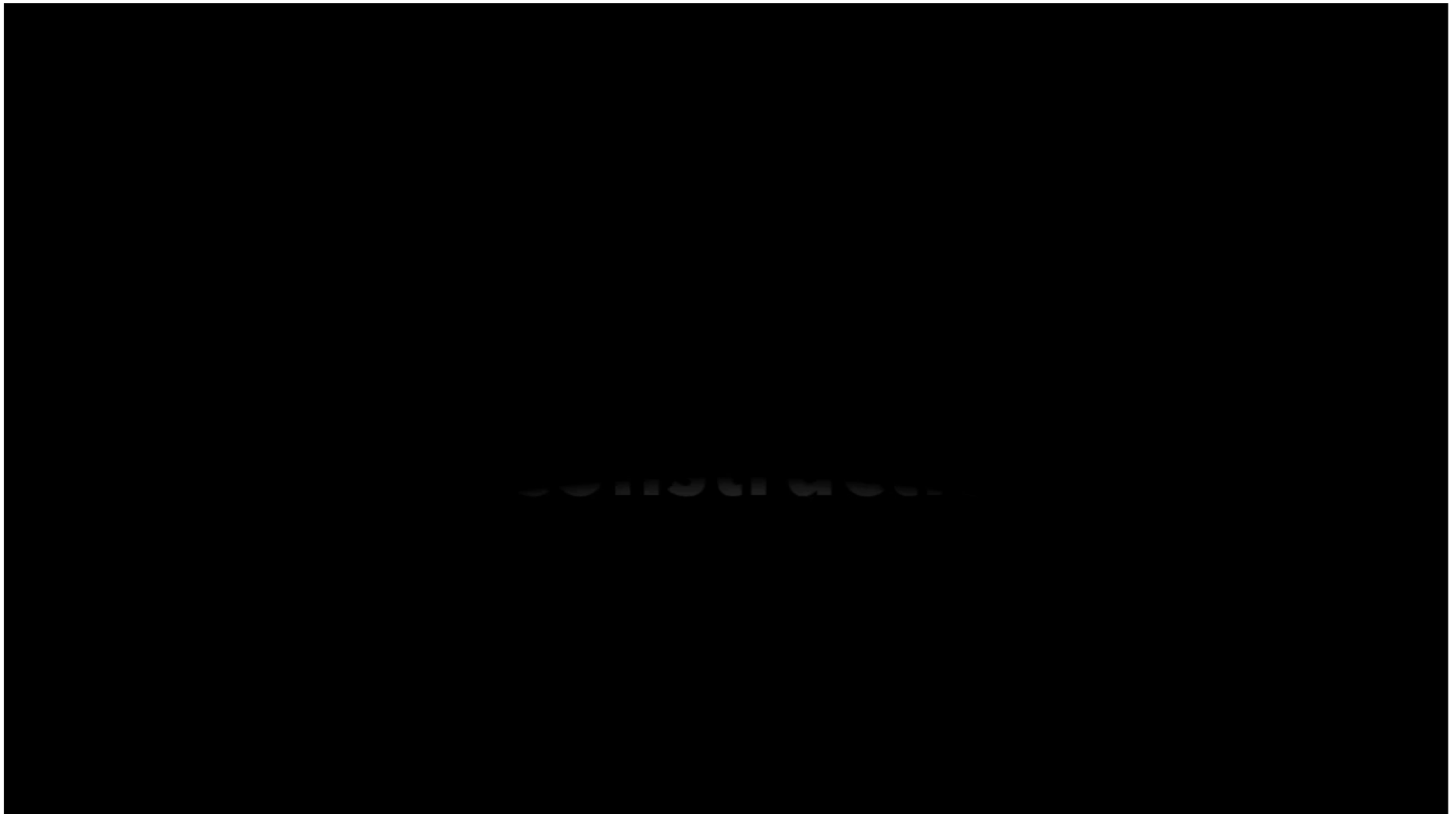
- Concatenation Cost
 - » Based on another model of transition between notes
 - » Between last note of unit and first note of next unit
 - » As learnt by a 2nd LSTM Recurrent Network

- Combination of the two criteria by a heuristic process

- As for Mozart Dice Music
- BUT with constraints on combination



Unit Selection – Demo



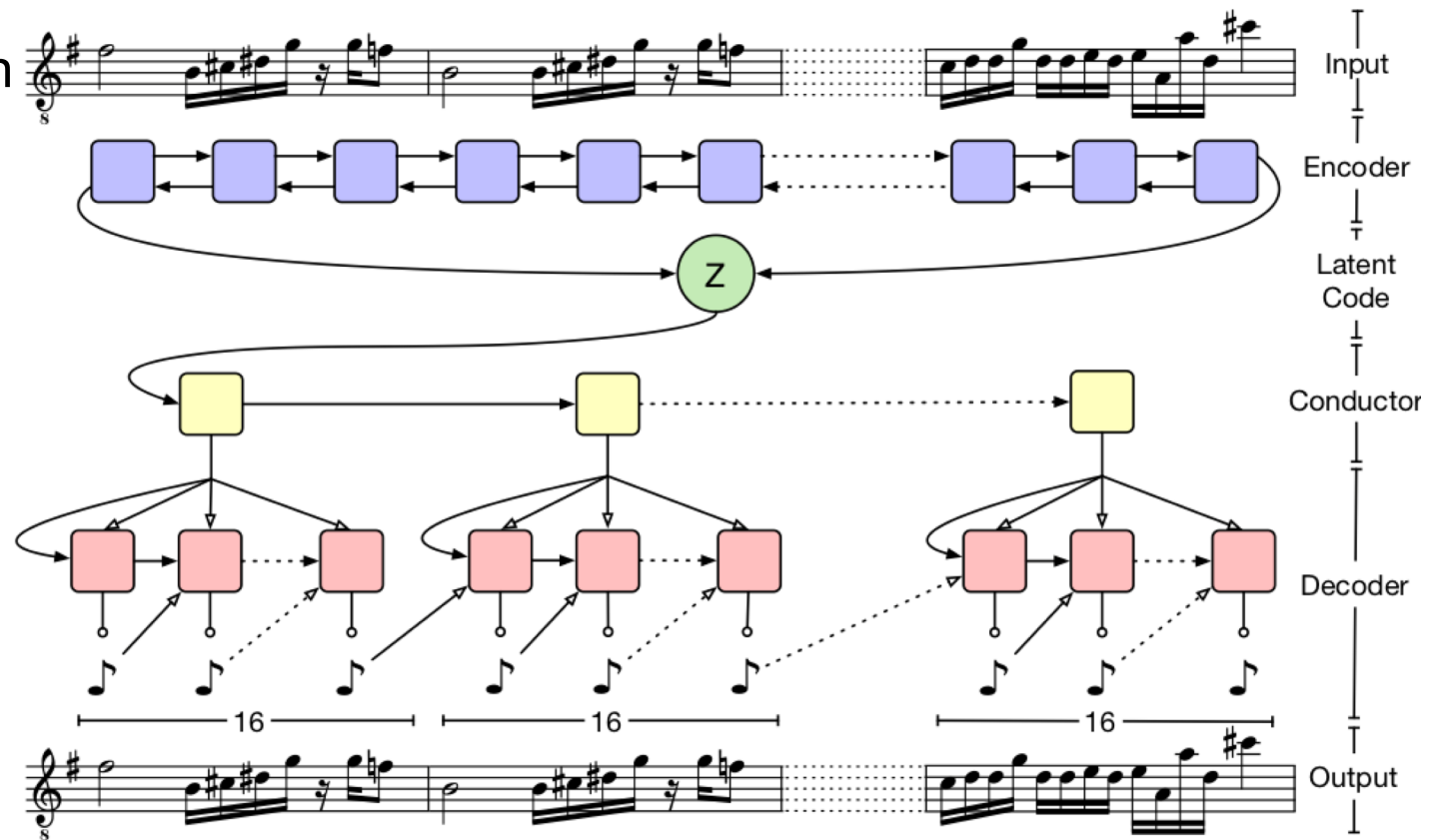
<https://www.youtube.com/watch?v=BbyvbO2F7ug&feature=youtu.be>

#5 Limitation – Structure

- No Sense of Direction
- No Structure

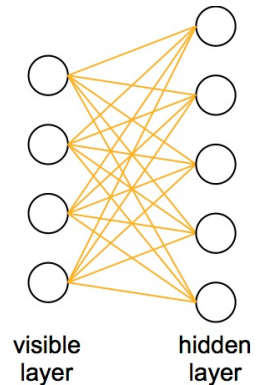
#5 Limitation – Structure – #1 Solution – Hierarchical RNN

- MusicVAE [Roberts et al., 2018]
- Hierarchical
 - Conductor RNN
 - Bottom RNN
- Long term generation
- Structure



#5 Limitation – Structure – #2 Solution – Structure Imposition

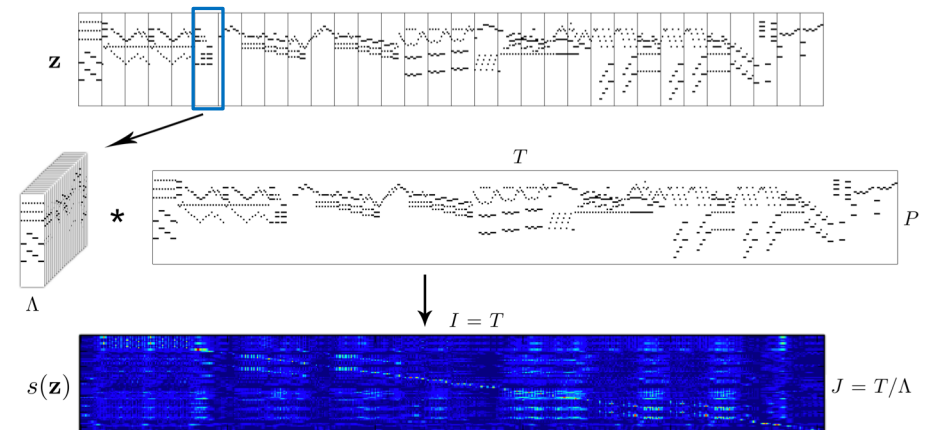
- Constrained sampling, C-RBM [Lattner et al., 2016]
- Convolutional Restricted Boltzmann Machine (RBM)
- Combination of:
 - **Input Manipulation** guided by **Gradient Descent** of current sample
 - » to impose Higher-Level Structure/Constraints:
 - Structure
 - Tonality
 - Meter
 - **Sampling Control**, by **Selective Gibbs sampling (SGS)**
 - » at a Selected Low-Level (subset of variables)
 - » to realign selectively the sample to the learnt distribution



Input Manipulation & Constrained sampling

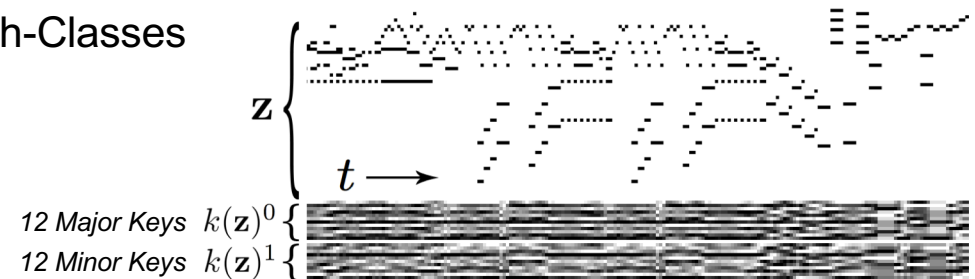
- Structure (Repetition Structure, Ex: AABA)

- » Self-Similarity Matrix
- » For each Music Slice



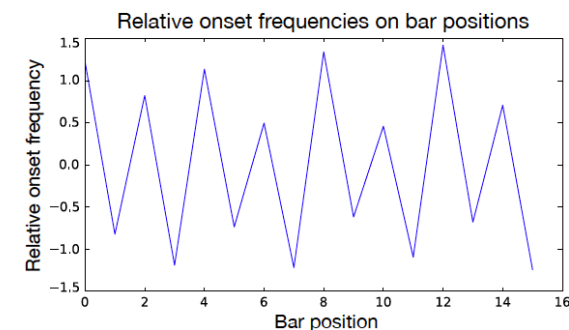
- Tonality, via Similarity of Distribution of Pitch-Classes

- » Key Estimation Vectors over Time



- Meter

- » Duration and Accent Patterns (ex: on 1st and 3rd Beats)
- » Via Relative Occurrence of Note Onsets



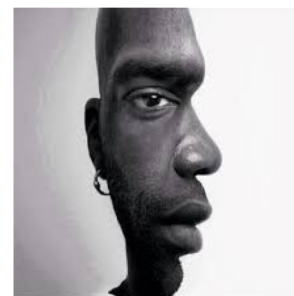
#6 Limitation – Originality

- Deep Learning will favor Conformance to a Corpus
- How to favor Originality/Creativity?
- Ex: CAN (Creative Adversarial Networks) architecture
[Elgammal et al., 2017] to favor the generation of new Styles

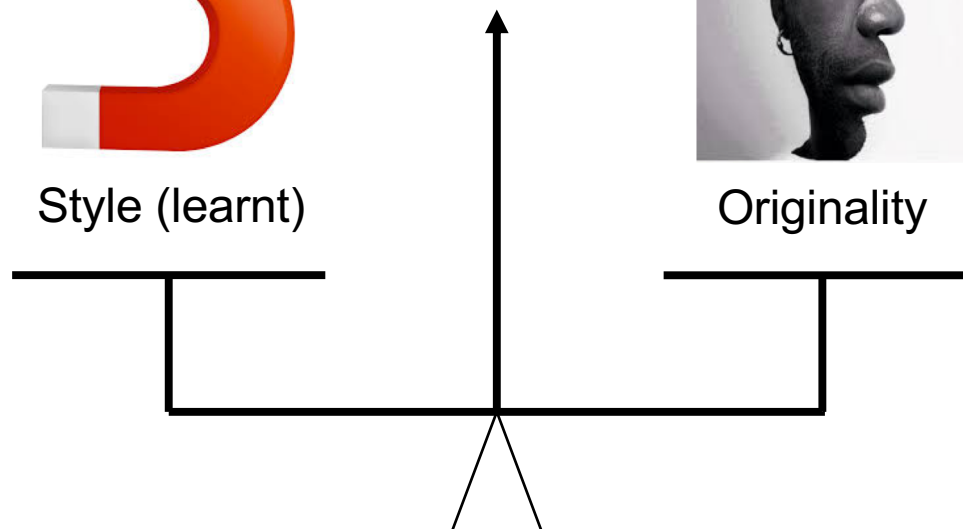
Style vs/and Originality



Style (learnt)



Originality



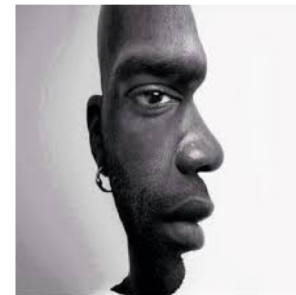
Style vs/and Originality



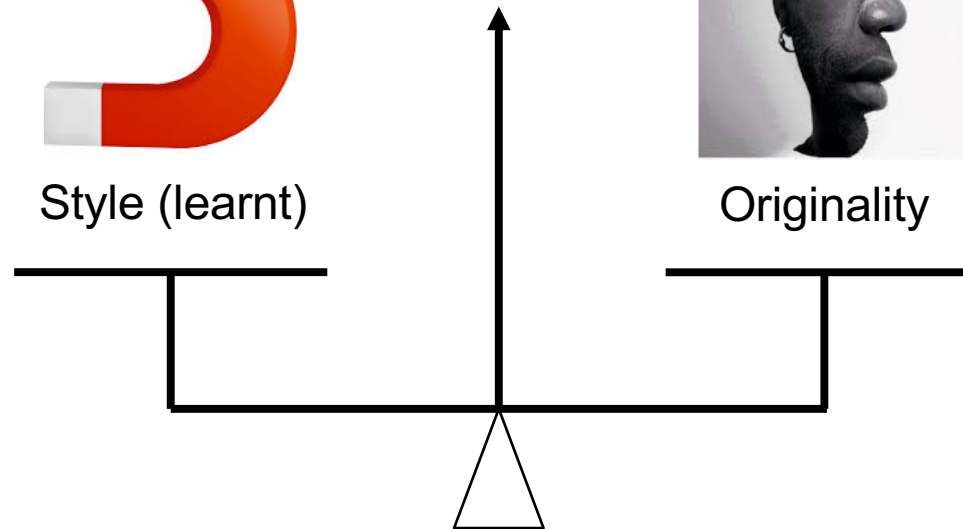
[Mimi & Eunice]



Style (learnt)



Originality



Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

- Training Simultaneously 2 Neural Networks
 - Generator
 - » Transforms Random noise Vectors into *Faked* Samples
 - Discriminator
 - » Estimates probability that the Sample came from training data rather than from G
 - Minimax 2-player game

$$\min_G \max_D V(G, D) = \log(D(x)) + \log(1 - D(G(z)))$$

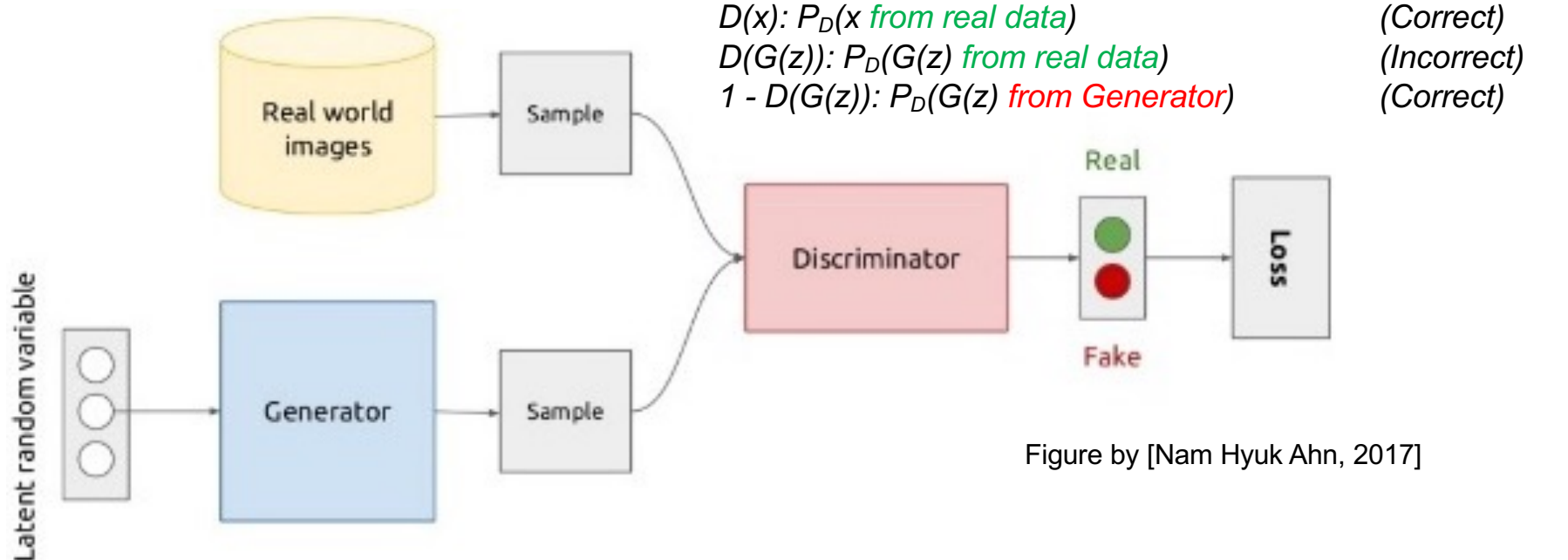
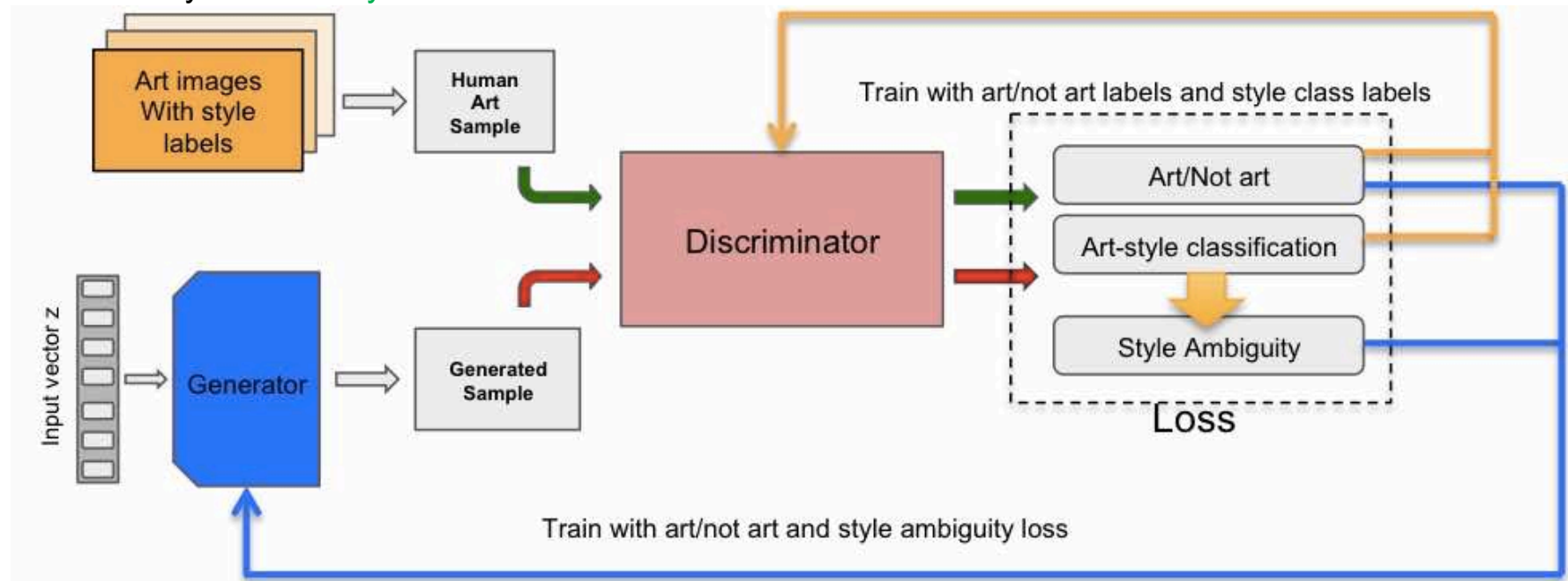


Figure by [Nam Hyuk Ahn, 2017]

#6 Limitation – Originality – #1 Partial Solution: Creative Adversarial Networks (CAN) [Elgammal et al., 2017]

- Extension of GAN
- Combining 2 (Contradictory) Objectives:
 - How Discriminator believes that the sample comes from the training dataset (GAN)
 - How **Easily** the Discriminator can classify the sample into established styles (classes)
 - » If there is strong **ambiguity** (i.e., various classes are **equiprobable**), this means that **the sample is difficult to fit within the existing art styles**
 - » Maybe **a new style** has been created...



Creative Adversarial Networks (CAN) – Ex. of Paintings Generated

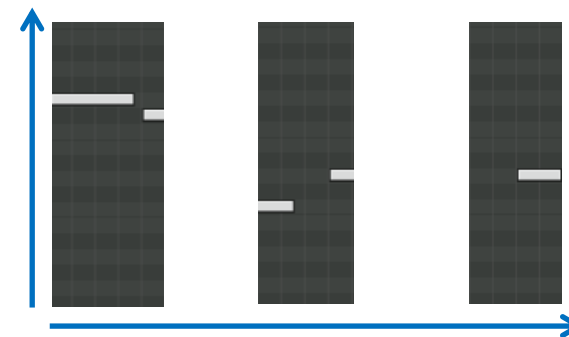
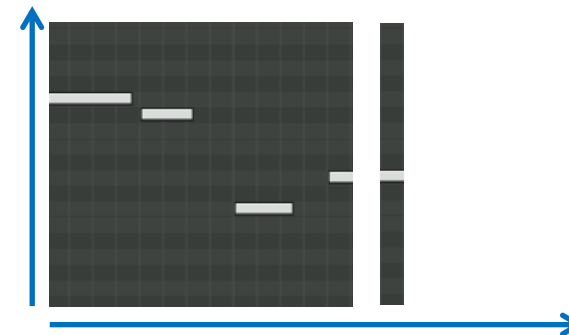
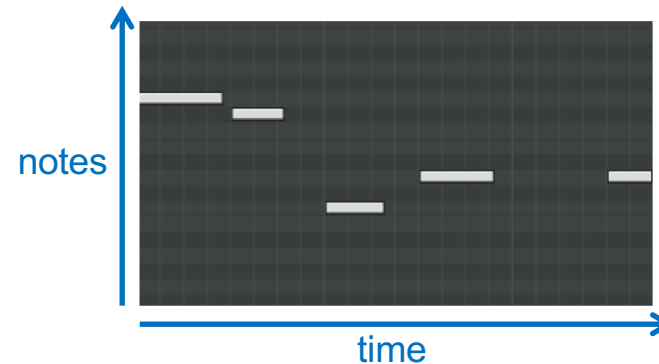
Table 1: Artistic Styles Used in Training

Style name	Image number	Style name	Image number
Abstract-Expressionism	2782	Mannerism-Late-Renaissance	1279
Action-Painting	98	Minimalism	
Analytical-Cubism	110	Naive Art-Primi	
Art-Nouveau-Modern	4334	New-Realism	
Baroque	4241	Northern-Renai	
Color-Field-Painting	1615	Pointillism	
Contemporary-Realism	481	Pop-Art	
Cubism	2236	Post-Impression	
Early-Renaissance	1391	Realism	
Expressionism	6736	Rococo	
Fauvism	934	Romanticism	
High-Renaissance	1343	Synthetic-Cubis	
Impressionism	13060	Total	

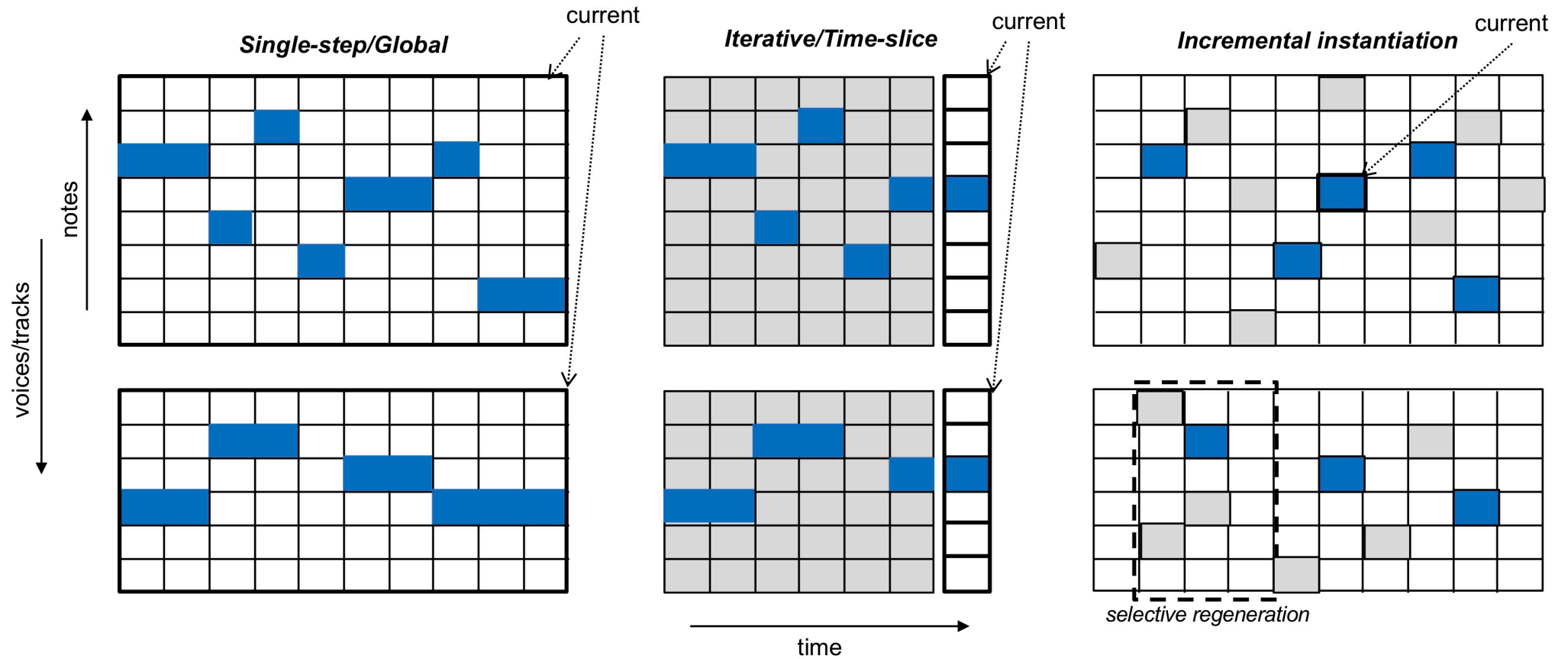


#7 Limitation – Incrementality

- Instantiation of Notes
 - Feedforward Models
 - » Global/One shot
 - Recurrent Models
 - » Note by Note
 - » Following Time Steps
 - ?
 - » Arbitrary Part/Direction



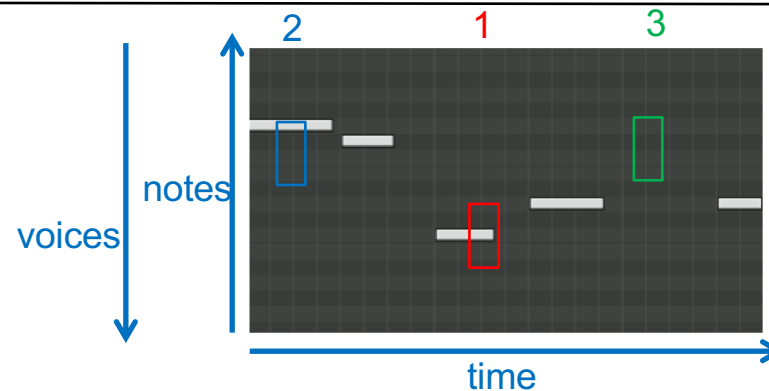
#7 Limitation – Incrementality



#7 Limitation – Incrementality

– #1 Solution: DeepBach [Hadjeres et al., 2017]

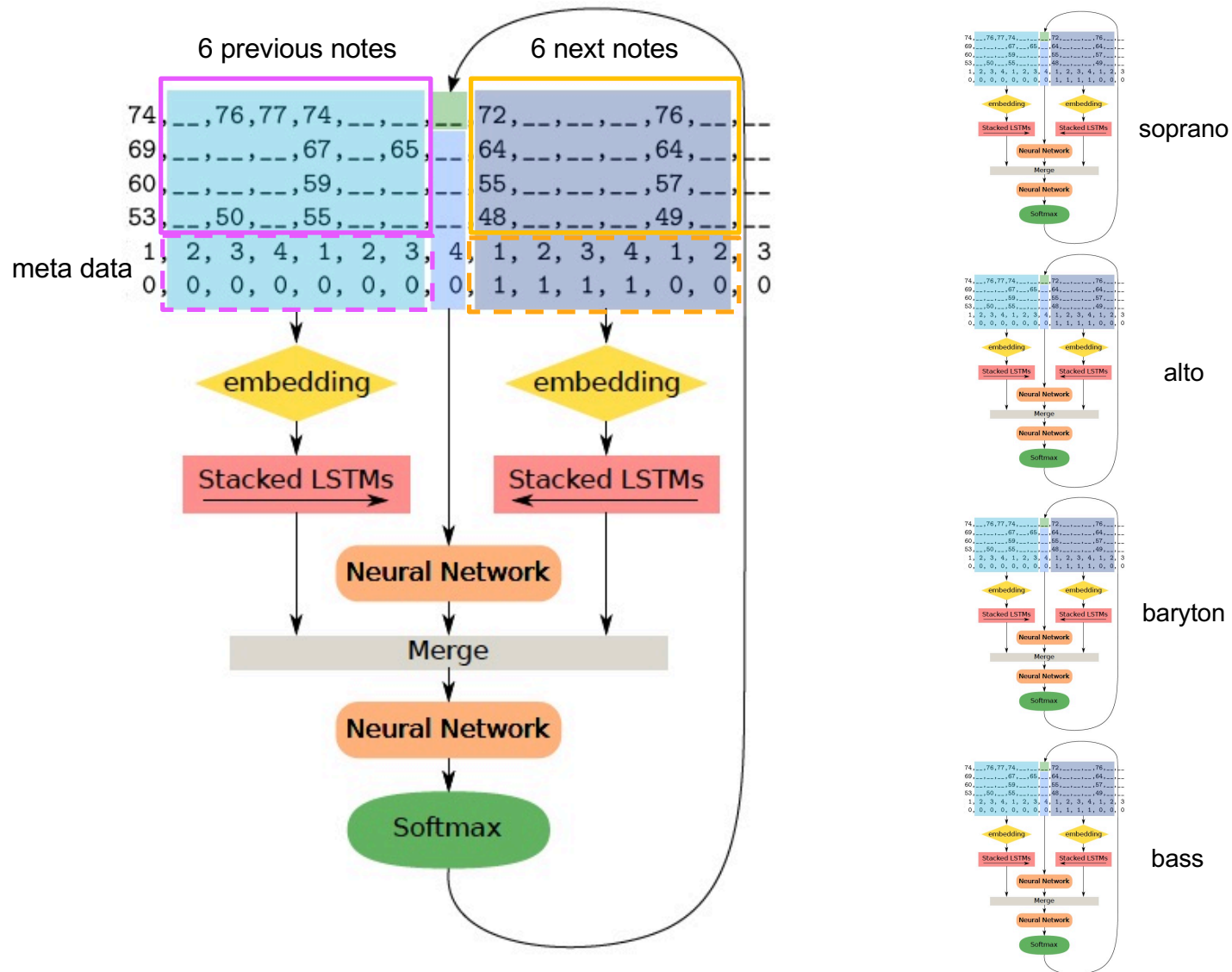
- Incremental Sampling
 - Iterative
 - Pseudo-Gibbs Sampling
 - Iterative
 - « Random »



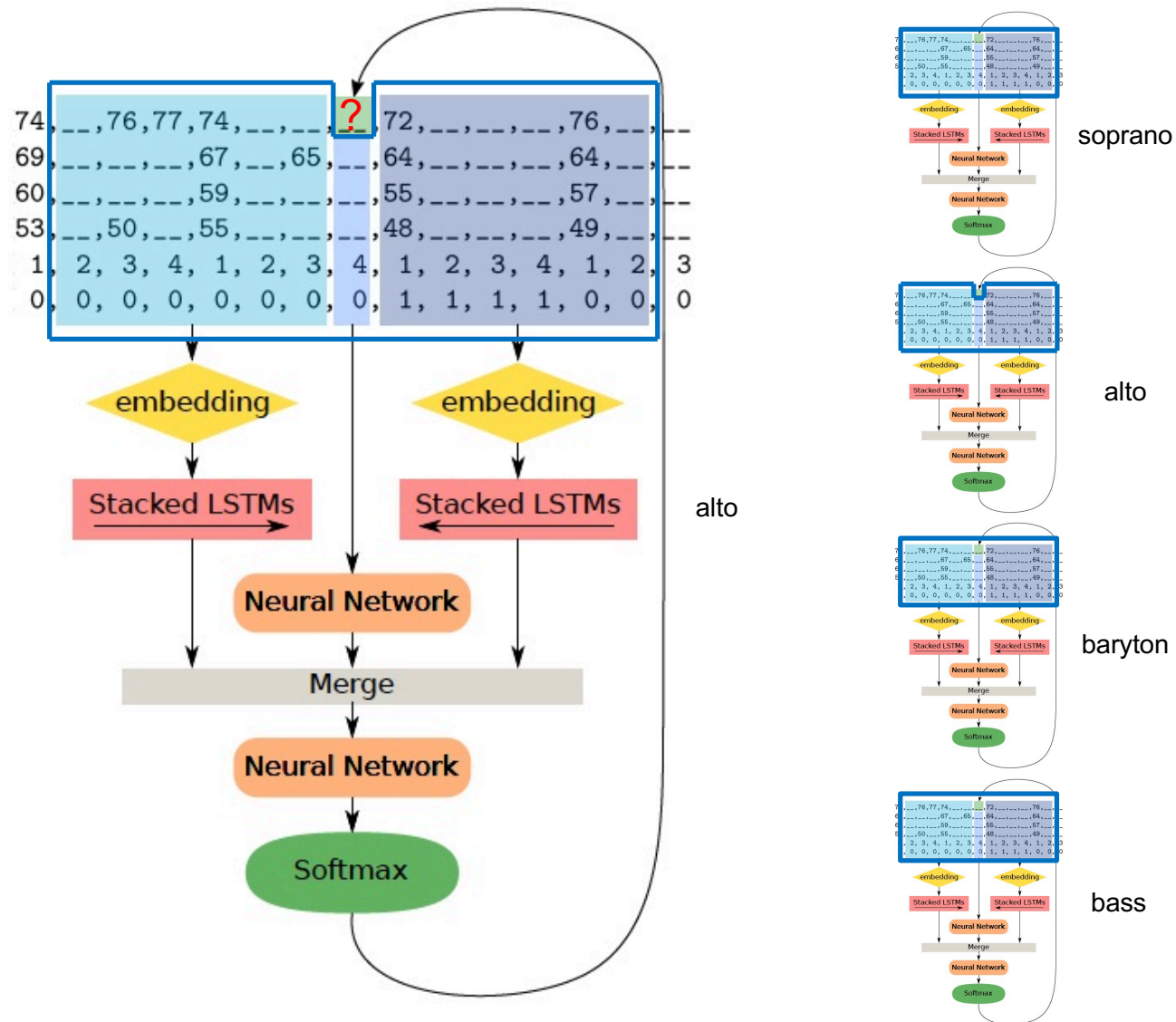
- Bach Chorales
 - 4 voices
- for m from 1 to M do
Choose voice i uniformly between 1 and 4
Choose time t uniformly between 1 and L
Re-sample $\mathcal{V}_i^t$ from $p_i(\mathcal{V}_i^t | \mathcal{V}_{\setminus i, t}, \mathcal{M}, \theta_i)$



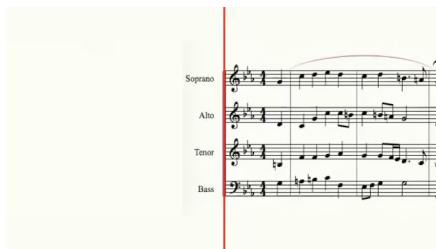
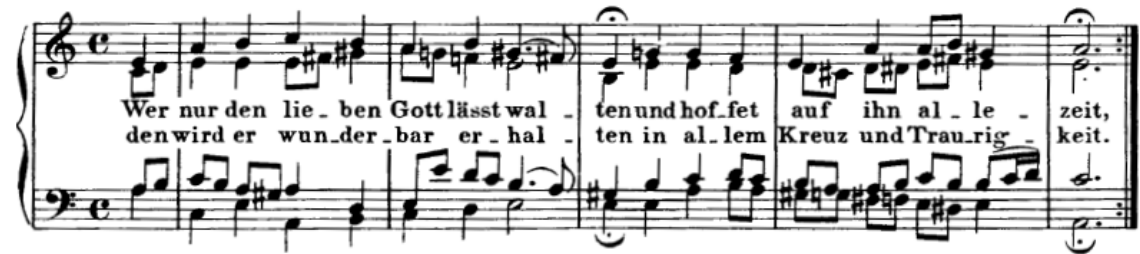
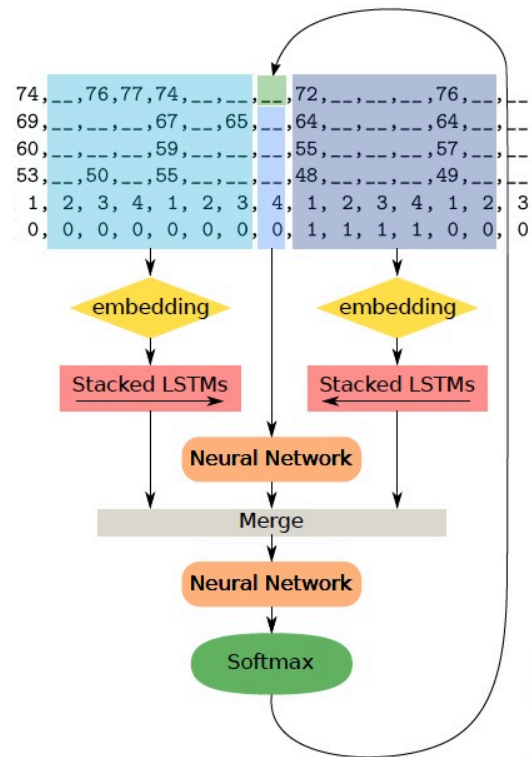
DeepBach



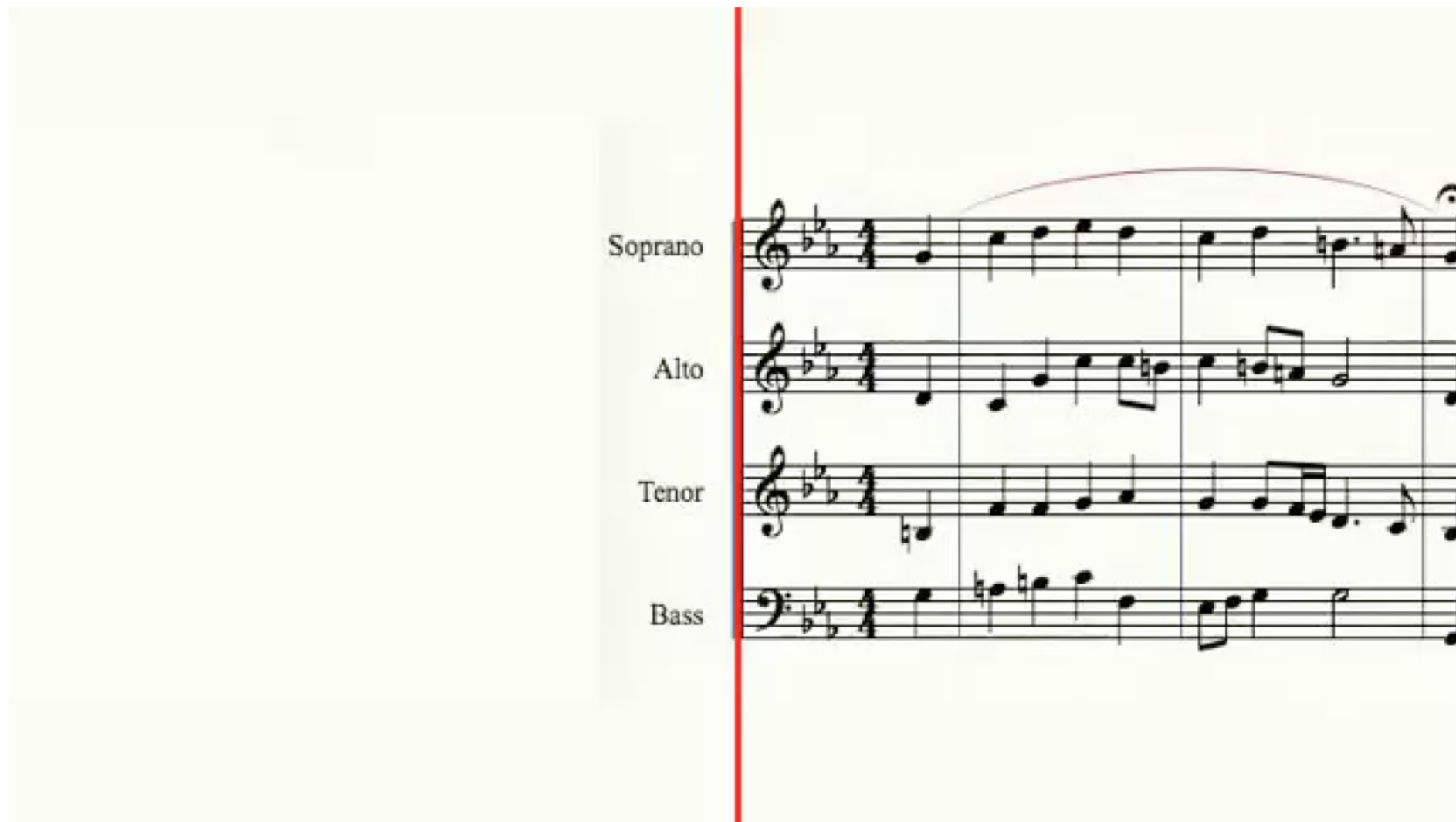
DeepBach



DeepBach



DeepBach – Demo



A musical score for four voices: Soprano, Alto, Tenor, and Bass. The score is displayed on four staves, each with a treble or bass clef and a key signature of two flats (B-flat and E-flat). The time signature is 4/4. A red vertical line is positioned between the Soprano and Alto staves. A red horizontal line is positioned between the Tenor and Bass staves. A red curved line (phrasing slur) is placed over the Soprano staff, spanning from the first measure to the end of the piece. The music is written in a modern, clean style with black notes and stems on a white background.

<https://www.youtube.com/watch?v=QiBM7-5hA6o>
Deep Learning – Music Generation – 2019

Turing Bach Test



<https://www.youtube.com/watch?v=DPNHbtiXGEM>

#8 Limitation – Interactivity

– #1 Solution: DeepBach [Hadjeres et al., 2017]



#8 Limitation – Interactivity

– #1 Solution: DeepBach [Hadjeres et al., 2017]

The screenshot displays the DeepBach web application interface. On the left, a sidebar contains the application title "Deep Bach" and a "Server address" field set to "http://localhost:5000/". Below this is a dropdown menu showing "deepbach" and a "Load" button. A "Compose" button is also present. A log of messages is displayed in green text:

- Retrieving models list at http://localhost:5000/
- No models found
- Retrieving models list at http://localhost:5000/
- No models found
- Retrieving models list at http://localhost:5000/
- Models list loaded
- currently loaded model is deepbach
- Loading model deepbach
- Model deepbach loaded
- Composing...
- Got Empty Response when composing
- Loading model deepbach
- Model deepbach loaded
- Composing...
- Done composing

The main area shows a music score titled "Music21 Fragment_recomposed_by_deepBach". The score is displayed in a 4-staff format (two treble clefs and two bass clefs) with a key signature of one sharp (F#) and a time signature of 4/4. The score is composed of 11 measures, with the first measure highlighted. The interface includes a toolbar at the top with various music notation symbols and a "Tonalité de concert" dropdown menu.

https://www.youtube.com/watch?time_continue=28&v=OkkKjy3WRNo

#8 Limitation – Interactivity

– #2 Solution: Flow Composer [Papadopoulos et al., 2016]

Fully Autonomous Automated Generator

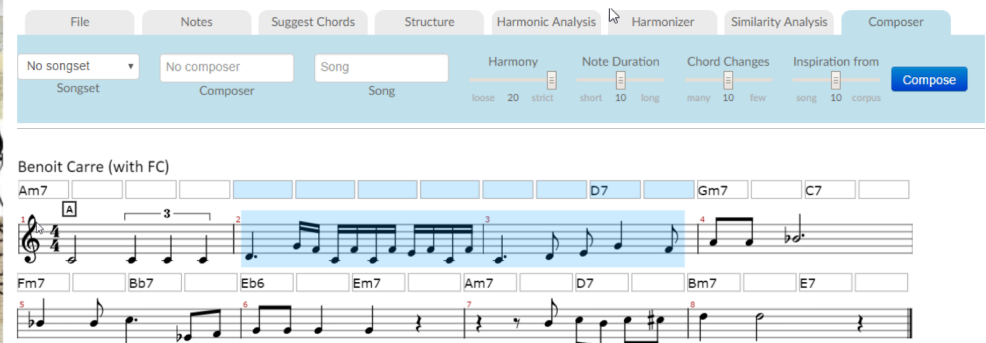
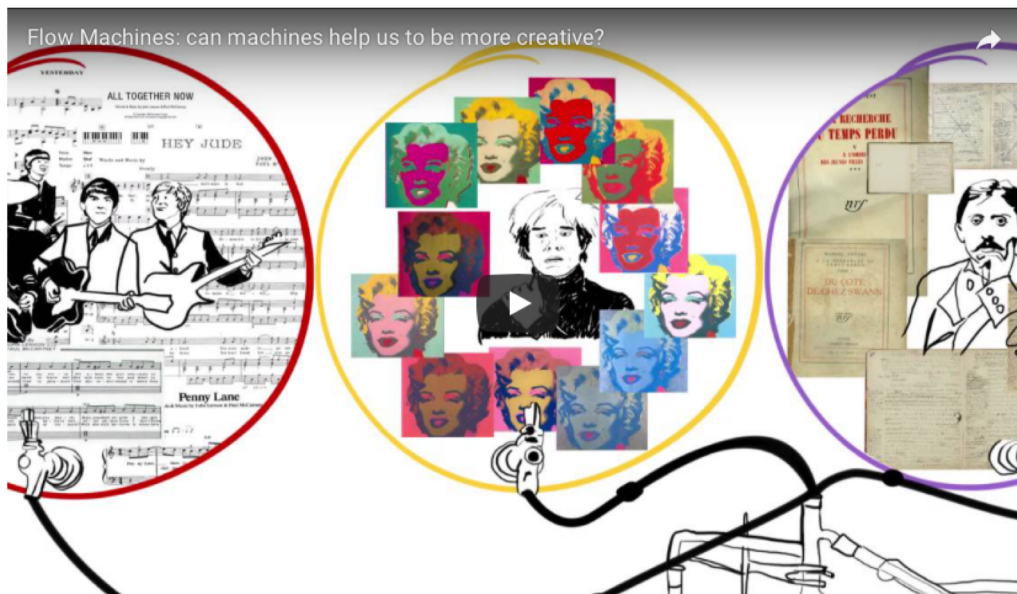
vs Assistant for Human Artists (Composers, Arrangers, Instrumentists...)

Flow Machines Project [Pachet et al., 2012]

Flow Machines are cutting-edge algorithms, made to explore new ways to create.

Flow Machines collaborate with musicians to compose the future.

Flow Machines are AI music-making.



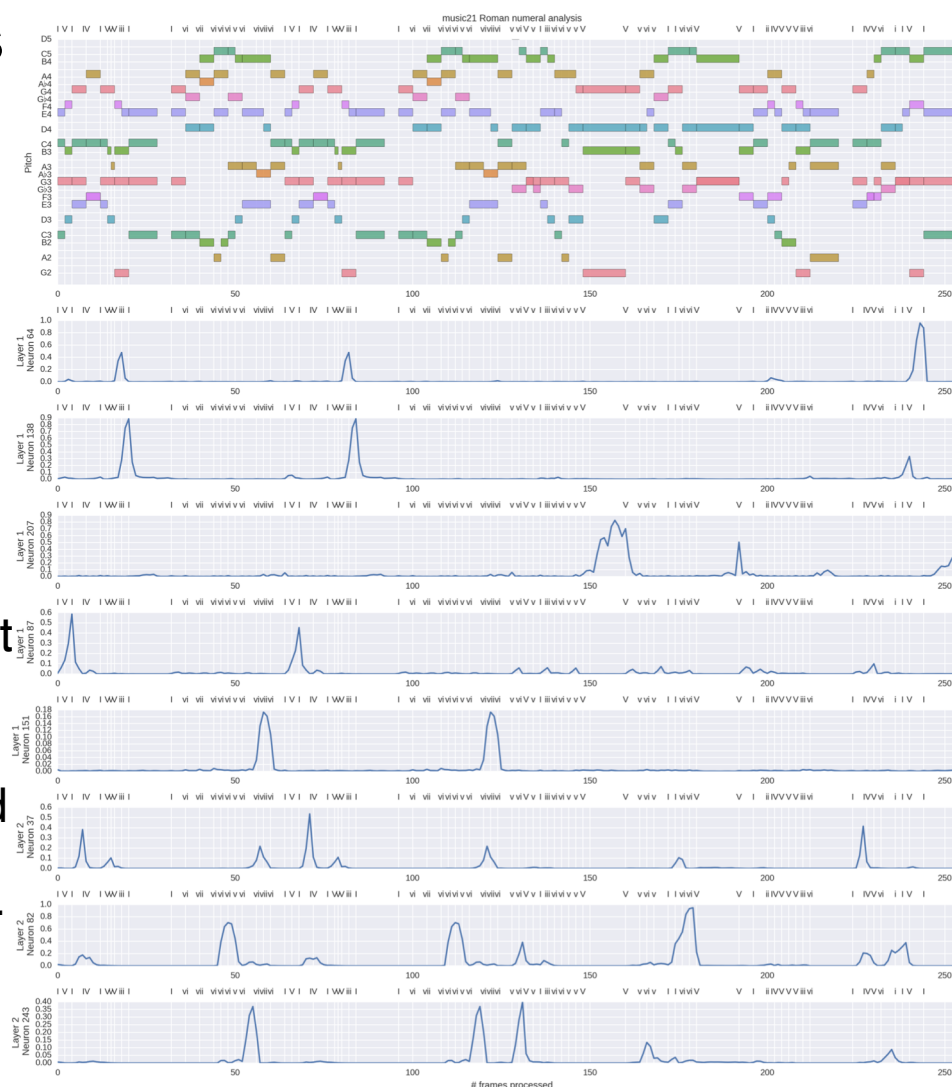
FlowComposer – Demo

#9 Limitation – Explainability

– #1 Partial Solution: BachBot [Liang, 2016]

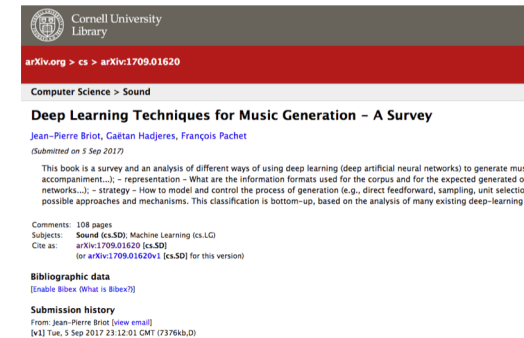
- Ex: BachBot correlation analysis [Liang, 2016]

- The first two (Layer 1/Neuron 64 and Layer 1/Neuron 138) seem to pick out (specifically) perfect cadence with root position chords in the tonic key
- There are no imperfect cadences here; just one interruption into bar 14
- Layer 1/Neuron 87: the I6 chords on the first downbeat and its reprise 4 bars later
- Layer 1/Neuron 151: the two equivalent a minor (originally b minor) cadences that end phrases 2 and 4
- Layer 2/Neuron 37: Seems to be looking for I6 chords: strong peak for a full I6; weaker for other similar chords (same bass)

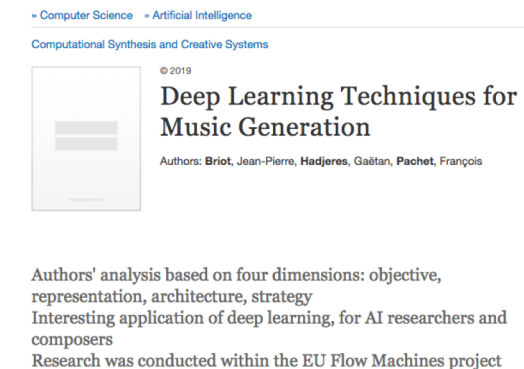


References

J.-P. Briot, G. Hadjeres, F.-D. Pachet, Deep Learning Techniques for Music Generation – A Survey, ArXiv:1709.01620, September 2017.



J.-P. Briot, G. Hadjeres, F.-D. Pachet, Deep Learning Techniques for Music Generation, Computational Synthesis and Creative Systems Series, Springer Nature, 2019.

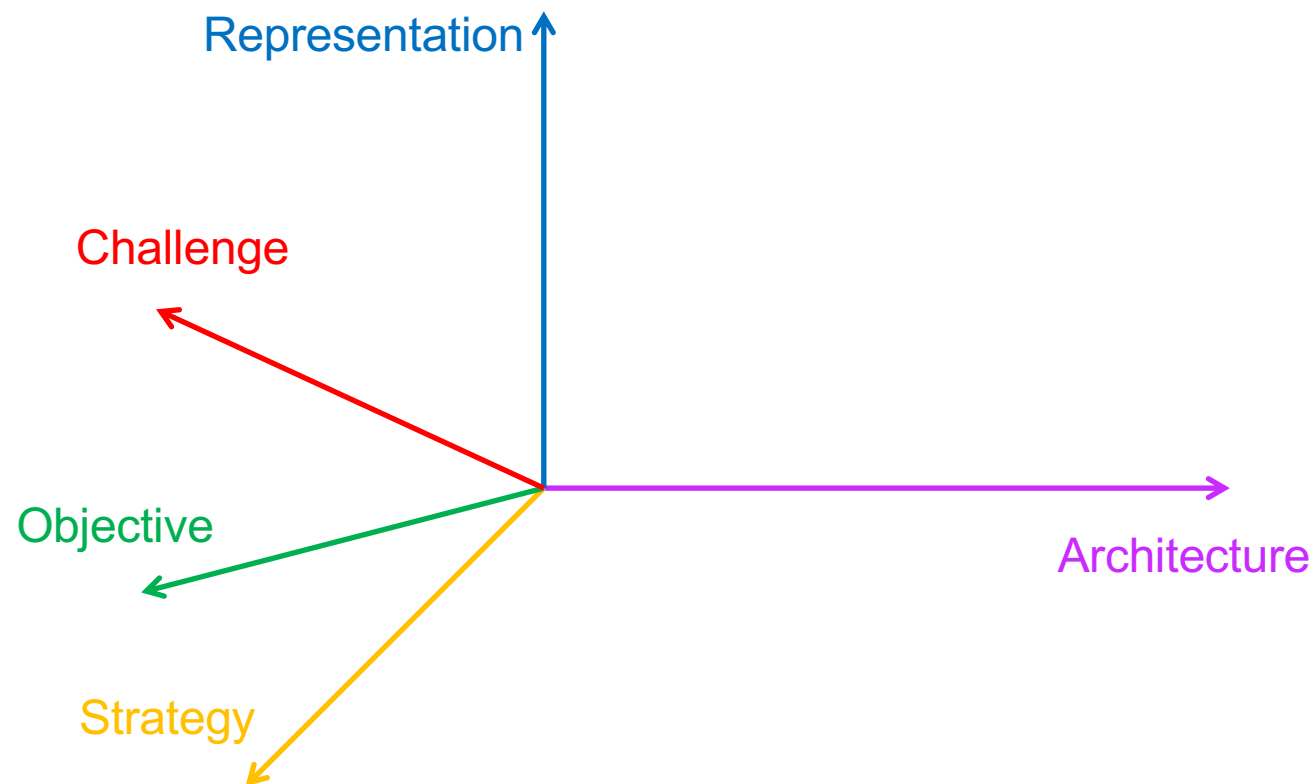


J.-P. Briot, F. Pachet, Music Generation by Deep Learning – Challenges and Directions, Neural Computing and Applications (NCAA), Special Issue on Deep Learning for Music and Audio, October 2018.



Survey/Analysis

4+1 dimensions



Objective

- Melody
 - Monodic
 - Polyphonic
- Polyphony (Multiple Voices/Tracks)
- Accompaniment
 - Counterpoint
 - » Melody
 - » Melodies (Chorale)
 - Chords
- Melody + Harmony/Chords



- Leadsheet



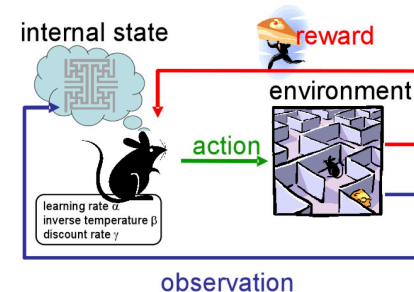
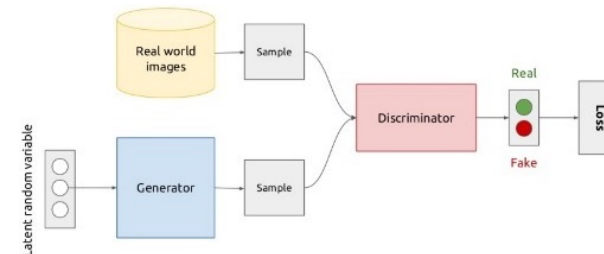
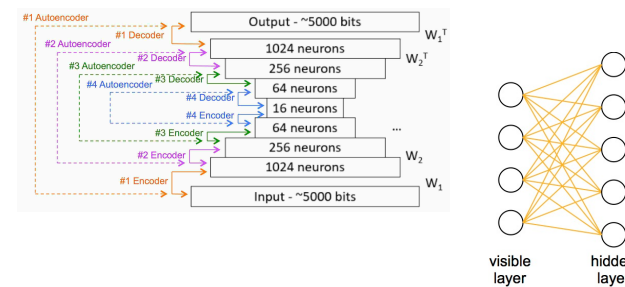
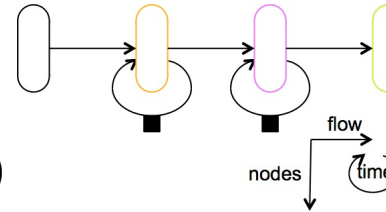
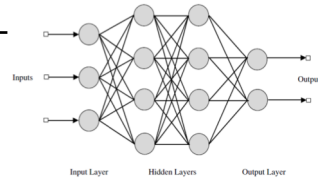
Representation

- Rhythm



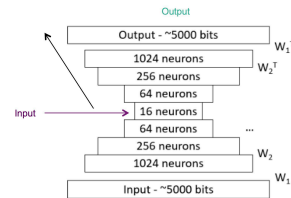
Architecture

- Feedforward
- Recurrent (RNN)
 - Long Short-Term Memory (LSTM)
- Autoencoder
 - Stacked Autoencoders
- Restricted Boltzmann Machine (RBM)
- Variational Autoencoder (VAE)
- Patterns
 - Convolutional
 - Conditioning
 - Generative Adversarial Networks (GAN)
- Reinforcement Learning
- Compound
 - RNN Encoder-Decoder
 - ...



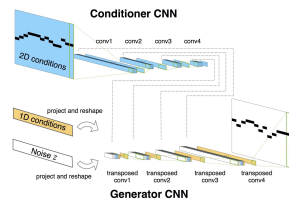
Strategy

- Feedforward
 - Single-Step Feedforward
 - Iterative Feedforward
 - Decoder Feedforward



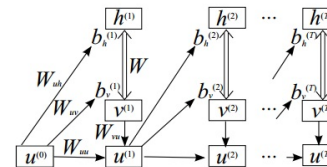
[Sun, 2016]

- Conditioning



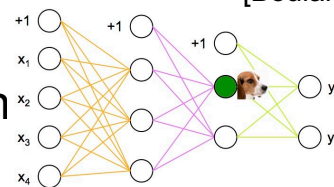
[Yang et al., 2017]

- Sampling

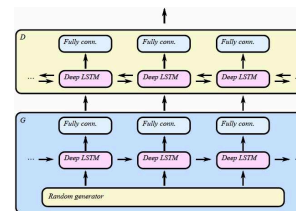


[Boulanger-Lewandowski et al., 2012]

- Input Manipulation

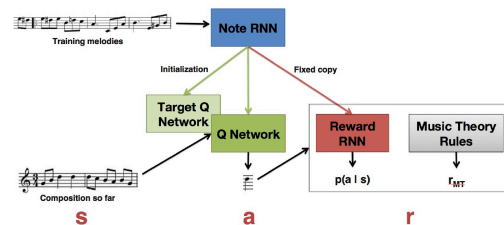


- Adversarial

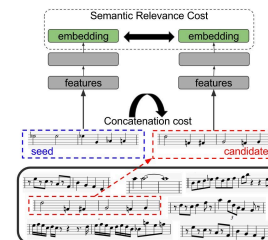


[Mogren, 2016]

- Reinforcement



[Jaques et al., 2016]



[Bretan et al., 2016]

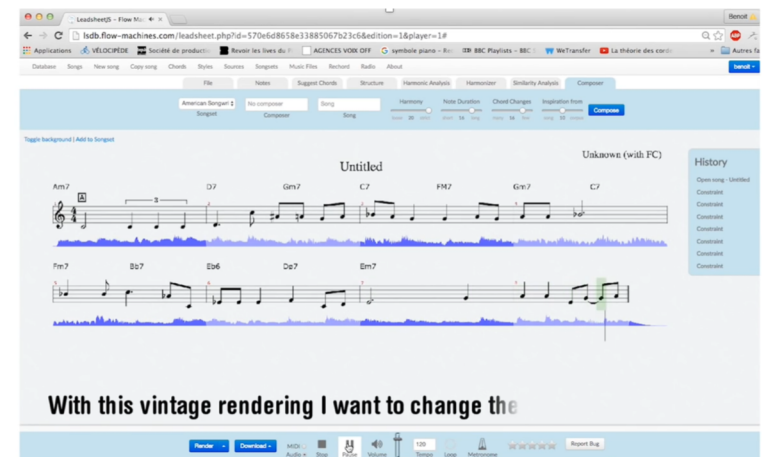
- Unit Selection

Configuration and Control Issues

- Corpus (Curation): Training Examples -> Style
- Architecture(s)
 - Single or Compound
 - Conditioning (Parameterization)
 - Configuration (Hyperparameters)
 - Loss Function
 - » From Prediction or Reconstruction Error to Incorporating more and more Constraints
 - External Loss/Control, ex: Adversarial/GAN
- Strategy(ies)
 - Data/Input Manipulation, Ex: Latent Variables
- Improbable Settings – Imagination Limits?
- Interactivity

Interactive Creation Environment

- A Deep Learning-Based Flow Composer Analog ?
- Slower Learning than for Markov Models
 - But GPUs, etc.
 - And Corpus Pre-Training
- No (or not yet) Exact Control Method (Markov Constraints)
- Various Architectures/Strategies
- Inspiration, RNN-based
- Complementation, Feedforward-based
- Control, VAE-based
- Inpairing, (V)AE+RNN-based



Acknowledgements

- Gaëtan Hadjeres
- François Pachet
- CNRS
- LIP6
- ERC Flow Machines
- Sony CSL-Paris
- Spotify CTRL
- PUC-Rio
- CAPES
- UNIRIO
- UNIRIO Course Students

Thank You – Questions
