Progress and Challenges for Music Generation By Deep Neural Networks (Deep Learning)

Jean-Pierre Briot

Jean-Pierre.Briot@lip6.fr

LIP6

Sorbonne Université – CNRS

Programa de Pós-Graduação em Informática (PPGI)

UNIRIO

Why/Outline

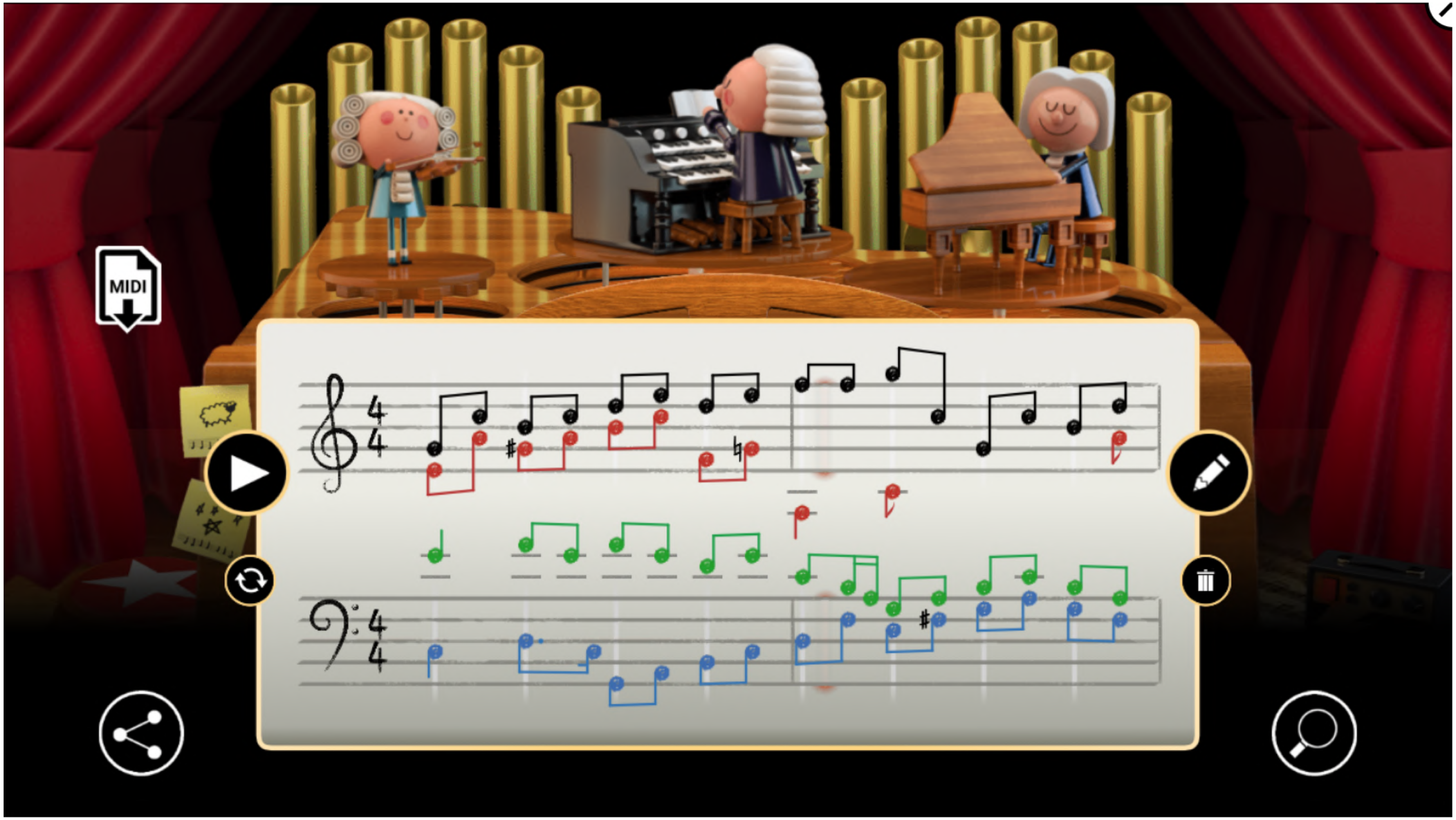
- Music Generation
- Recent boom using Deep Learning Techniques
- Very active domain
  - Ex: Google Magenta Project
- What is New?
  - From Initial Neural Networks
- Generative Architectures
  - Variational Autoencoders (VAE)
  - Generative Adversarial Networks (GAN)
- Issues
  - Interaction, Control, Creativity, Structure
- Prospects

Outline

- Deep Learning Music Generation Recent Achievements
- Neural Networks
- A First Example of Music Generation
- *Pioneering Work of Neural Network-based Music Generation (1988)*
- From Neural Networks to Deep Learning
- Deep Learning Progress and Architectures
- Variational Autoencoders (VAE)
- Generative Adversarial Networks (GAN)
- *Autonomous Generation vs Creation Support*
- Issues/Challenges
- Control
- Conclusion

Recent Creations

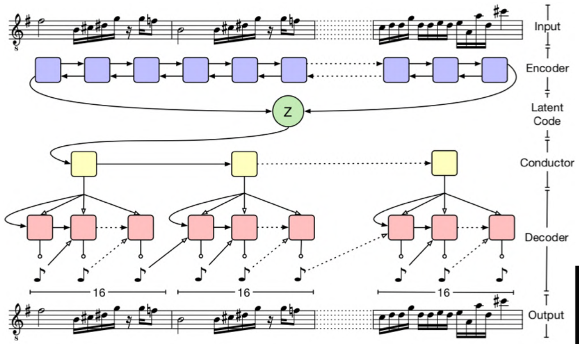
Doodle Bach Chorales

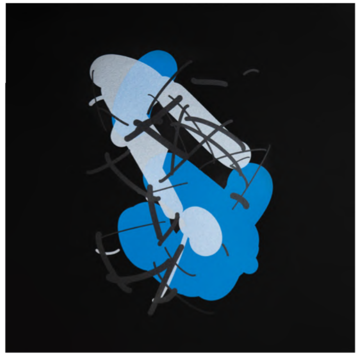


https://www.google.com/doodles/celebrating-johann-sebastian-bach

Electro Dance-Pop Music

- YΔCHT (Young Americans Challenging High Technology)
- Chain Tripping Album, 30 August 2019
- Composed with Magenta MusicVAE [Roberts et al., 2018]





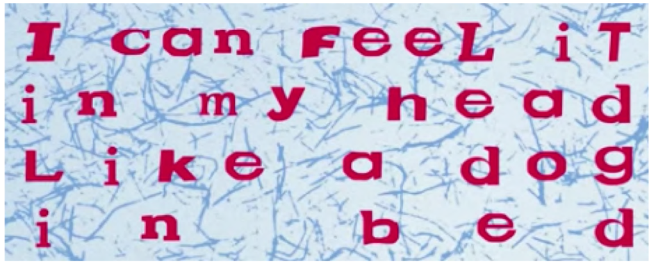
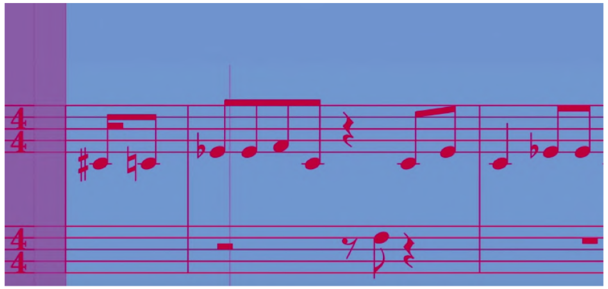
I'm so in love
I can feel it in my car
I can feel it in my heart,
I can feel it so hard
I want your phone to my brain
I want you to call my name
I want you to do it too
Oh, won't you come, won't you come
Won't you work on my head
Be my number nine

(Downtown) Dancing

Loud Light

YΔCHT + Magenta – Chain Tripping Album

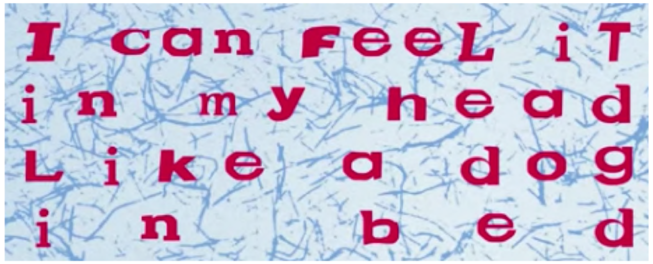
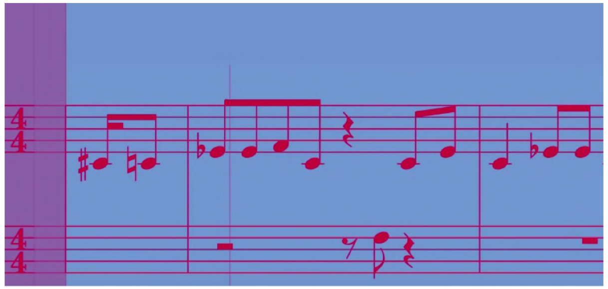
- Melody/Chords/Rhythm Loops
  - MusicVAE (VRAE)
  - Training Corpus: Previous music by YΔCHT
- Lyrics
  - LSTM
  - Training Corpus: YΔCHT + Liked Lyrics
- Sounds
  - Nsynth (Signal VAE)
- Images and Videos
  - GAN



https://arstechnica.com/gaming/2019/08/yachts-chain-tripping-is-a-new-landmark-for-ai-music-an-album-that-doesnt-suck/

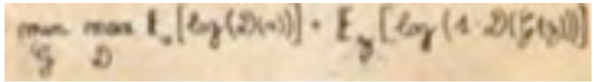
YΔCHT + Magenta – Chain Tripping Album

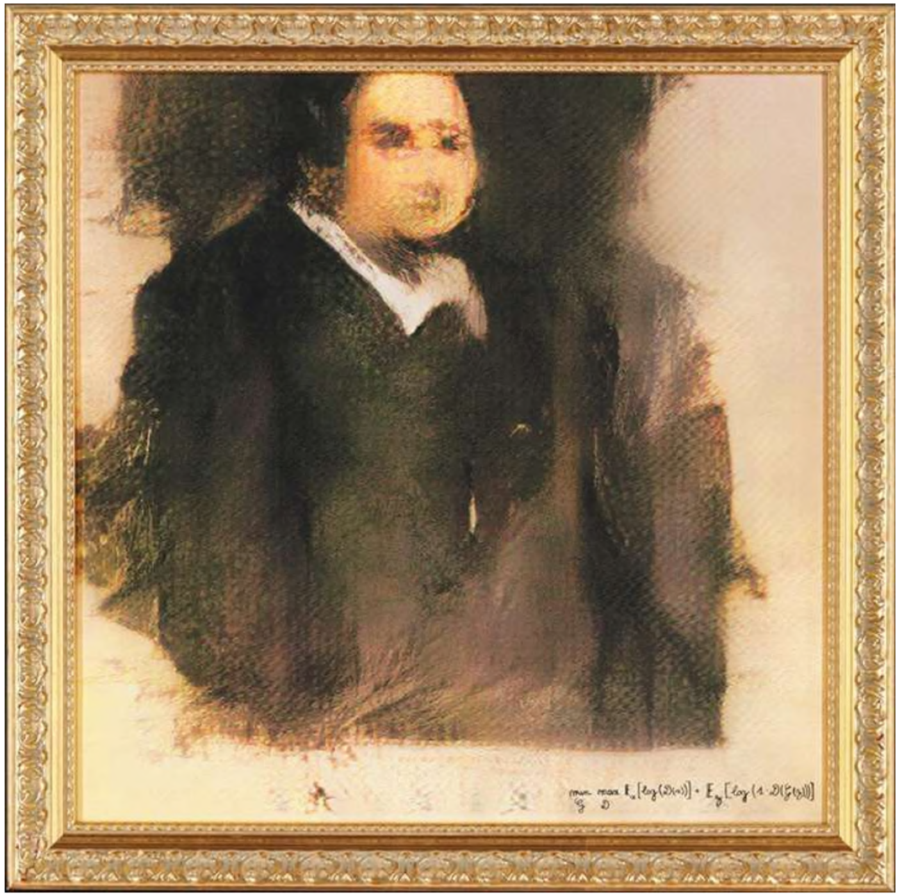
- Rules:
  - Every new song interpolated from existing YΔCHT melodies
  - 4 measures-long loops
  - Cannot add any note, harmony
  - Only substractive or transpositional changes
  - Structure and collage allowed
  - Assignment (to vocal, bass line…)
- Human Production and Arrangements



https://www.youtube.com/watch?time_continue=1378&v=pM9u9xcM_cs&feature=emb_logo

Painting

- 26 October 2018, Christie's Auction, New York, US$ 432 500
- Edmond de Belamy, Obvious (Collective) 
- Created with Deep Learning (GAN)
- Trained with 15 000 paintings (XIV – XX centuries)



Hello World

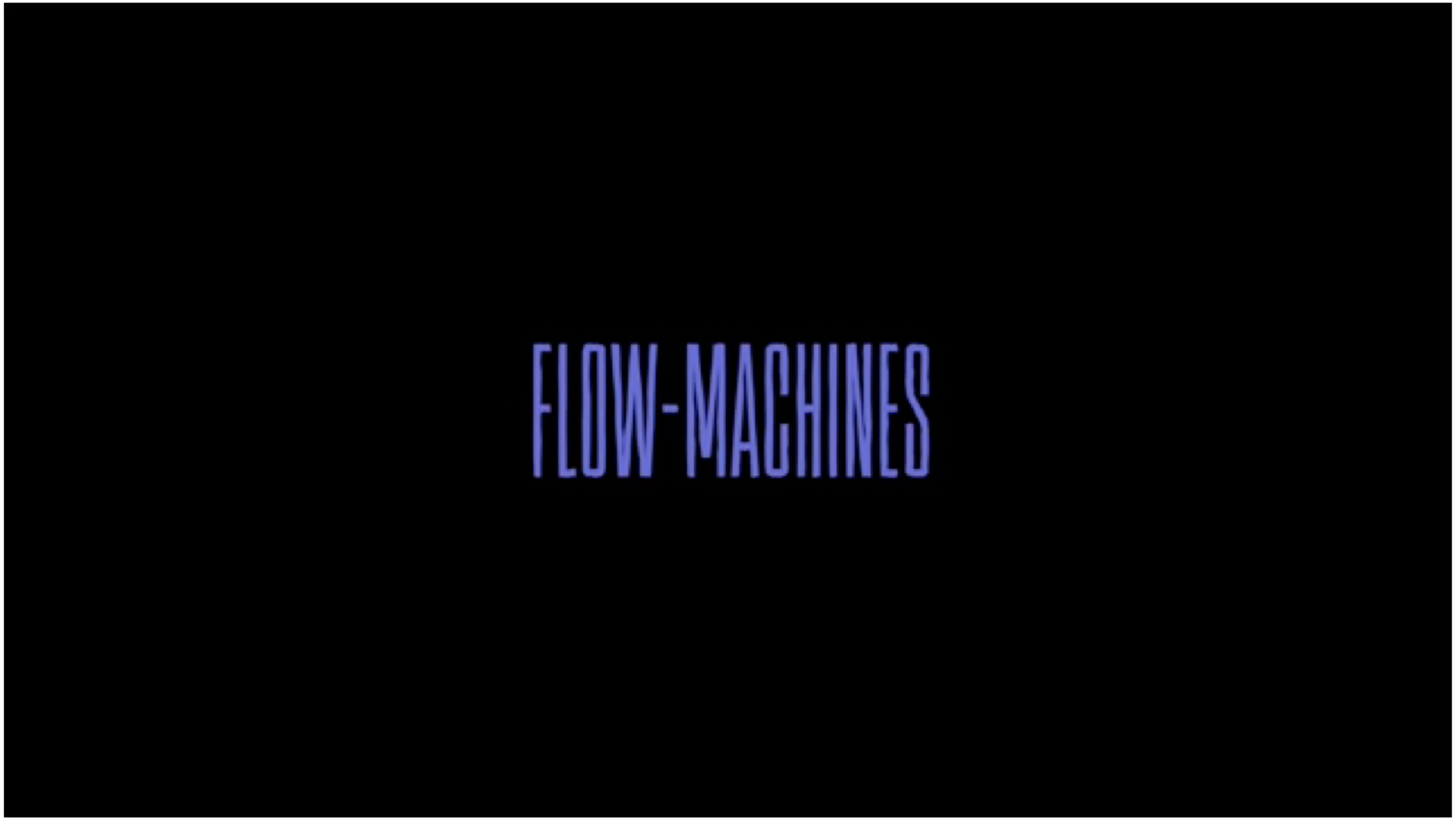
- January 2018, Hello World
- Created by Musicians (Musical Direction: Skygge – aka Benoît Carré)
- with FlowComposer [Pachet et al., 2014]
- ERC Project Flow Machines [Pachet et al., 2012-2017]
- Various Techniques (Markov Constraints, Rules, …)

European Research Council

https://www.youtube.com/watch?v=iuWYQe3aGlg

Hello World

- January 2018, Hello World, Flow Records
- Making Off



https://www.youtube.com/watch?v=yxTF-UFvoHU

"Beyond the Fence" Musical

- PropperWryter
- The Cloul Lyricist
- Folk-RNN
- Flow Machines

- Arts Theater, London, February-March 2016

https://www.youtube.com/watch?v=lzeSDIoI-7I

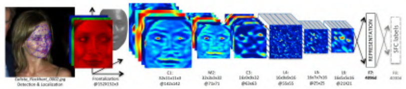
https://www.youtube.com/watch?time_continue=75&v=VZzl4sfCFjc

Deep Learning

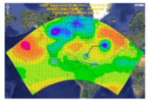
Deep Learning

- Boom Since 2012 (Imagenet Breakthrough)

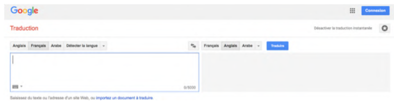

- Image Recognition



- Weather Prediction



- Translation

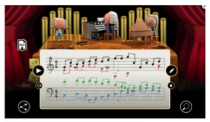


- Speech Recognition

- Speech Synthesis

- Source Separation

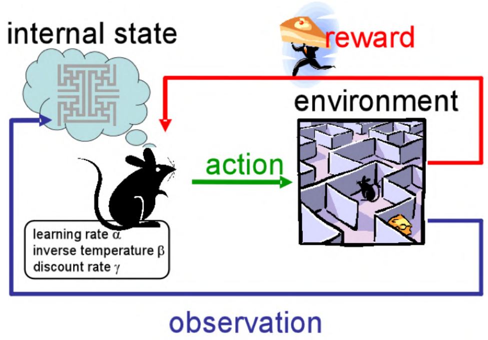
- Music Creation



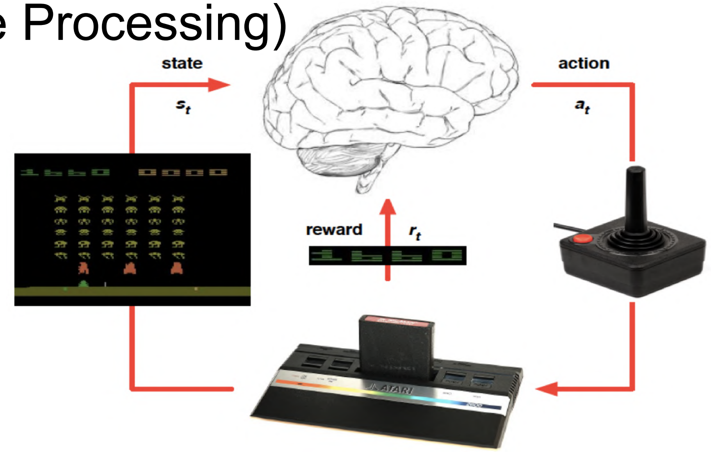
- Image Creation

Deep Reinforcement Learning [Silver et al., 2013]

- Deep Learning improves Other Machine Learning Paradigm Implementation:
  - Reinforcement Learning



- Deep Reinforcement Learning
  - Efficient Estimation of Gain (Q-Learning Q-Table)
  - Massive Simulation/Evaluation (Massive Processing)
  - Replay Mechanism (Massive Memory)

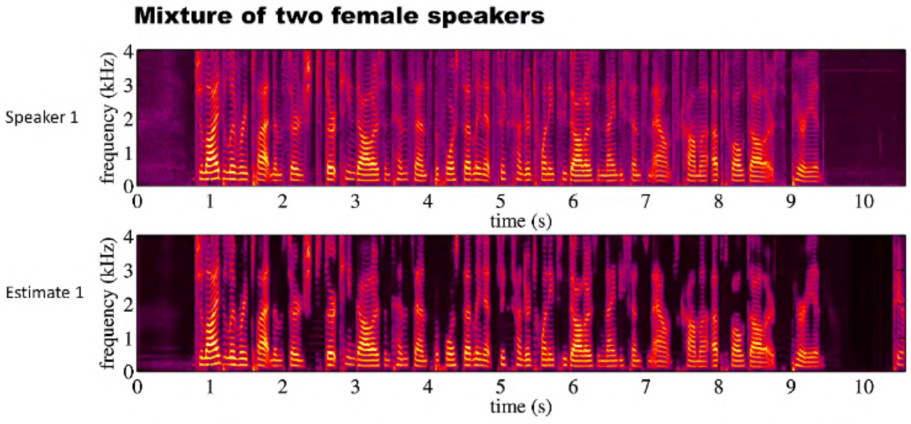


- First Application: Atari Games

- Second Application: Go
  - Alpha Go, AlphaZero Go

Speech/Music Separation

- Long Time Very Hard Problem, Now Resolved
- Cocktail Effect Voice Separation



Mitsubishi Electric Research Labs (MERL)
Publicado em 1 de ago de 2016

https://www.youtube.com/watch?v=vW51cG1Ox98

- Music/Voice Separation

deep neural networks for music source separation

deep neural networks for music source separation

Hexa (Namju) Kim
11 inscritos

This work is from Jeju Machine Learning Camp 2017

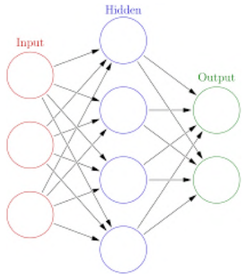
https://www.youtube.com/watch?time_continue=2&v=Cx7Me0Ayz1I

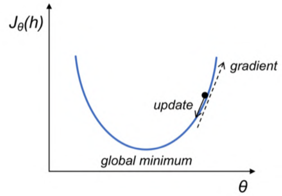
From Neural Networks to Deep Learning

Deep Learning

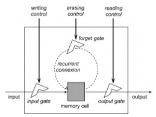
- Overwhelming Success

- Simple Basic Receipt
  - Linear/Logistic Regression
  - Loss Function Minimization





- Technical Improvements (since First Neural Networks)
  - Backpropagation, LSTM, Batch Normalization…
  - Loss Function Wide Application
    - » Meta-Level, ex: LSTM
    - » Constraints, ex: VAE

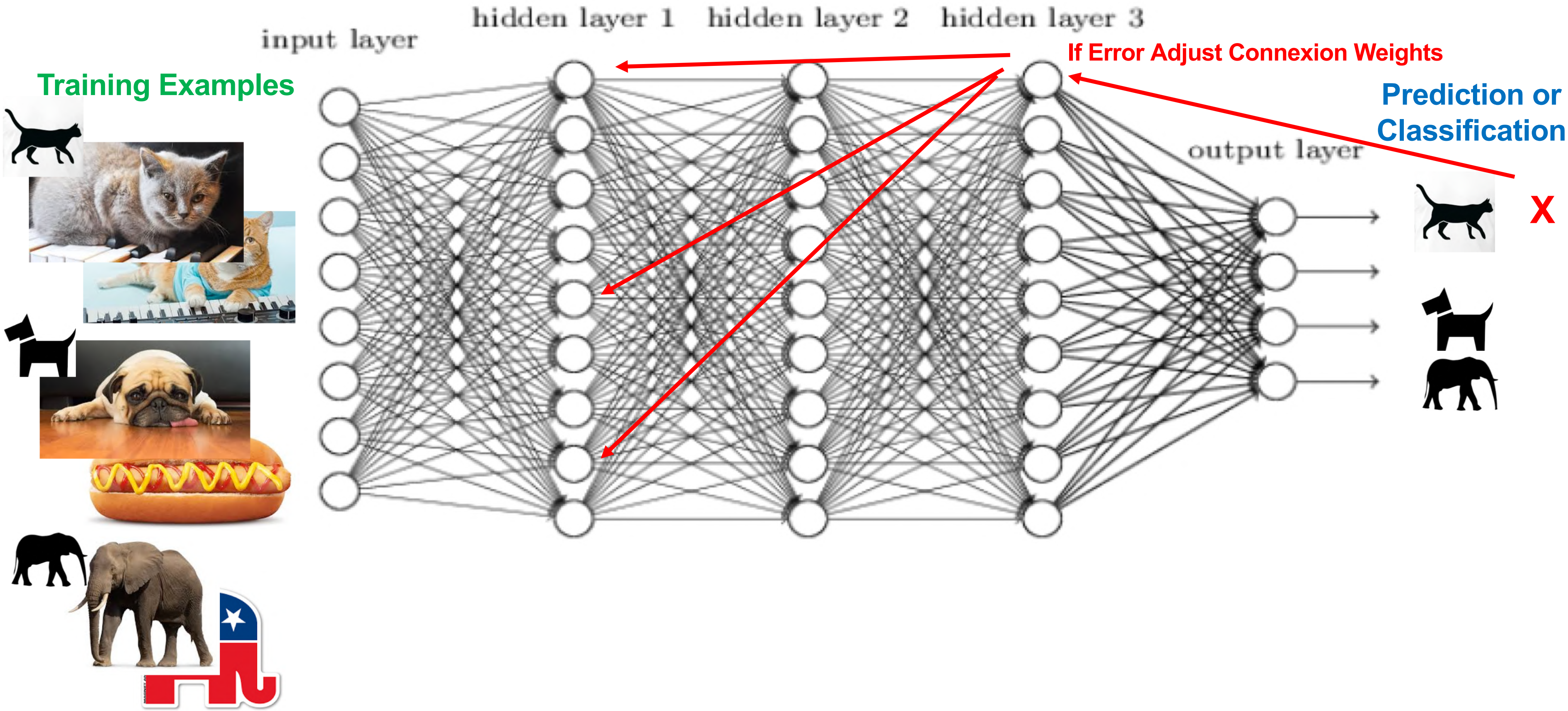


  - Optimized Implementations/Platforms
- Scale+
  - CPU

  - Data

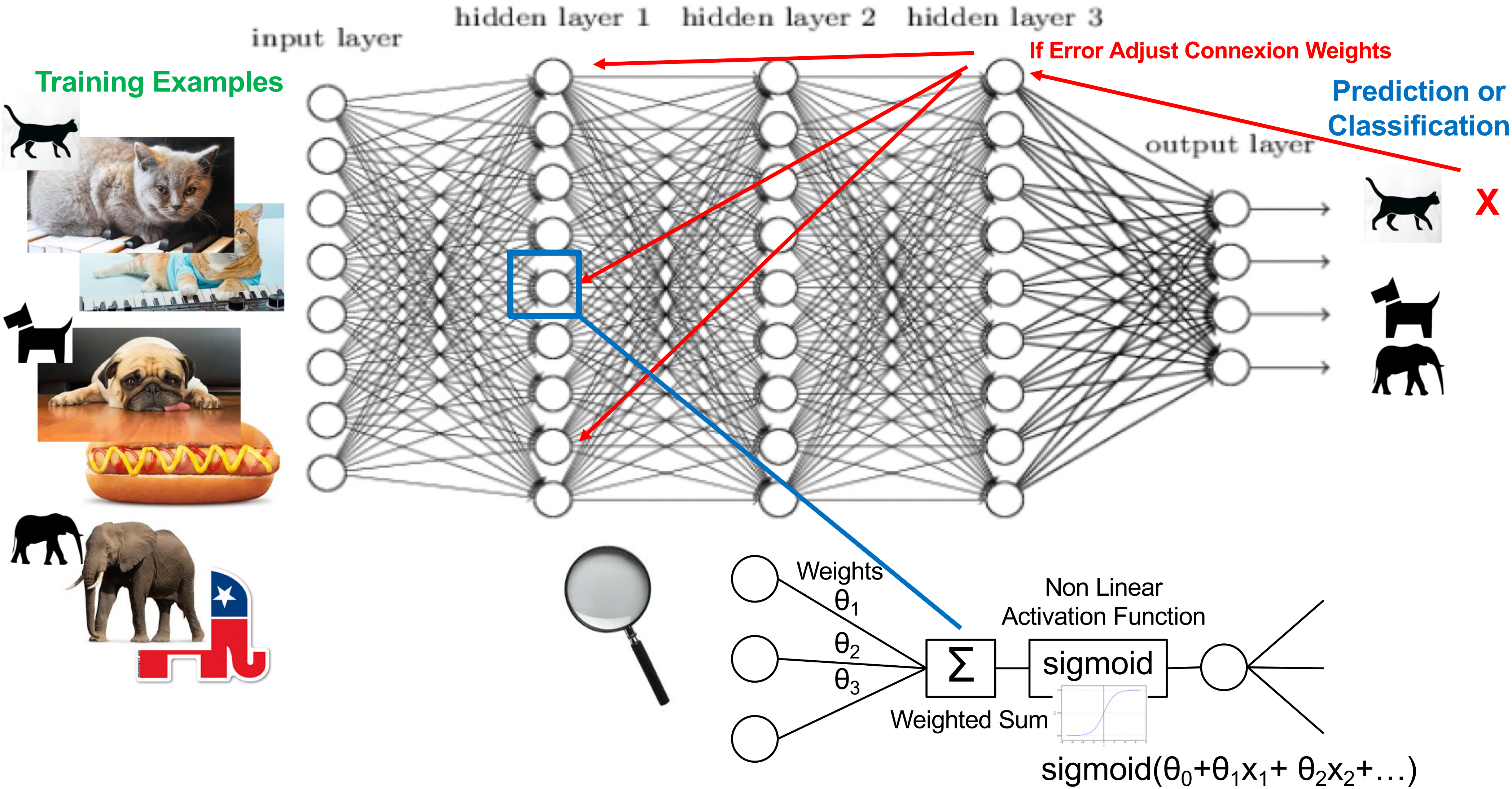
Neural Networks in One Slide

Principle – Error Prediction/Classification Feedback



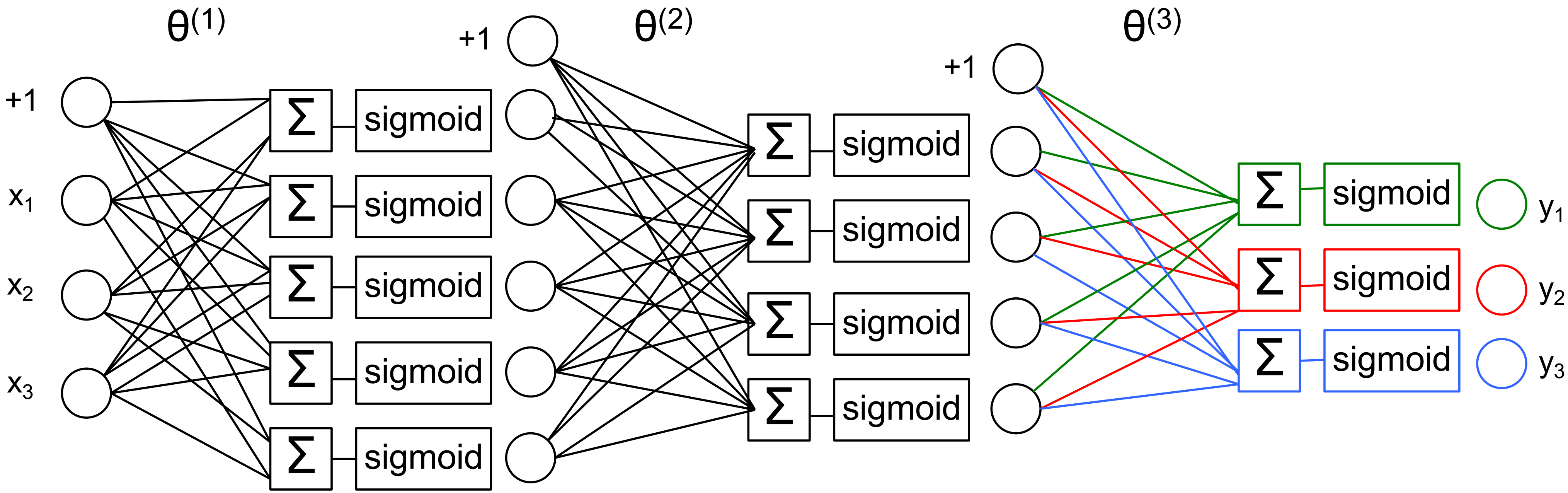
Neural Networks in ~~One~~ Two Slides

Principle – Error Prediction/Classification Feedback



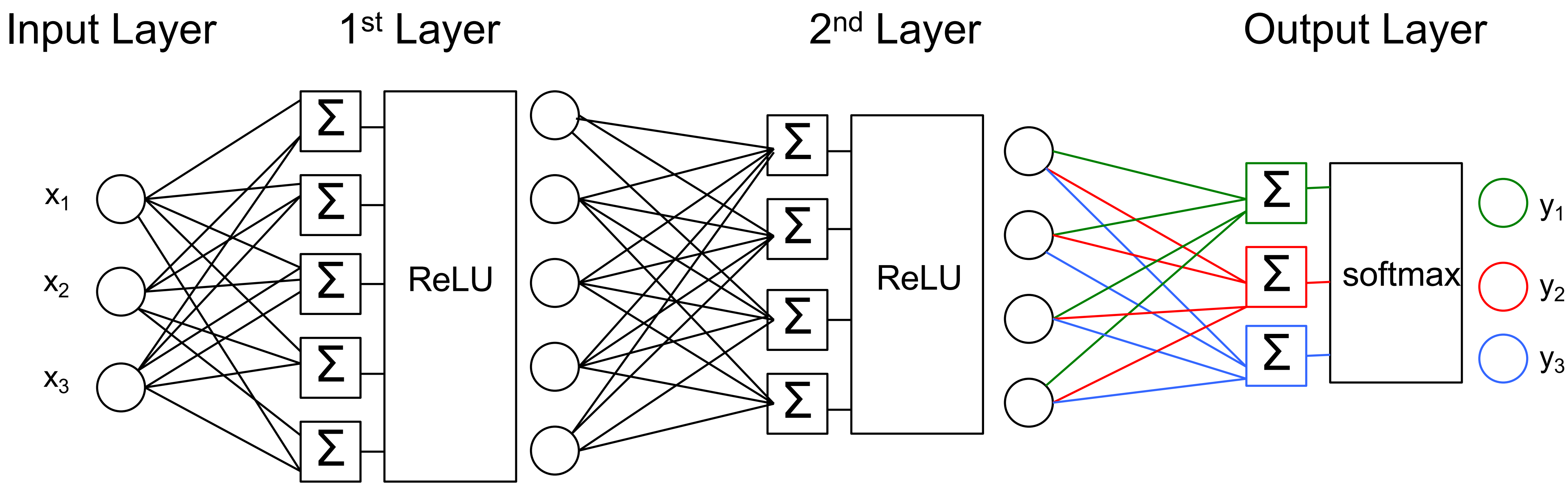
Foundation

- Neural Network =
- Successive Layers of Logistic Regression =
- Successive Layers of Linear Regression + Non Linear Activation Function

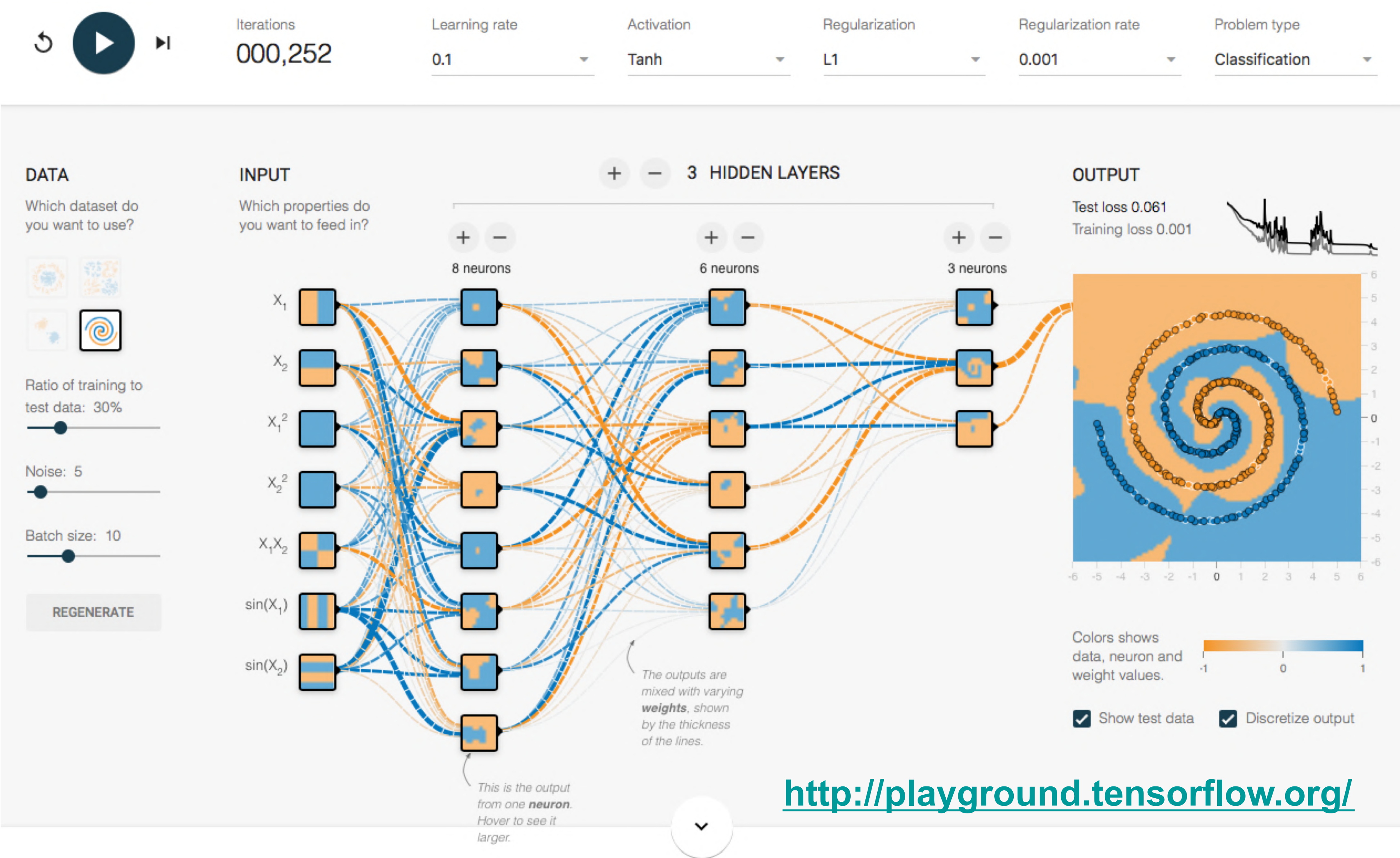


Foundation

- Neural Network =
- Successive Layers of Logistic Regression =
- Successive Layers of Linear Regression + Non Linear Activation Function

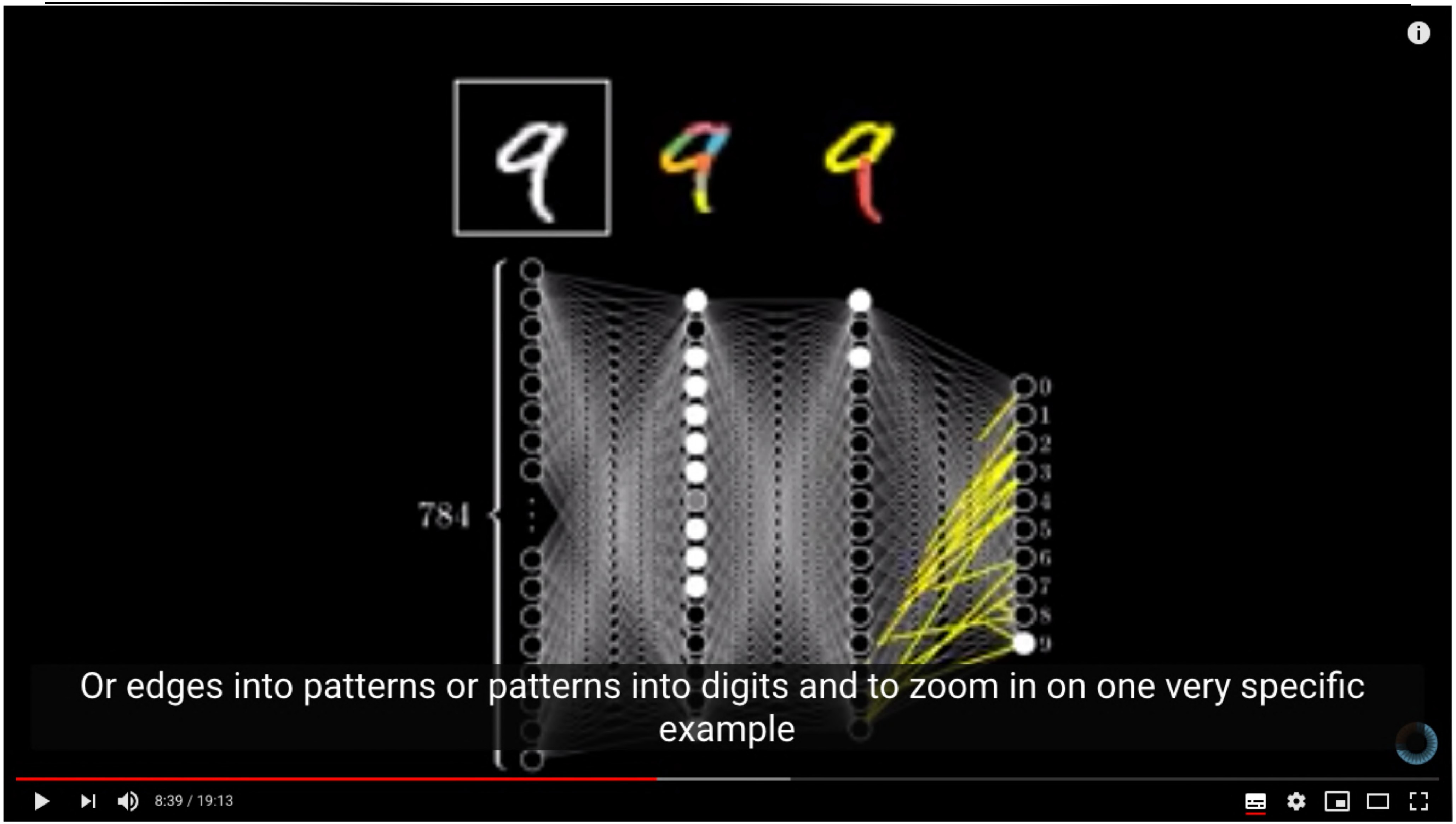


Example (TensorFlow PlayGround)



http://playground.tensorflow.org/

Nice Animation [3Blue1Brown]

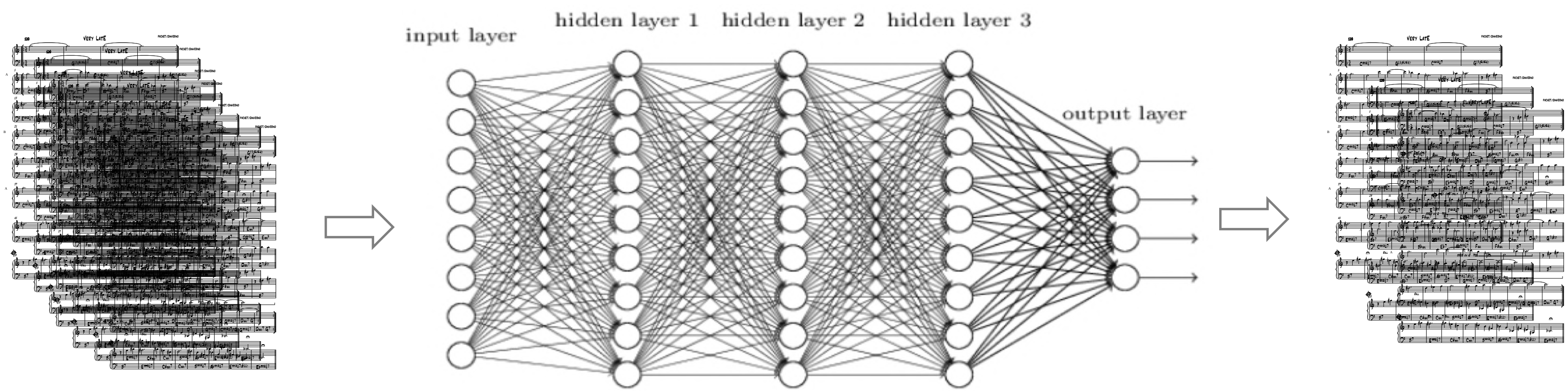


https://www.youtube.com/watch?v=aircAruvnKk

A First Example of Music Generation

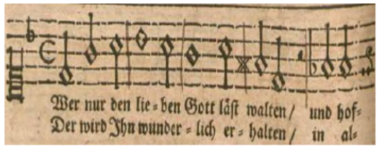
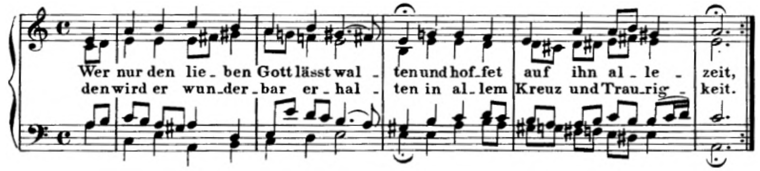
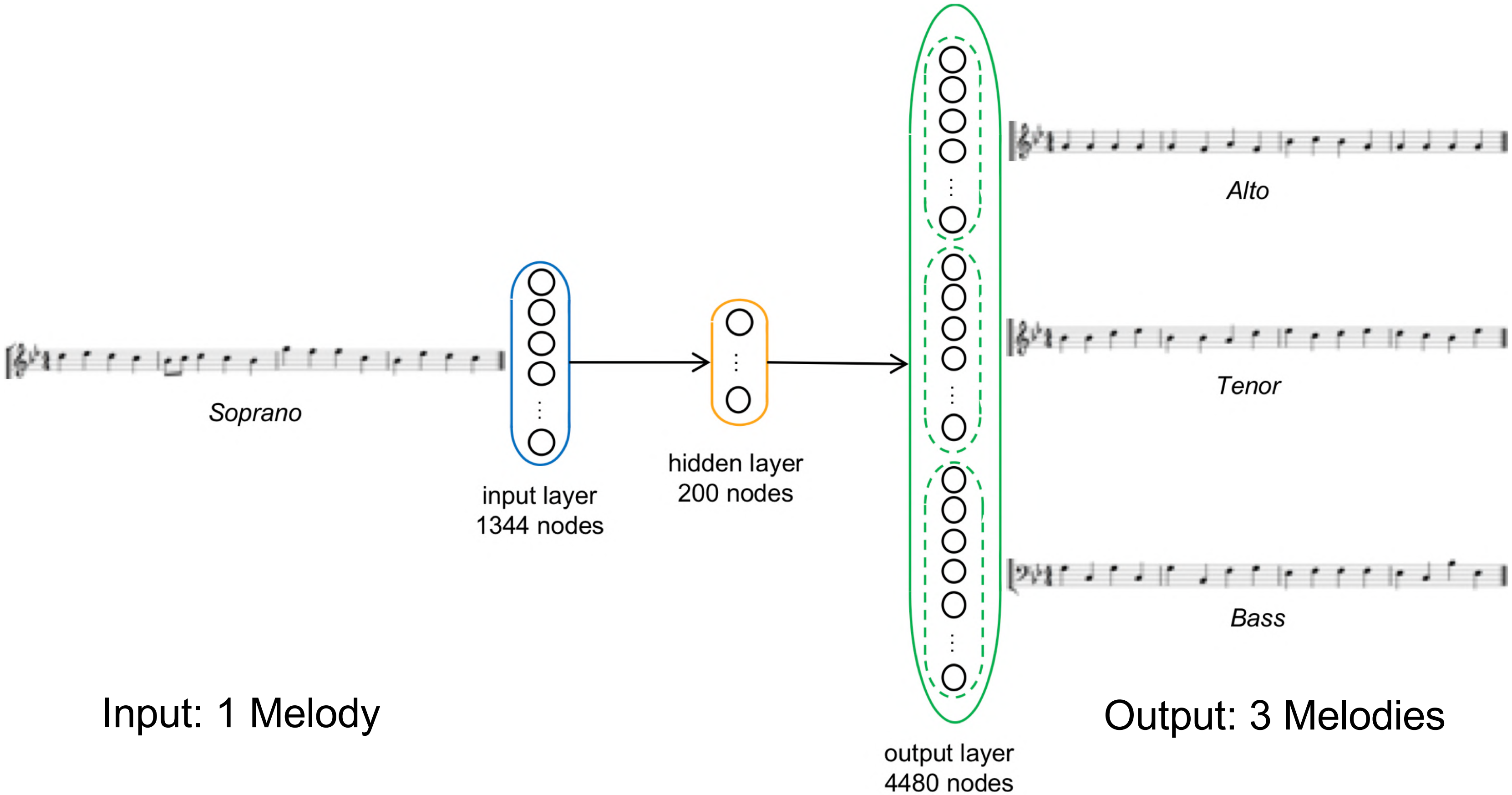
Artistic Content Generation Basic Cycle

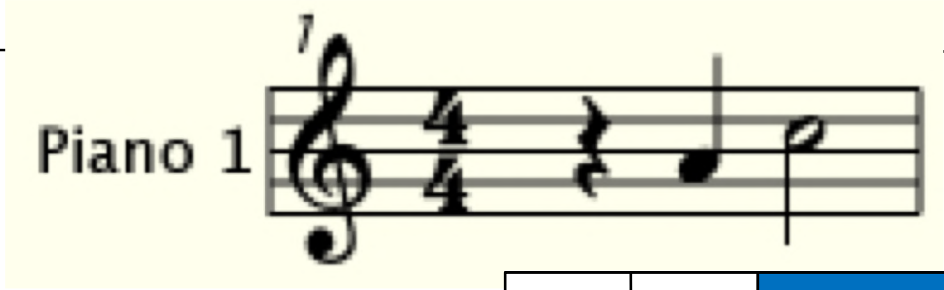
- Curation
  - Collecting Examples (Training Set)
  - *Extensional Definition* of the Style
- Configuration
  - of the (Selected) Learning Model/Architecture
- Selection
  - Among Results Generated

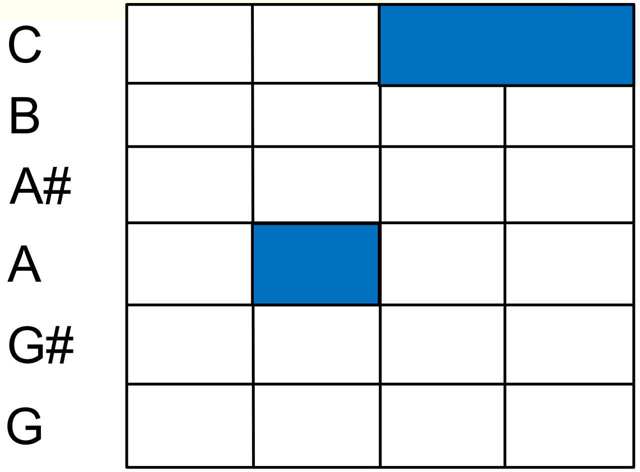


Curation Configuration Selection

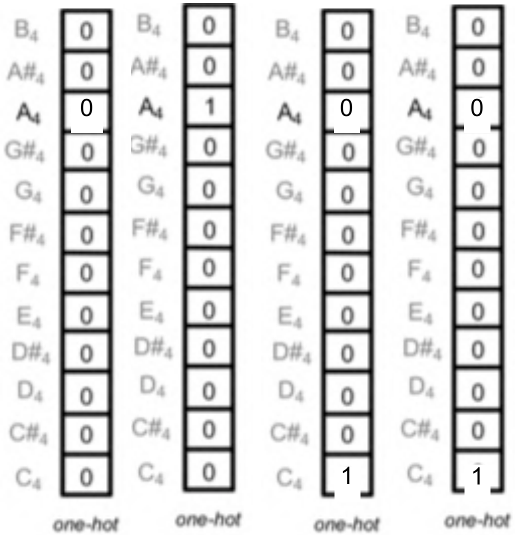
Neural Network Direct Application

- Feedorward Architecture
- Classification Task (What Notes)
- Counterpoint (Chorale) Generation
- Training on the Set of (389) J. S. Bach Chorales (Choral Gesang)



Representation

Score

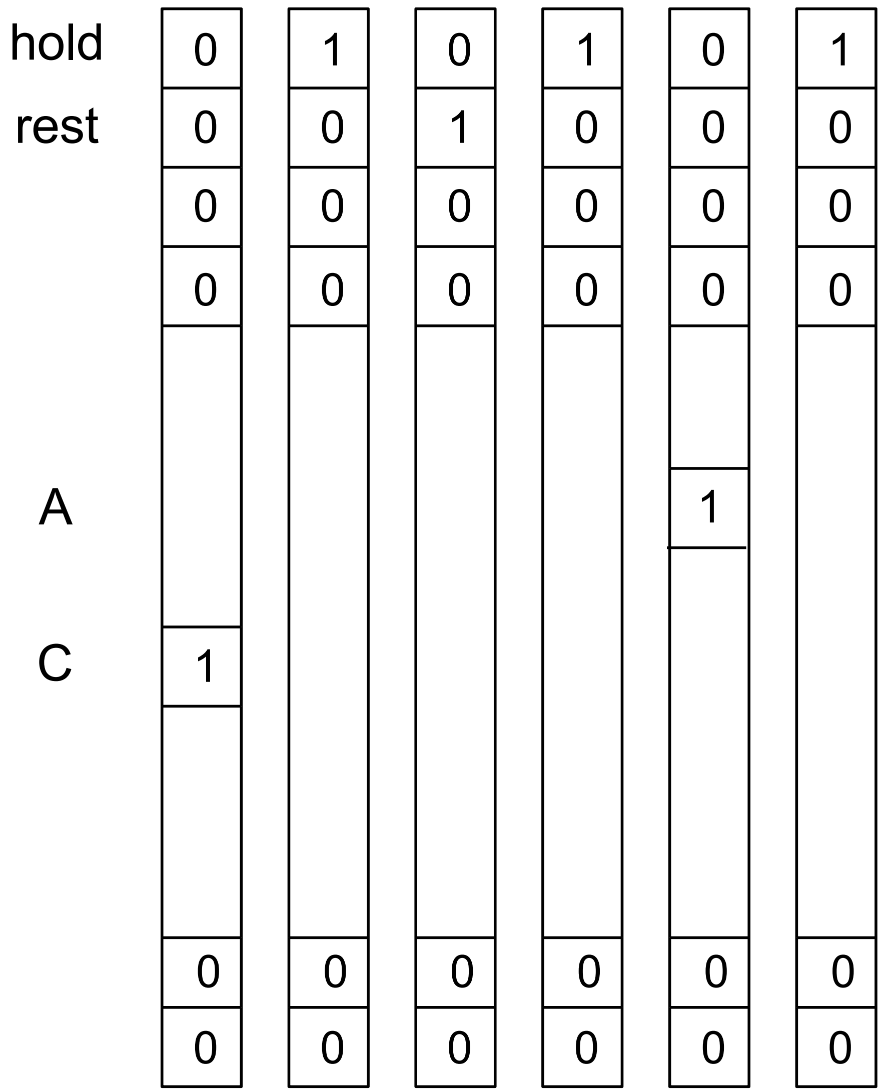


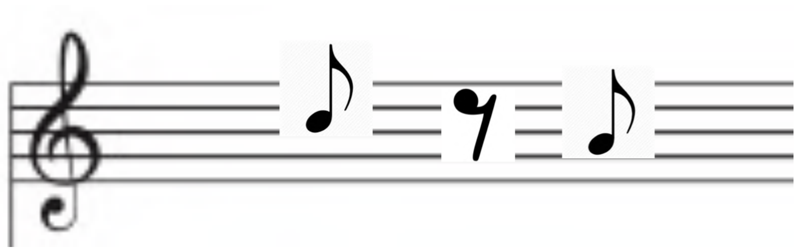
Piano Roll

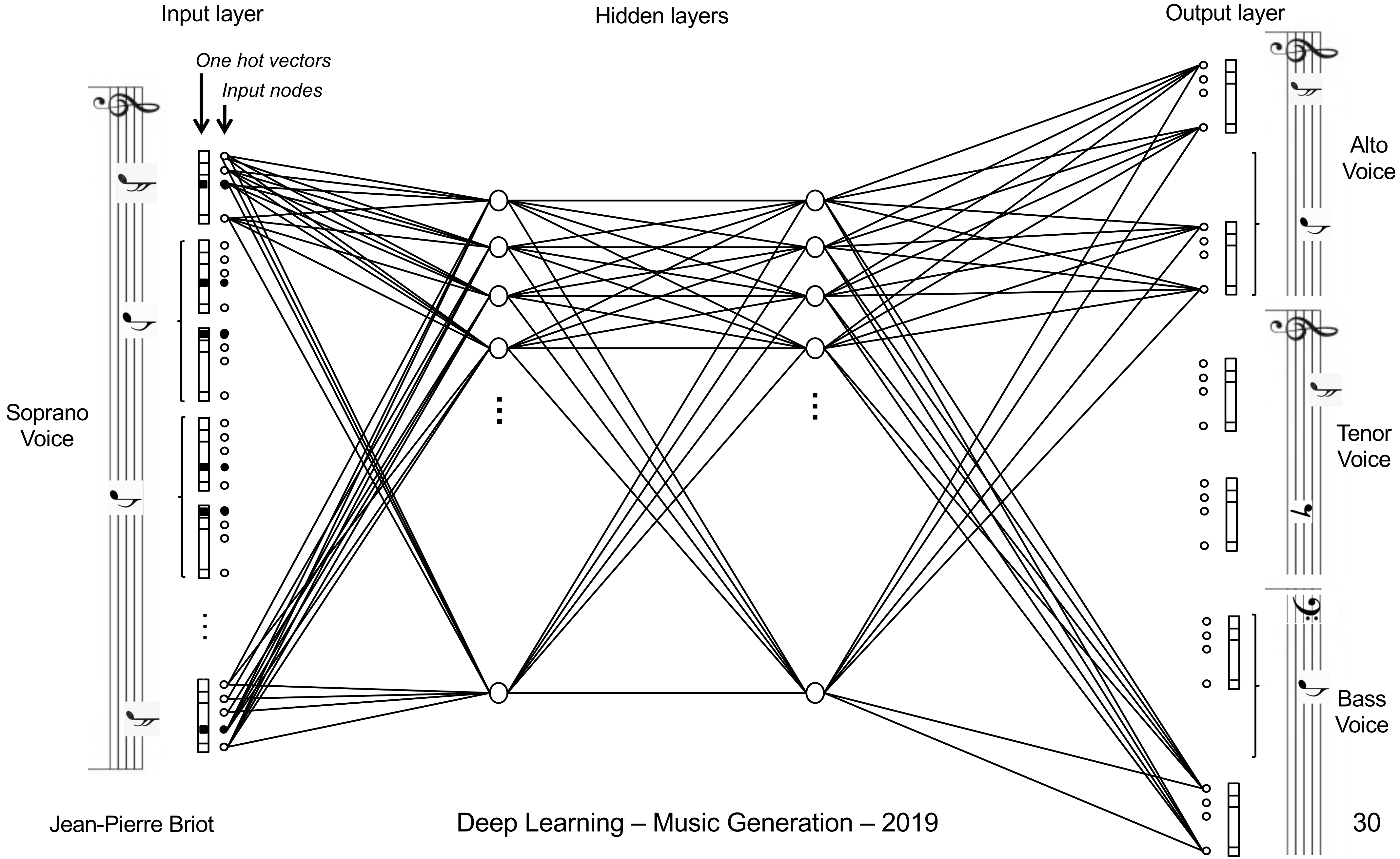


One hot Encoding

Representation

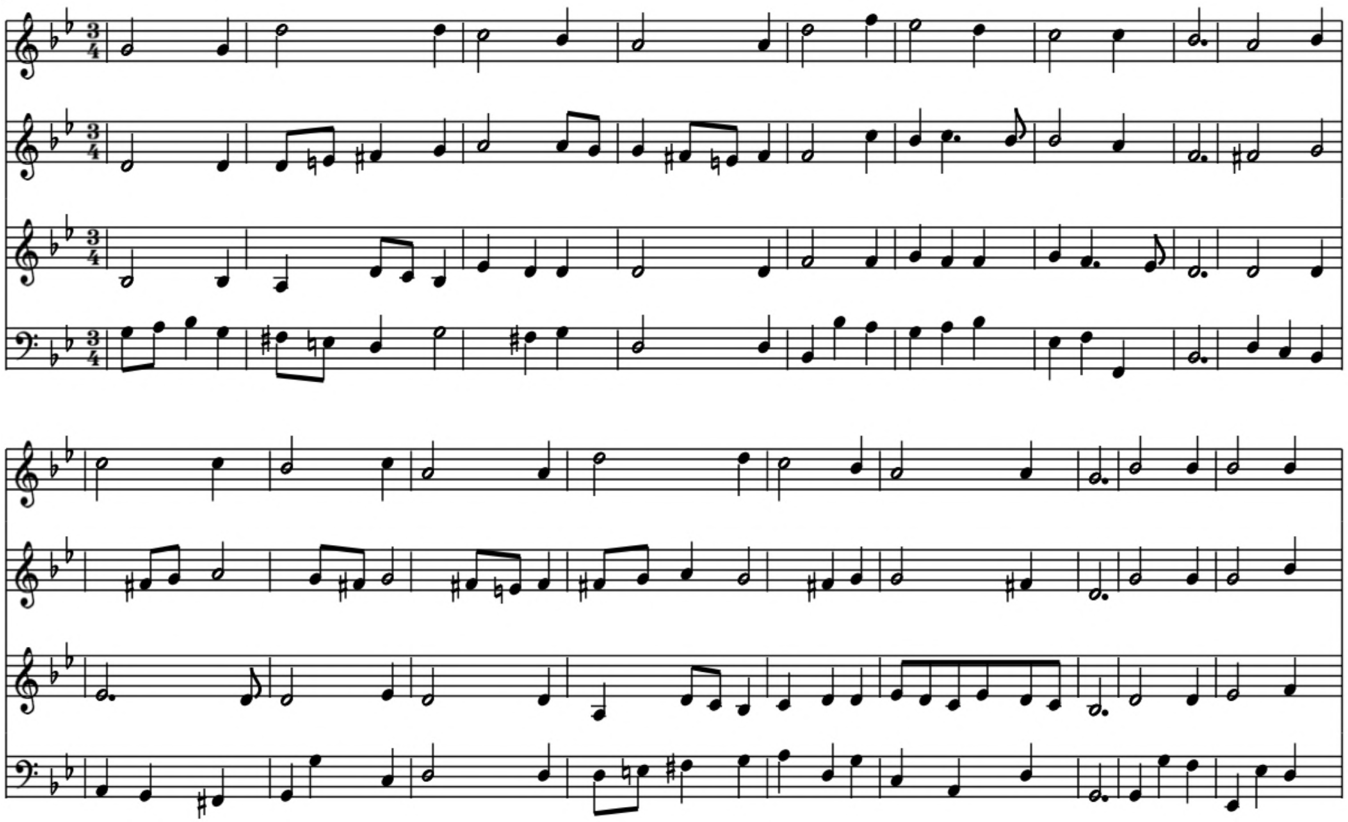
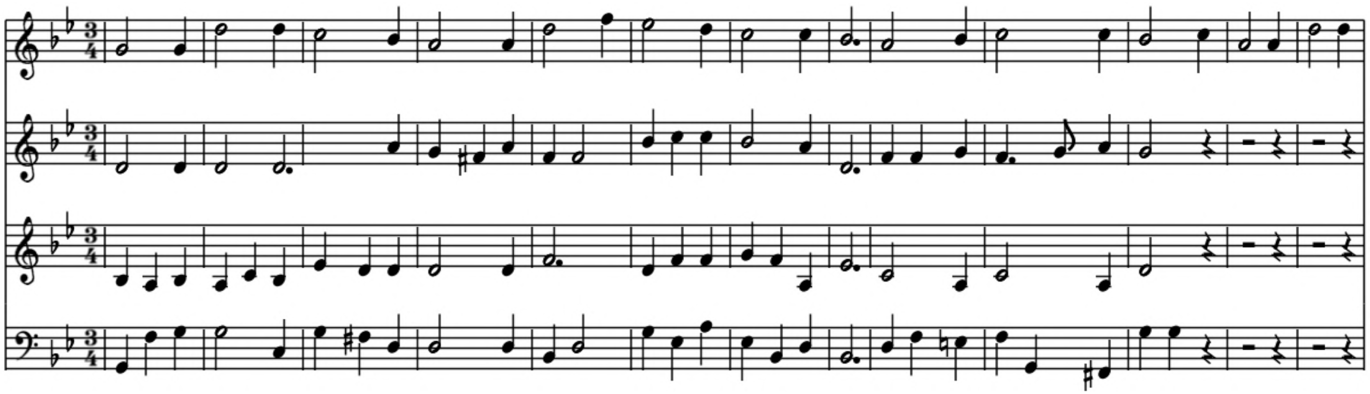
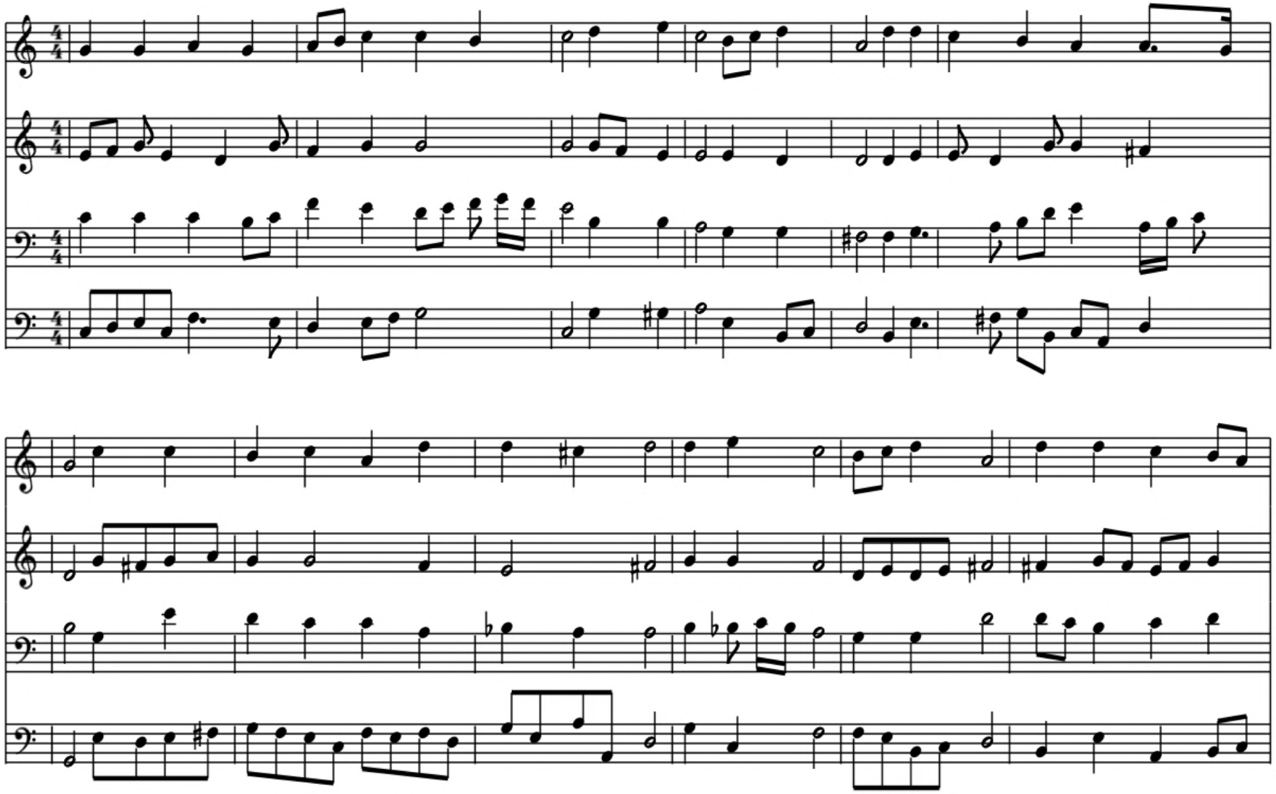


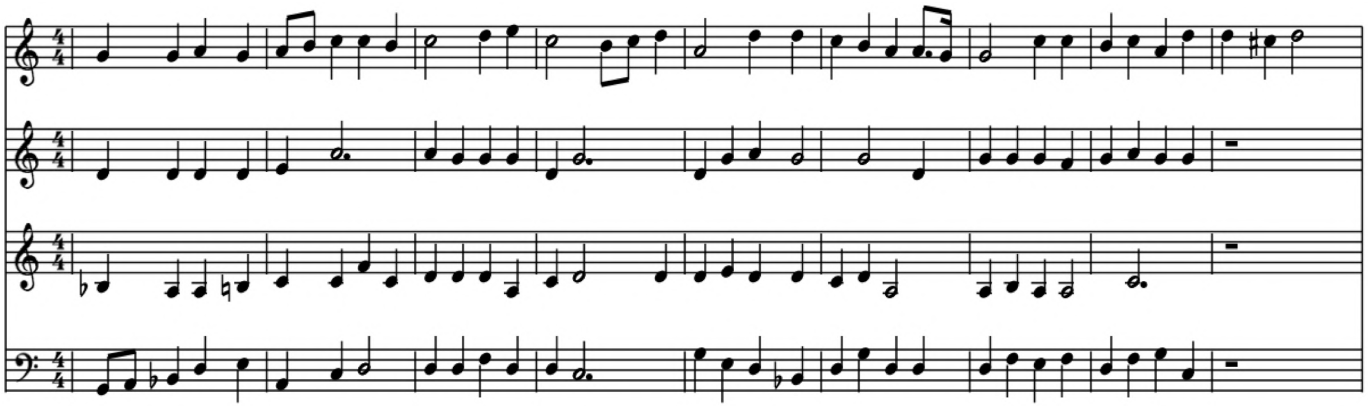
If time slice = sixteenth 

Music / Representation / Network

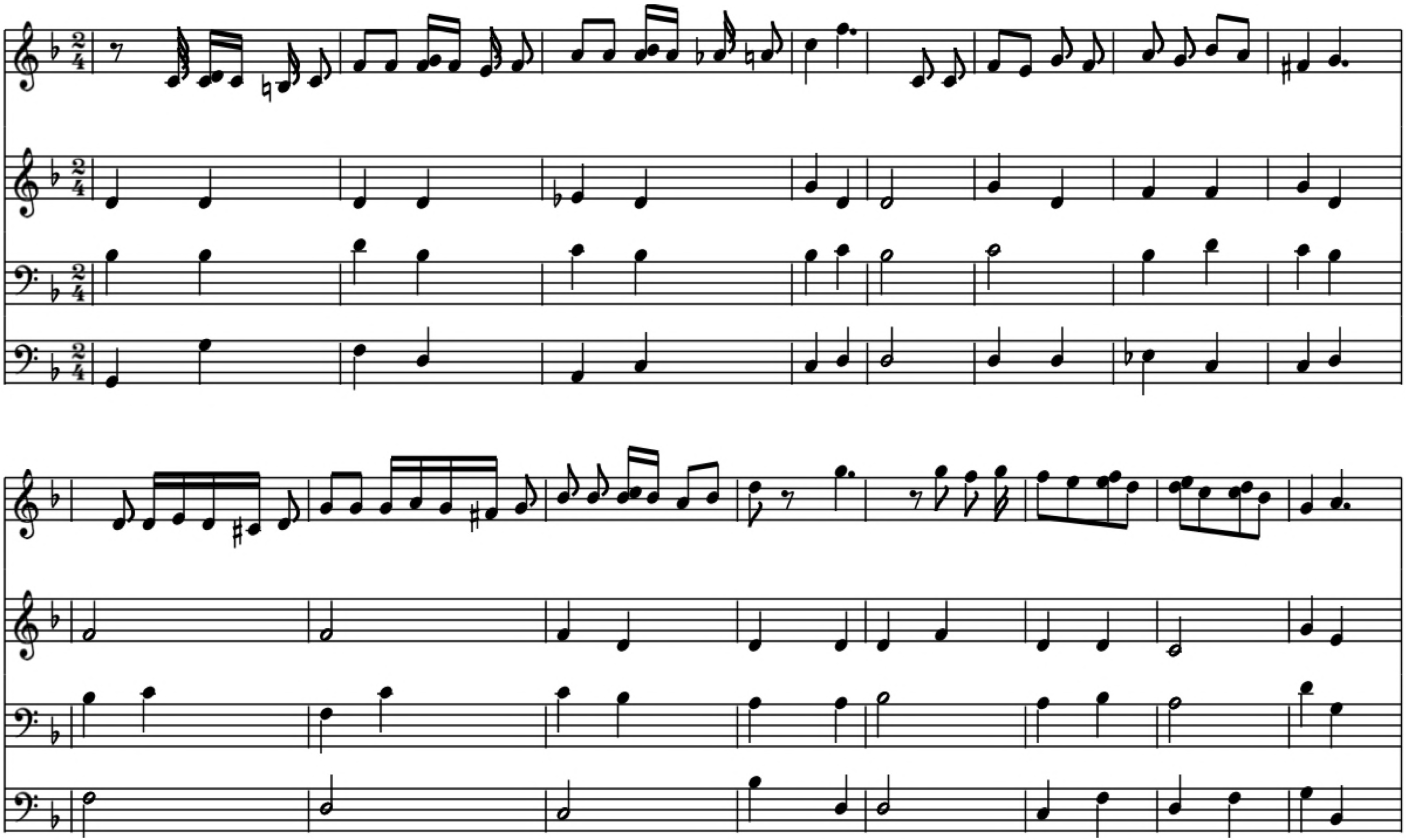


ForwardBach

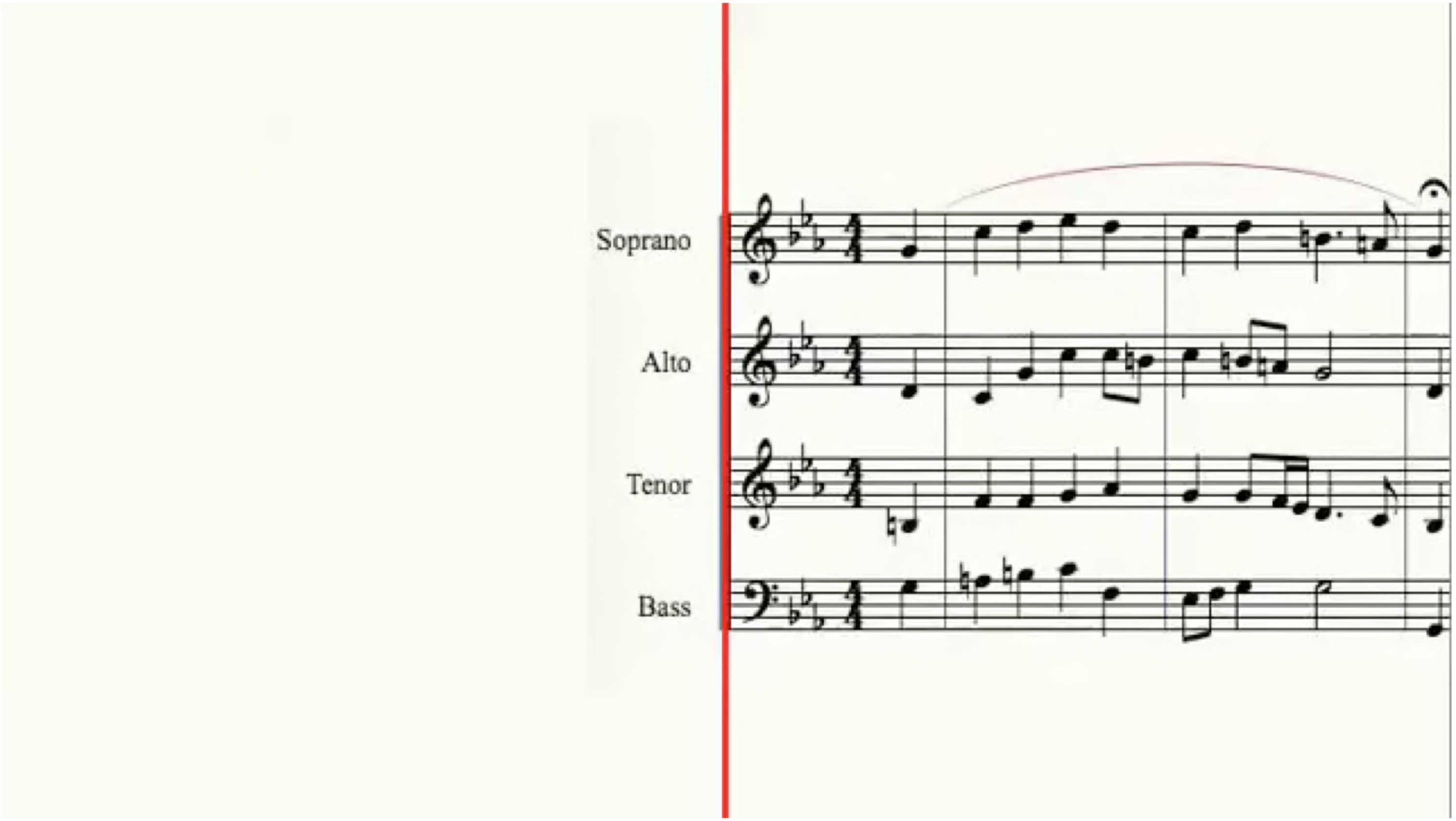
Bach BWV 344 Chorale
(Training Example)

Original

Regenerated

ForwardBach

Bach BWV 423 Chorale
(Test Example)

Original

Regenerated

ForwardBach Brazilian Hymn Counterpoint

Bach Chorales

- December 2016, DeepBach, Gaëtan Hadjeres
- Deep Learning
- Training Set = 352 Chorales

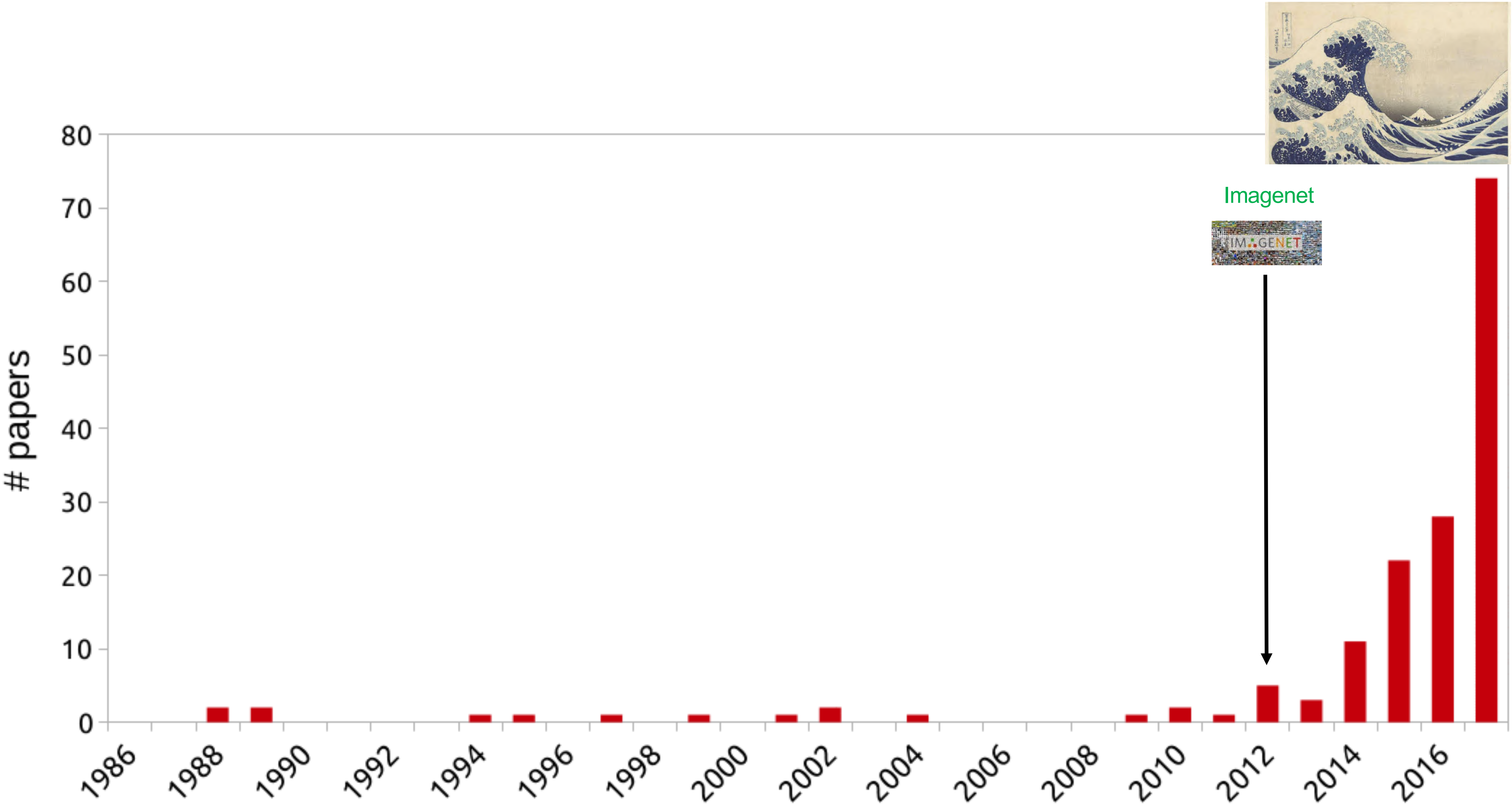


https://www.youtube.com/watch?v=QiBM7-5hA6o

Reorchestration of God Save the Queen by DeepBach [Hadjeres, 2017]

History of Neural Network-based Music Generation

Number of Scientific Papers about Neural Networks and Music (Generation, Classification…) [Pons, 2018]



[Pons, 2018]

#Citations

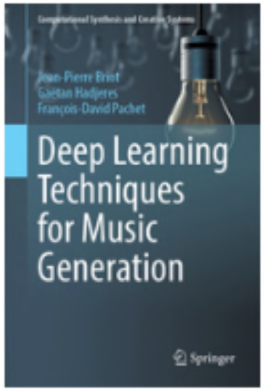
Google Scholar

Samedi 19 octobre 2019

	#Citations	Year
Deep learning techniques for music generation-a survey JP Briot, G Hadjeres, F Pachet arXiv preprint arXiv:1709.01620	~~85~~ **105**	2017

» Computer Science » Artificial Intelligence

Computational Synthesis and Creative Systems

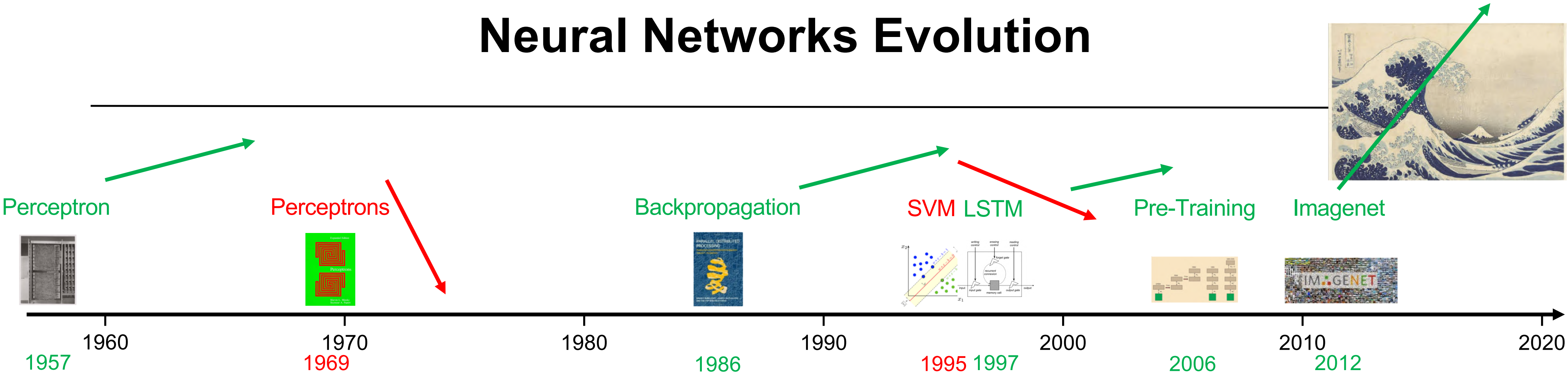


© 2019

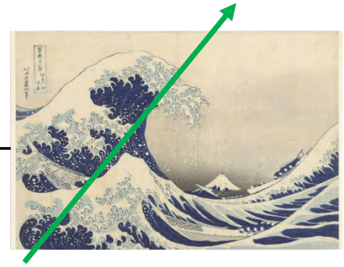
Deep Learning Techniques for Music Generation

Authors: **Briot**, Jean-Pierre, **Hadjeres**, Gaëtan, **Pachet**, François-David

Neural Networks Evolution

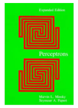


Neural Networks 4 Music Generation Evolution

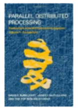


Perceptron

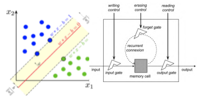
Perceptrons



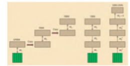
Backpropagation



SVM LSTM



Pre-Training



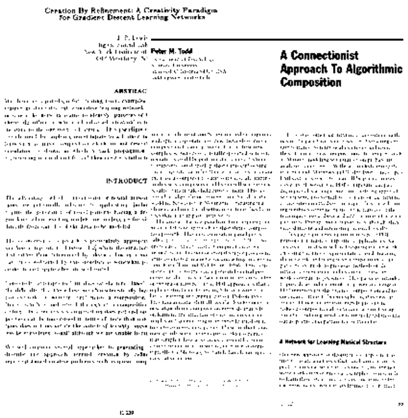
Imagenet

1960 1970 1980 1990 2000 2010 2020

1957 1969 1986 1988 1989 1995 1997 2002 2004 2006 2012 2016

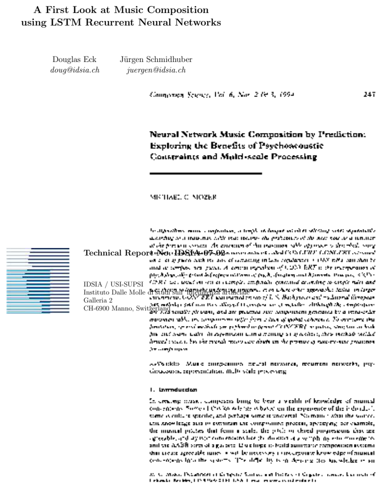
Creation by Refinement

Sequential network



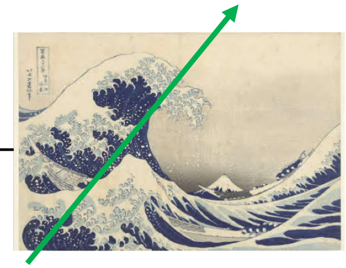
LSTM Blues

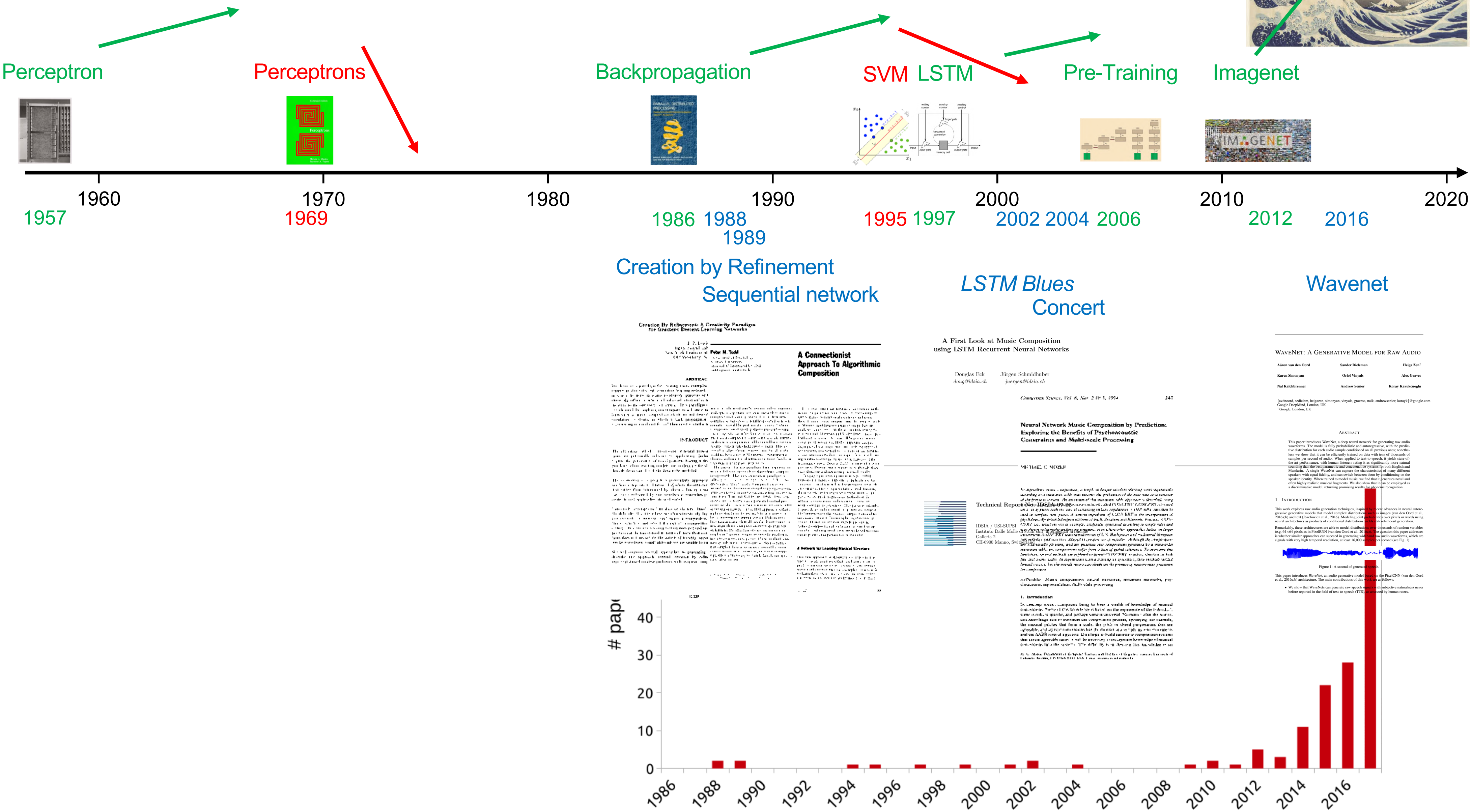
Concert



Wavenet

Neural Networks 4 Music Generation Evolution

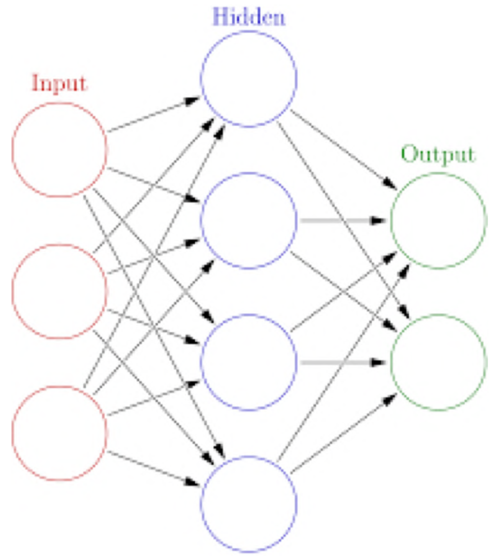




The Old Emperor Old Clothes

The Old Emperor Old Clothes (Neural Networks)

- Single Hidden Layer Neural Network
- Hand Made
- Technical Limitations
- Slow CPU
- Small memory
- Few Examples



First Experiments in Using Artificial Neural Networks for Music Generation

1988–1989

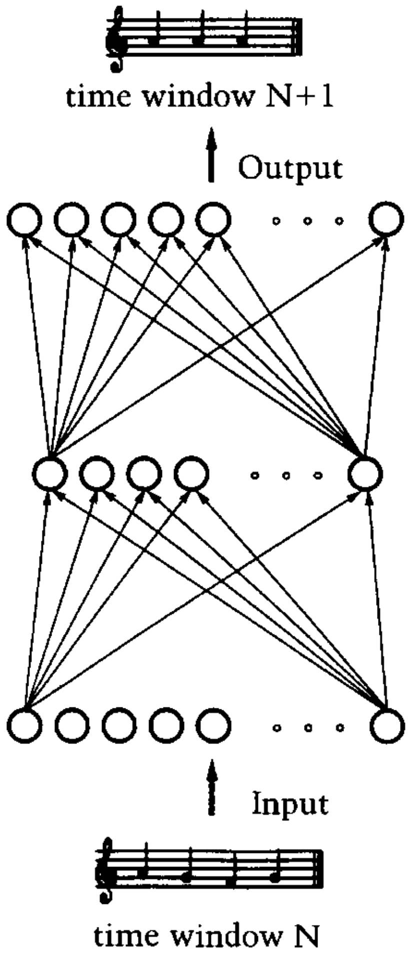
- Lewis, J. P., Creation by Refinement: A Creativity Paradigm for Gradient Descent Learning Networks, International Conference on Neural Networks, San Diego, CA, USA, July 1988, pp. II-229–233.
- Todd, Peter M., A Sequential Network Design for Musical Applications, Proceedings of the 1988 Connectionist Models Summer School, CMU, June 1988, Touretsky, D., Hinton, G., Sejnowski, T. (eds), Morgan Kaufmann, pp. 76–84, 1989.
- Todd, Peter M., A Connectionist Approach to Algorithmic Composition, Computer Music Journal (CMJ), MIT Press, 13(4):27–43, 1989.

2004

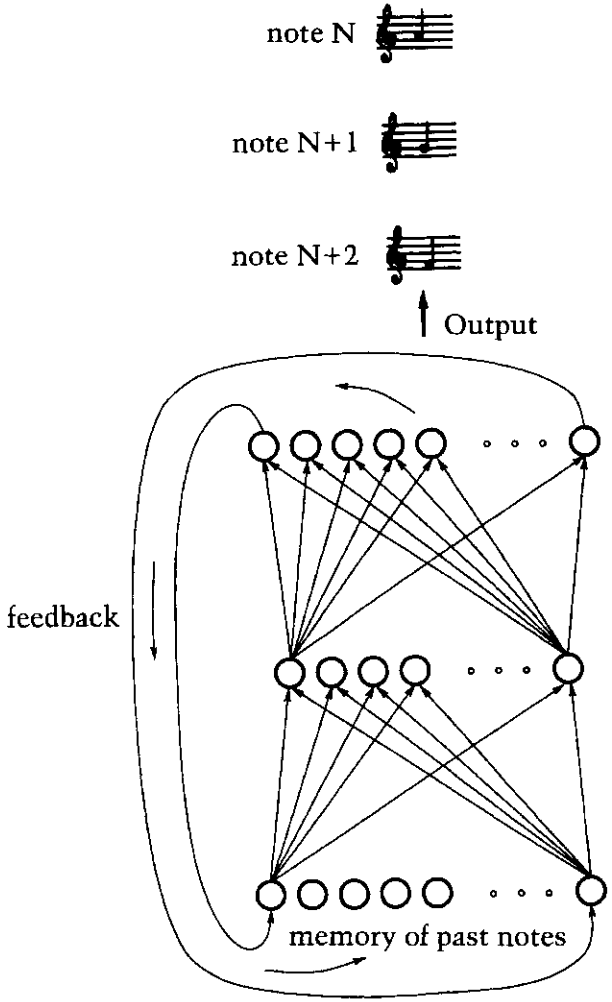
- Mozer, M. C., Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing, Connection Science, 6(2&3):247–280, 1994

Todd's Architecture Variation [Todd, 1989]

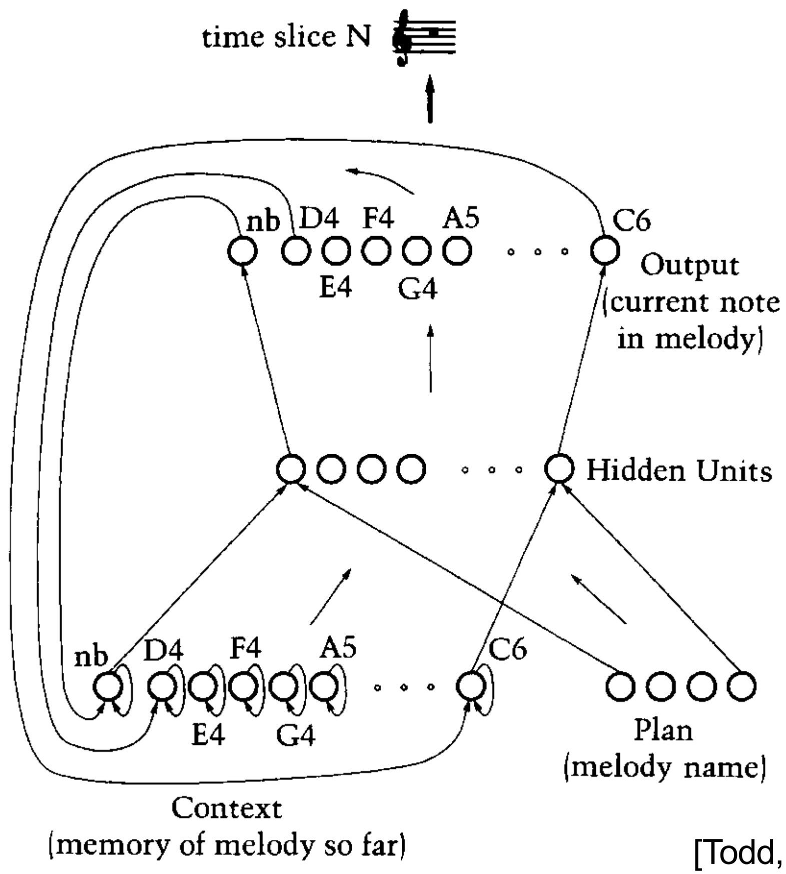
Feedforward architecture
Iterative generation



Recurrent architecture
Iterative generation



Recurrent + Conditioning architecture
Iterative generation



[Todd, 1988]

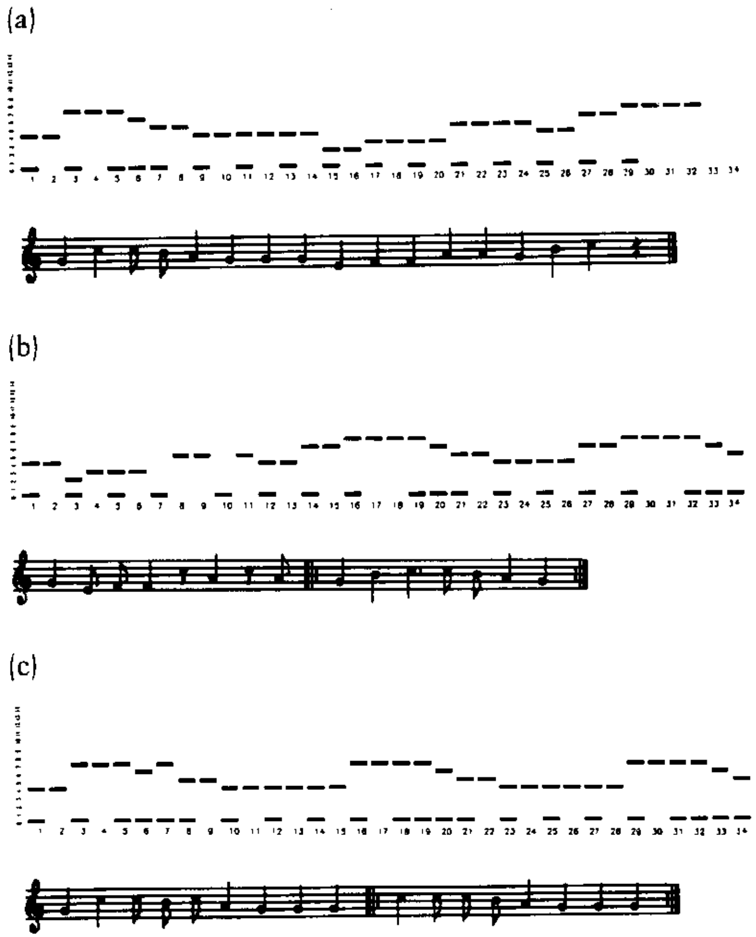
Todd’s Conditioned Generation

Extrapolation

Original melody (plan 1.0)

New melody (plan 0.0)

New melody (plan 2.0)

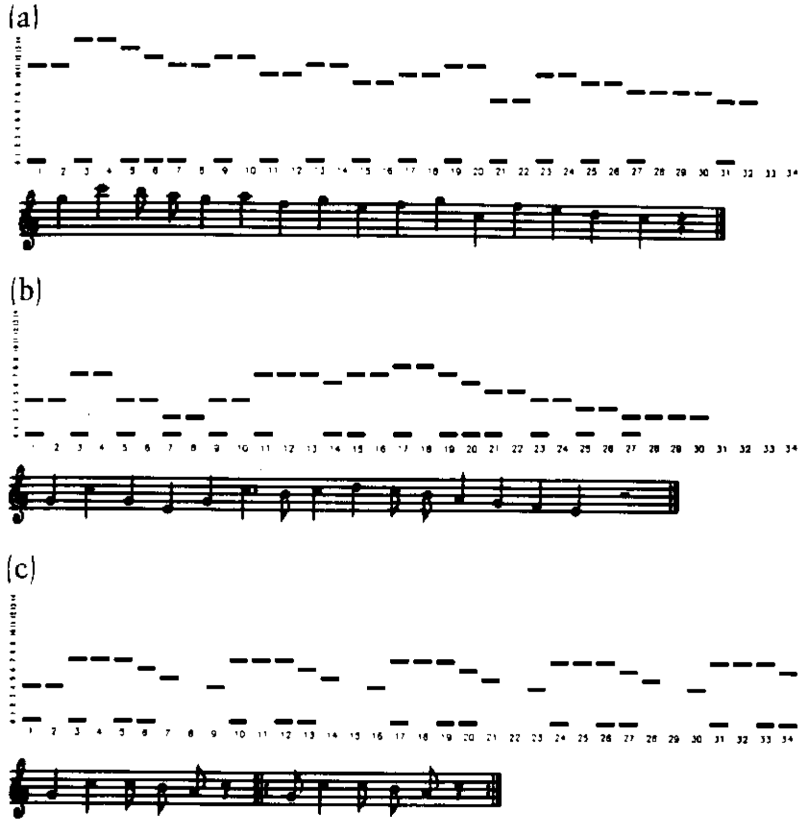


Interpolation

Original melody_A (plan 0.0)

Original melody_B (plan 1.0)

New $\text{melody}_{A\text{-}B}$ (plan 0.5)



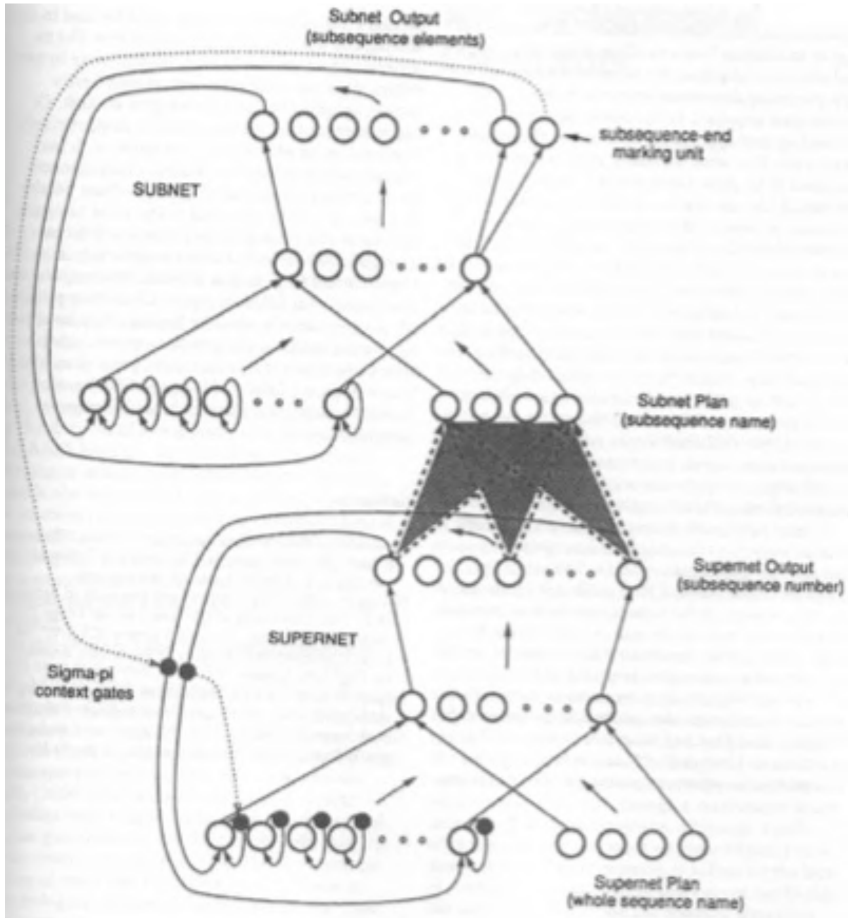
Todd's Architecture Prospects/Addendum (1/2) [Todd, 1989]

- Structure

One of the largest problems with this sequential network approach is the limited length of sequences that can be learned and the corresponding lack of global structure that new compositions exhibit. Hierarchically organized and connected sets of sequential networks hold promise for addressing these difficulties. Several ways of passing control back and forth between the interconnected networks will be described and the remaining issue of learning hierarchical structures will be addressed in this addendum.

- Hierarchy

One solution to these problems is first to take the sequence to be learned and divide it up into appropriate chunks (for instance, in the case of the sequence just presented, these could be A-B-C-D, E-E-E, A-B-C-D, and G-G). Next, train a sequential network to produce each of these subsequence chunks with a different plan. Finally, give this network the appropriate sequence of subsequence plans so that it will produce the chunks in the proper order to recreate the entire original pattern.



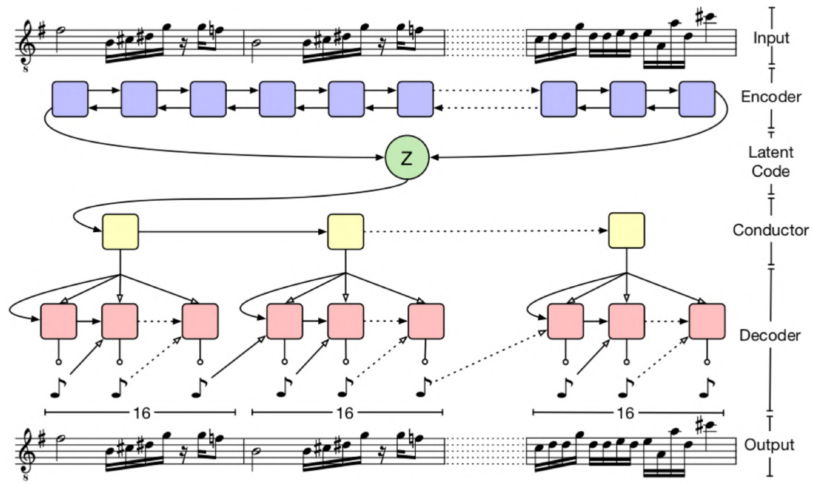
- Multiple Time/Clocks

Of course, one way to present this subsequence-generating network with the appropriate sequence of plans is to generate *those* by another sequential network, operating at a slower time scale. Then,

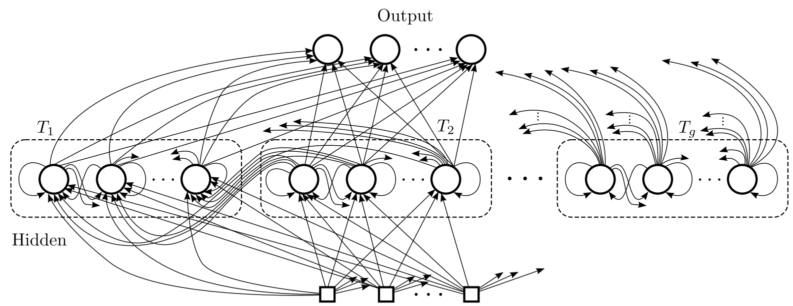
Todd's Architecture Prospects/Addendum (2/2) [Todd, 1989]

- Precursor of

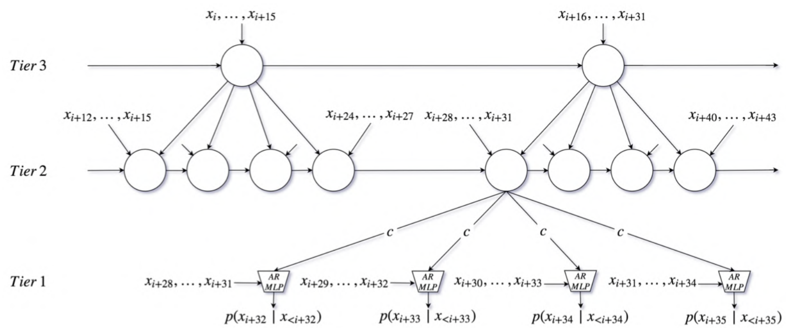
- Hierarchy
  - Ex: MusicVAE [Roberts et al., 2018]



- Multiple Time/Clocks
  - Ex: Clockwork RNN [Koutnik et al., 2014]

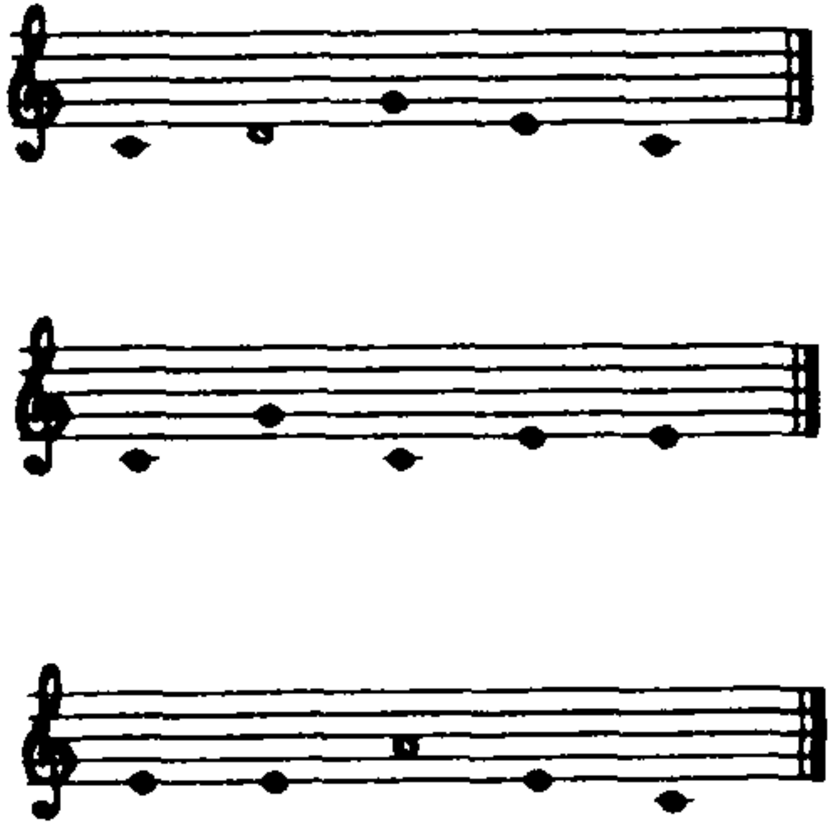


  - SampleRNN [Mehri et al., 2017]



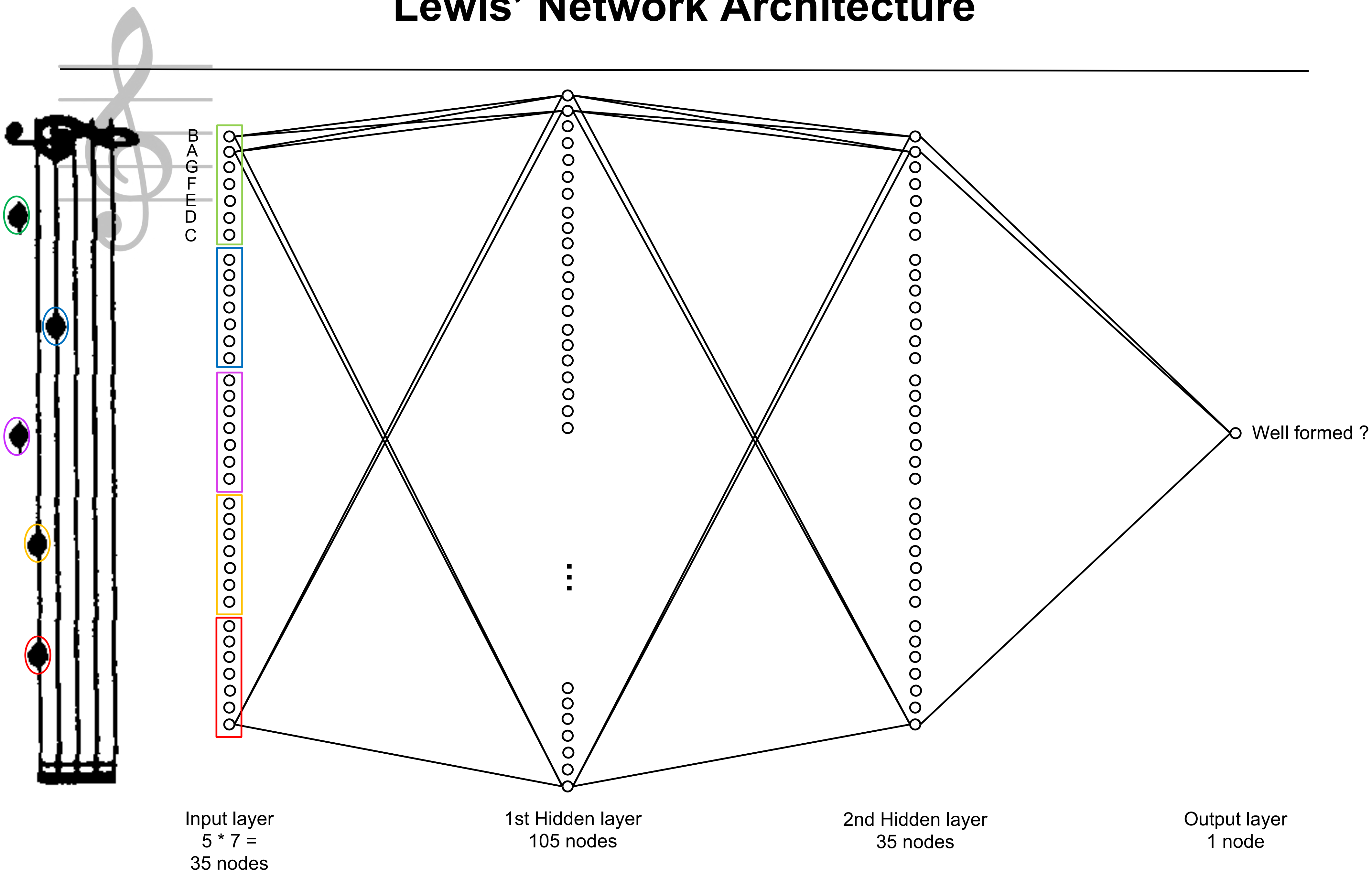
Lewis' Creation by Refinement (1/4) [Lewis, 1988]

- Training on 30 Manually Generated 5-Note Melodies
- 7 Possible Notes (from C to B, without alteration)
- Well Formed
  - Possible Intervals:
    - » Unison, 3rd, 5th,
    - » Scale Degree Stepwise Motion
- Poorly Formed
  - Excessive Motion or Excessive Repetition
- Binary Classification Training
  - Well or Poorly Formed

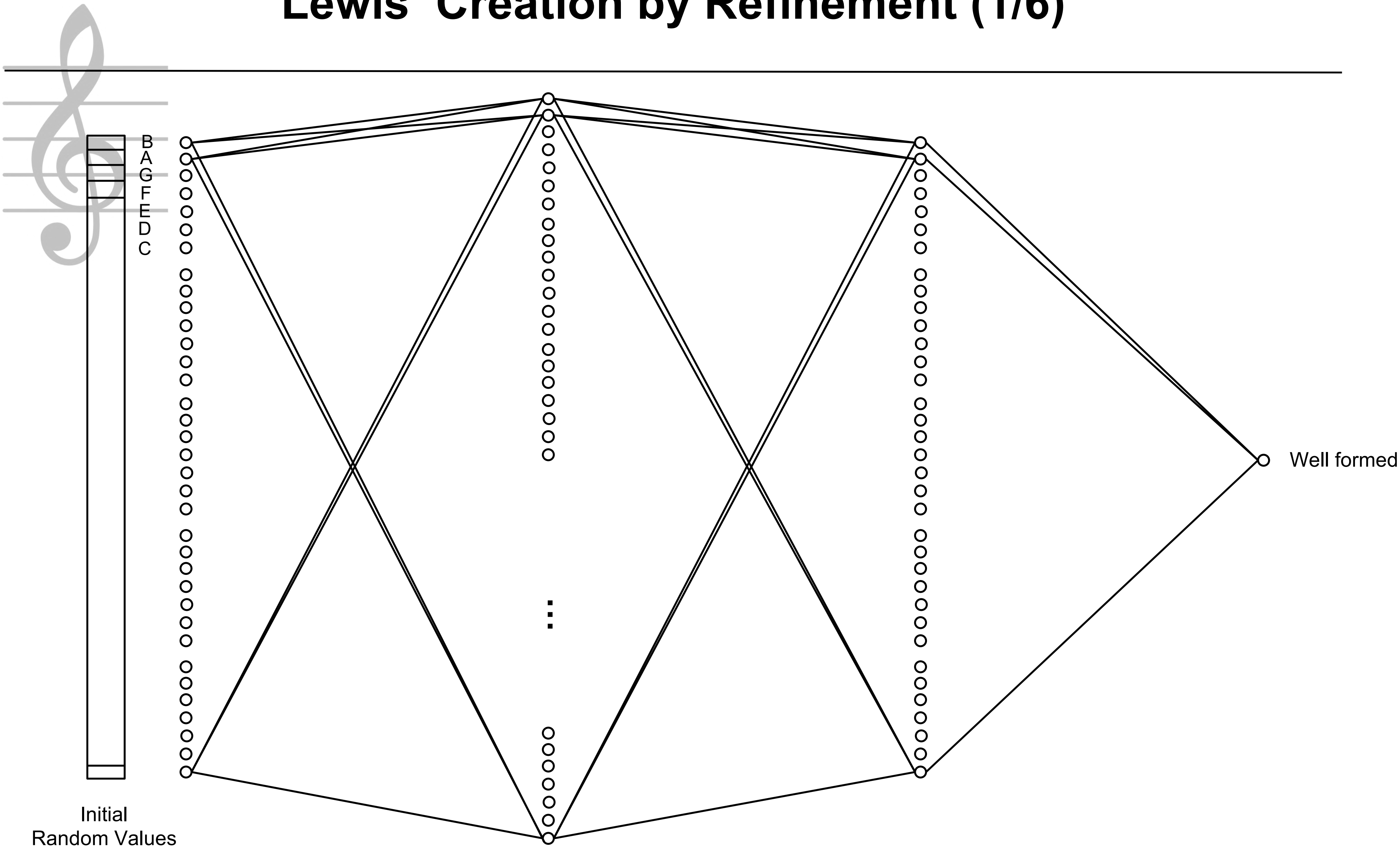


Ex. of Training Examples

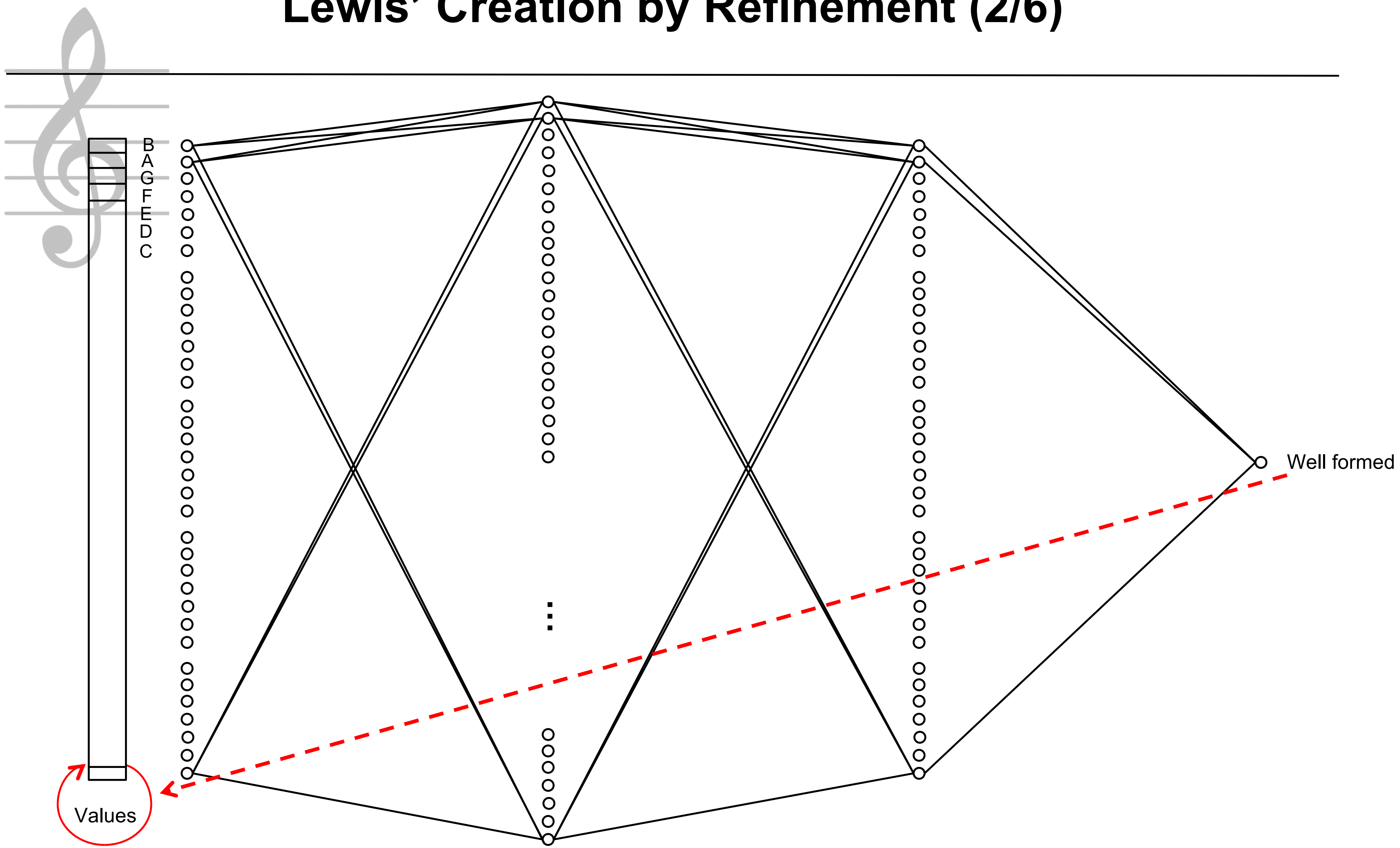
Lewis' Network Architecture



Lewis' Creation by Refinement (1/6)



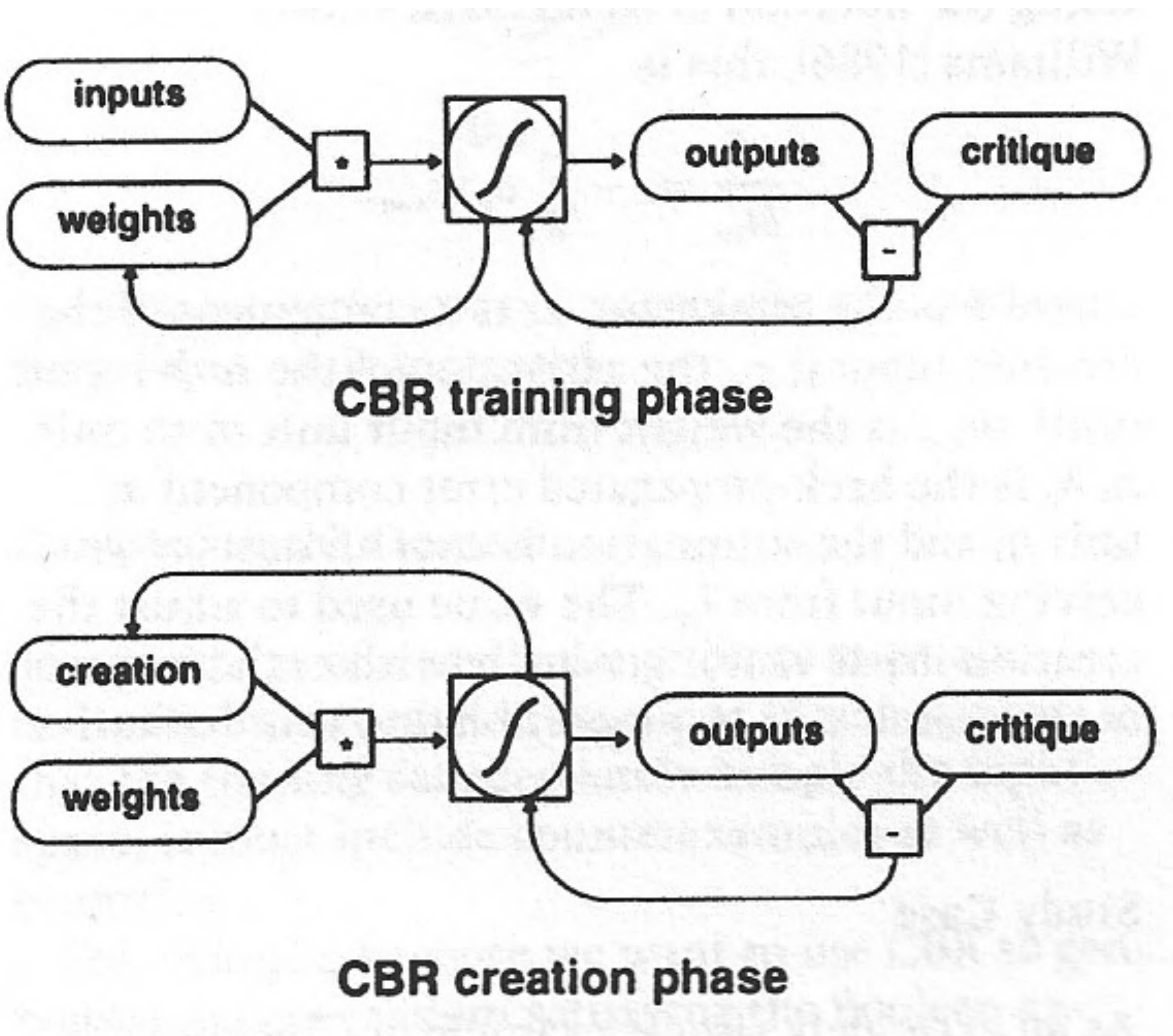
Lewis' Creation by Refinement (2/6)



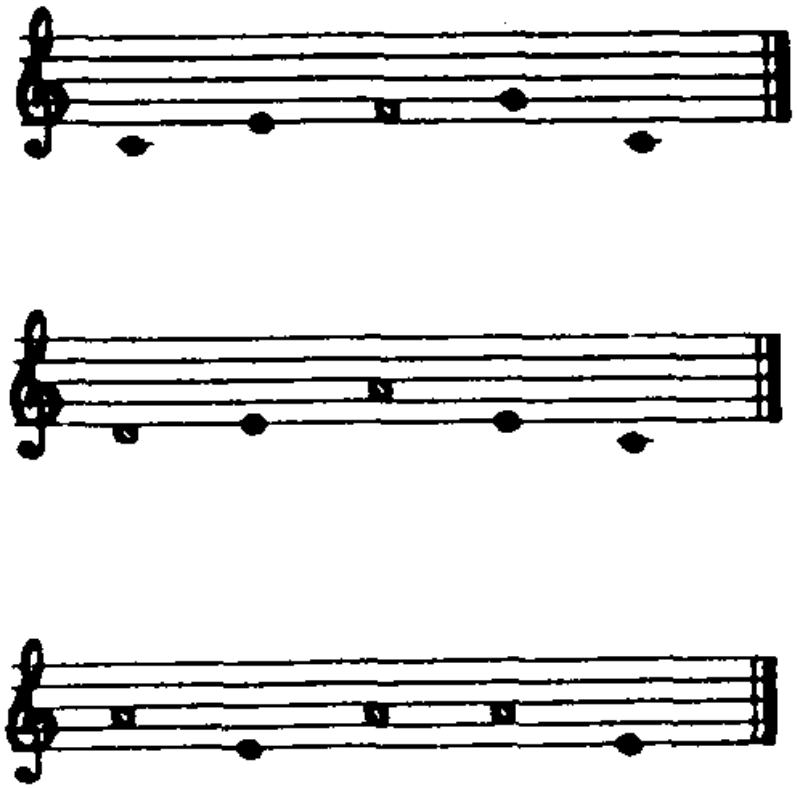
Input Values are Incrementally Manipulated

Under the Control of a Gradient Descent on Error in Predicted Well Formed

Lewis' Creation by Refinement (3/6)



Lewis' Creation by Refinement (4/6)



Ex. of Melodies Created by Refinement

- The Network Learned Preference for Stepwise and Triadic Motion

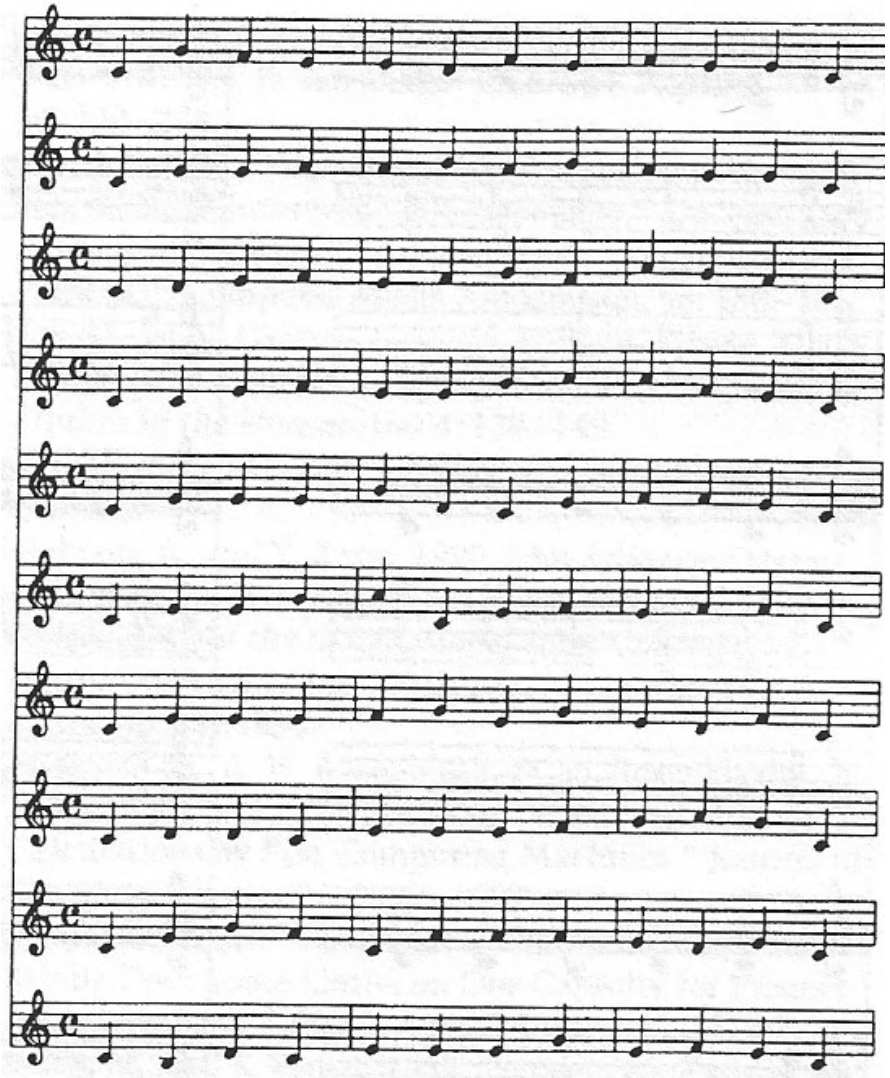
Lewis' Creation by Refinement (5/6)

- Attention

Attentional CBR

In order to partition a large problem into manageable subproblems, we need to provide both an attention mechanism to select subproblems to present to the network and a context mechanism to tie the resulting subpatterns together into a coherent whole. A context mechanism can be provided by context inputs, which during the creation phase are clamped to the values of the surrounding and previously constructed pattern. As an example, to produce elaborations on a short phrase, the training set inputs would consist of sample phrases paired with corresponding embellished phrases (possibly using a suitable null-note representation to allow different phrase lengths), and the critique would (as usual) consist of some critique of the character of the embellishment. In the creation phase, the embellished inputs would be set to random values, but the context inputs would be clamped to the phrase itself.

- Hierarchy

The author's experiments have employed *hierarchical* CBR. In this approach, a developing pattern is recursively filled in using a scheme somewhat analogous to a formal grammar rule such as $ABC \rightarrow AxByC$, which expands the string without modifying existing tokens. That is, three tokens (for example, musical notes) labeled A,B,C will be expanded with two additional tokens x,y inserted in the indicated positions. The expanded string $AxByC$ may be rewritten further using a suitable scheme.

Ex. of Melodies Created by Hierarchical Refinement (ABCD -> ABxCD scheme)

Lewis' Creation by Refinement (6/6)

- Reinforcement

Reinforcement CBR

Developing the training set is probably the most difficult aspect of employing CBR (and other supervised learning algorithms). In *reinforcement* CBR some or all of the training set is produced automatically, by completing the domain, rather than being compiled by the experimenter as in the standard supervised learning paradigm. In this scheme, the training phase is interrupted at intervals, and the creation phase is invoked. The resulting creations are evaluated by the experimenter and are added to the training set with a corresponding critique if they are judged to extend the existing training set. After the training set is extended, the net is retrained, followed by the accumulation of new examples, etc., until all sample creations are judged satisfactory by both the experimenter and the network.

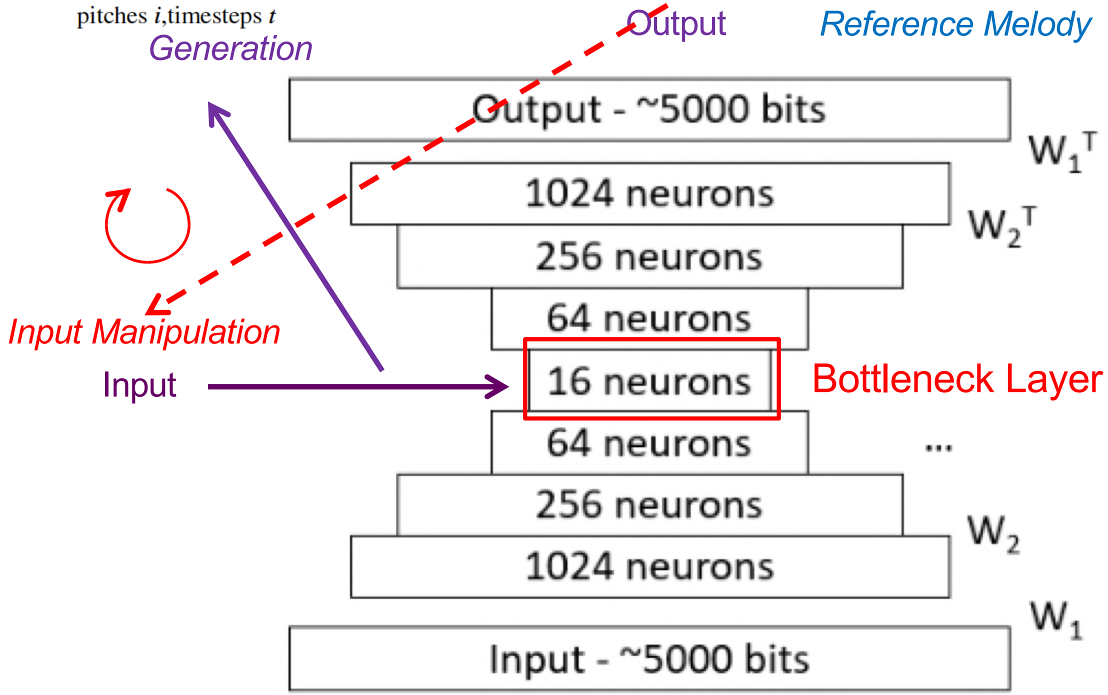
Not Reinforcement learning

Created Melodies which are Liked are Added to the Training Set

Lewis' Creation by Refinement Pioneering (1/3)

- Precursor of
- Gradient Descent Input Manipulation [Briot et al., 2017]
- Ex: DeepHear [Sun, 2016]
  - Melody Consonant Accompaniment Creation

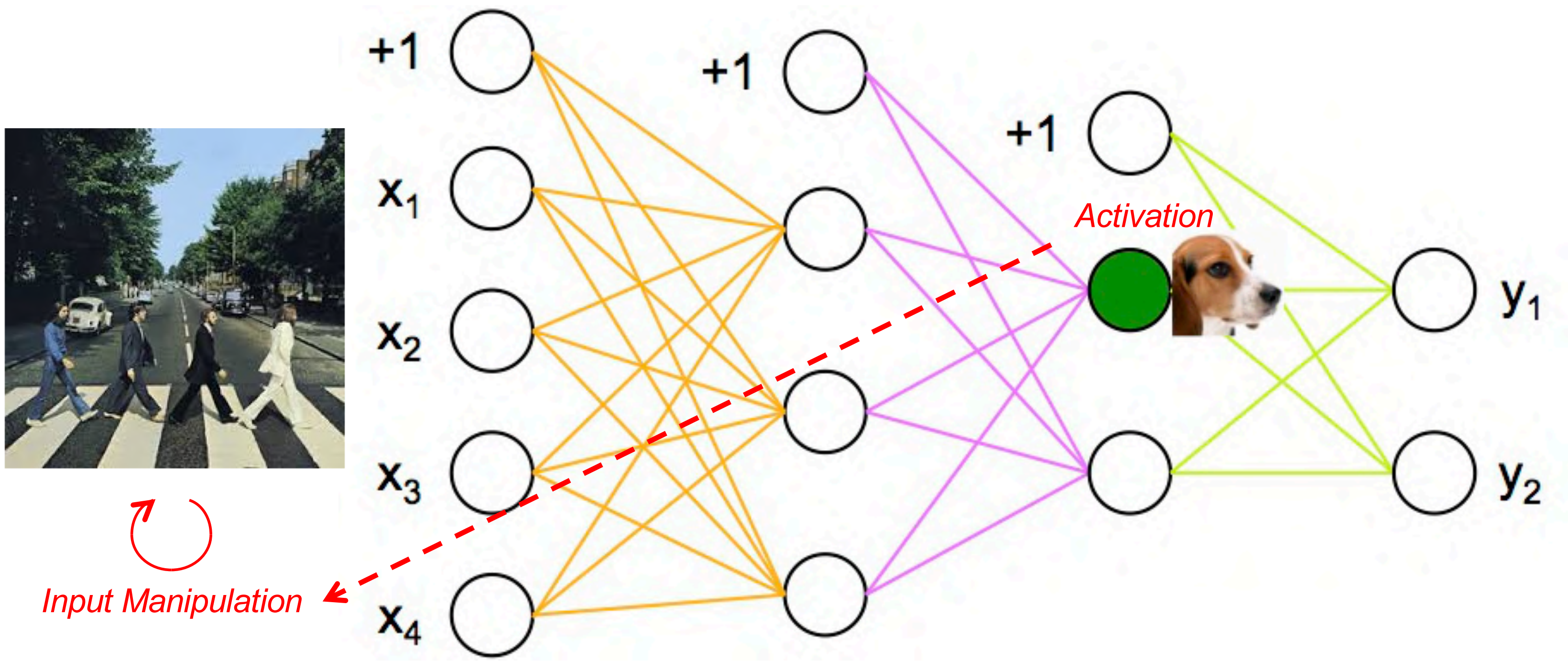
$$err(S) = \sum_{\text{pitches } i, \text{timesteps } t} ((i, t) \text{ in melody}) \cdot (1 - S[i, t])^2$$



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

Lewis' Creation by Refinement Pioneering (2/3)

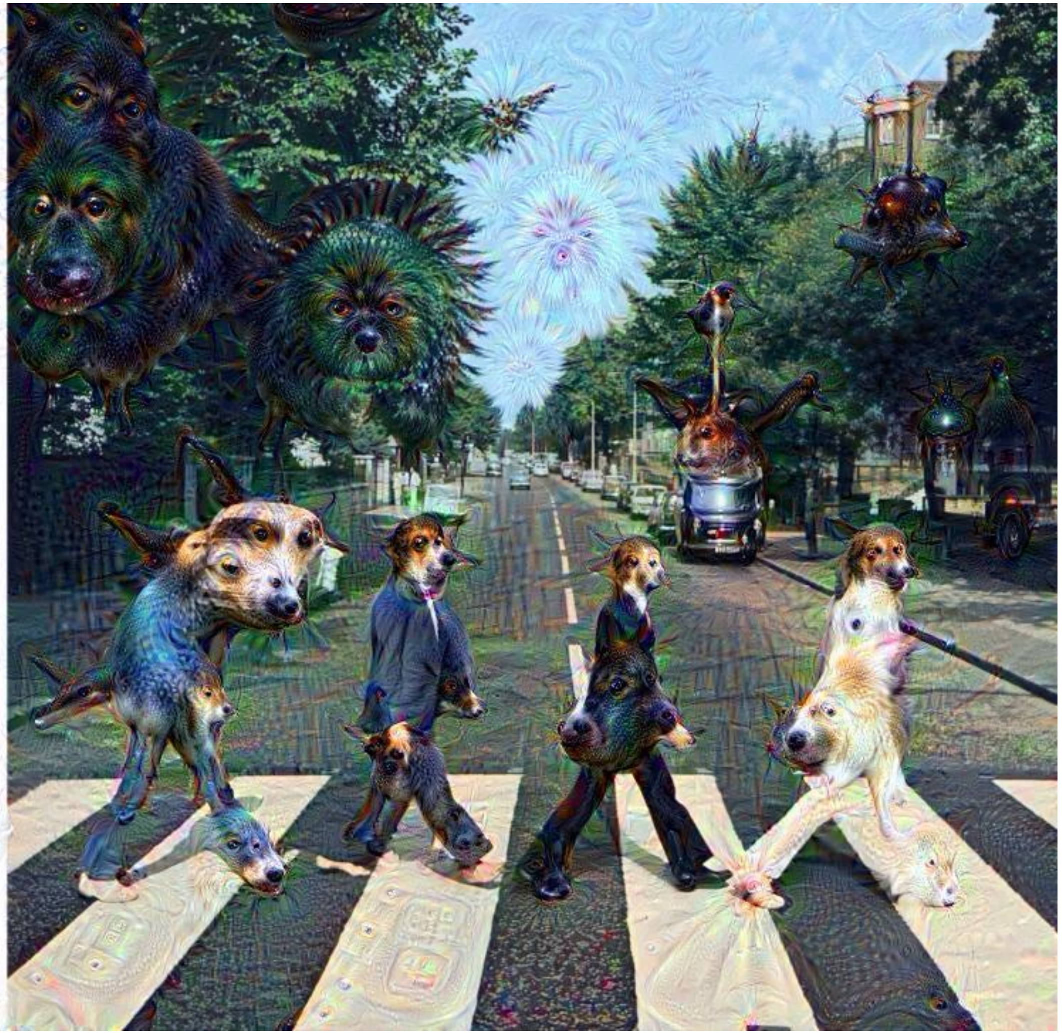
- Precursor of
- Gradient Ascent Input Manipulation [Briot et al., 2017]
- Ex: DeepDream [Mordvintsev et al. 2015]
  - Motif Detector Neuron Activation Maximization



Lewis' Creation by Refinement Pioneering (3/3)

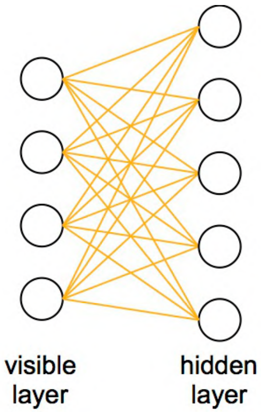
Initial Image

Deep Dream Image

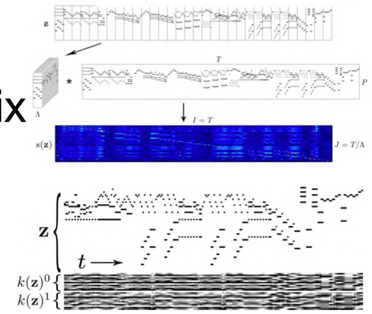


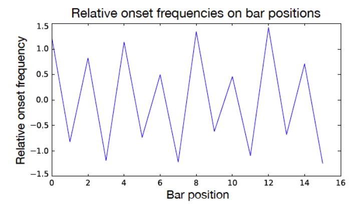
Structure Imposition (1/2) [Lattner et al., 2016]

- 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)

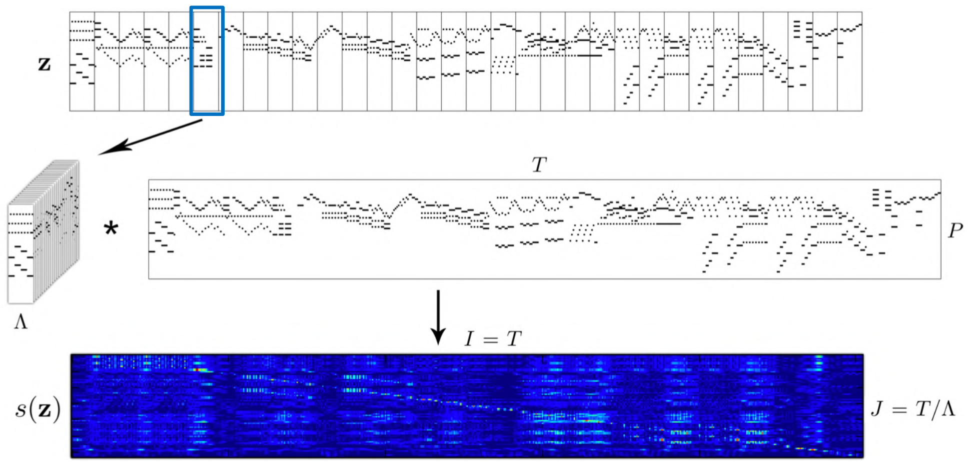




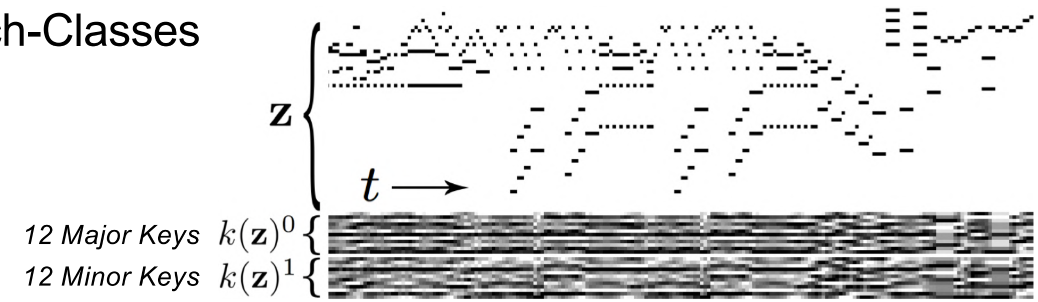
  - Sampling Control, by Selective Gibbs sampling (SGS)
    - » at a Selected Low-Level (subset of variables)
    - » to realign selectively the sample to the learnt distribution
  - Alternate IP/GD and SGS, controlled by Simulated Annealing
  - But not exact as, e.g., Markov Constraints [Pachet & Roy, 2011]

Structure Imposition

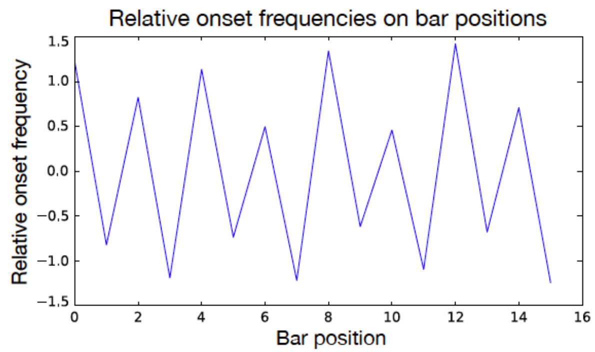
- Structure (Repetition Structure, Ex: AABA)
  - » Self-Similarity Matrix
  - » For each Music Slice



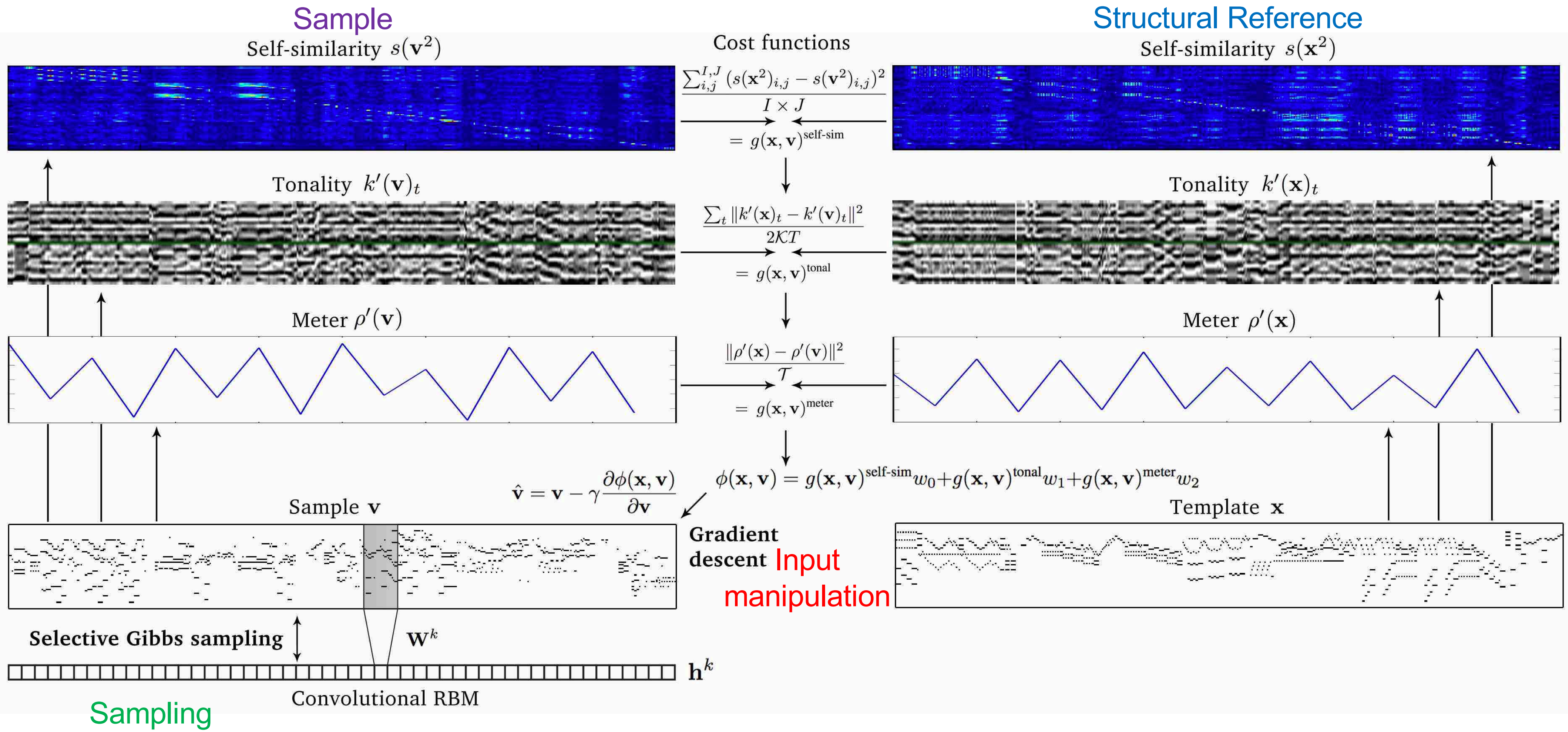
- Tonality, via Similarity of Distribution of Pitch-Classes
  - » Key Estimation Vectors over Time



- 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

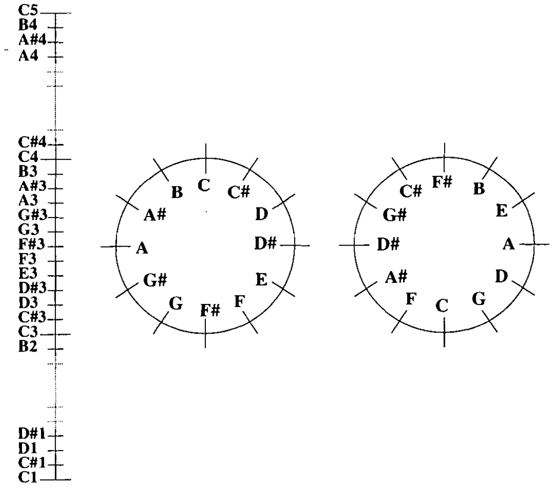
C-RBM Examples

- RNN-RBM Sample
- Unconstrained Sample
- Template Piece
- Constrained Sample

https://soundcloud.com/pmgrbm

Mozer's Rich Representation Model [Mozer, 1994]

Note/Harmony

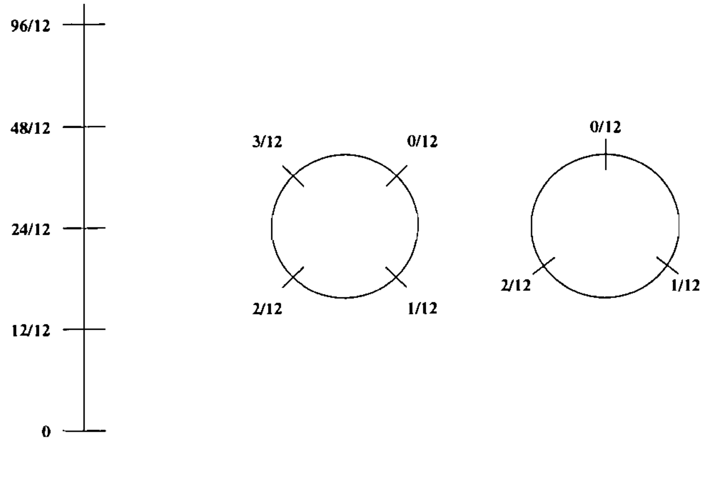


Pitch Height Chroma Circle Circle of Fifths

Pitch	PH	CC						CF					
C1	−9.978	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
F#1	−7.349	−1	−1	−1	+1	+1	+1	+1	+1	+1	−1	−1	−1
G2	−2.041	−1	−1	−1	−1	+1	+1	−1	−1	−1	−1	+1	+1
C3	0	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
D#3	1.225	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
E3	1.633	−1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1
A4	8.573	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
C5	9.798	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
Rest	0	+1	−1	+1	−1	+1	−1	+1	−1	+1	−1	+1	−1

[Mozer, 2004]

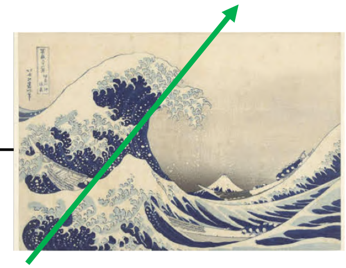
Duration/Rhythm

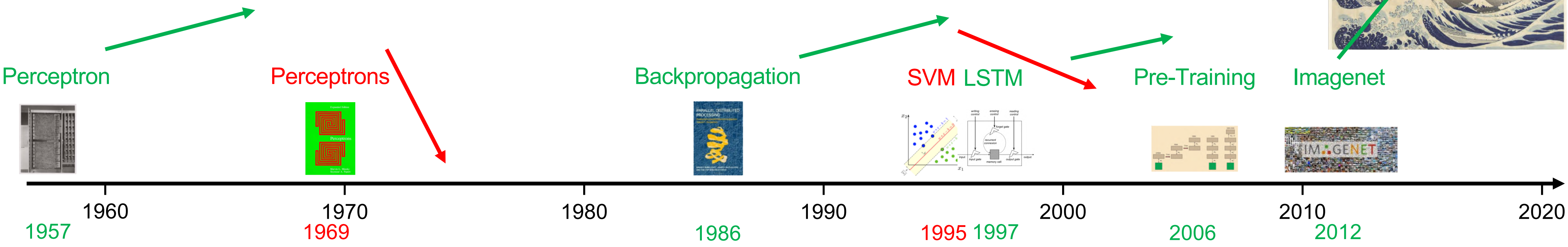


Duration Height *log(duration)* 1/3 Beat Circle *mod(duration, 1/3)* 1/4 Beat Circle *mod(duration, 1/4)*

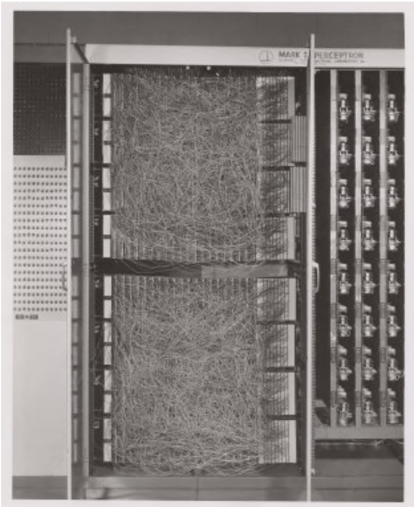
From Neural Networks to Deep Learning

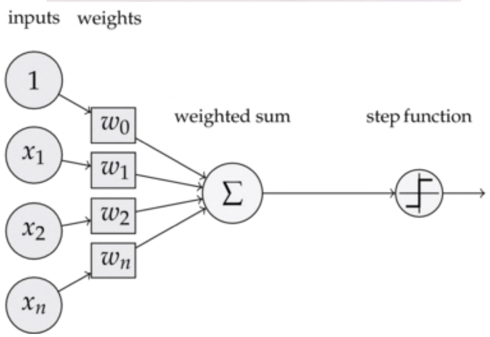
Neural Networks Evolution



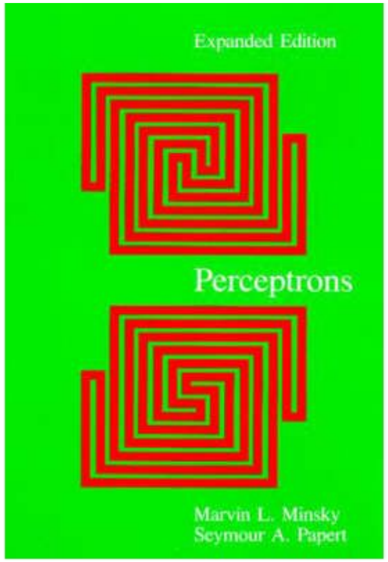


History: From Perceptron to Artificial Neural Networks to Deep Learning (1/4)



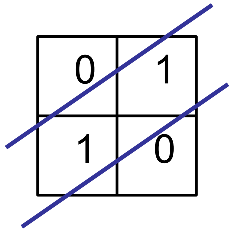


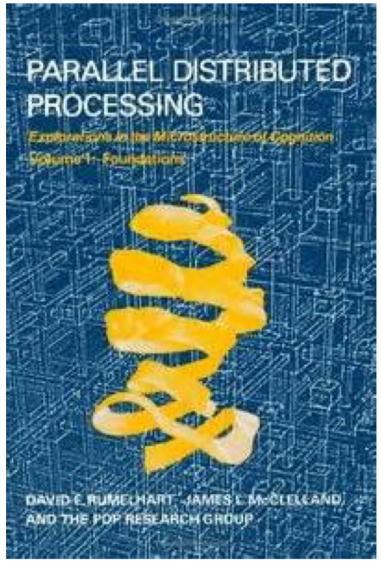
Perceptron
[Rosenblatt 1957]



Perceptrons (Book)
[Minsky & Papert 1969]

Linear Separable only
XOR counter example



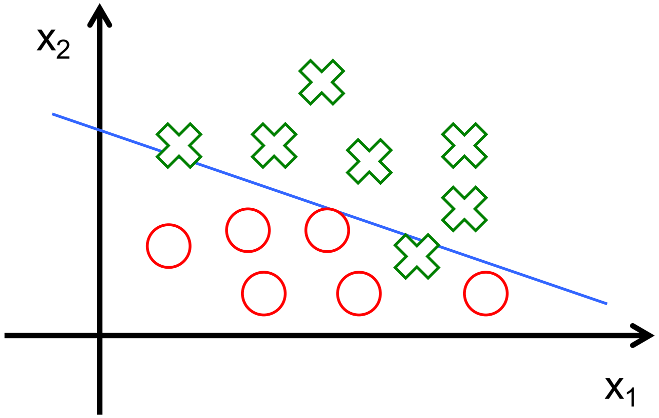


PDP (Books)
[Rumelhart et al. 1986]

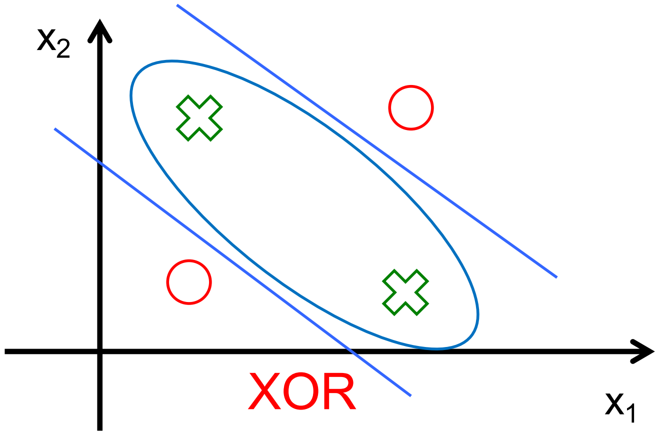
Multi-layer networks
Backpropagation

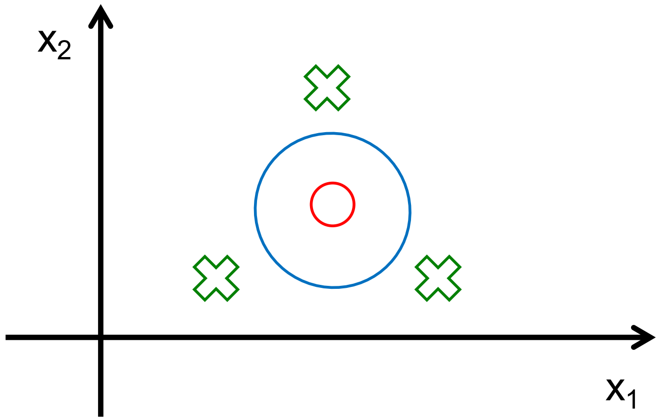
Linear vs Non Linear Decision Boundary

- Linear



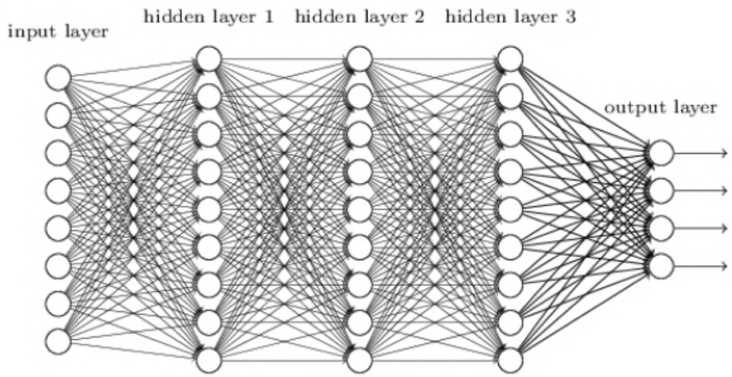
- Non Linear

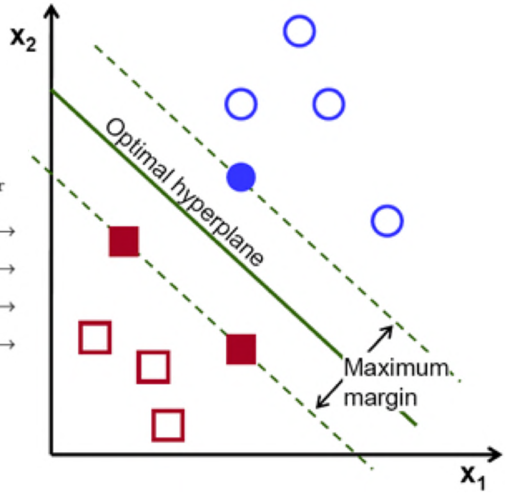


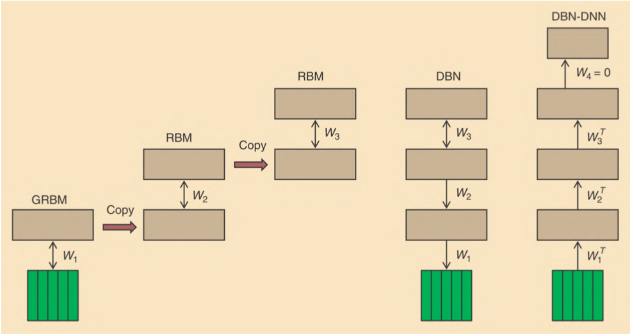


- Argument (XOR) used by [Minsky & Papert 1969] to criticize Perceptrons [Rosenblatt 1957] (and advocate Symbolic Artificial Intelligence)
- This stopped research on Perceptrons/Neural Networks for a long while
  - until Hidden Layers and Backpropagation or/and Kernel Trick (see later)

History: From Perceptron to Artificial Neural Networks to Deep Learning (2/4)







Pre-Training [Hinton et al. 2006]
Layer-Wise Self-Supervised Training/Initialization

Difficulty to Efficiently Train Networks with many Layers

Unstable Gradients

SVM [Vapnik 1963]
SVM + Kernel Trick [Vapnik et al. 1992]

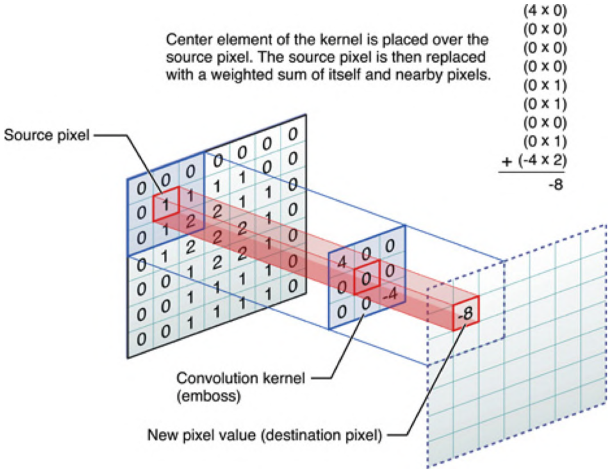
Nice Model and Optimized Implementation

Margin Optimization

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

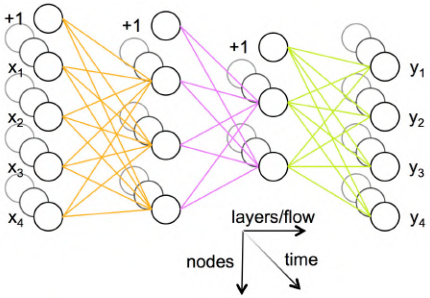
ImageNet 2012 Image Recognition Challenge Breakthrough

History: From Perceptron to Artificial Neural Networks to Deep Learning (3/4)

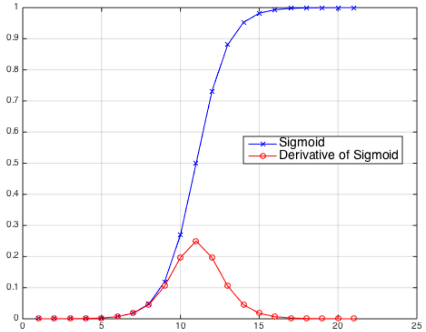


Convolutional Networks
[Le Cun et al. 1998]

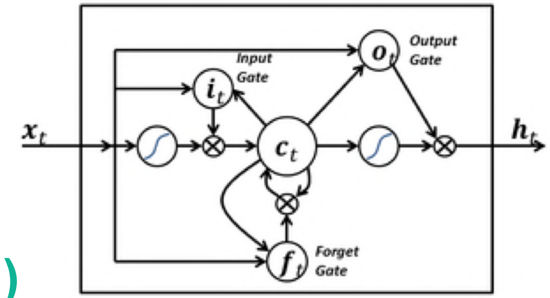
Equivariance (to translation)
& Invariance (to small transformations)



Recurrent Neural Networks (RNN)
(1986)
Temporal Invariance



Gradient Vanishing
or Explosion (1991)



Long Short-Term Memory
(LSTM)
[Hochreiter &
Schmidhüber 1997]

History: From Perceptron to Artificial Neural Networks to Deep Learning (4/4)

Massive Data Available

\+

theano

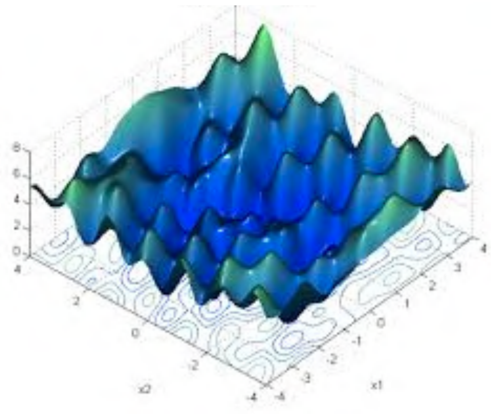
Efficient Implementation Platforms

\+

Affordable Efficient Parallel Processing (Graphic Cards GPU)

Power Increase

- Brute Force



Loss Minimization

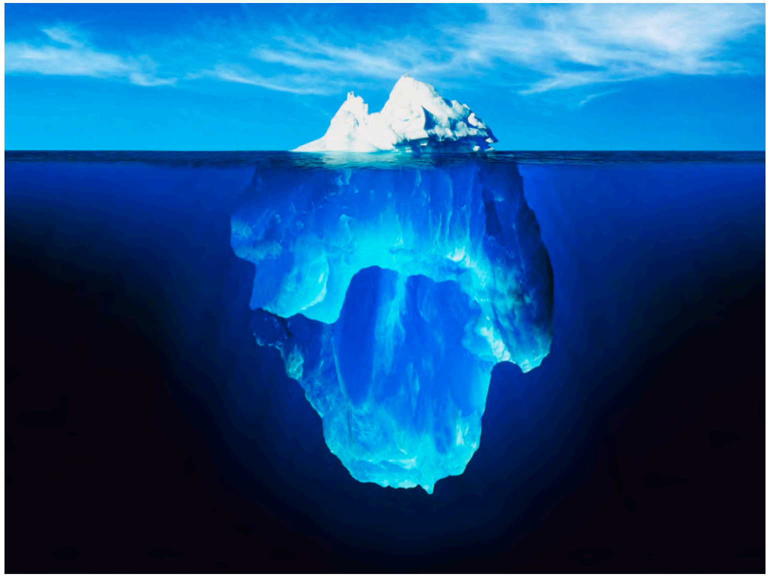
- Hypervitamined Brute Force

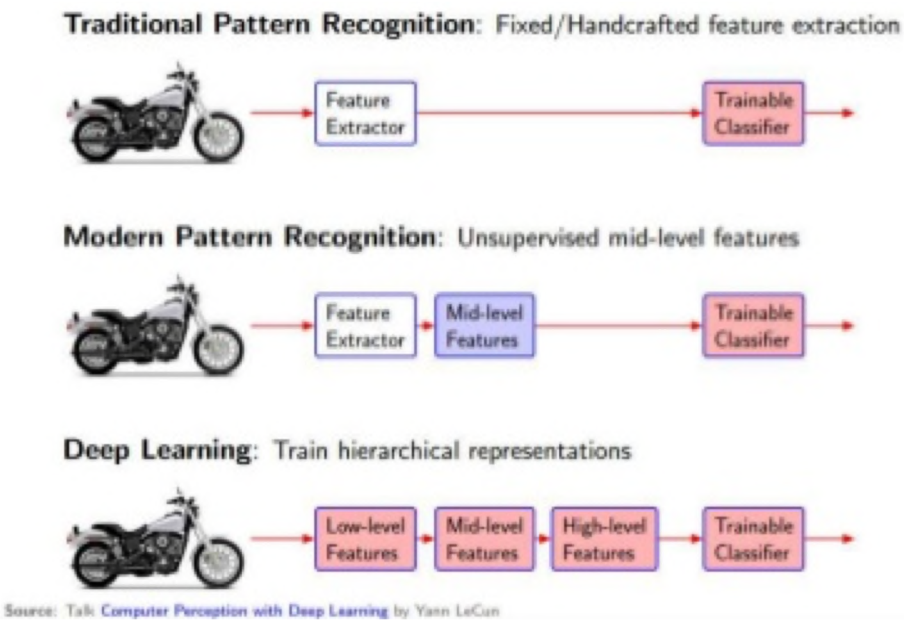
GPUs

TensorFlow PyTorch

Why Deep ?

- More Complex Models
- Learns better Complex Functions
- Hierarchical Features/Abstractions
- No Need for Handcrafted Features
  - (Automatically Extracted)



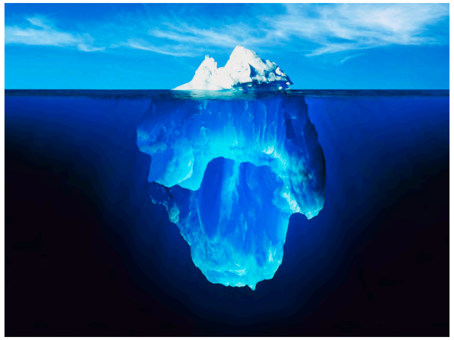


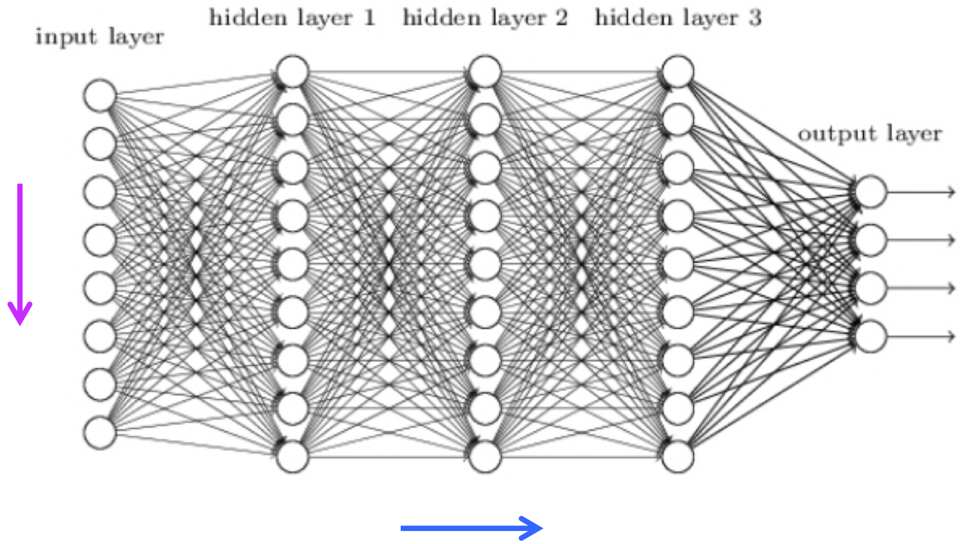
Distributed Representations

End-to-End Architecture

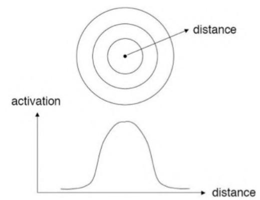
Why Deep ?

- Theorem [Eldan & Shamir 2016]
  - There is a simple radial function on R^d, expressible by a 3-layer net, but which cannot be approximated by any 2-layer net to more than a constant accuracy unless its width is exponential on the dimension d
  - Depth $\longrightarrow$ vs/and Width $\downarrow$

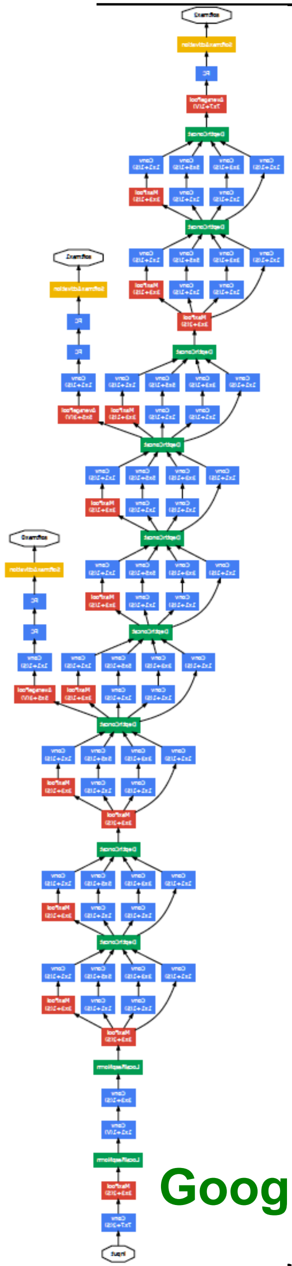




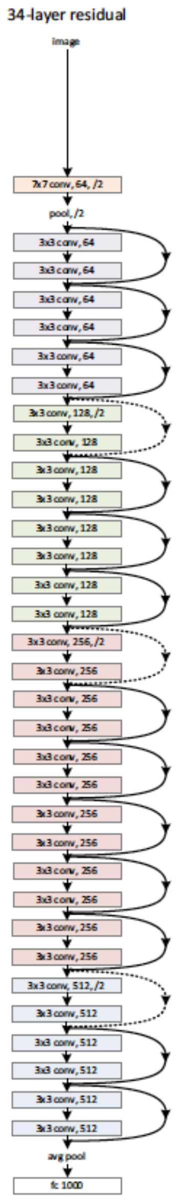
Radial function = Function whose value at each point depends only on distance between point and origin



Very Deep Learning



GoogLeNet (2014)



ResNet (2015)

(Very) Deep Networks

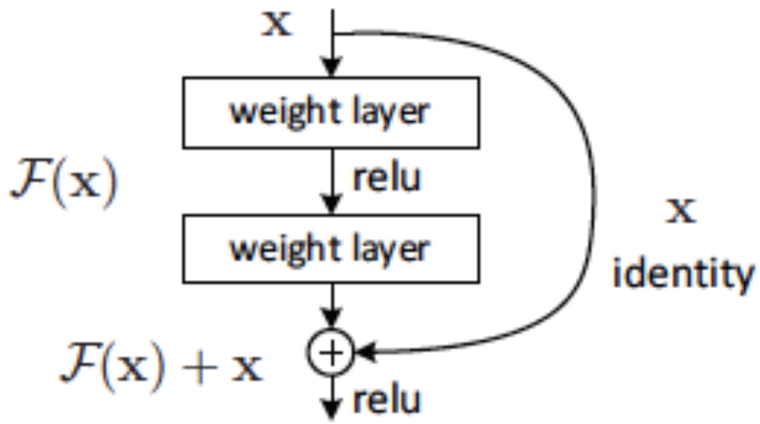
Upto 152 Layers !

New Techniques (Tricks ?!) e.g.,

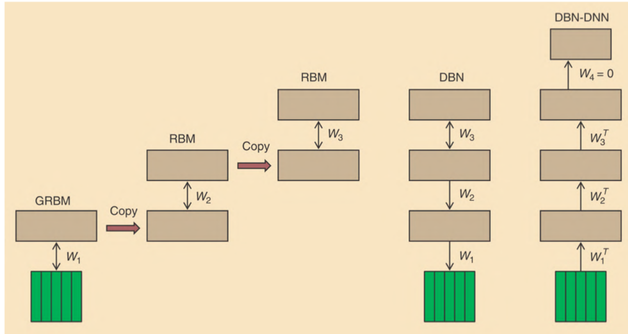
Batch Normalization [Ioffe & Szegedy, 2015]

Deep Residual Learning [He et al., 2015]

Replaced Pre-Training (less in vogue)



The Groundbreaking Start of Deep Learning



Pre-Training [Hinton et al. 2006]
Layer-Wise Self-Supervised Training/Initialization

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

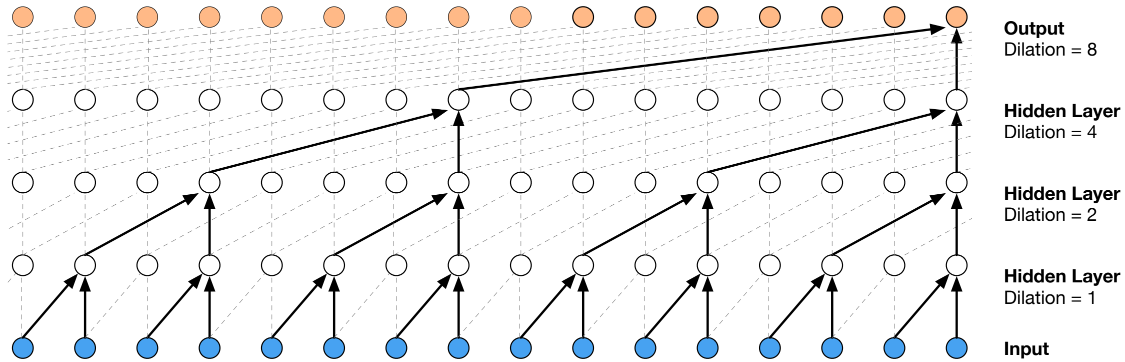
ImageNet 2012 Image Recognition Challenge Breakthrough

WaveNet Audio End-to-End Generation [van den Oord et al., 2017]

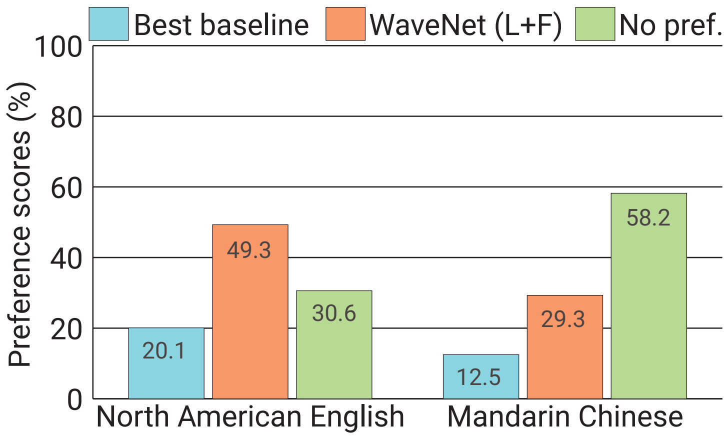
- Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., WaveNet: A Generative Model for Raw Audio, arXiv:1609.03499, December 2016.

- Waveform

- End to end architecture

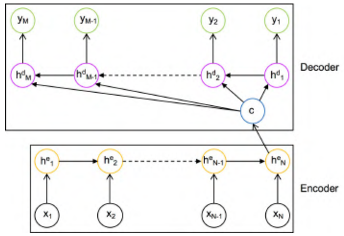


[van den Oord, 2016]

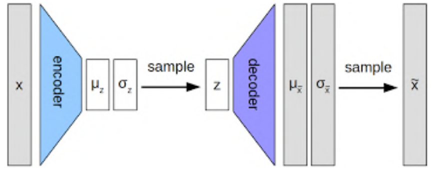


New Architectures

- New Architectures and Mechanisms
- RNN Encoder Decoder

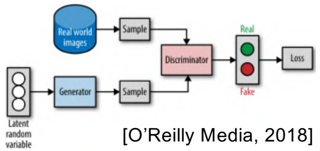


- Variational Autoencoders



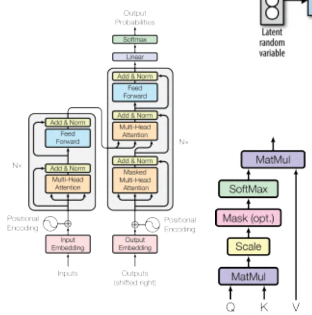
[Bechberger, 2018]

- Generative Adversarial Networks



[O'Reilly Media, 2018]

- Transformer
- Attention Mechanism

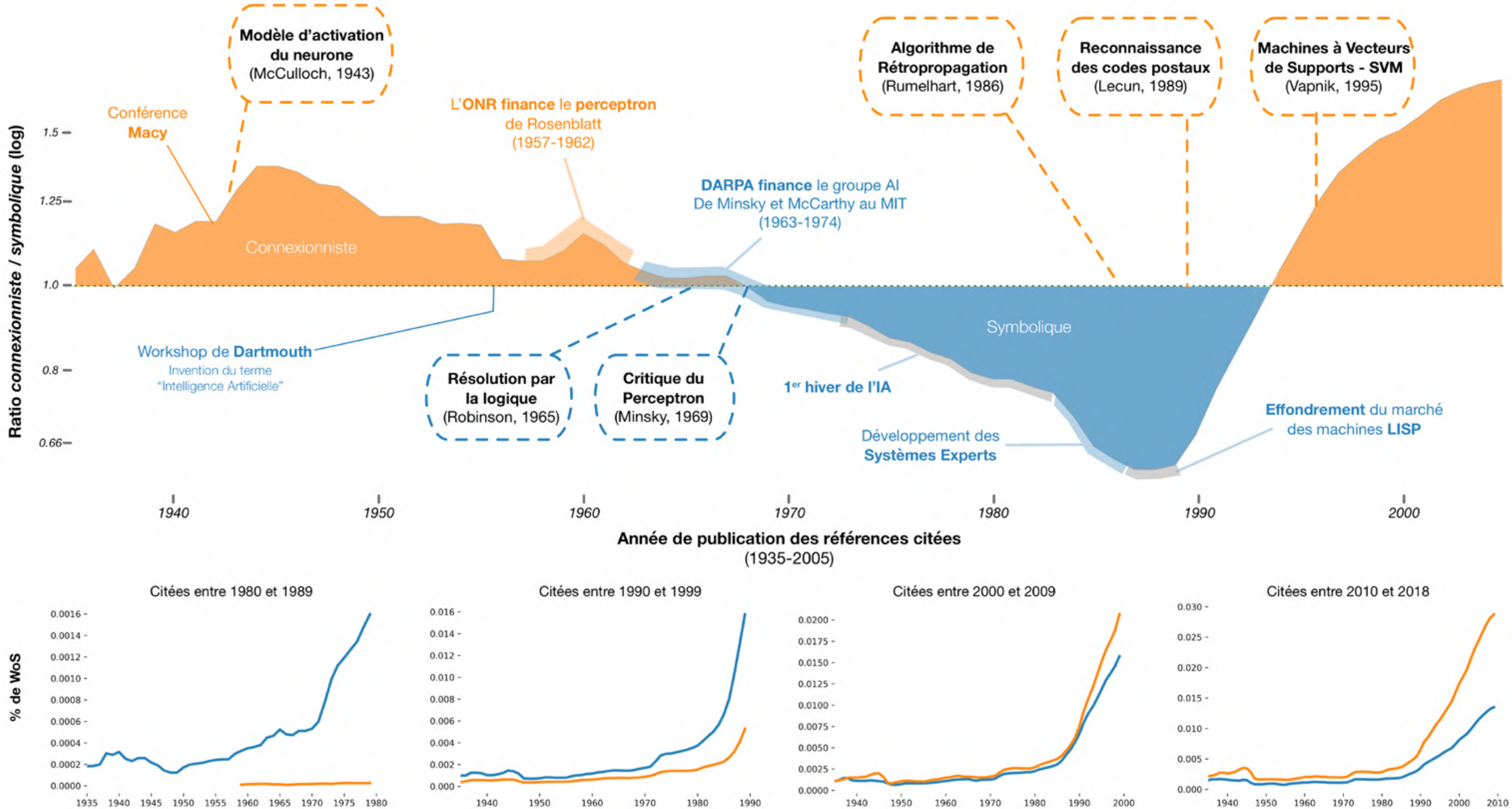


[Vaswani et al., 2017]

- …

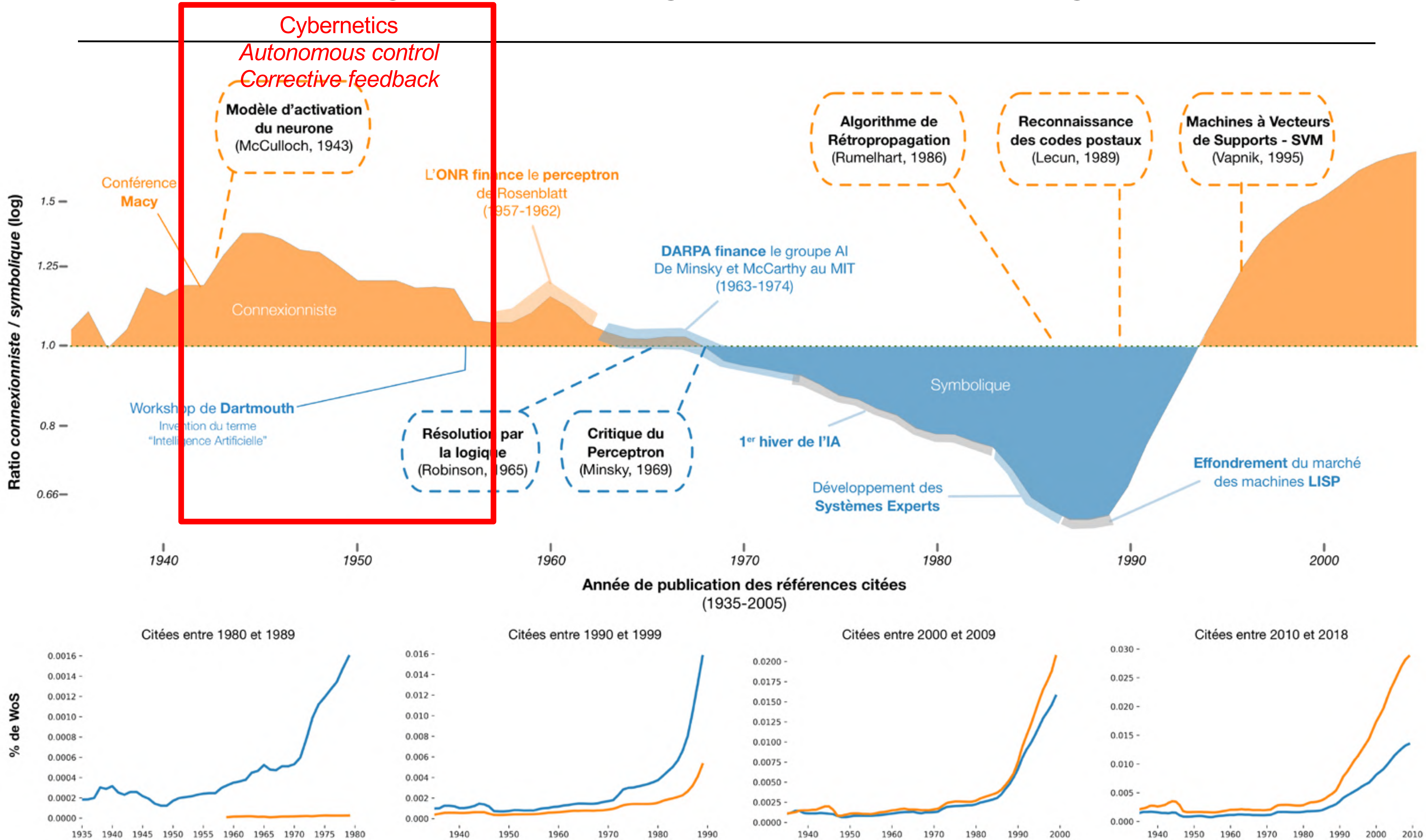
Artificial Intelligence and Machine Learning

Symbolic vs Connexionist AI – History



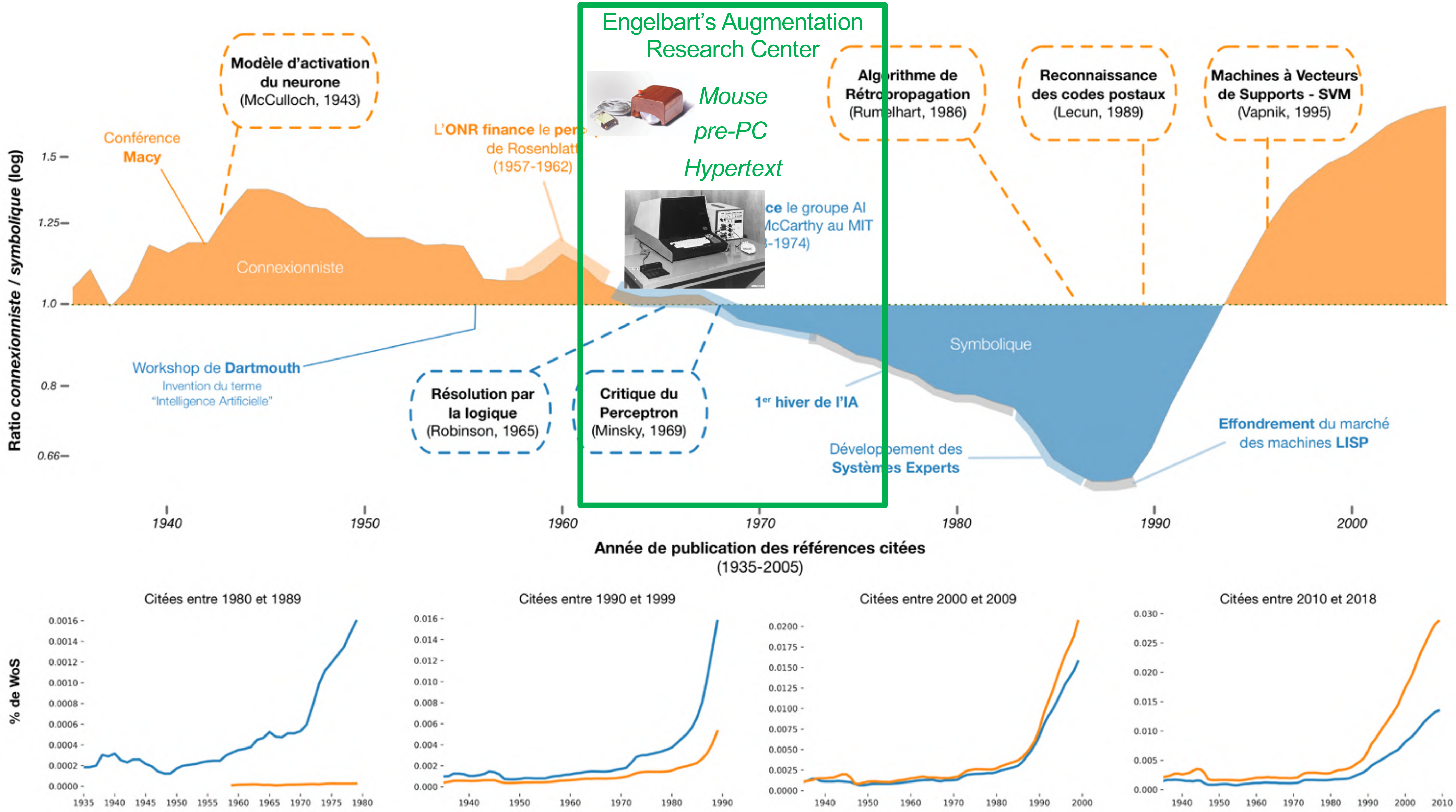
[Cardon et al., 2018]

Symbolic vs Cybernetics – History



[Cardon et al. 2018]

Artificial Intelligence vs Intelligence Augmentation – History



[Cardon et al. 2018]

Machine Learning and Artificial Intelligence

- Various Forms of Machine Learning

- Statistical
- Neural Networks, Bayesian Networks, Clustering…

- Decision
- Reinforcement learning

- Symbolic – Learning Concepts from Examples
- Inductive Logic Programming (ILP)

- Learning and Adapting from Cases
- Case-Based Reasoning

Machine Learning and Artificial Intelligence

- Machine Learning is Part of Artificial Intelligence Techniques

But also:

- Reasoning
- Planning
- Knowledge Representation
- User Modeling and Interaction
- Collaboration (Multi-Agent Systems)

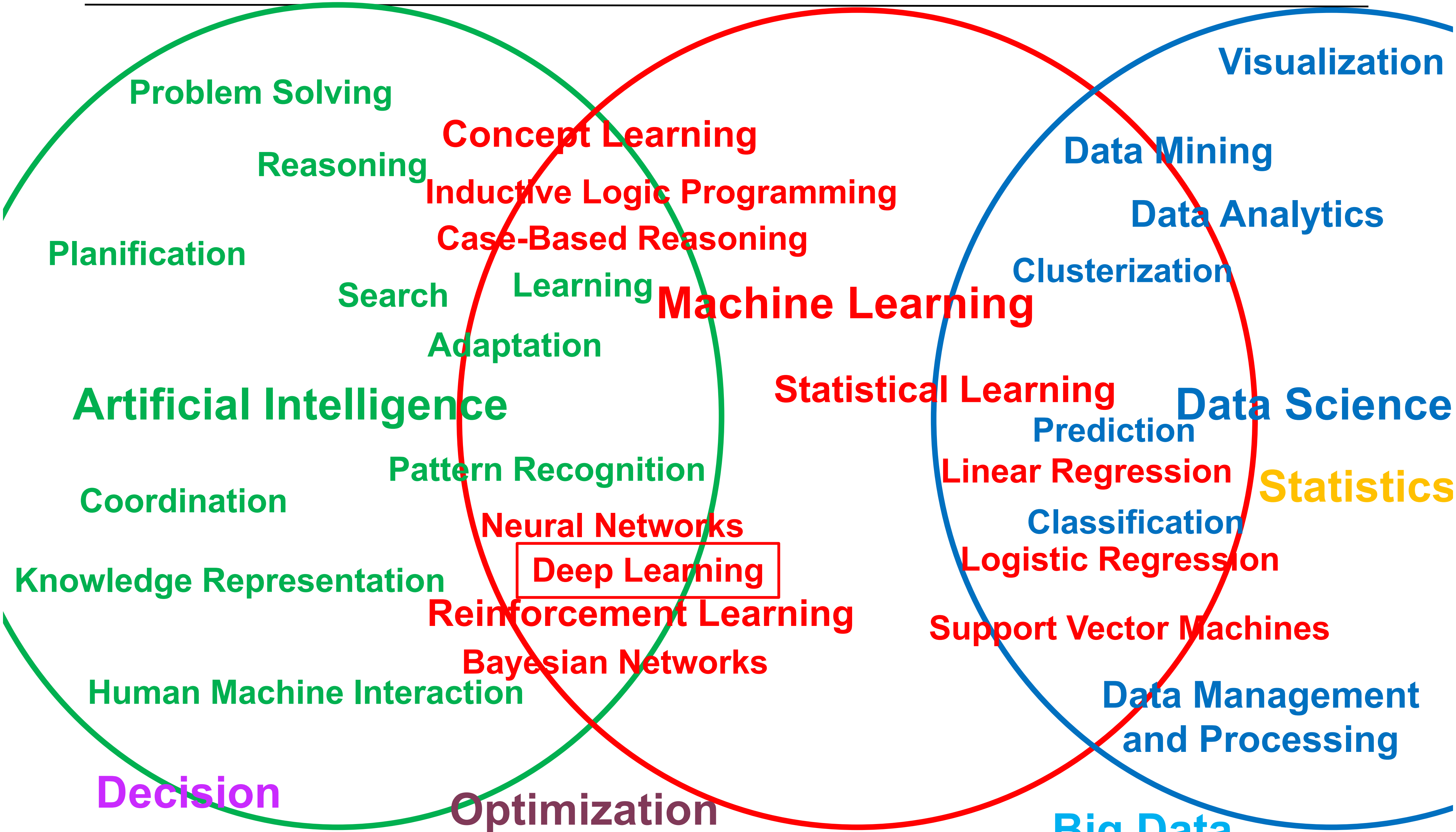
- Natural Language Processing
- Dialogue
- Speech Processing

- Decision
- Game Theory
- Optimization
- Robotics

Machine Learning and Artificial Intelligence

- Backfire (Irony) of History
- In 1960, Minsky and Papert founded AI (Artificial Intelligence) based on Concepts, Symbols, Logic, Reasoning…, Against Cybernetics (Feedback) and Connexionism (Neural Networks)
- In 1969, they "Killed" Connexionism/Neural Networks (Sound Critic of Perceptron)
- In 2006, Start of Deep Learning
- Now, AI is synonym of Deep Learning
- When Actually, Neural Networks are somehow based on Statistical (Correlation) Brute Force

Terminology



Why Using Computer and Machine Learning (for Creating Music)?

Why Using Computer for Music

- Bad Reasons (Fears)
  - Lead human musicians to unemployment
  - Lower the quality of music ☺

- Good reasons
  - Facilitate storing, indexing, delivering and sharing of music (MIDI, MP3, Spotify…)
  - New instruments and interaction (Synthesizers, Interactive music performances…)
  - New sounds (Synthesizers and Signal processing)
  - Analysis tools and algorithms (Spectrum, Patterns Discovery…)
  - Initiation and Education (Band in the Box, Garage Band…)

- Production
  - Partially automate tasks (Ex: Mixing, etc.)
- Composition, Analysis and Arrangement
  - Algorithmic composition
  - Harmonization
  - Analysis

Why Using Computer (and Machine Learning) for Music

- Vast Associative Memory
  - More systematic than Human memory
- Representation of Musical pieces, Style, Patterns…
- Associations and Correlations
- Knowledge (Theory, Rules, Heuristics…)

- Can Help Human musicians

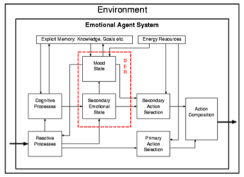
- Human musicians rarely compose from scratch – They borrow from others
  - Consciously
    - Plagiat, Citation…
  - Unconsciously
    - Influence
  - Recombinations
  - Historical Evolution/Extension
    - Modal monophonic -> Polyphonic (Counterpoint) -> Tonal Music (Harmony) -> Extended Harmony (Debussy, Jazz…)
  - Ruptures (Dodecaphonism, Free Jazz, Punk…)
    - Rare and often transient

Some Preconcepts Against Deep Learning / AI

- No Emotion
  - Create Emotion to the Human Target ?
  - Or/And Internal Model of Emotion ?

[Image: BBC]

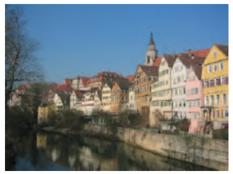
[Karras et al., 2018]



[Bryson et al., 2004]

- No Creativity
  - Exploratory
    - » AlphaZero used successful strategies yet unconsidered

  - Recombination
    - » Concept and Conjecture Discovery (ex: Numbers, Prime Numbers, Prime Numbers Decomposition) AM and Eurisko [Lenat, 1976; 1983]
    - » Style Transfer [Gatys et al., 2015]


+

=


  - Paradigm Reformulation
    - » Ex: Quantum Physics, Algebraic Geometry, Dodecaphonism…
    - » More difficult

Handcrafted vs Learnt Models

- Handcrafted
  - Tedious
  - Error-Prone

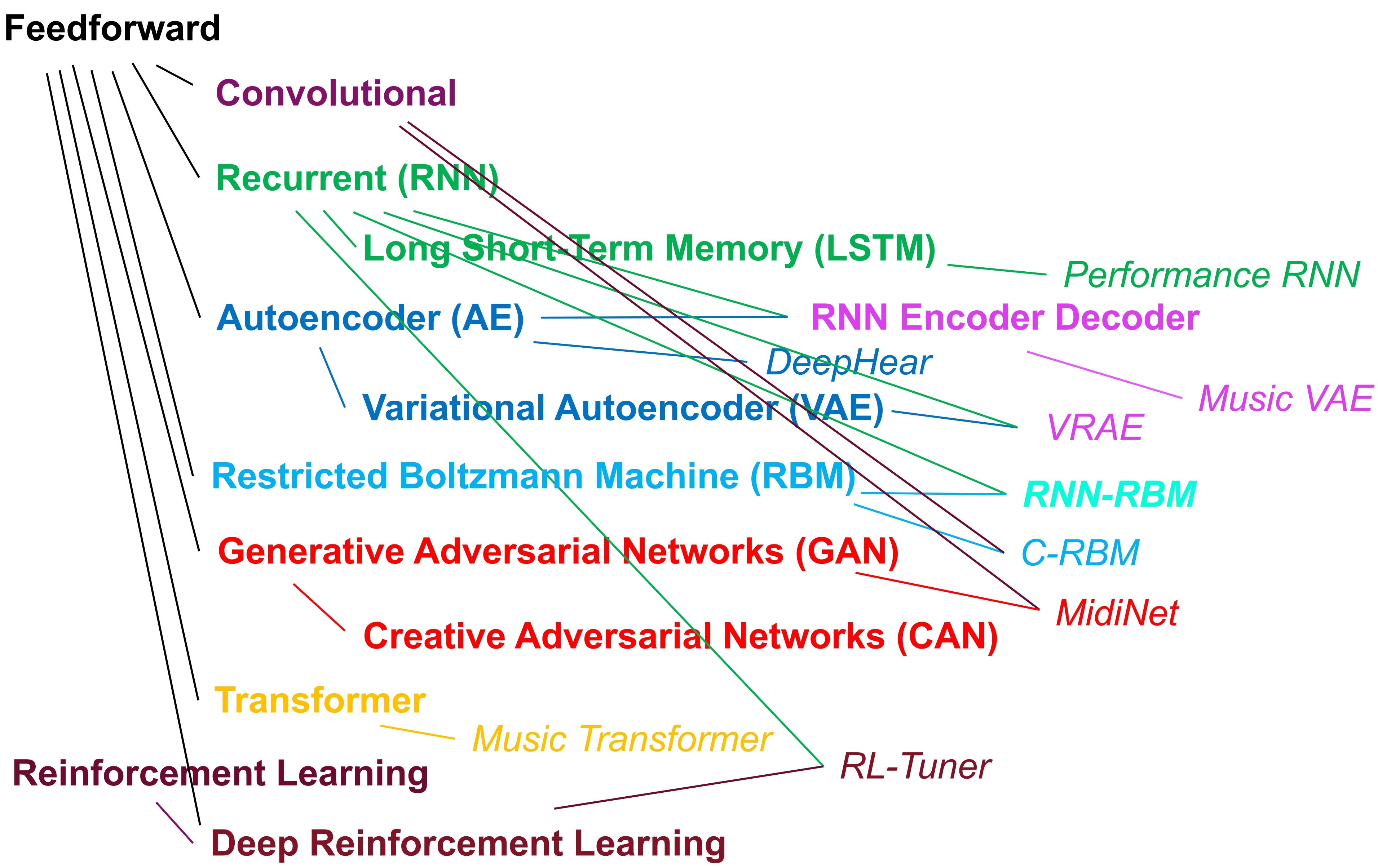
- Automatically Learnt (Induction)
  - Markov Models
  - Neural Models

- Style Automatic Learned from a Corpus (Composer, Form, Genre…)
  - Melody
  - Harmony
  - Counterpoint
  - Orchestration
  - Production

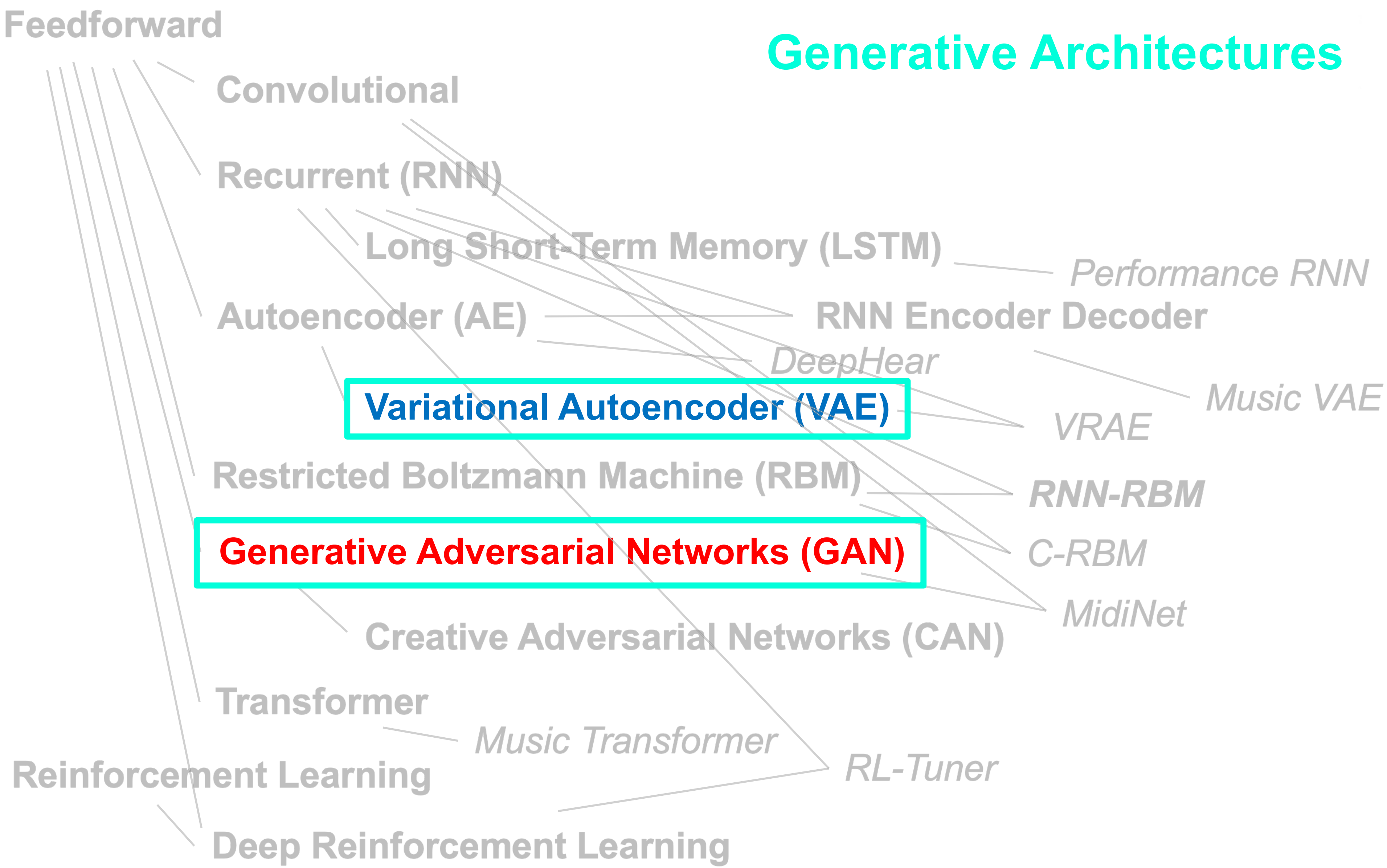
Flow Machines [Pachet et al. 2012]

- Machine Learning Techniques
  - Neural Networks, Deep Learning, Reinforcement Learning
  - (and other models/techniques, Ex: Markov Models)

Deep Learning Phylogenetics



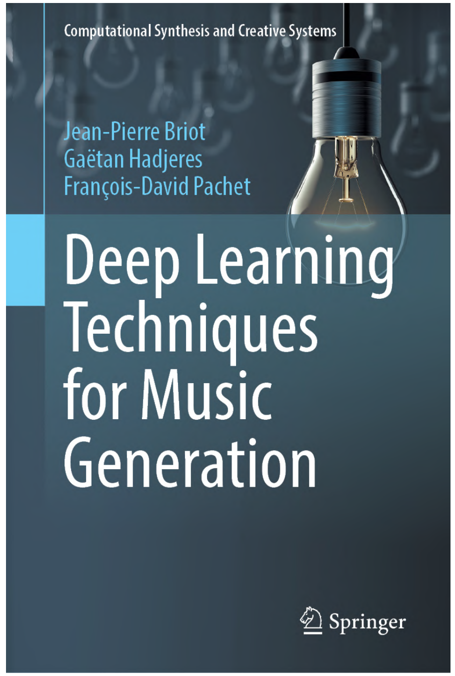
Deep Learning Phylogenetics



Self-References for More Information

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

https://www.springer.com/br/book/9783319701622



ArXiv version:

https://arxiv.org/abs/1709.01620

UNIRIO Course:

http://www-desir.lip6.fr/~briot/cours/unirio3/

Slides and programs

0. General Introduction

Slides

1. Introduction to Computer Music

Slides

2. Introduction to Deep Learning

Slides

MNIST handwritten digit classification Code
Version without one hot Code
Version with one hidden layer Code
Version with convolutions Code

3. Generation by Feedforward Architectures

Slides

DeepMusic Representation Code
DeepMusic Config Code
DeepMusic Metrics Code
Deep Music README

DeepMusic Bach chorale counterpoint Feedforward generator Code

Original Bach chorale from training dataset Midi
DeepMusic Bach chorale from training dataset counterpoint regenerated Midi
Original Bach chorale from test dataset Midi
DeepMusic Bach chorale counterpoint from test dataset regenerated Midi
Brazilian hymn Midi
DeepMusic Brazilian hymn counterpoint generated Midi

4. Generation by Autoencoder Architectures

Slides

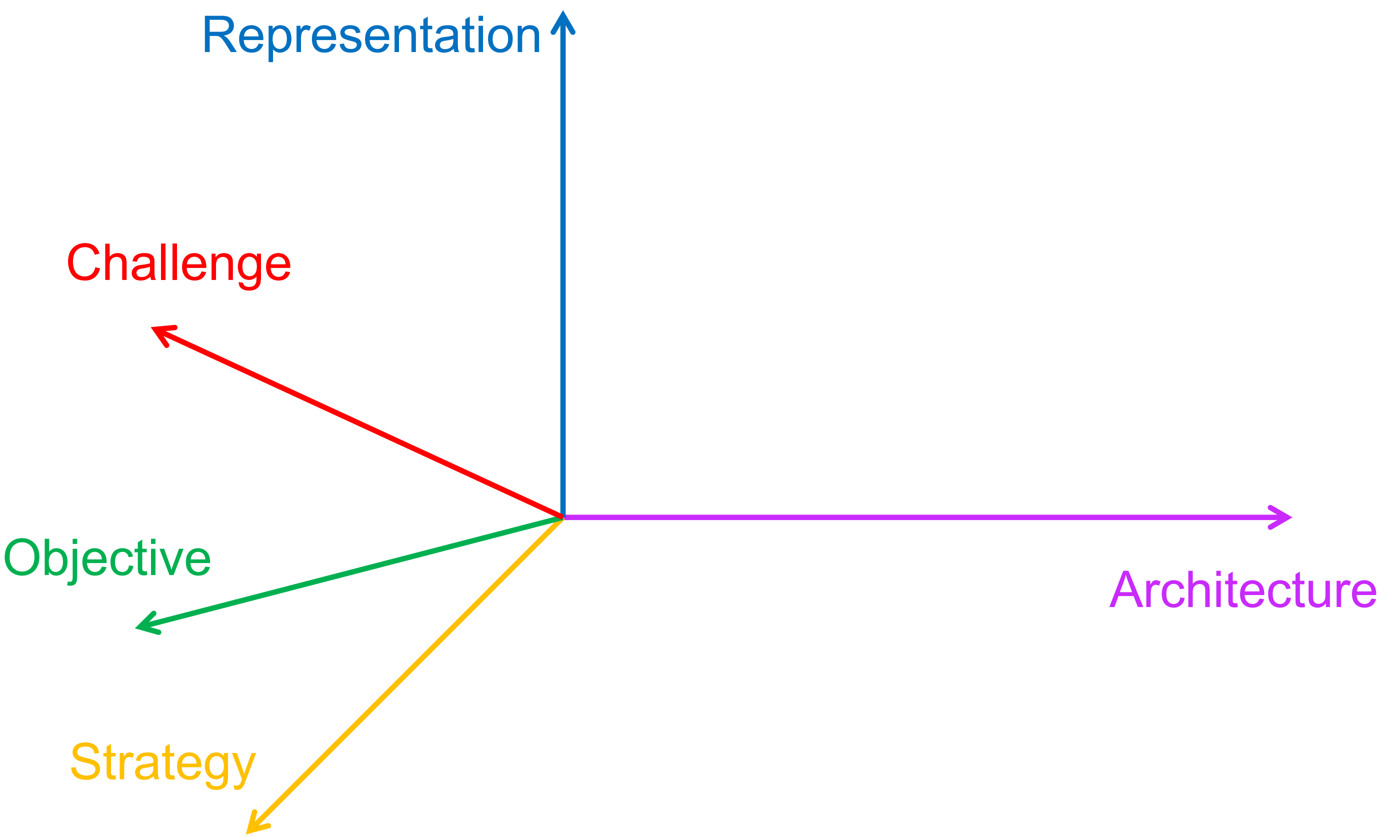
MNIST handwritten digit Autoencoder generator Code

DeepMusic Bach chorale melody Autoencoder generator Code

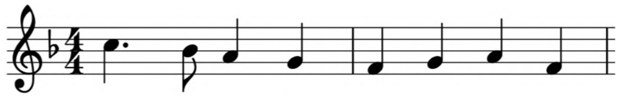
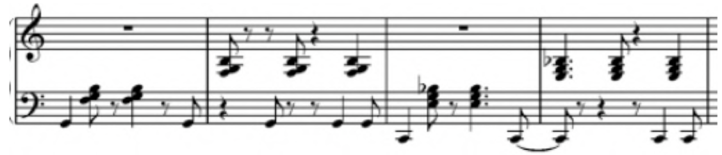
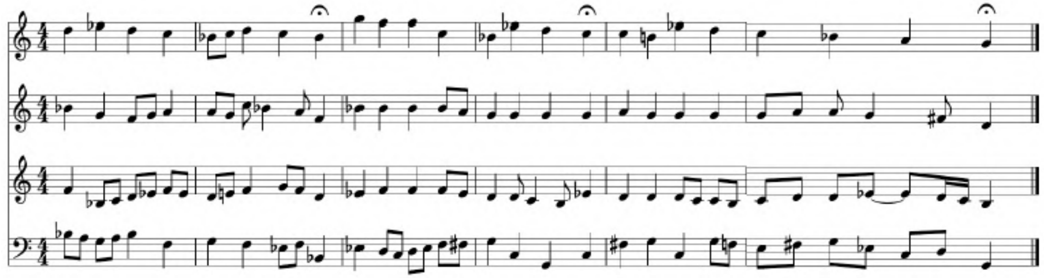
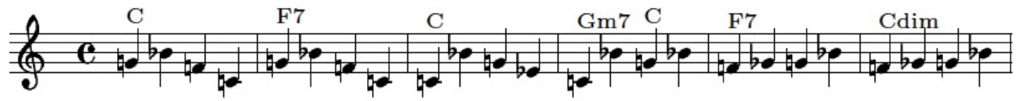
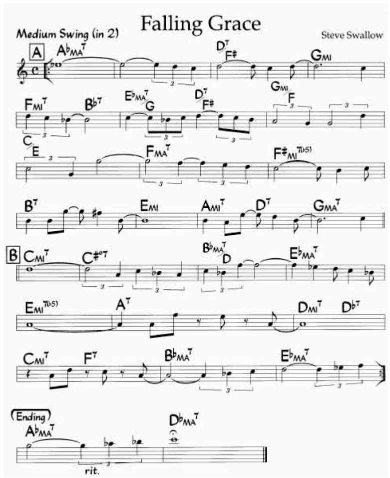
Melody generated - label elements all 0 Midi
Melody generated - label elements all 0 Midi
Melody generated - label elements random [0, 1] Midi

Survey/Analysis

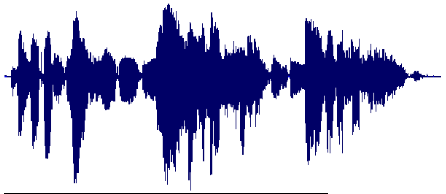
4+1 dimensions



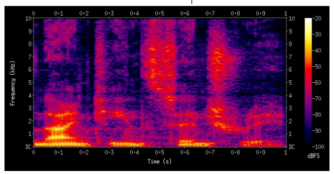
Objective

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

Representation

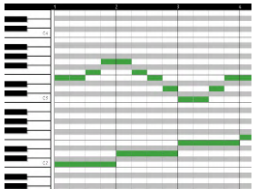
- Signal
  - Waveform



  - Spectrum



- Symbolic
  - MIDI
  - Piano roll

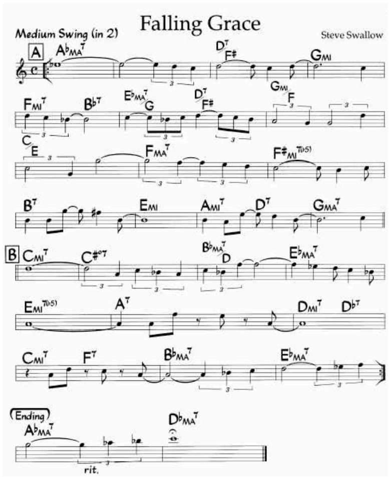


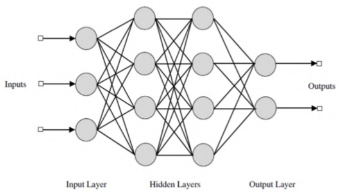
  - Text

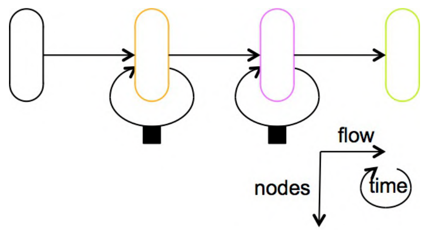
```
|:eA (3AAA g2 fg|eA (3AAA BGGf|eA (3AAA g2 fg|1afge d2 gf:|2afge d2 cd||
|:eaag efgf|eaag edBd|eaag efge|afge dgfg:|
```

  - Chord

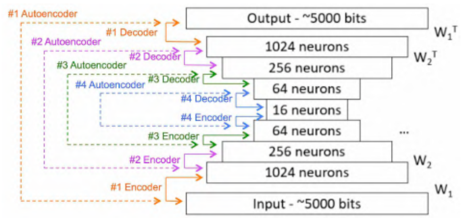
EbMaj7/G

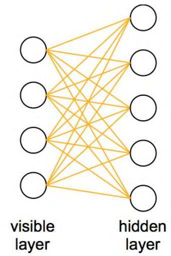
  - Lead sheet
  - Rhythm

Architecture



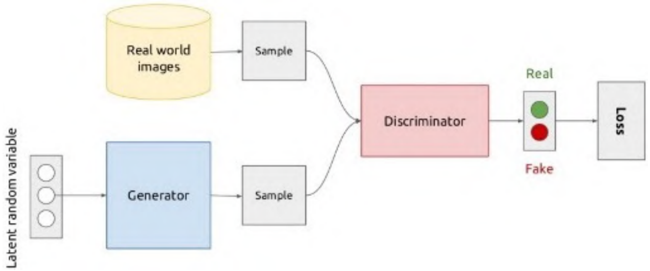


- Feedforward
- Recurrent (RNN)
  - Long Short-Term Memory (LSTM)
- Autoencoder
  - Stacked Autoencoders
- Restricted Boltzmann Machine (RBM)
- Variational Autoencoder (VAE)

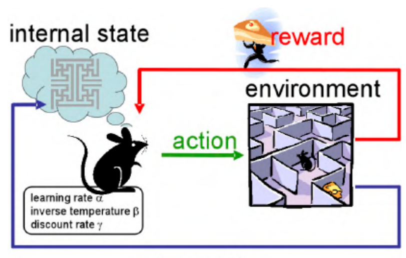




- Patterns
  - Convolutional
  - Conditioning
  - Generative Adversarial Networks (GAN)



- Reinforcement Learning



- Refinement and Compound
  - Ex: VRAE = Variational(Autoencoder(RNN, RNN)) = Variational(RNN Encoder-Decoder)

Refined and Compound Architectures

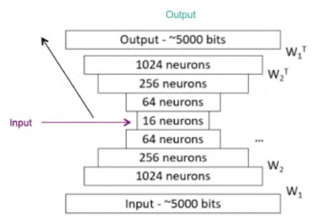
- Composition
  - Bidirectional RNN
  - RNN-RBM
- Refinement
  - Variational(Autoencoder) (VAE)
- Nested
  - Stacked Autoencoder
  - RNN Encoder-Decoder = Autoencoder(RNN, RNN)
- Pattern Instantiation
  - C-RBM = Convolutional(RBM)
  - C-RNN-GAN = GAN(RNN, RNN)
- Compound
  - VRASH = Variational(Autoencoder(RNN, Conditioning(RNN, History))).

Challenge

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**

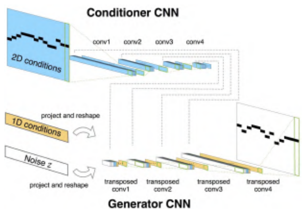
(Generation) Strategy

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



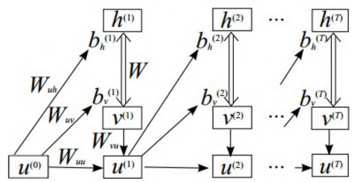
[Sun, 2016]

- Conditioning



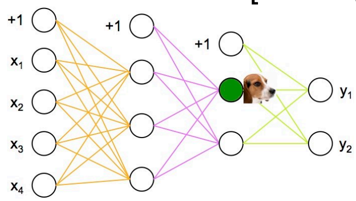
[Yang et al., 2017]

- Sampling

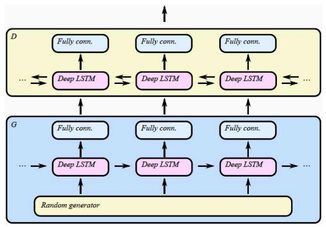


[Boulanger-Lewandowski et al., 2012]

- Input Manipulation

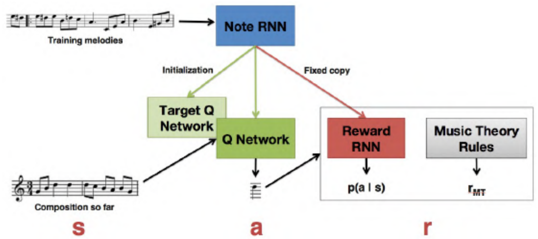


- Adversarial



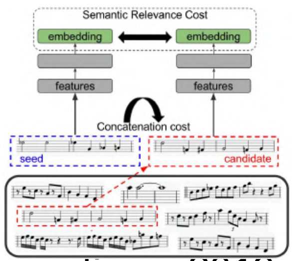
[Mogren, 2016]

- Reinforcement



[Jaques et al., 2016]

- Unit Selection



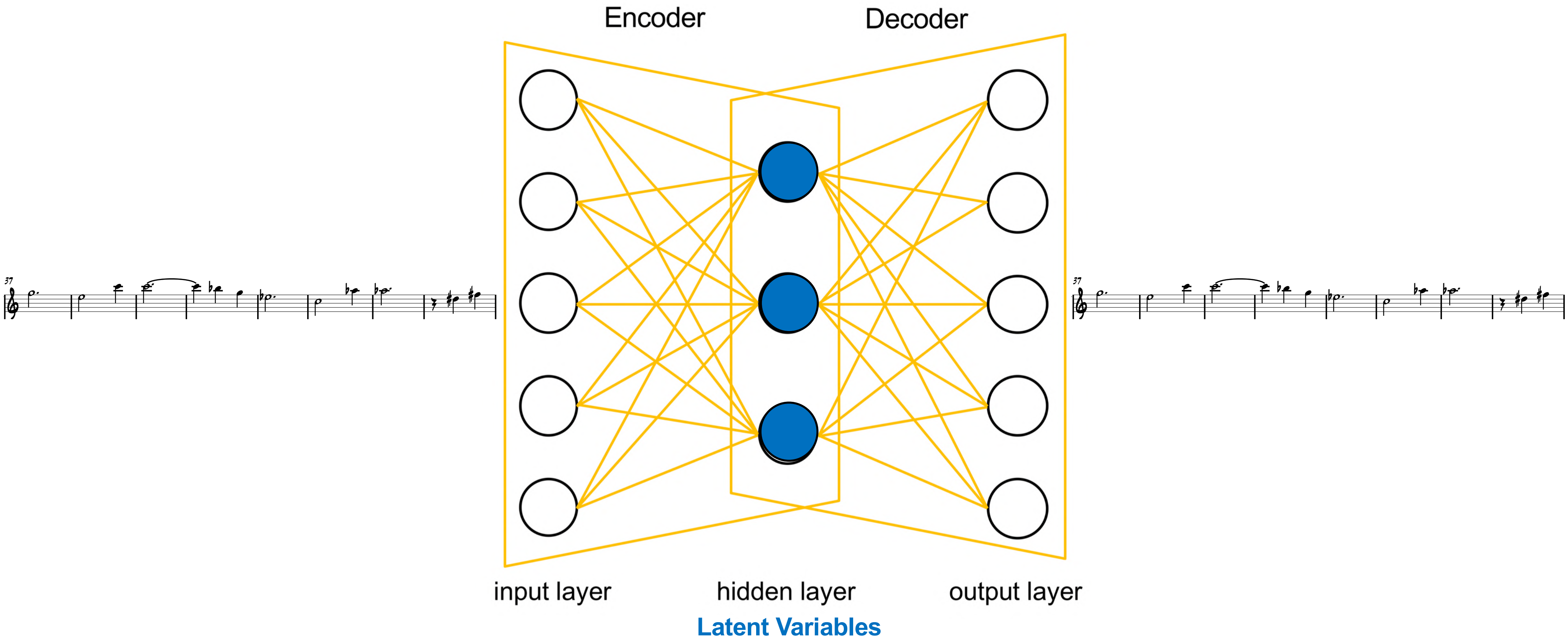
[Bretan et al., 2016]

Generative Architectures

Variational Autoencoder

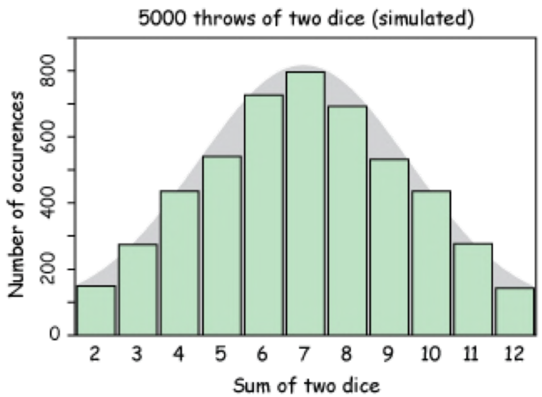
Autoencoder

- Symmetric Neural Network
- Trained with examples as input and output
- Hidden Layer will Learn a Compressed Representation at the Hidden Layer (Latent Variables)

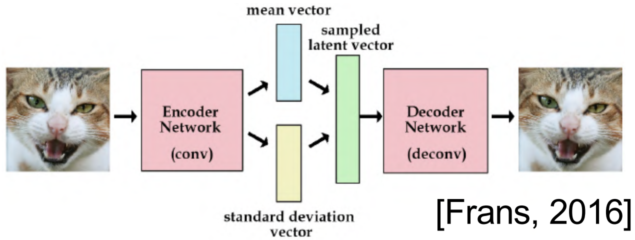


Variational Autoencoder (VAE) [Kingman & Welling, 2014]

- *Additional Constraint:*
- Encoded representation (latent variables *z*) follows some prior probability distribution *p(z)*, usually, a Gaussian distribution (normal law)



- *Reparameterization Trick*



[Frans, 2016]

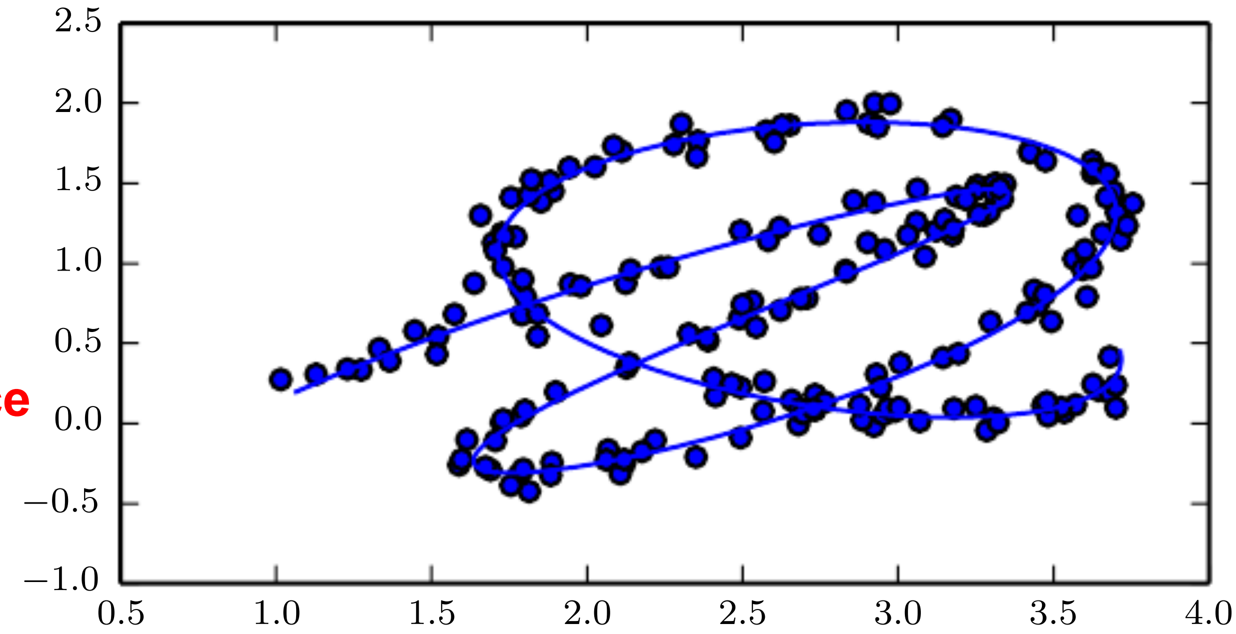
- The VAE decoder part will learn the relation between a Gaussian distribution of the latent variables and the learnt examples

- A VAE is able to learn a *smooth* latent space *mapping* to realistic examples

Representation/Manifold Learning

Manifold :
Set of connected points

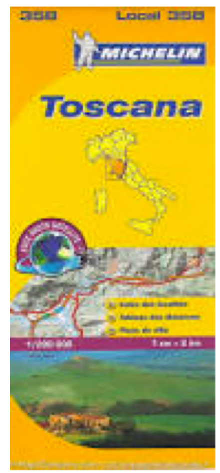
In a high-dimensional space



But can be approximated by a smaller number of dimensions, each dimension corresponding to a local variation

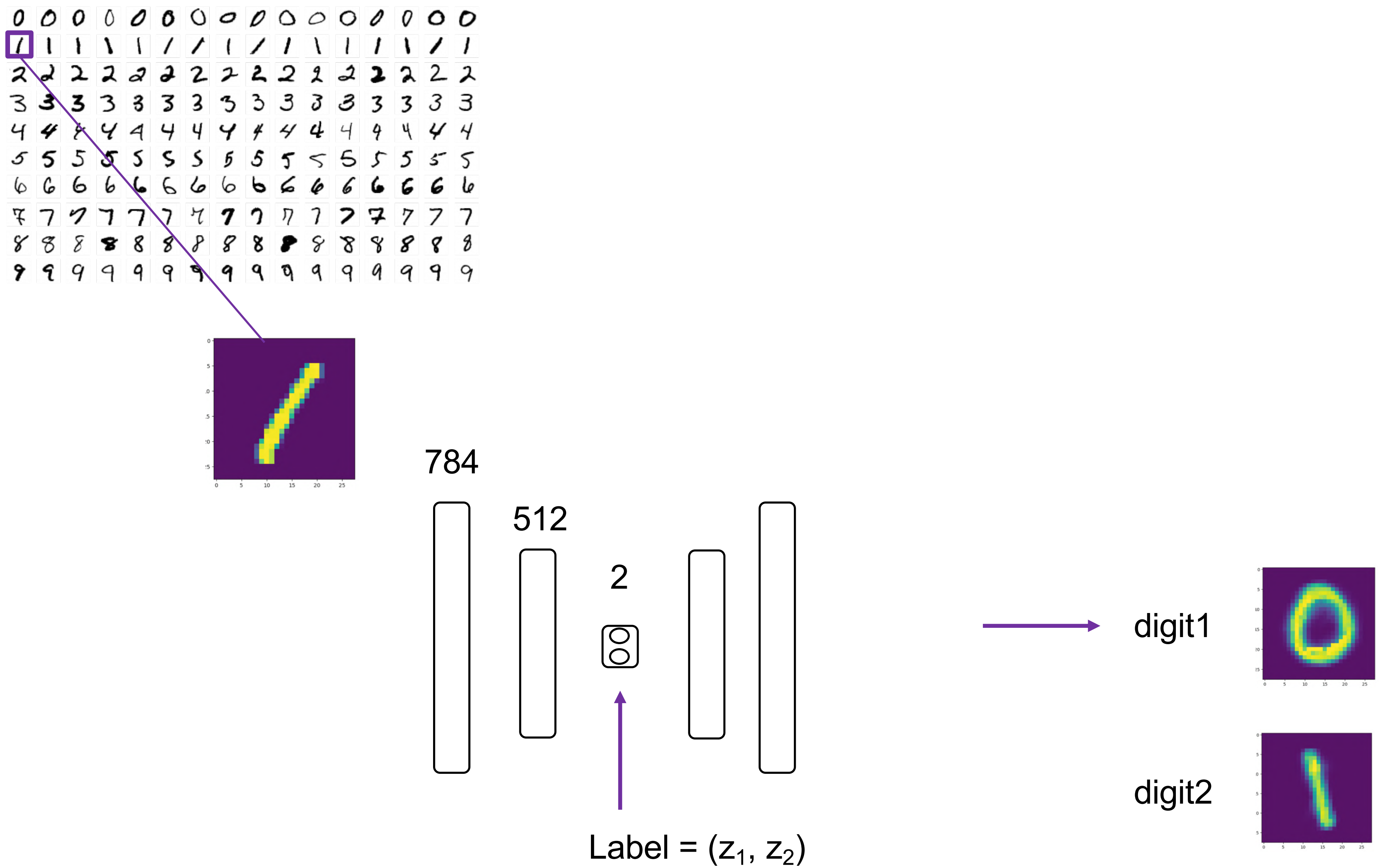
Analogy:

3D Earth

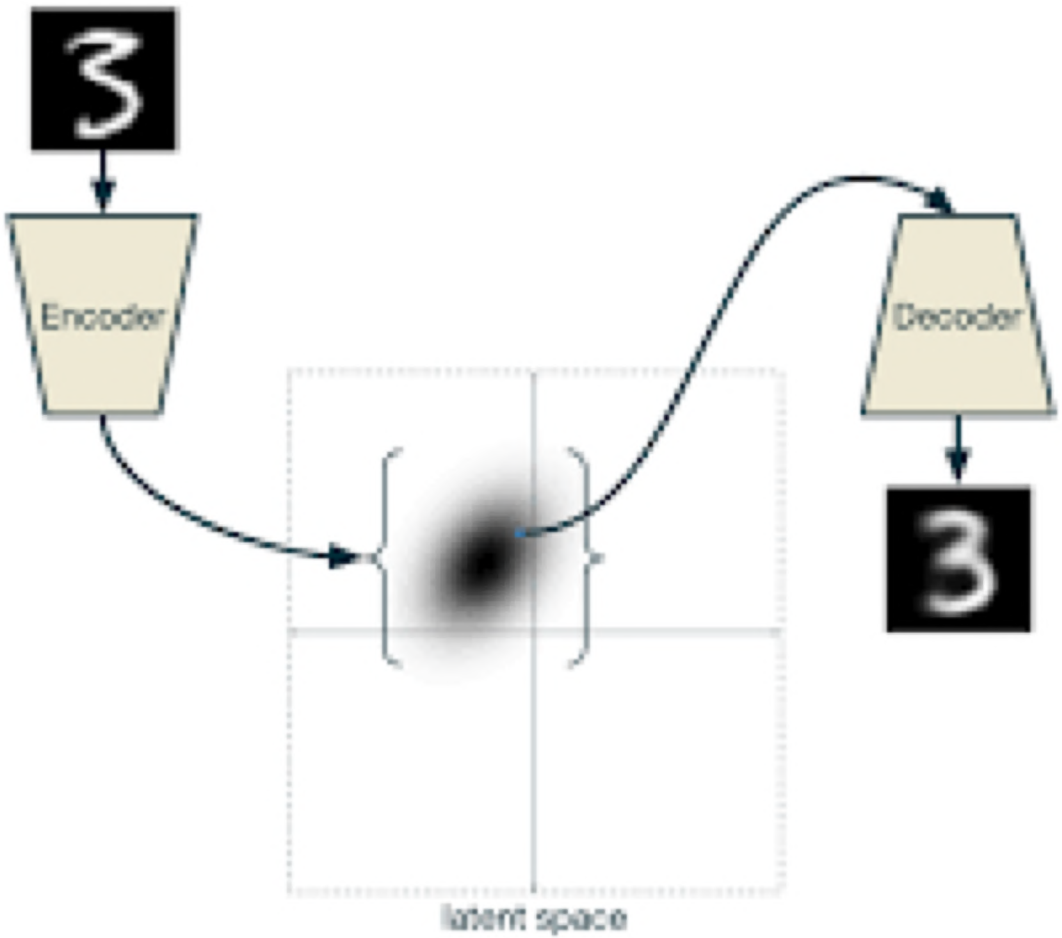


2D Map

VAE MNIST [Keras/Cholet, 2016]



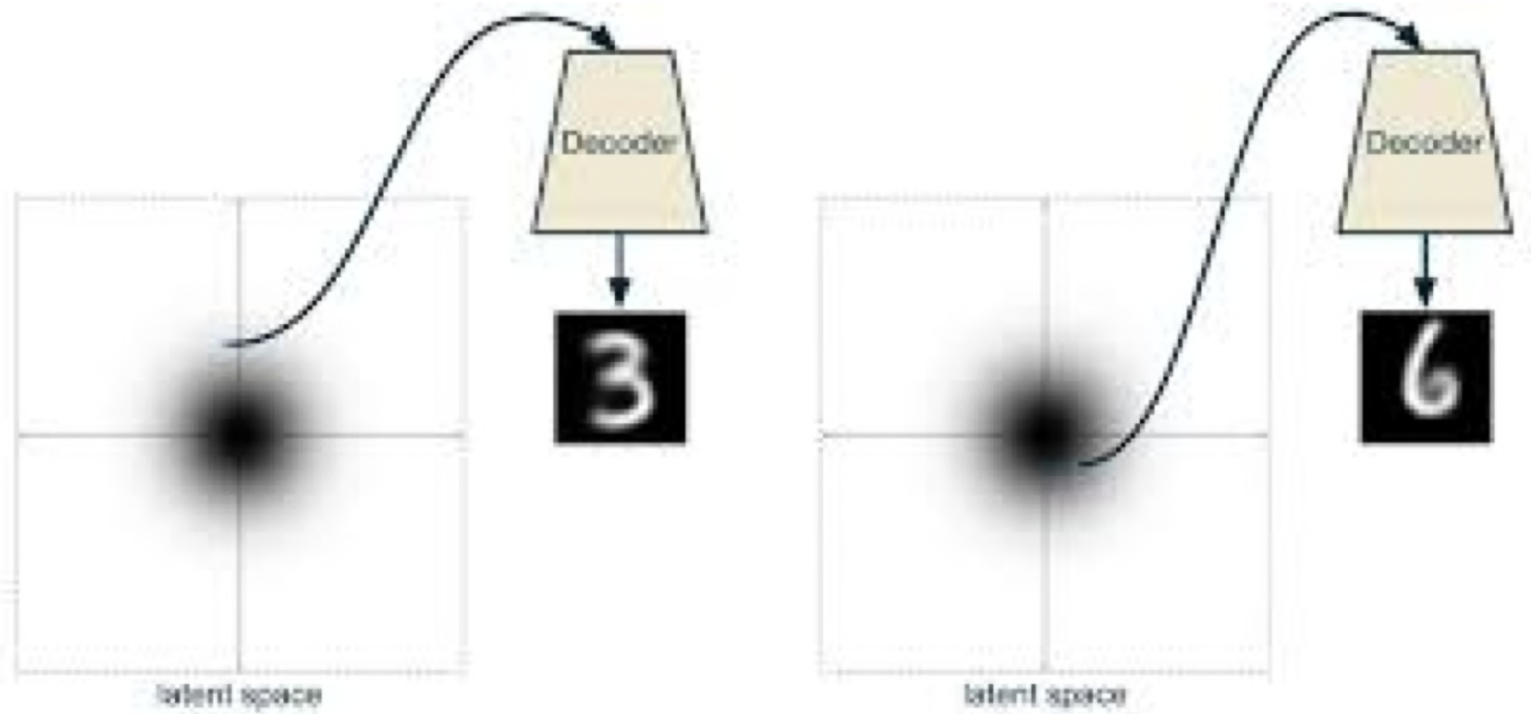
Variational Autoencoder



[Dykeman, 2016]

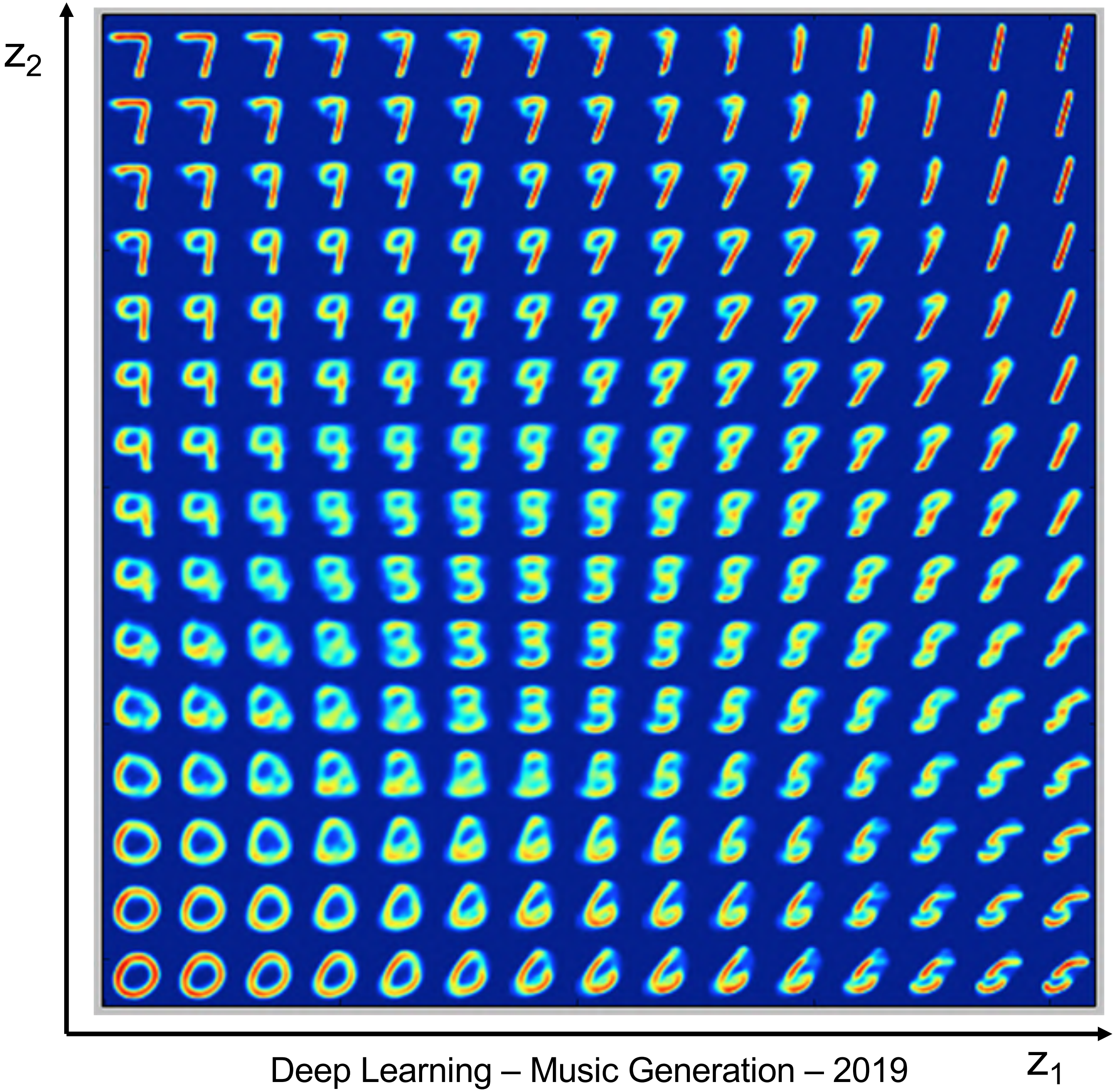
Variational Autoencoder

Generation
by Exploring the Latent Space
and Decoding



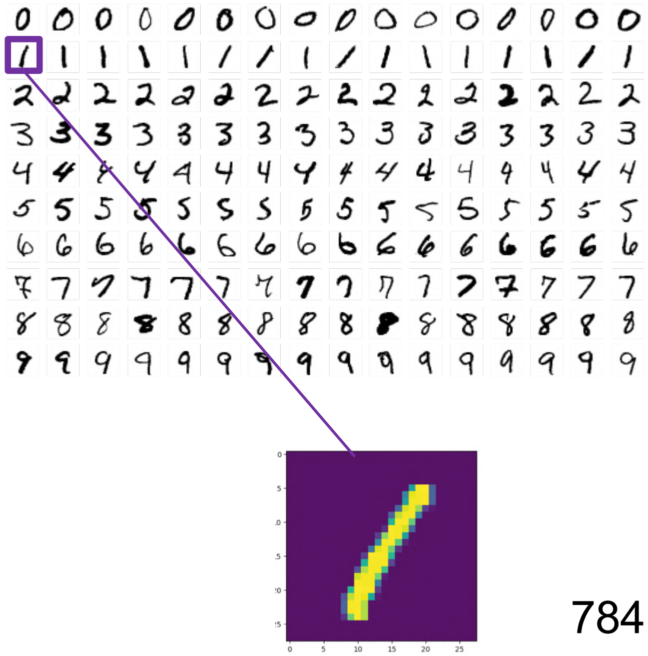
[Dykeman, 2016]

VAE MNIST [Keras/Cholet, 2016]

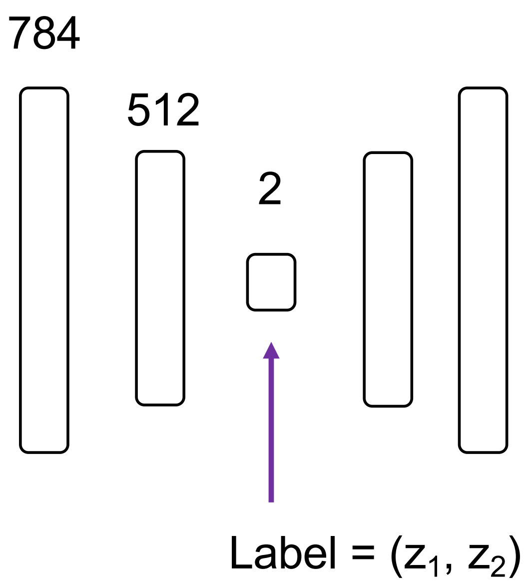


[Cholet, 2016]

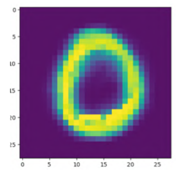
VAE Magic



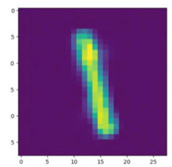
How is it possible ?
Compress 784 variables into 2
and reconstruct the original ?



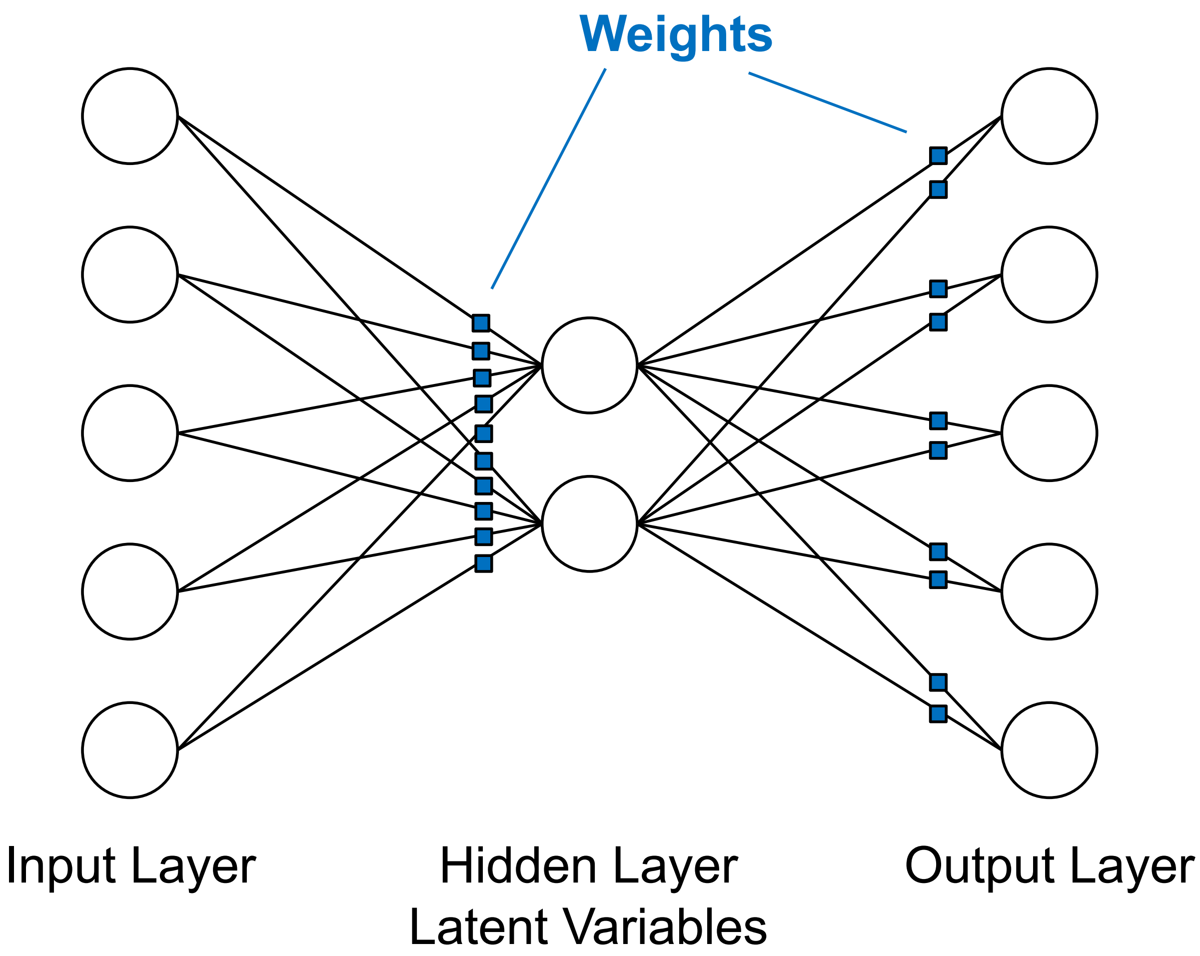
digit1



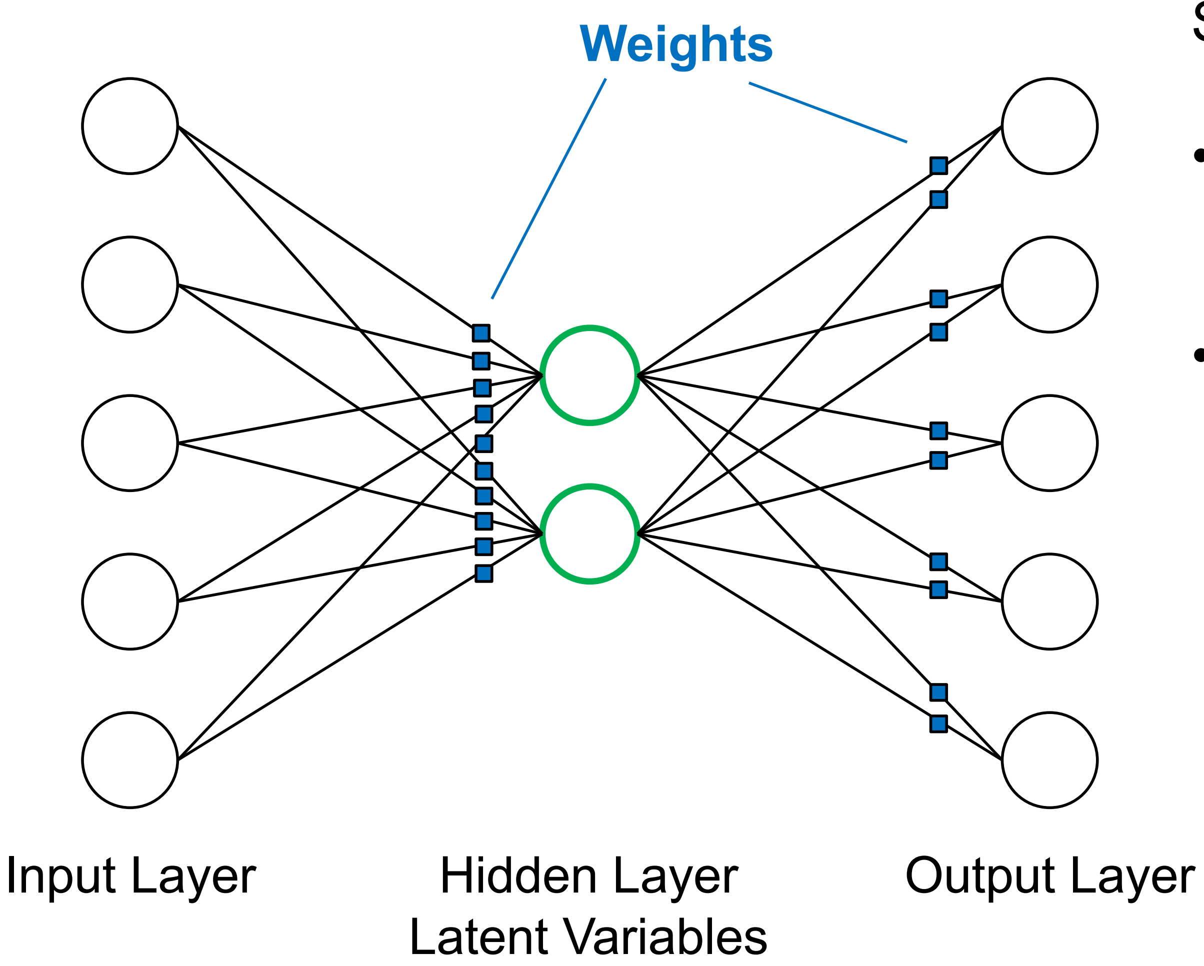
digit2



VAE Magic Revealed



VAE Magic Revealed



Split/Extract between

- Common Data:
  Weights
- Variable/Discriminative Data:
  Latent Variables

Variational Generation

Exploration of the latent space with various operations to control/vary the generation of content

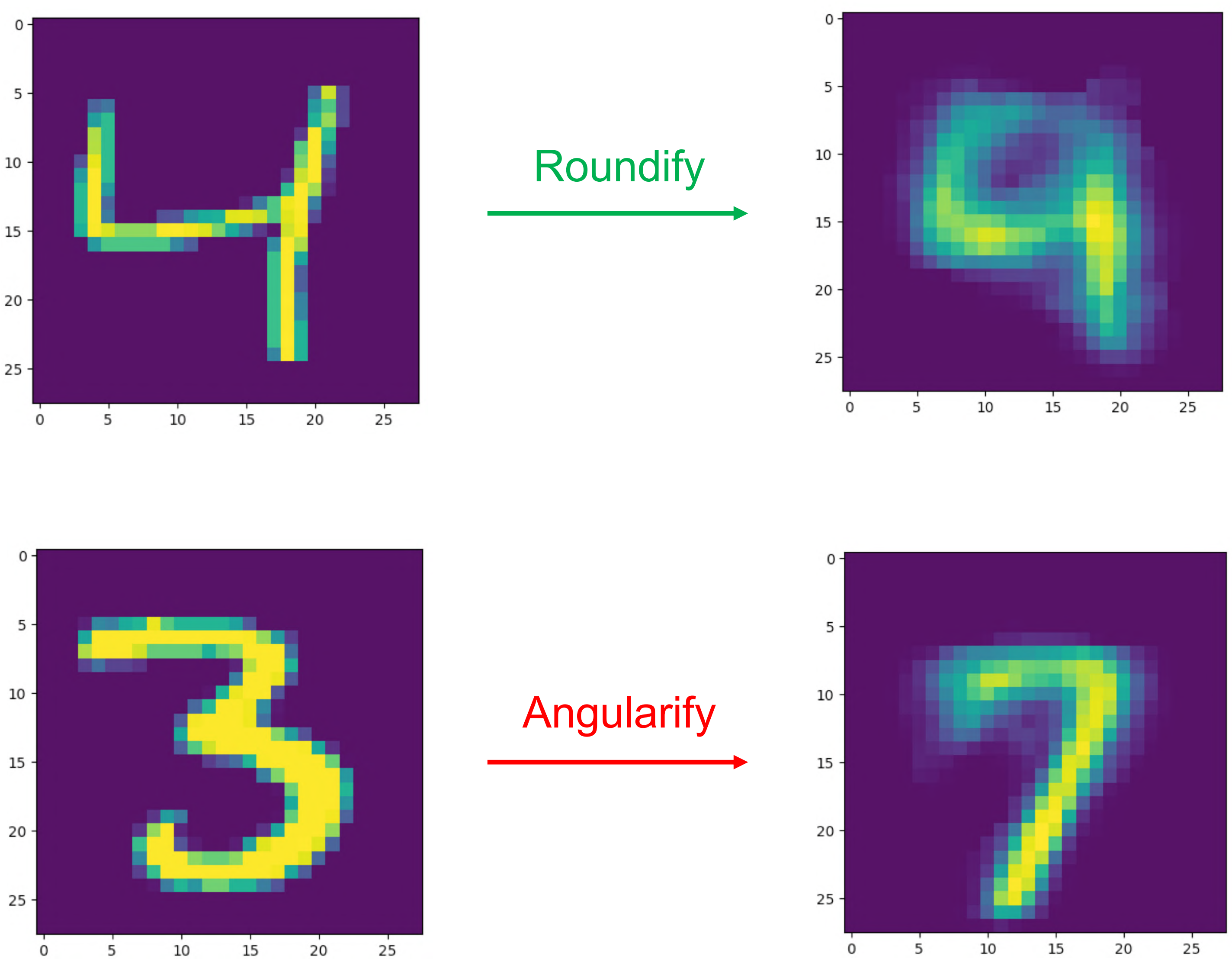
Ex:

- Translation
- Arbitrary path
- Interpolation (morphing) (between points)
- Averaging (of some points)
- Attribute arithmetic
  - Addition or subtraction of an attribute vector capturing a given characteristic
  - This attribute vector is computed as the average latent vector for a collection of examples sharing that attribute (characteristic)

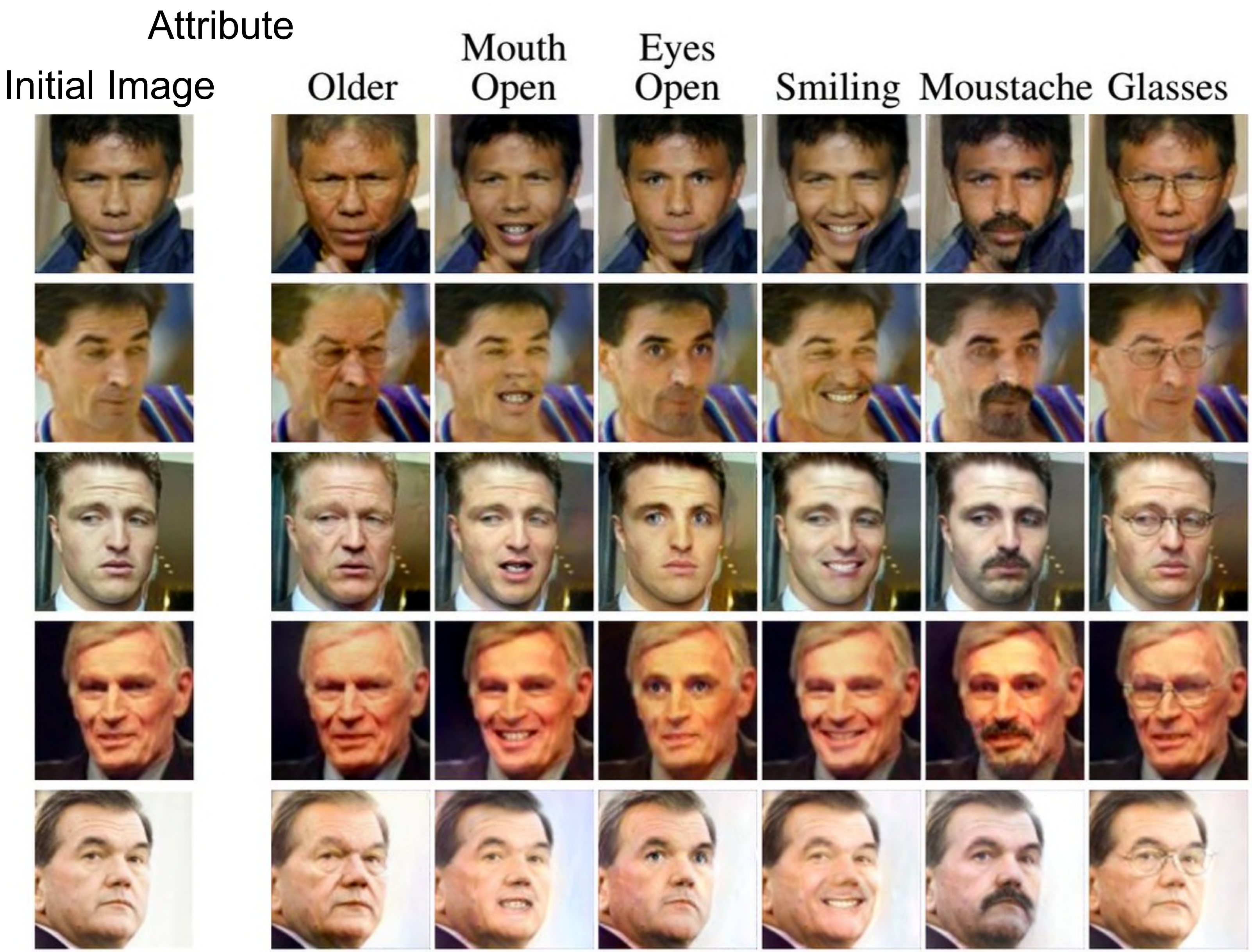
Attribute Arithmetic

- (Characteristics) Attribute Arithmetic
  - Addition or subtraction of an attribute vector capturing a given characteristic
  - This attribute vector is computed as the average latent vector for a collection of examples sharing that attribute (characteristic)
- Select a set of round and angular digits images
  - round_numbers = [3, 6, 8, 9]
  - angular_numbers = [1, 4, 7]
- Encode each one
  - _, _, z_round_elements = encoder.predict(np.array(round_elements))
  - _, _, z_angular_elements = encoder.predict(np.array(angular_elements))
- Compute the mean of the (z) corresponding latent variable values
  - z1_mean_round_elements = mean(z1_round_elements)
  - z1_mean_angular_elements = mean(z1_angular_elements)
  - …
- Do attribute arithmetic
  - def roundify(z):
  -     z_rounded = [z[0] + z1_mean_round_elements, z[1] + z2_mean_round_elements]
  -     return(decoder.predict(np.array([z_rounded]))[0])

Examples

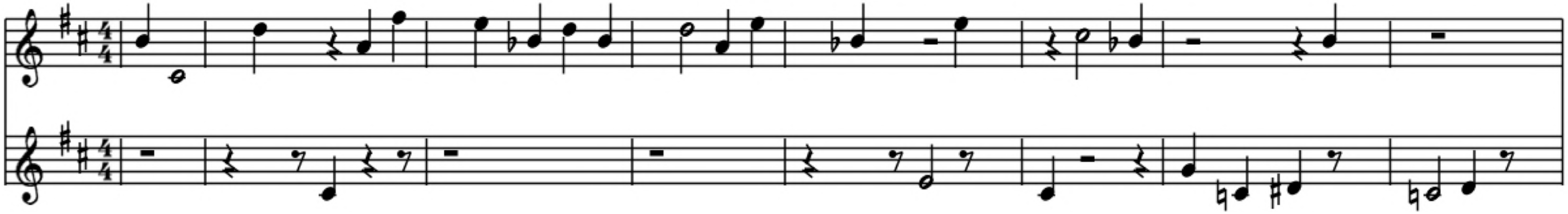
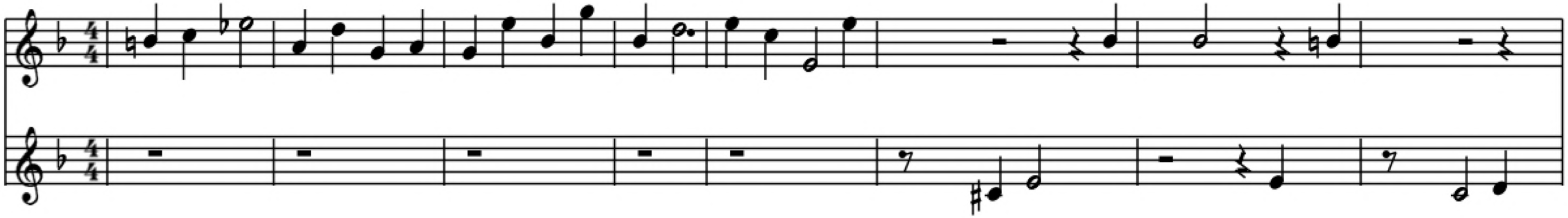
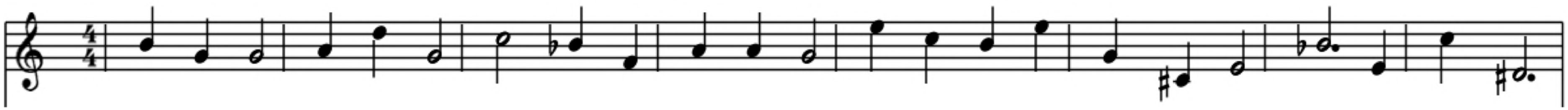
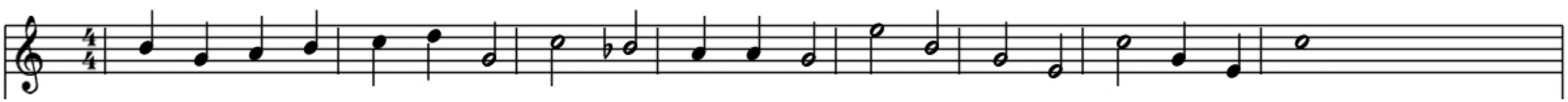
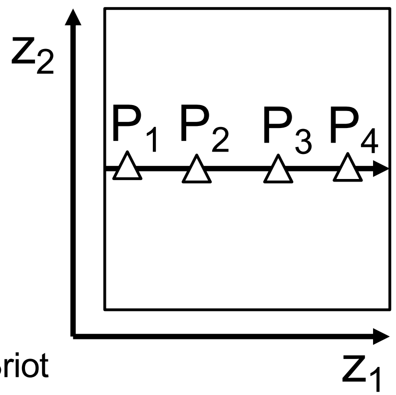


Variational Autoencoder
Ex. of Attribute Arithmetic

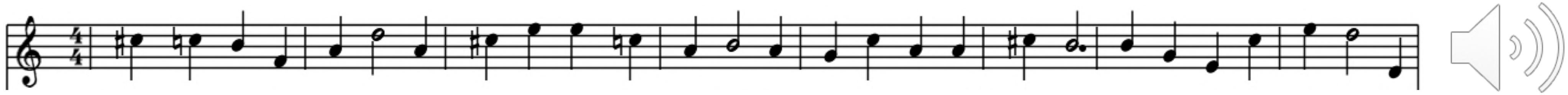
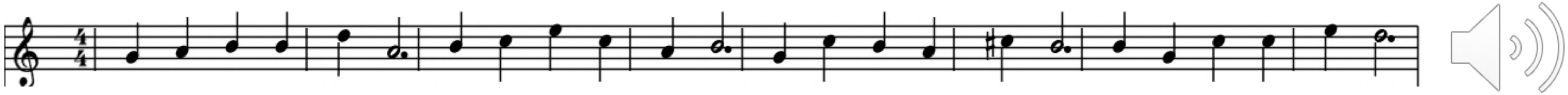
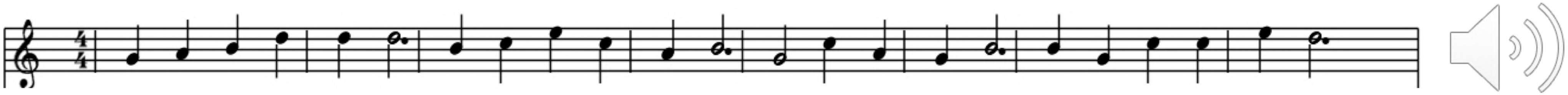
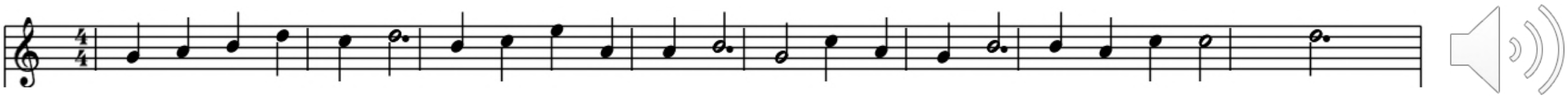
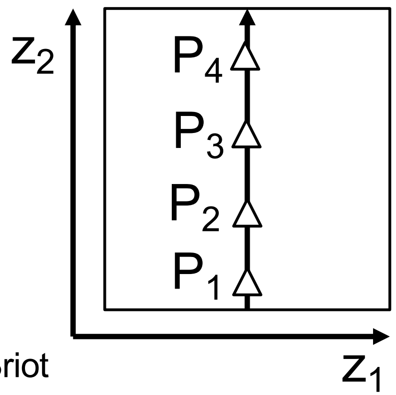


[Li et al., 2016]

Bach Choral Soprano Melodies
Z_1 Step Interpolation

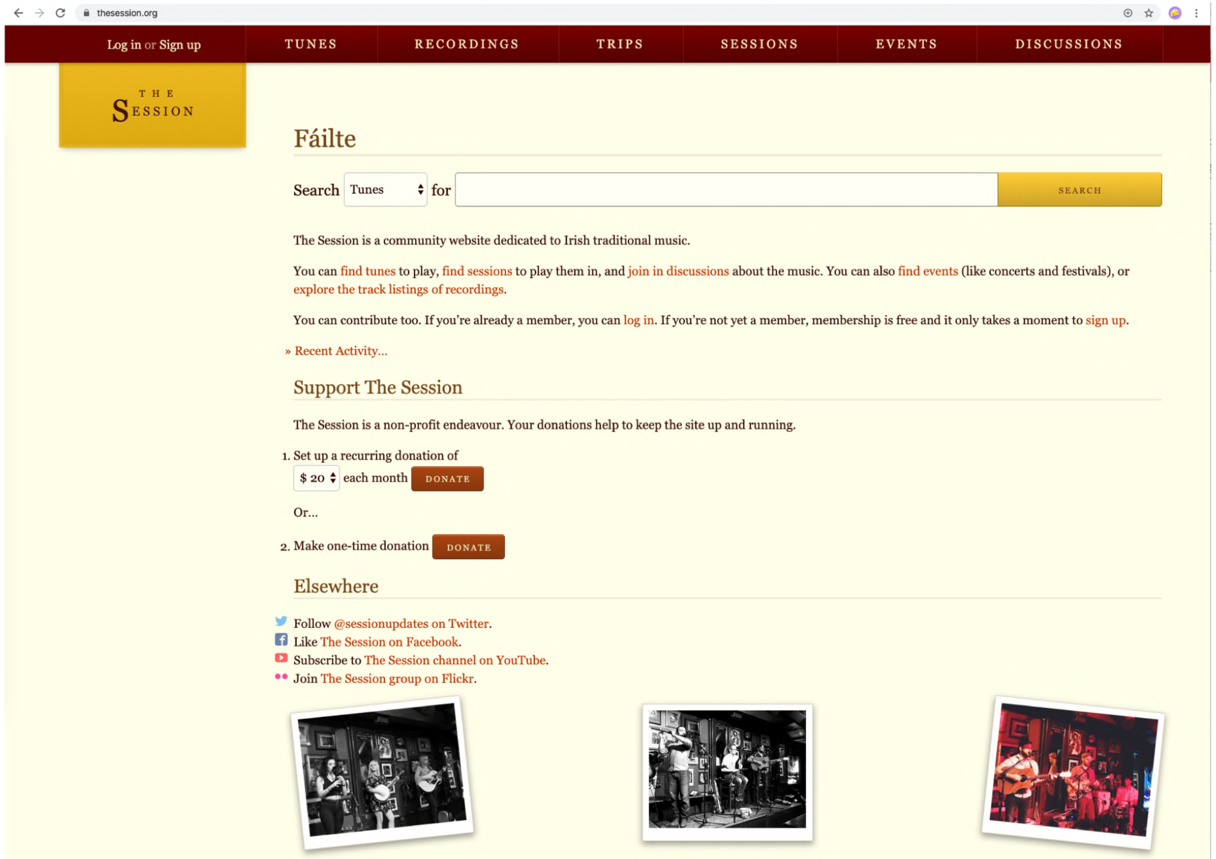
P_1

P_2

P_3

P_4



Bach Choral Soprano Melodies
Z_2 Step Interpolation

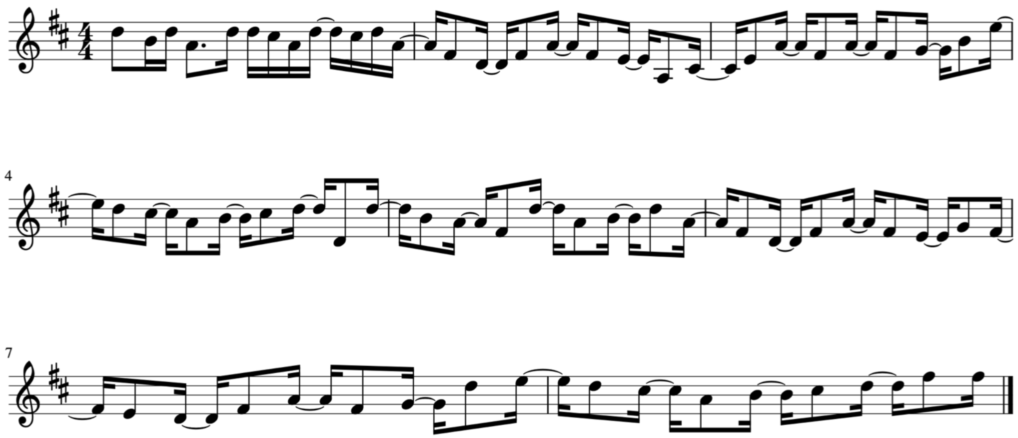
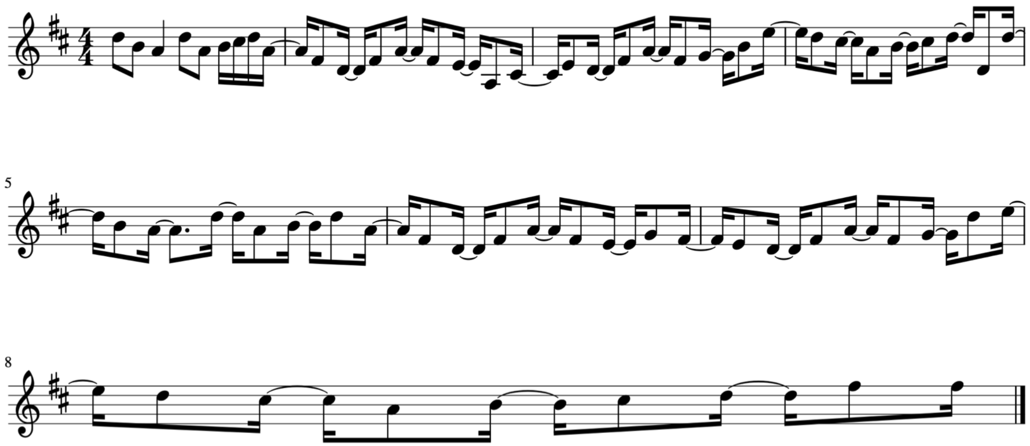
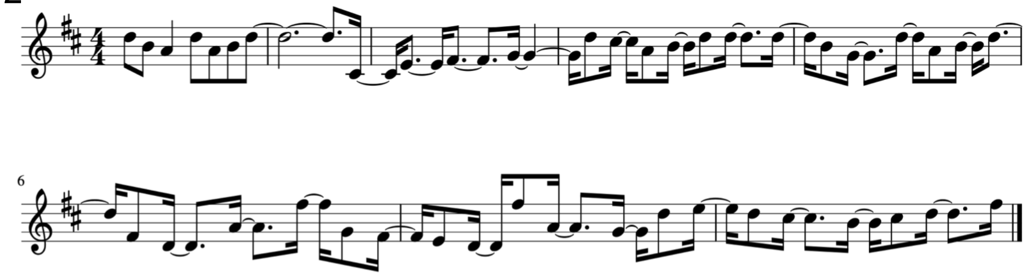
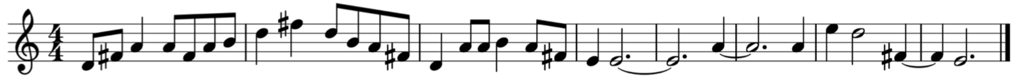
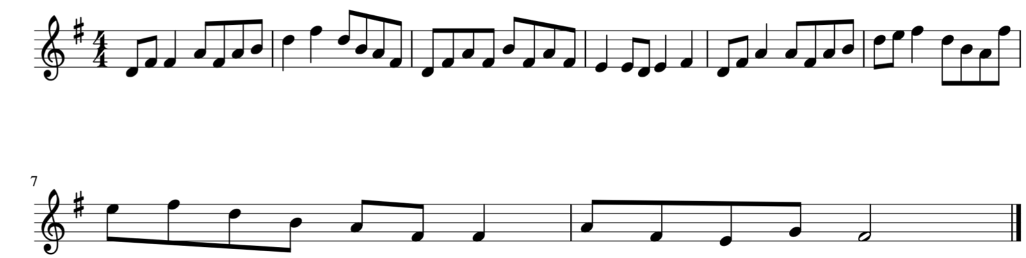
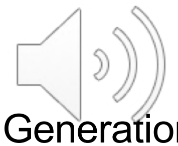
P_1

P_2

P_3

P_4



Celtic Music

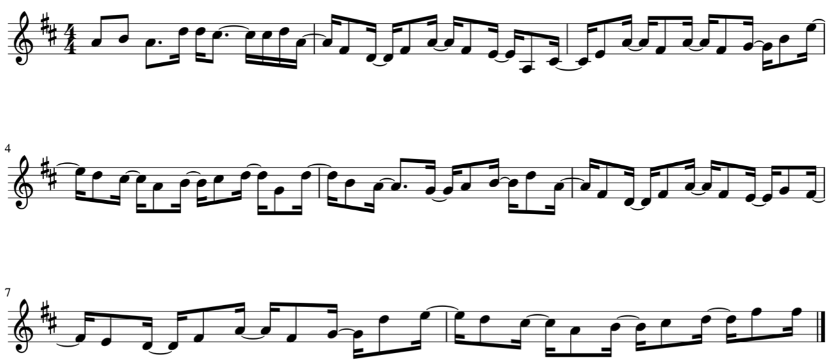
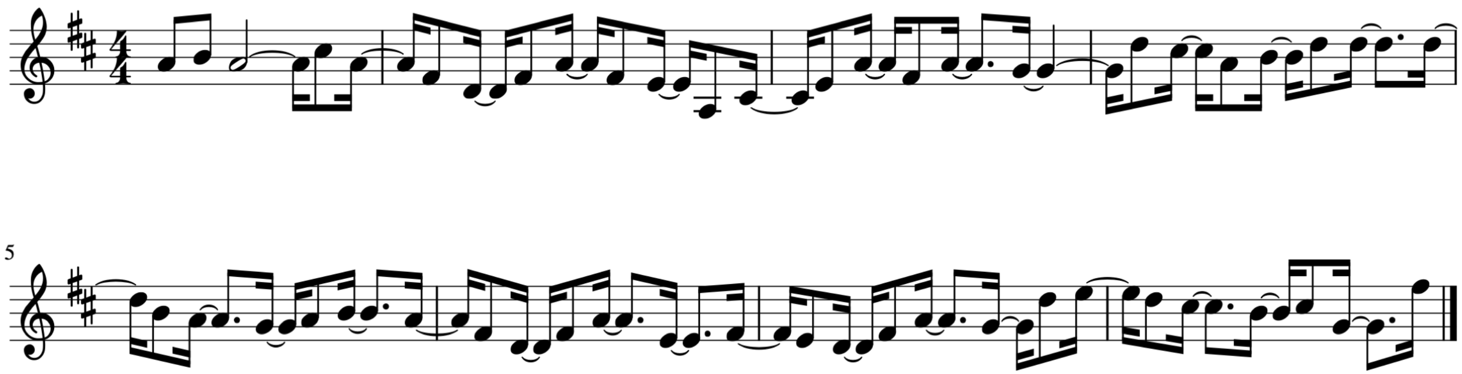
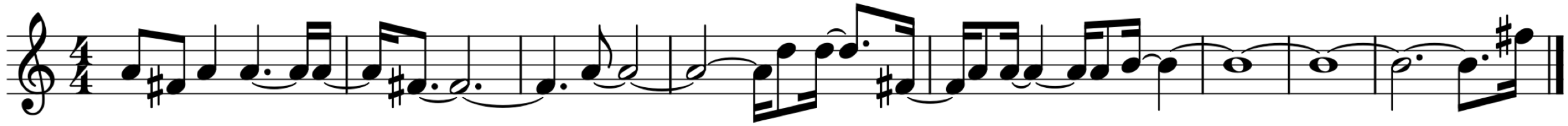
- Training Examples/Corpus:
- In ABC format (see later) -> Music21 -> representation
- 29 songs from the Session (https://thesession.org/)
- In the same key (D major) and the same rhythm metric (4/4)

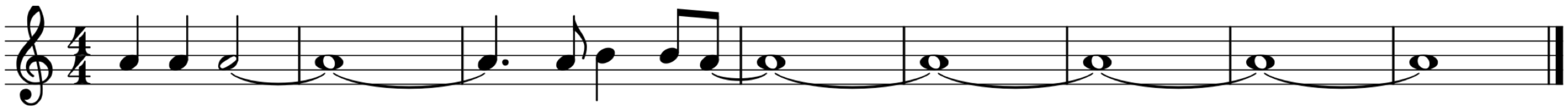


Celtic Melodies
Z_1 Step Interpolation

P_0

P_1

P_2

P_3

P_4



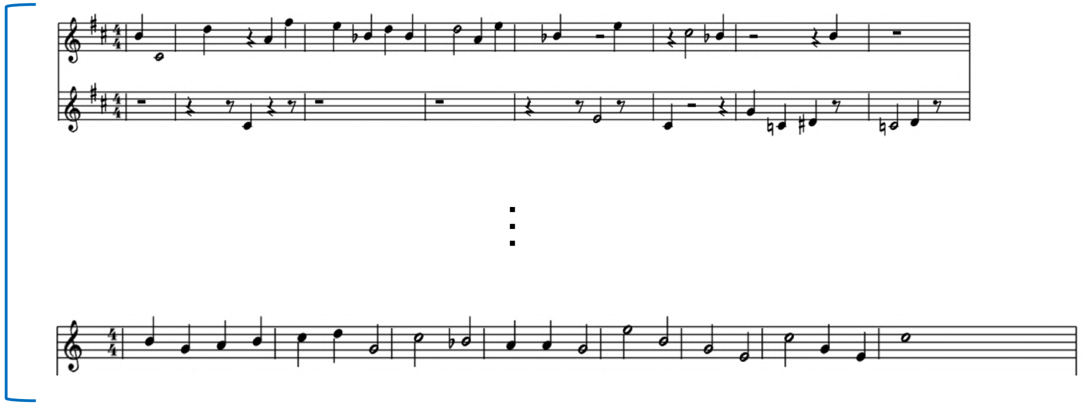
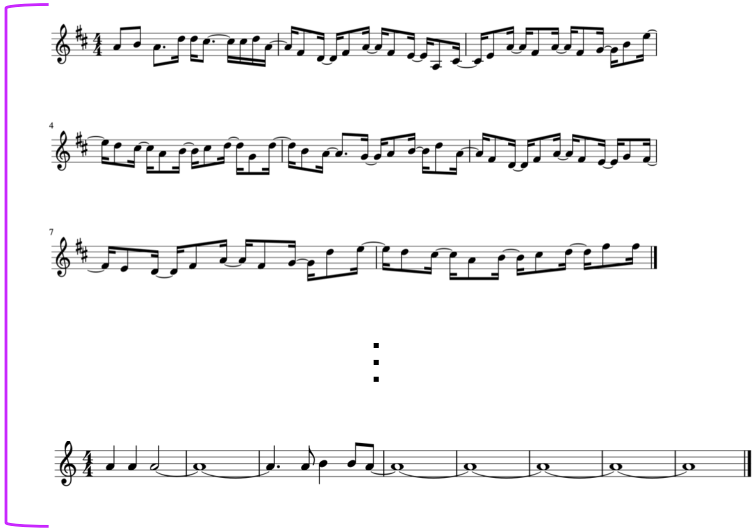
Celtic Melodies
Z_2 Step Interpolation

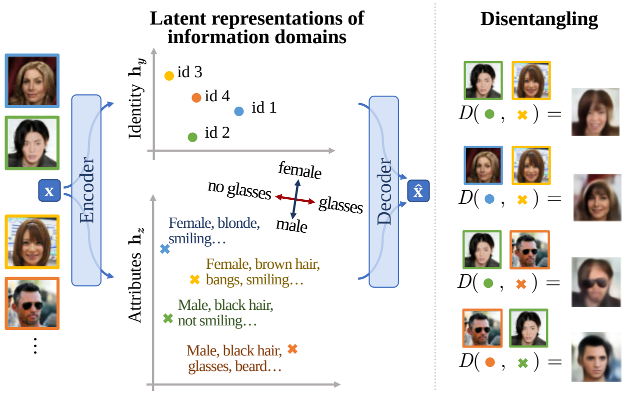
P_1

P_2

P_3

P_4



Disentanglement (1/3)

- $z_1 \perp z_2$

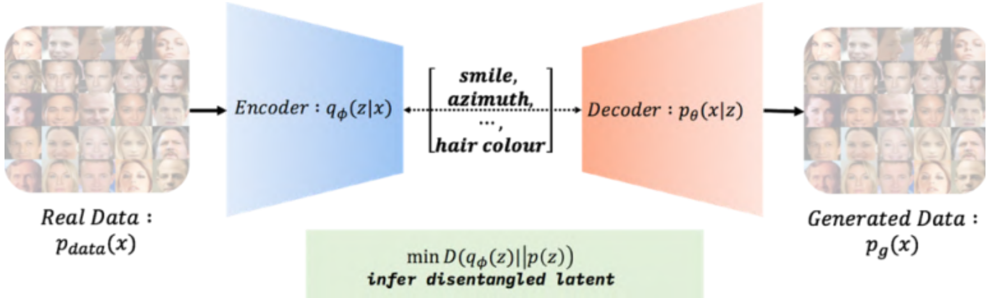
- Ex: Pitch Range $\perp$ Duration Range

- Ex: Gender $\perp$ Glasses



[Robert el al, 2019]

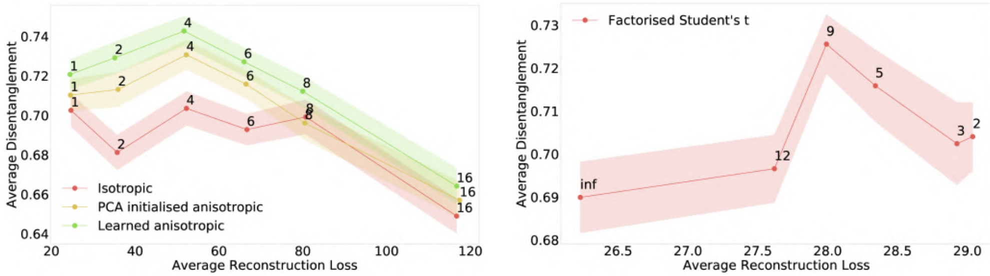
Disentanglement (2/3)

- Adding Term to the Reconstruction Loss [IBM Research, 2018]

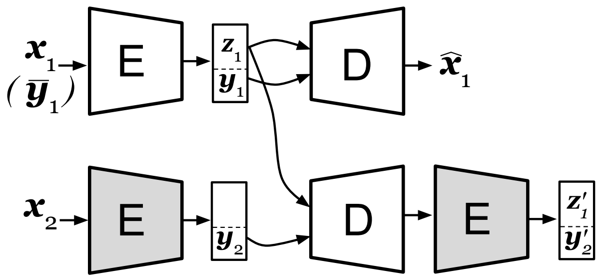


$$\max_{\theta,\phi} \mathbb{E}_{\mathbf{x}} \left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))\right] - \lambda D(q_\phi(\mathbf{z})\|p(\mathbf{z}))$$

- Deconstructing the β-VAE [Mathieu et al., 2019]

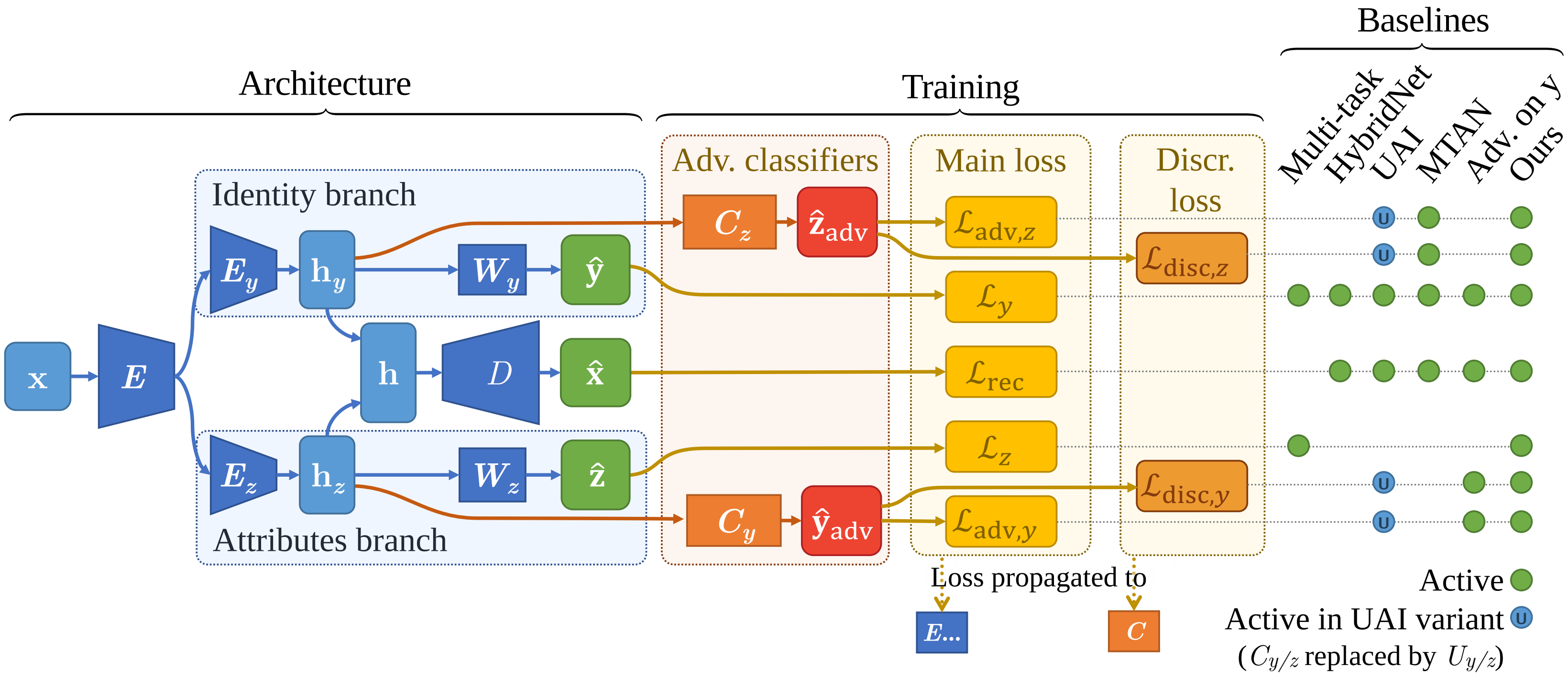


- Reconstruction Trade-off via Jacobian Supervision [Lezama, 2019]



Disentanglement (3/3)

- Dual Branch Adversarial (DualDIS) [Robert el al, 2019]
- Separate Dimensions in Distinct Autoencoders (E_y and E_z)
- Measure the Presence/Absence of the Dual Dimension through a Dual (Adversarial) Classifier (C_z and C_y)
- Objective(s): Not Being Able to Classify Properly the Dual Dimension

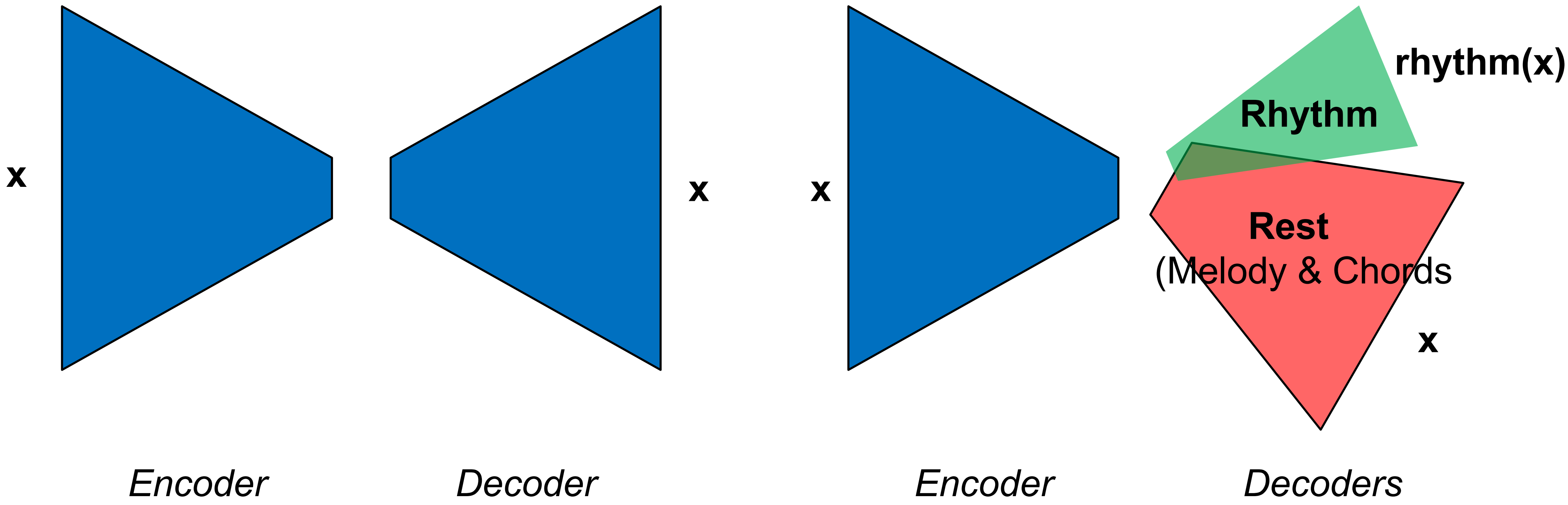


[Robert el al, 2019]

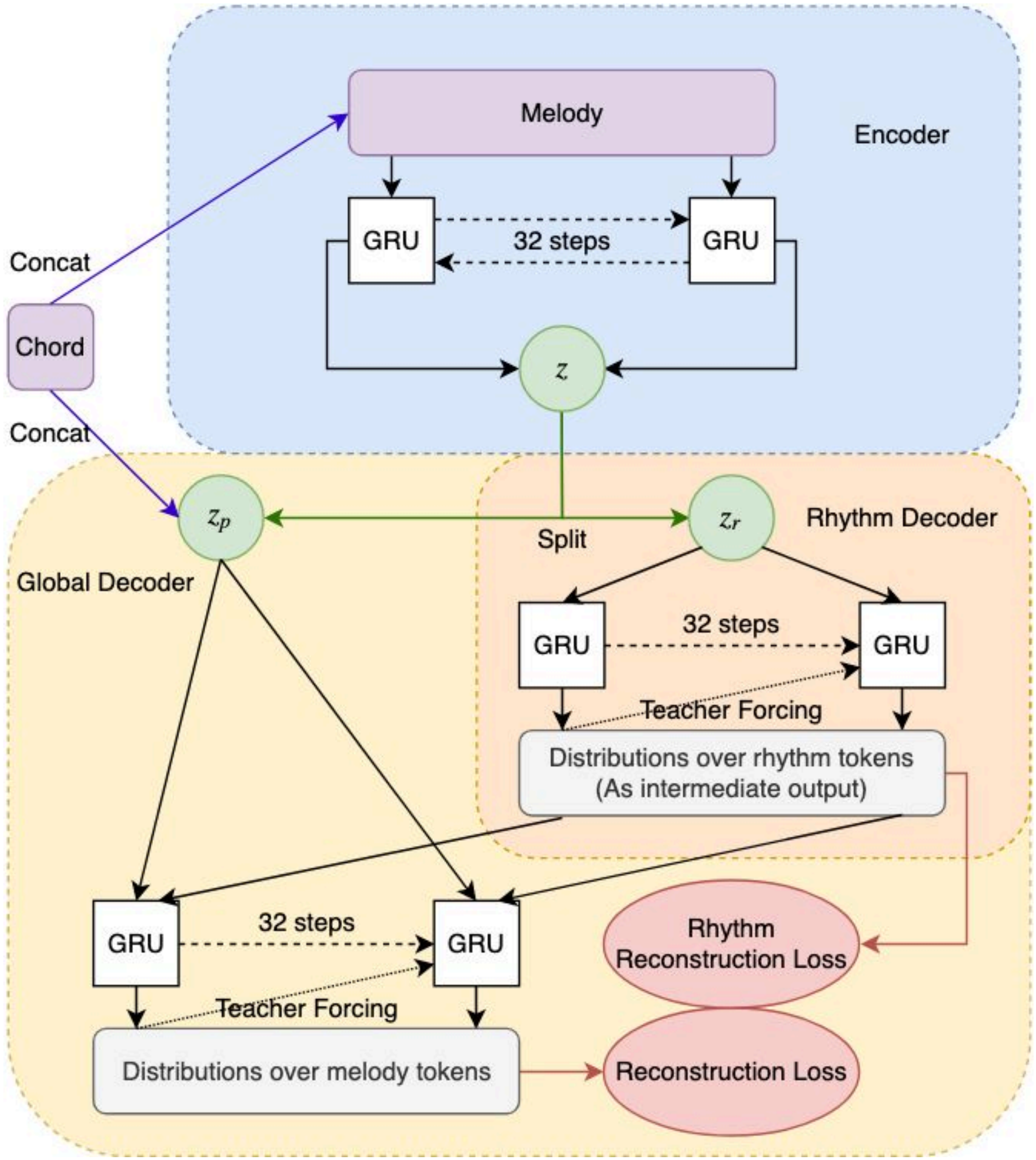
Implicit vs Explicit Dimensions (Disentanglement)

- Dimensions (ex: Pitch Range, Duration Range…) are « Chosen » by the Architecture

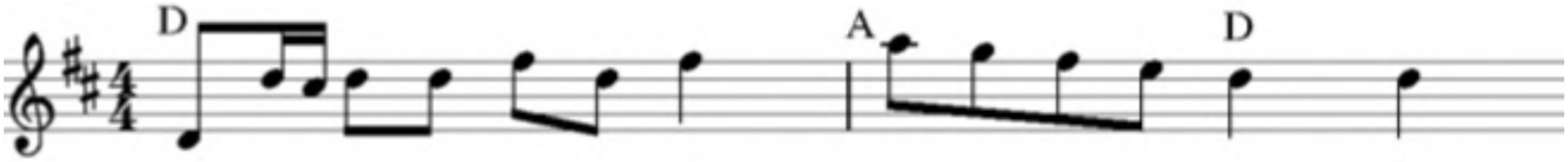
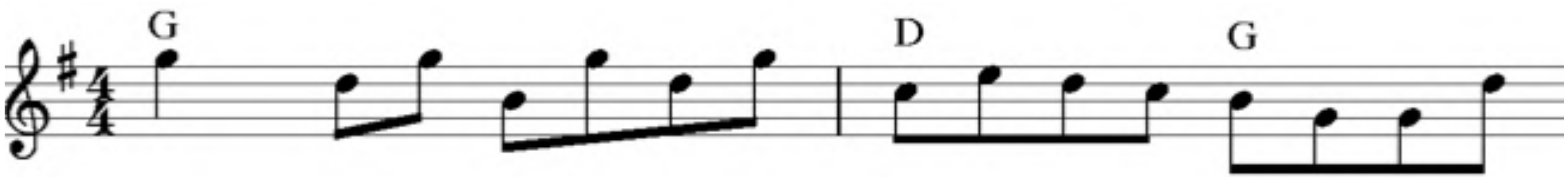
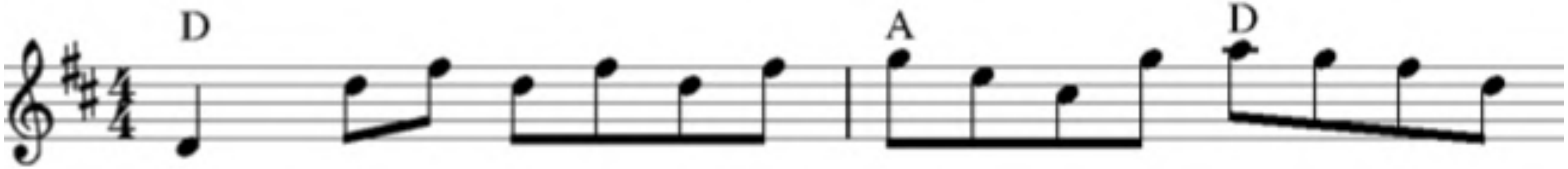
- But we can also Configure/Train the Architecture in order to « Force » some Dimensions

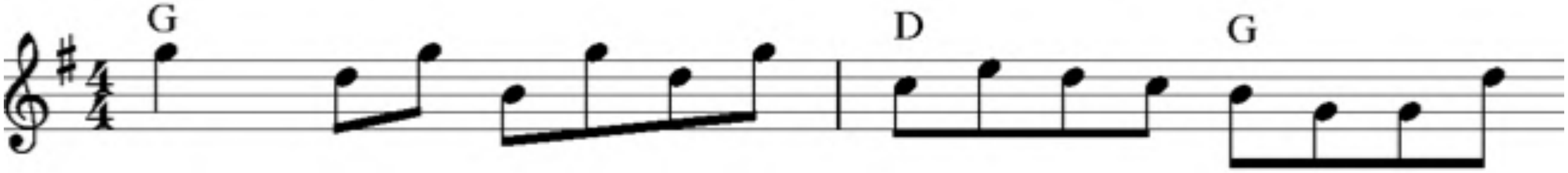
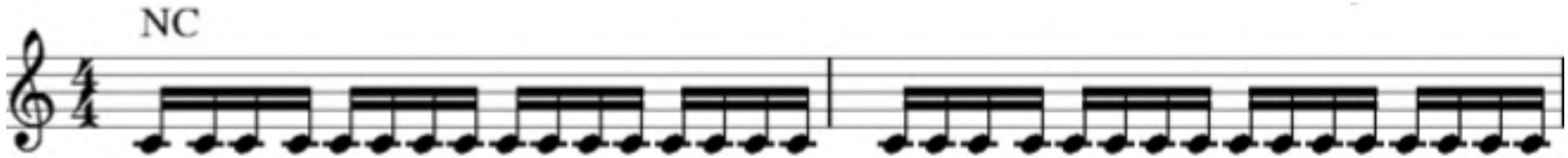
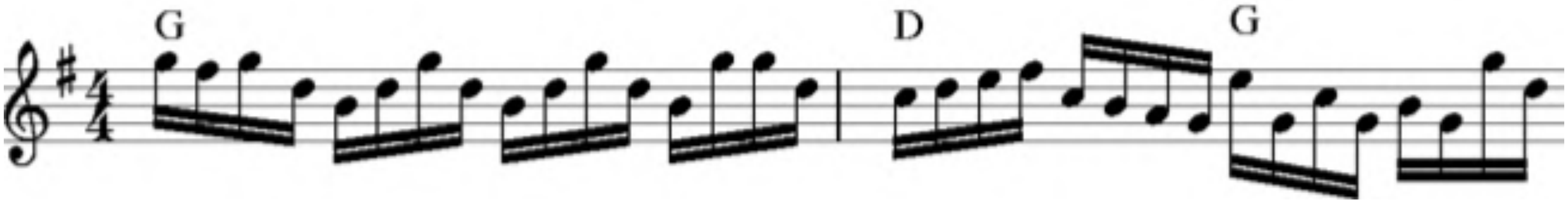


Ex: EC2-VAE [Yang et al., 2019]

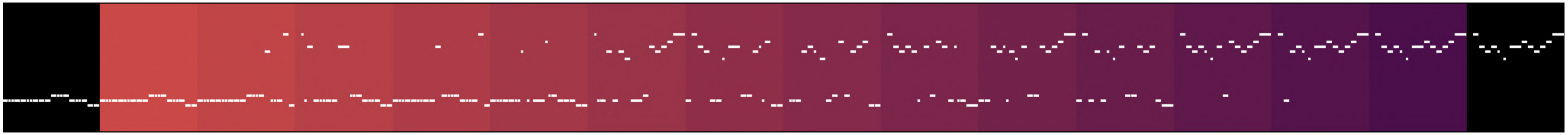


Examples EC2-VAE [Yang et al., 2019]

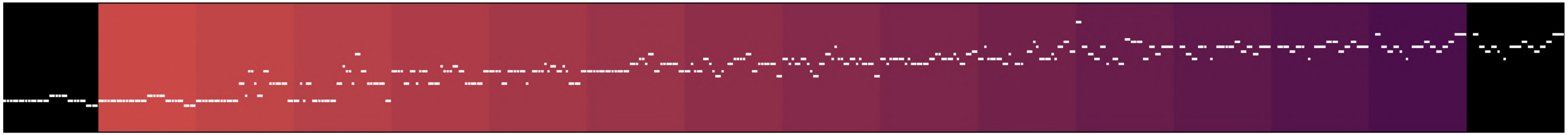
Original Melody **A**

Rythm Reference **B**

Rest(**A**) + Rythm(**B**)

Examples EC2-VAE [Yang et al., 2019]

Original Melody **B**

Rythm Reference **C**

Rest(**B**) + Rythm(**C**)

MusicVAE [Roberts et al., 2018]

- Comparing Interpolation
  - In the data space (melodies)

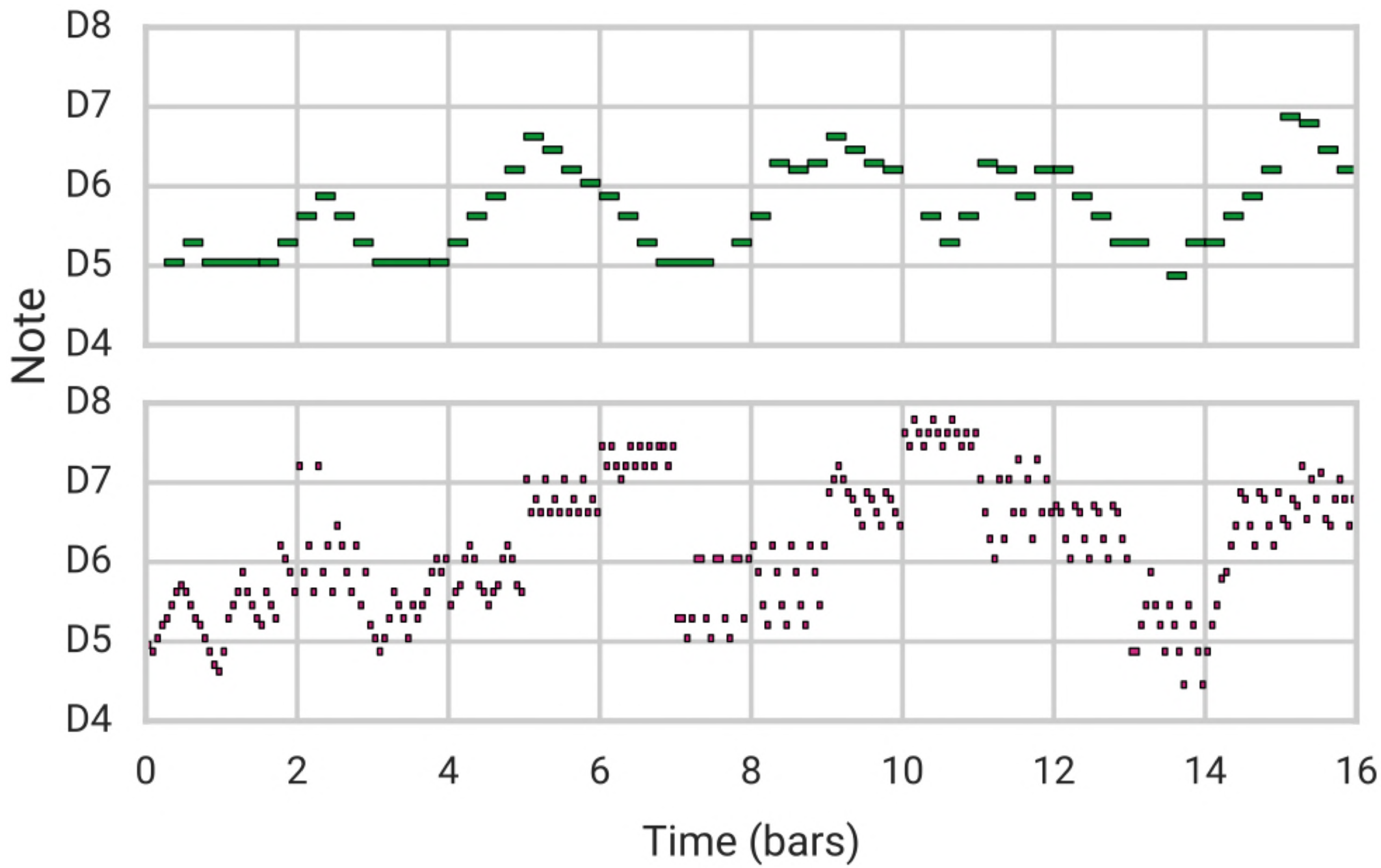


  - In the latent space

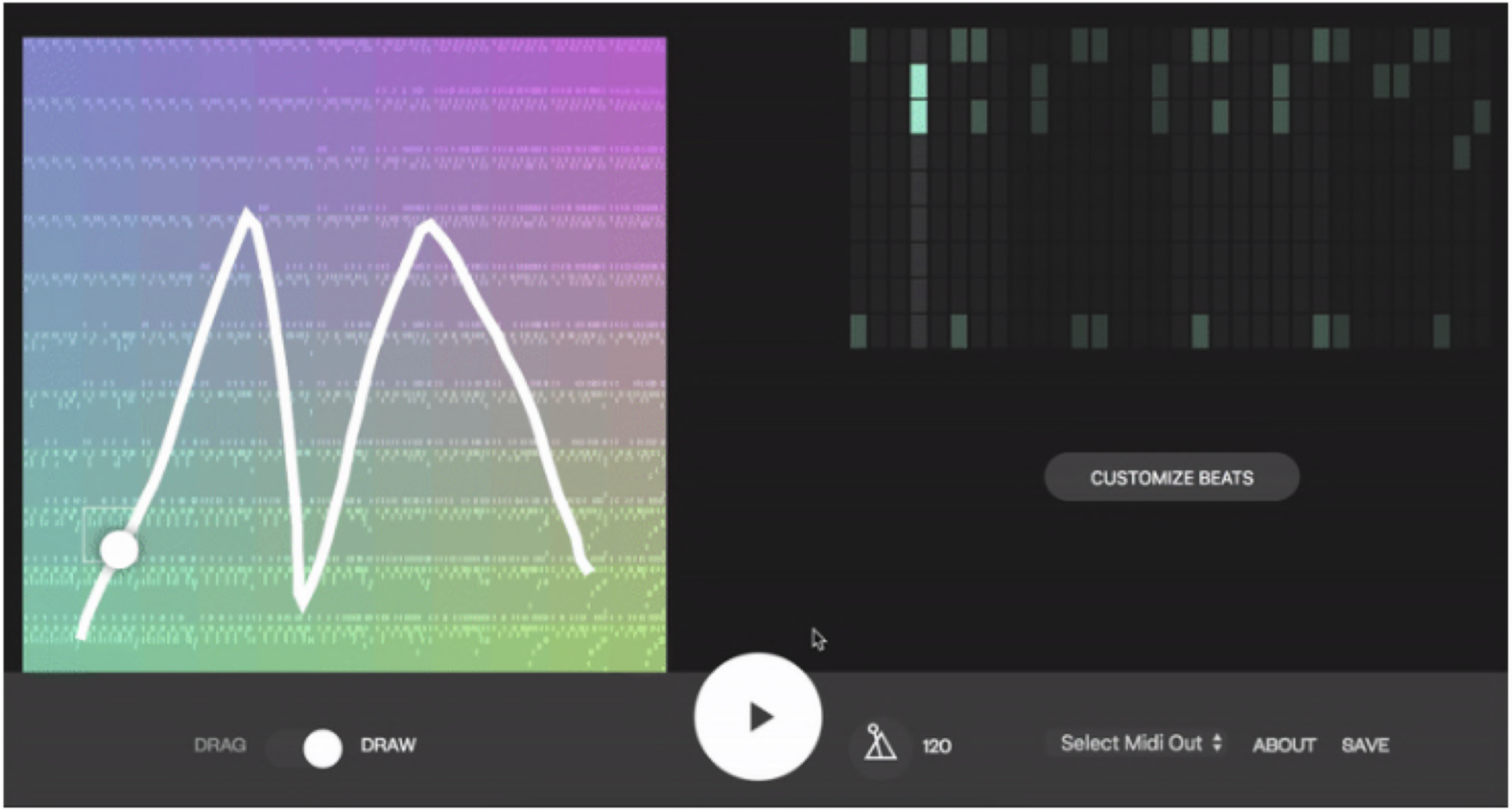


MusicVAE [Roberts et al., 2018]

- Adding a high note density attribute vector

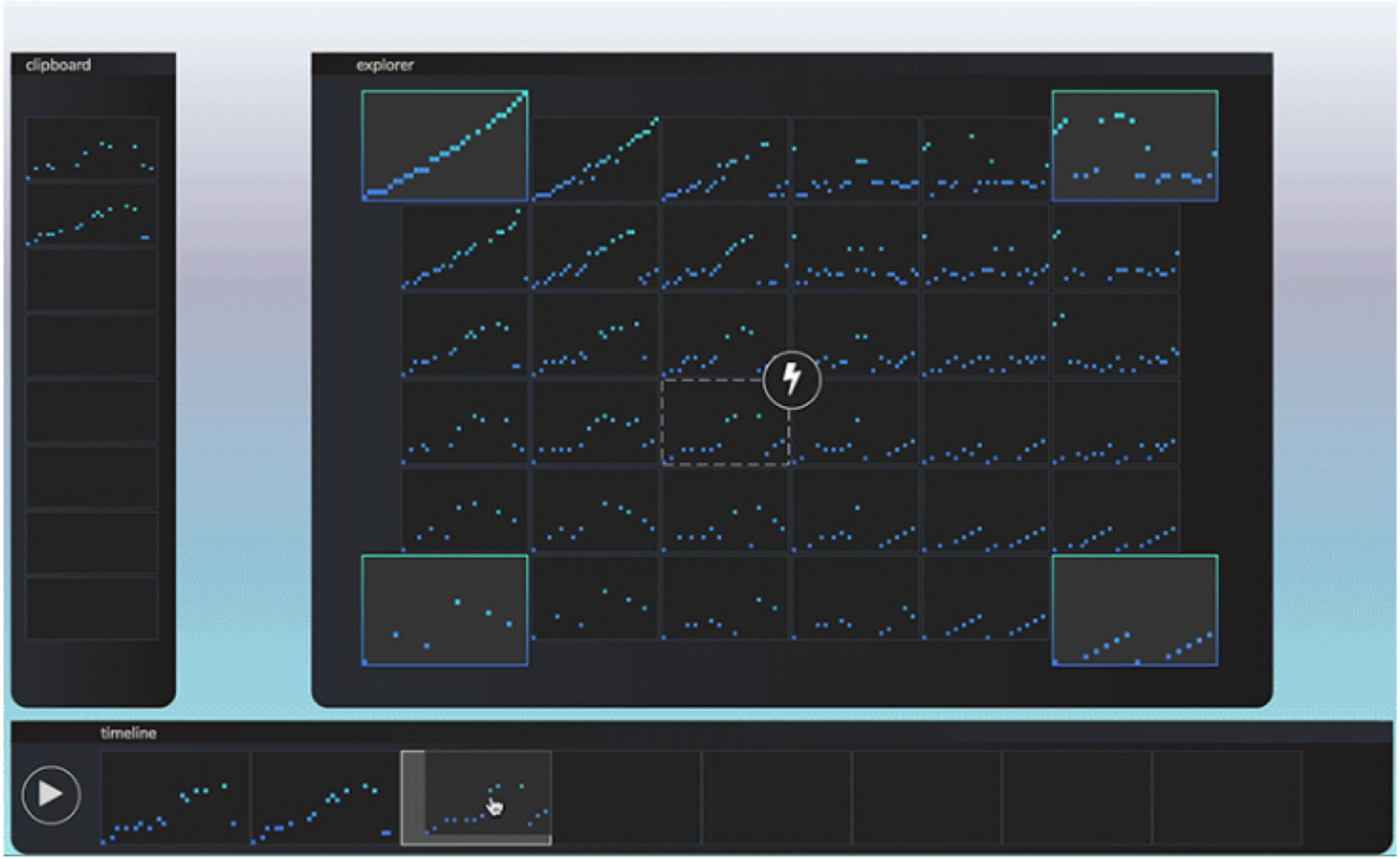


BeatBlender in TensorFlow.js
MusicVAE [Roberts et al., 2018]



https://experiments.withgoogle.com/ai/beat-blender/view/

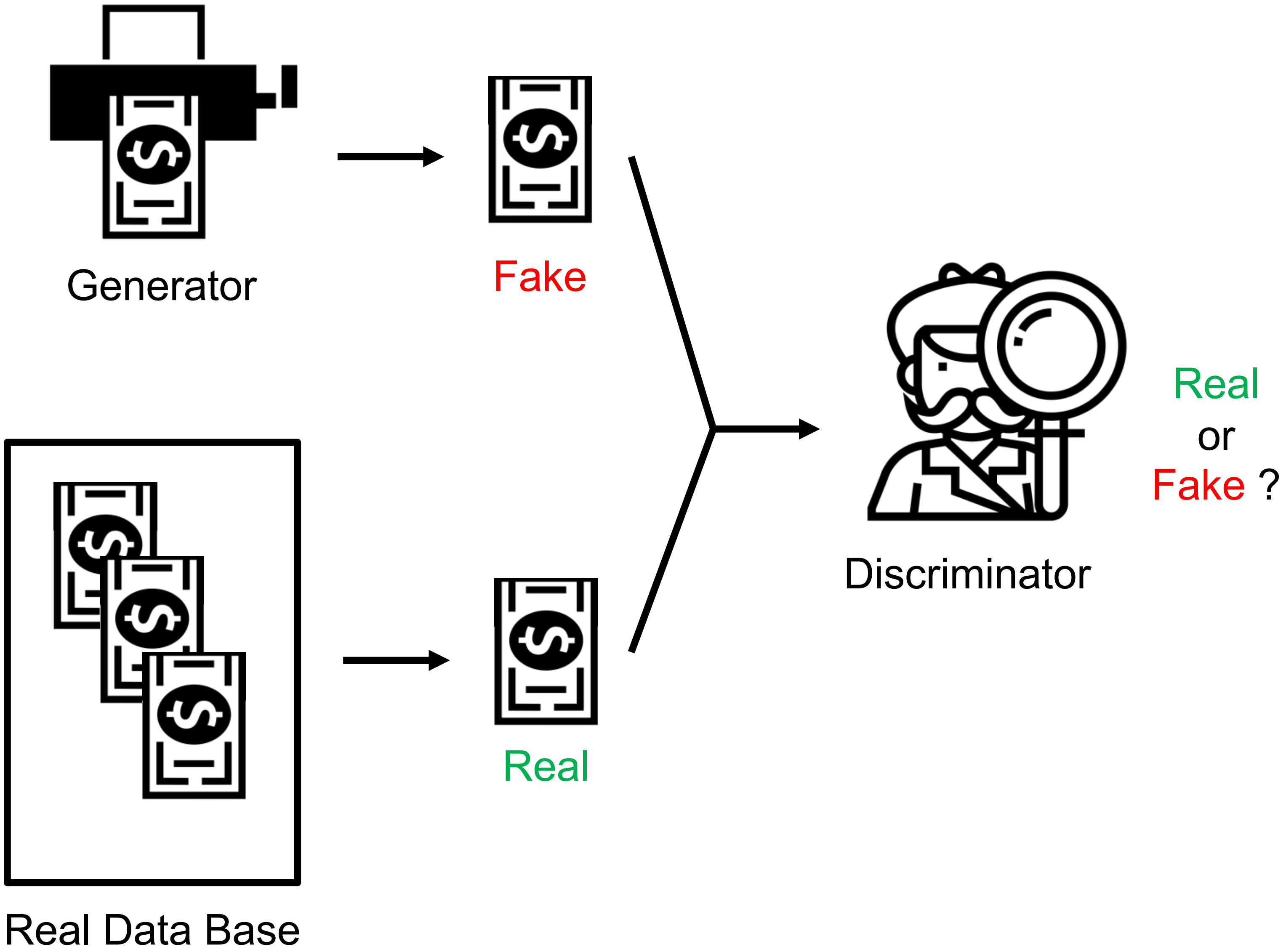
LatentLoops in TensorFlow.js
MusicVAE [Roberts et al., 2018]



https://teampieshop.github.io/latent-loops/

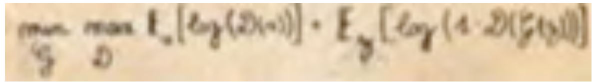
Generative Adversarial Networks

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



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 $\min_G \max_D V(G,D) = \mathbb{E}_{\mathrm{x}\sim p_{\mathrm{Data}}}[\log D(\mathrm{x})] + \mathbb{E}_{\mathrm{z}\sim p_{\mathrm{z}}(\mathrm{z})}[\log(1 - D(G(\mathrm{z})))]$

	Prediction by D
D(x): P_D(x from real data)	*(Correct)*
D(G(z)): P_D(G(z) from real data)	*(Incorrect)*
1 - D(G(z)): P_D(G(z) from Generator)	*(Correct)*

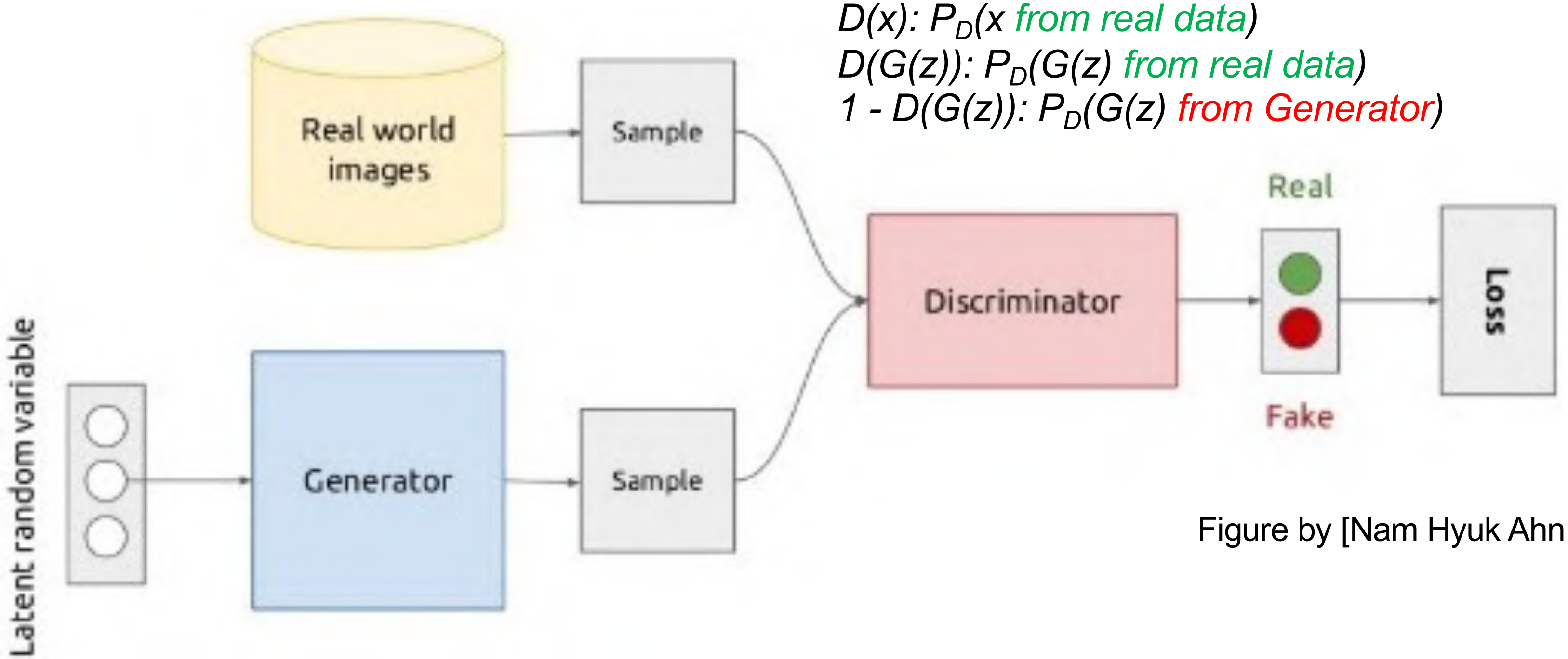


Figure by [Nam Hyuk Ahn, 2017]

Examples of GAN Generated Images

CelebFaces Attributes Dataset (CelebA)
> 200K celebrity images

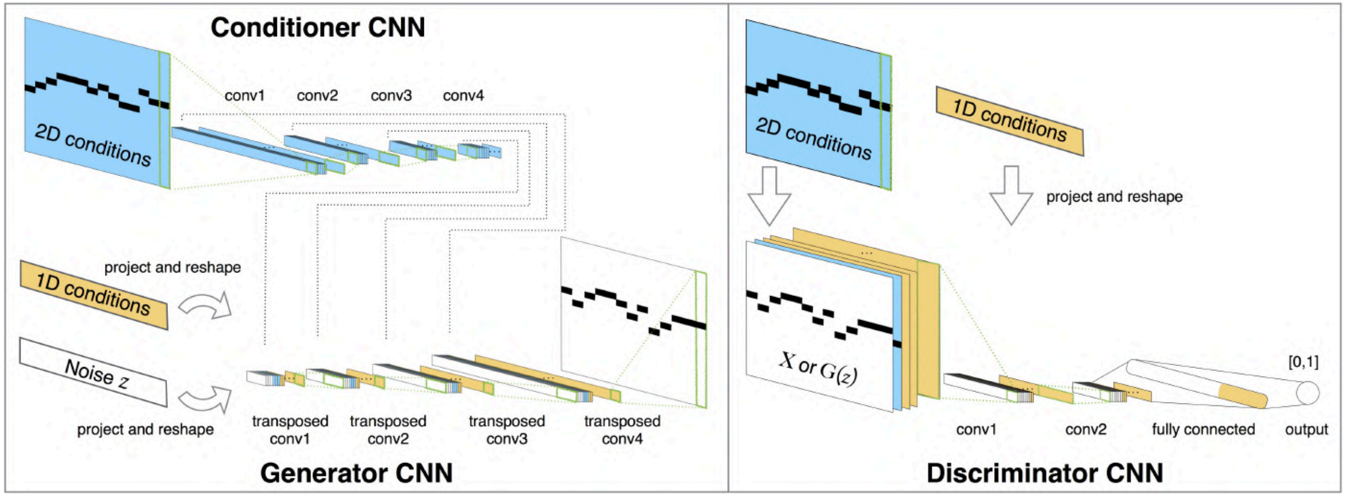
[Brundage et al., 2018]

Synthetic (Generated) Celebrity images

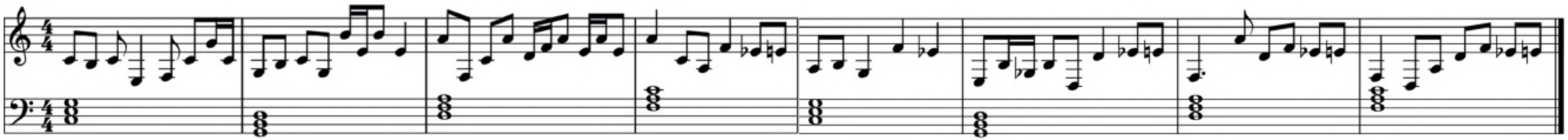
[Karras et al., 2018]

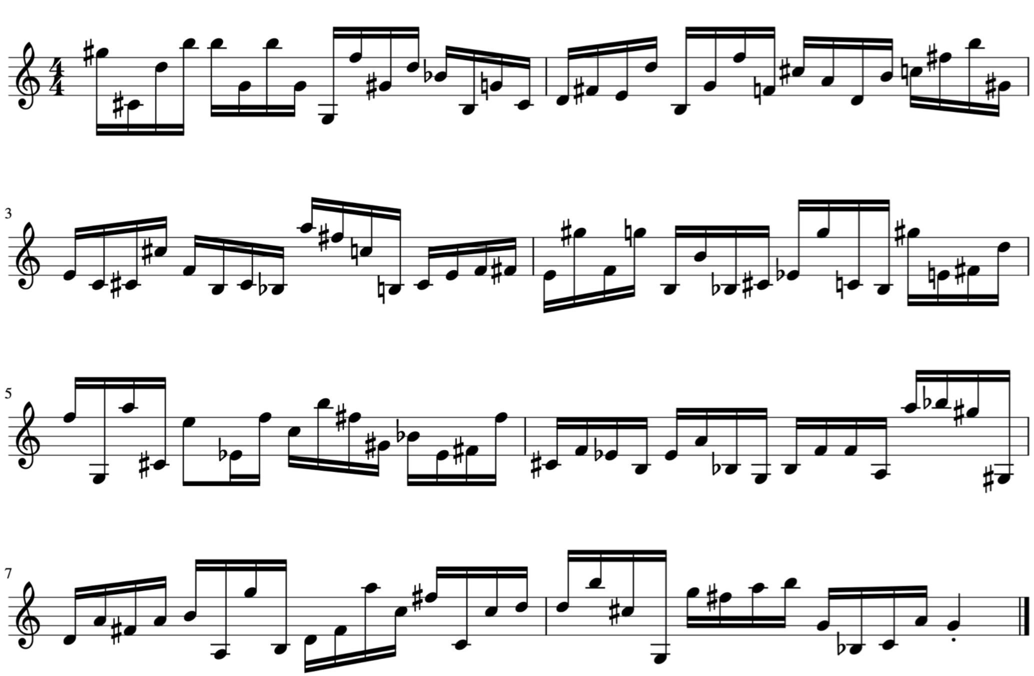
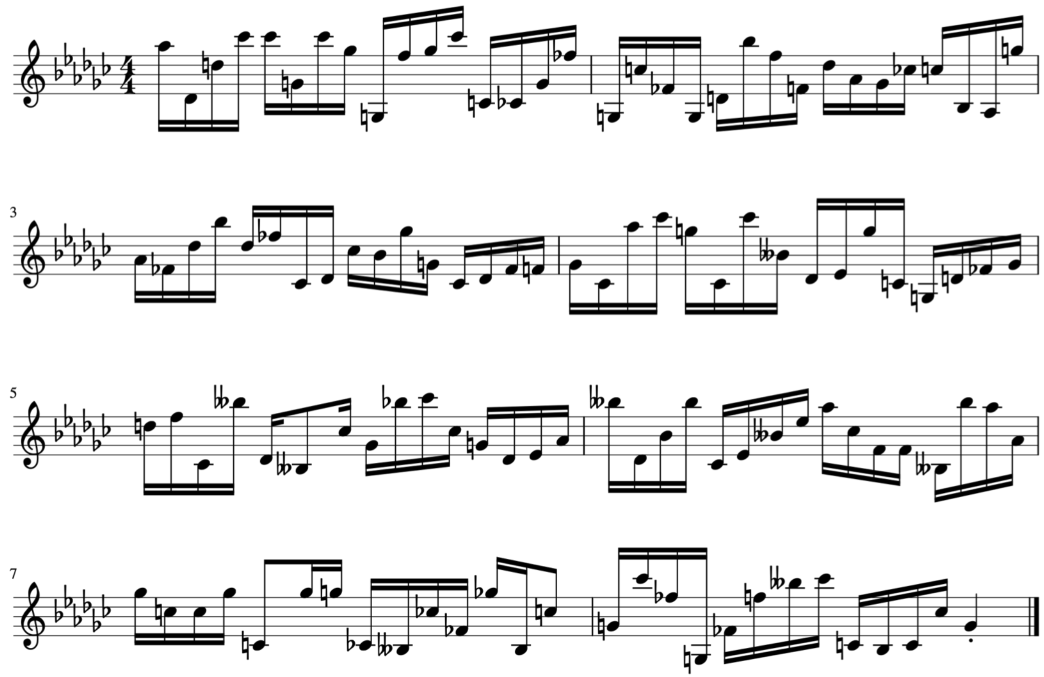
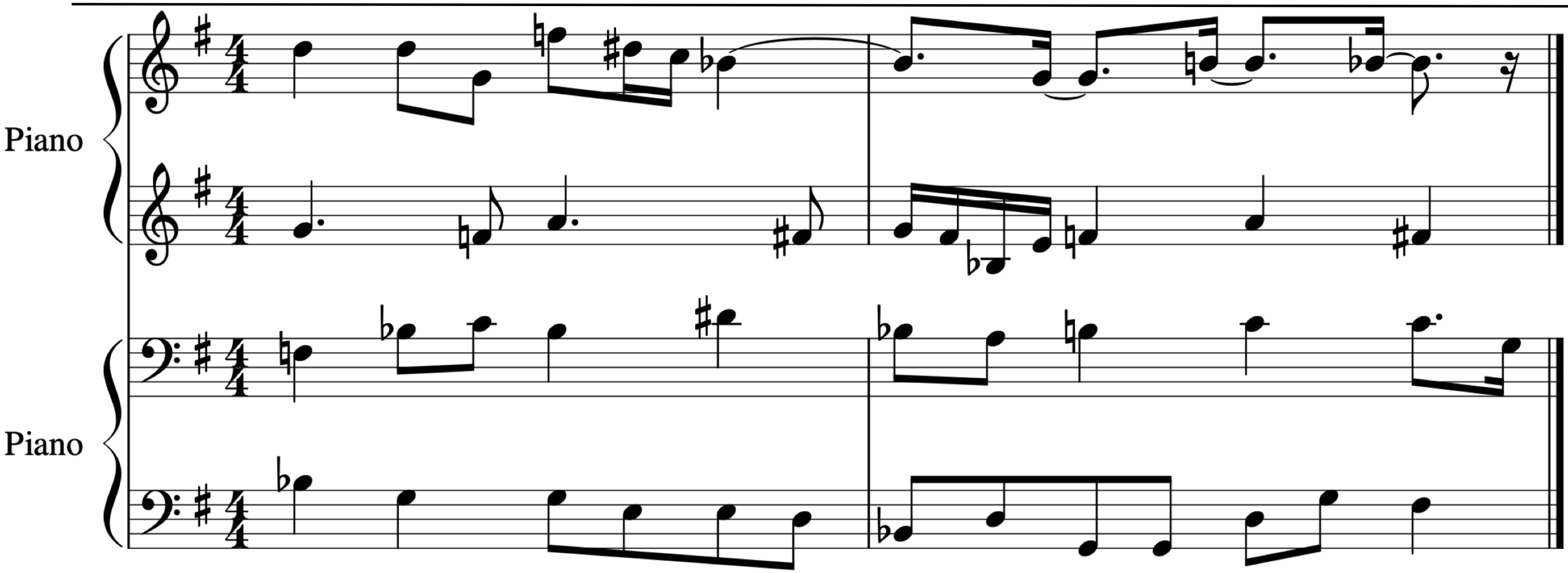
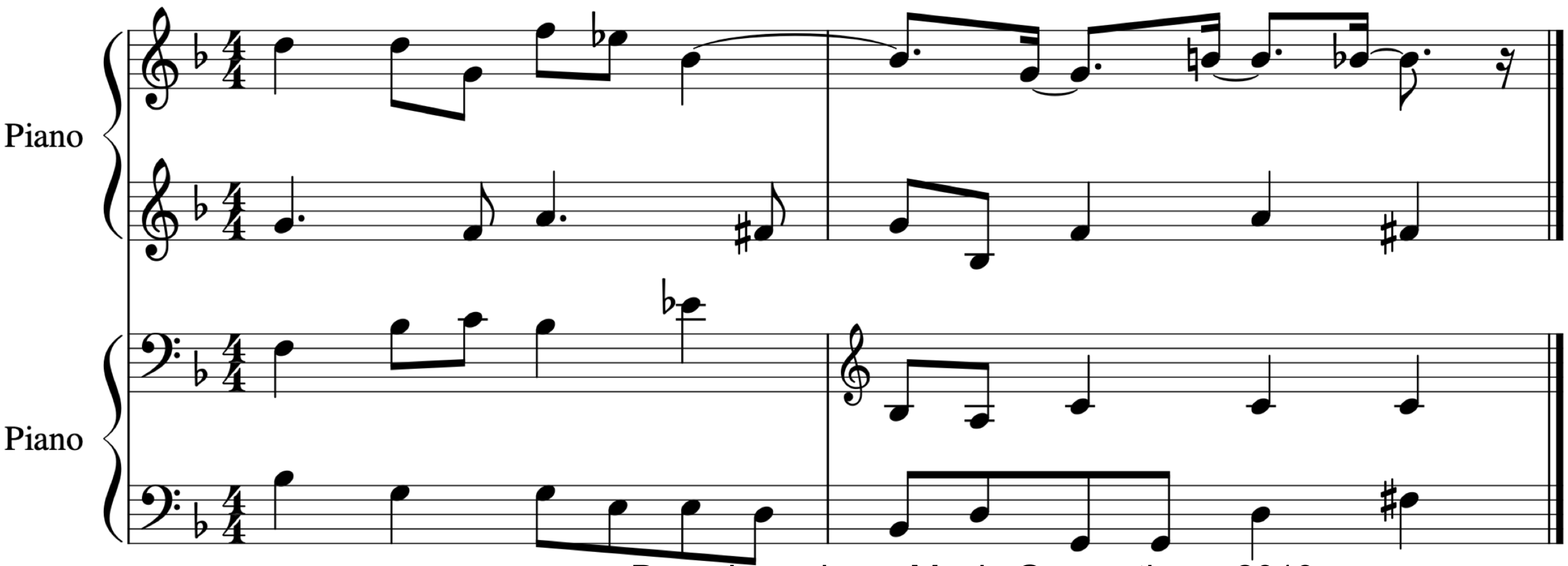
MidiNet [Yang et al., 2017]

- Conditioning information
  - Previous measure
  - Chord sequence

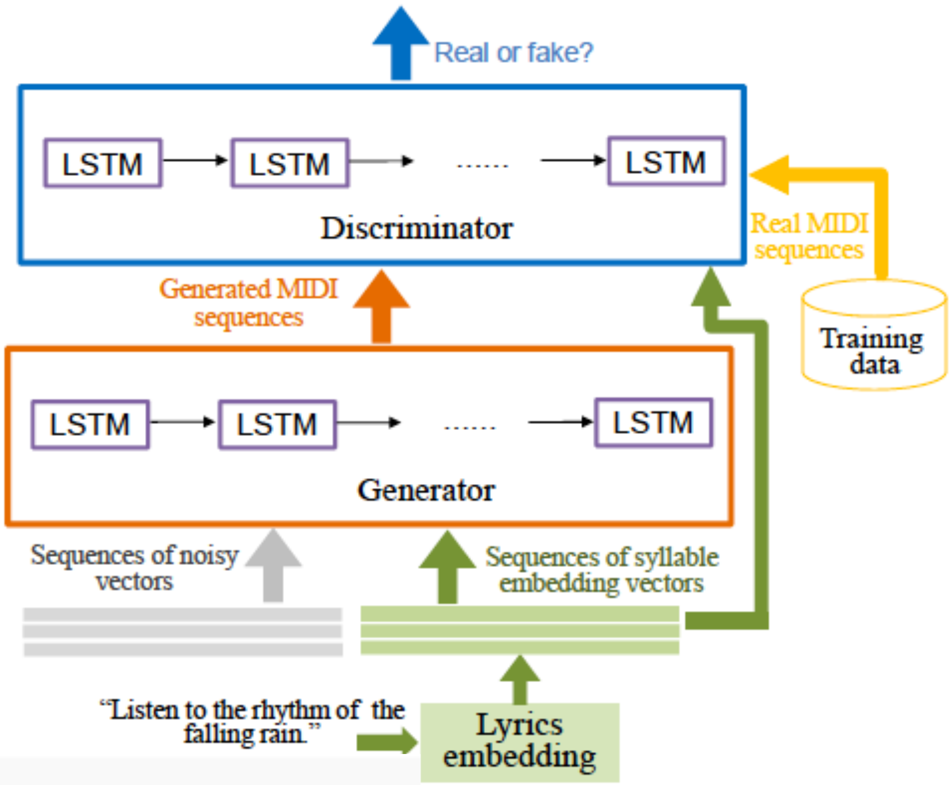
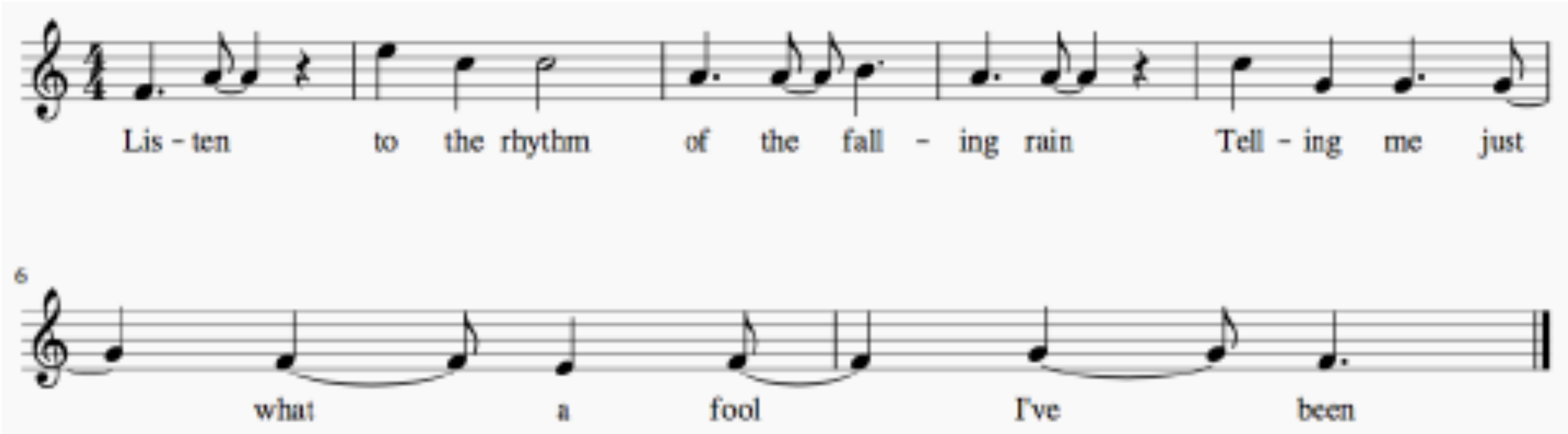


- Scope:
  - Previous measure (1D conditions)
  - Various previous measures (2D conditions)

- 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

GAN Examples – Celtic Melodies

GAN Examples – Bach Chorales

Conditional LSTM-GAN [Yu, 2019]

- Melody Generation from Lyrics
- GAN Architecture
- Conditional(GAN(LSTM))



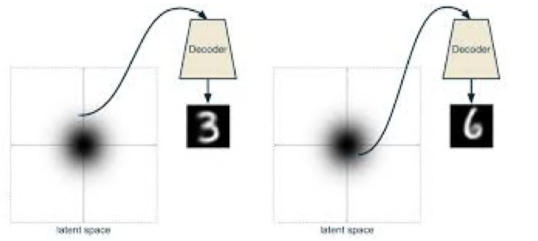
Conditional LSTM-GAN [Yu, 2019]

VAE vs GAN

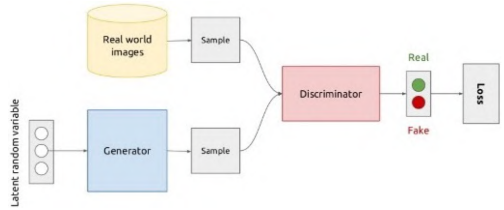
- VAE (Variational Autoencoder) and GAN (Generative Adversarial Networks)

Some Similarities:

- Are both generative architectures
- Generate from random latent variables



[Dykeman, 2016]



Differences:

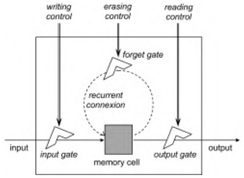
- VAE is representational of the whole training dataset
- GAN is not
- VAE Smooth control interface for exploring latent data space

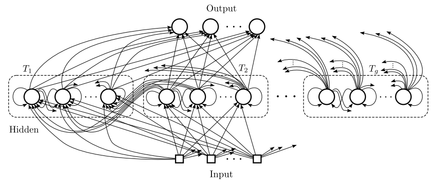
- GAN has some (ex: interpolation) but not as for VAE
- GAN produces better quality content (ex: better resolution images)
  - *Not a main issue for symbolic music representation*

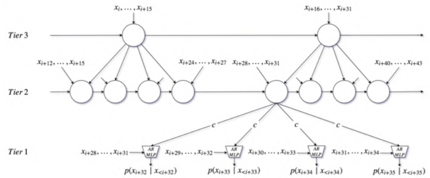
Issues

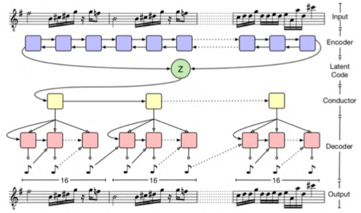
Open Issues

- Structure
  - Ex: LSTM [Hochreiter & Schmidhuber, 1997]
  - Clockwork RNN [Koutnik et al., 2014]
  - SampleRNN [Mehri et al., 2017]
  - MusicVAE [Roberts et al., 2018]

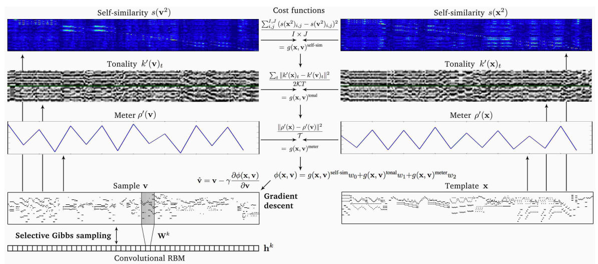




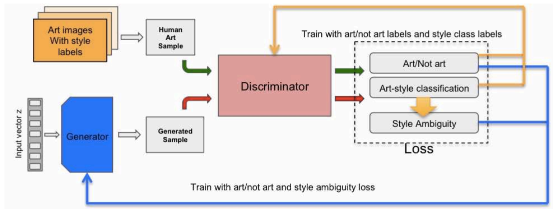


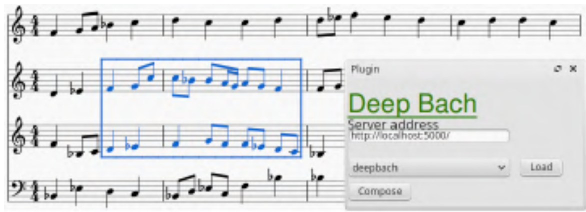


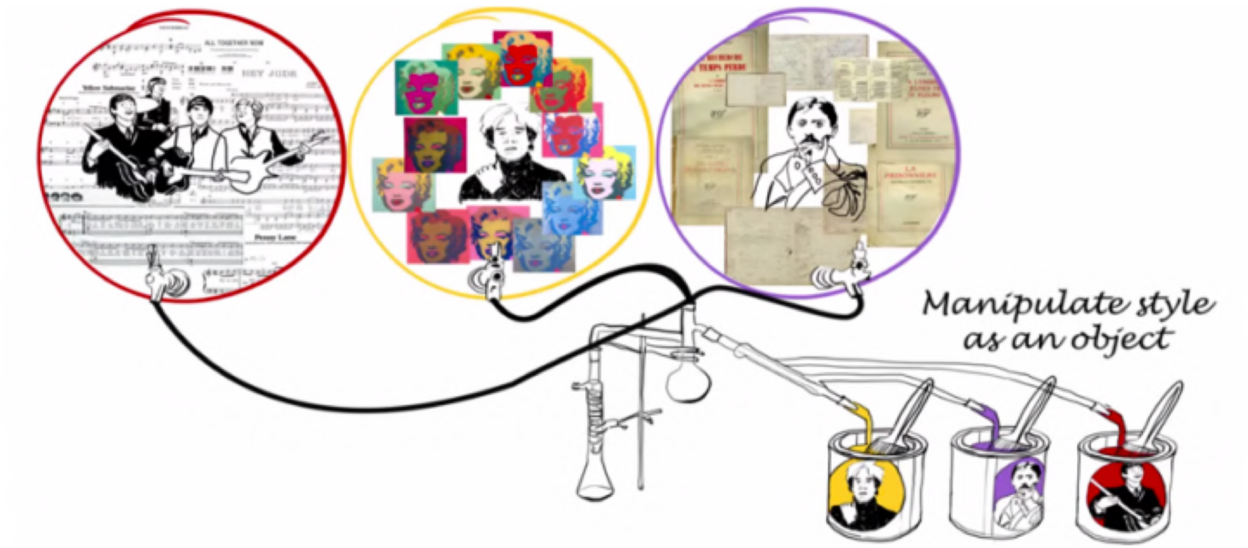
- Control
  - Tonality Conformance
  - Rhythm
  - Ex: C-RBM [Lattner et al., 2016]
  - Conditioning
  - Arbitrary Constraints

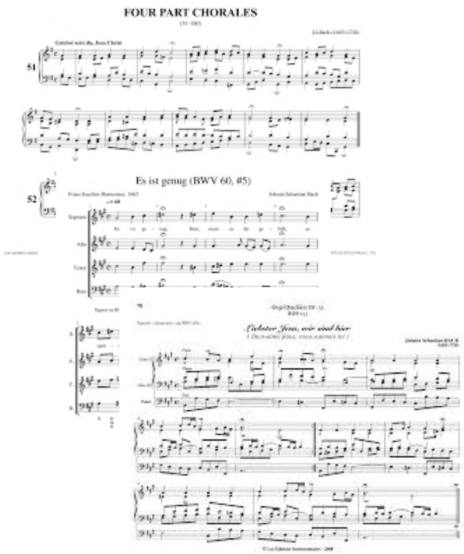
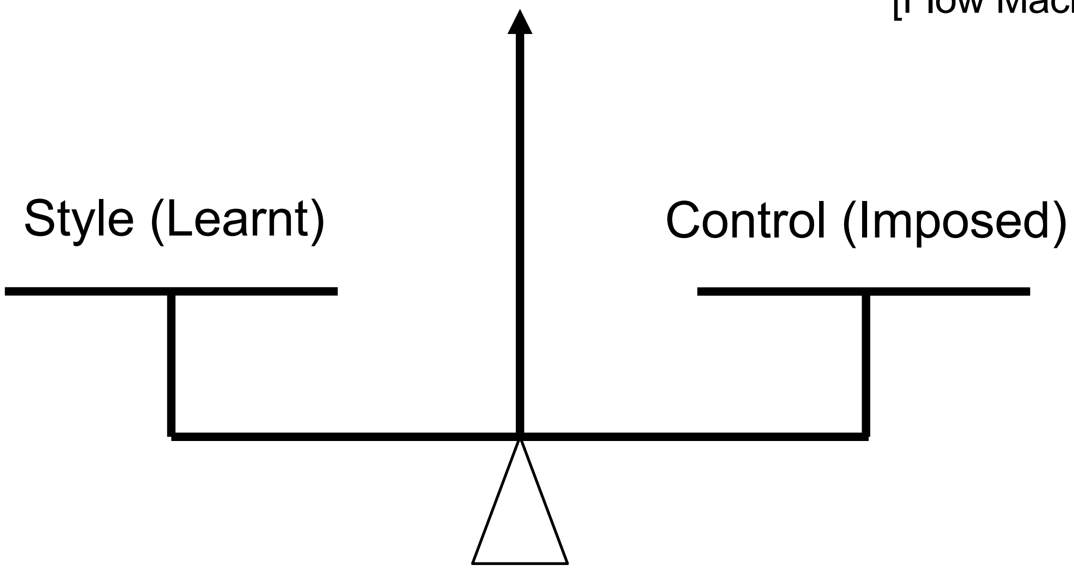


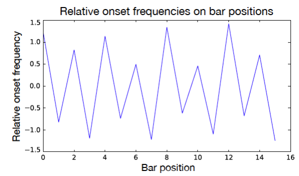
- Creativity Incentive
  - Vs Style Conformance
  - Ex: CAN [Elgammal et al., 2017]



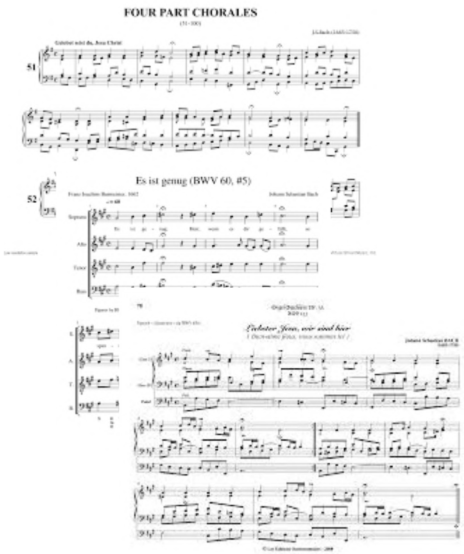
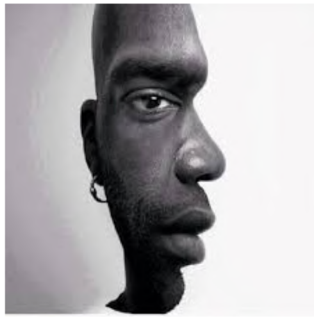
- Interactivity/Incrementality
  - Ex: DeepBach [Hadjeres et al., 2017]
  - Incremental Sampling

Style vs/and Control



[Flow Machines]



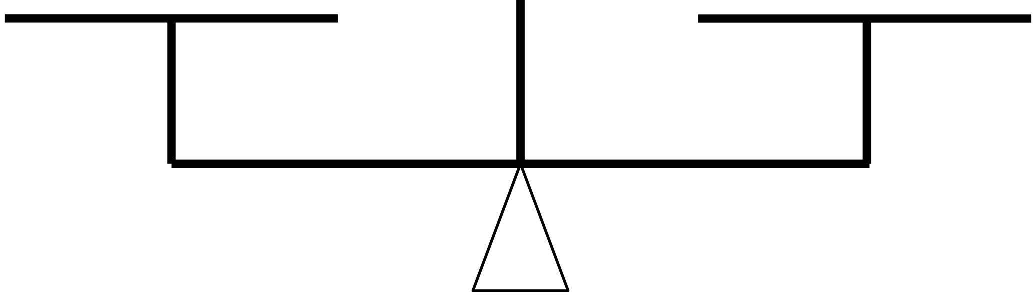


Style vs/and Originality

[Mimi & Eunice]



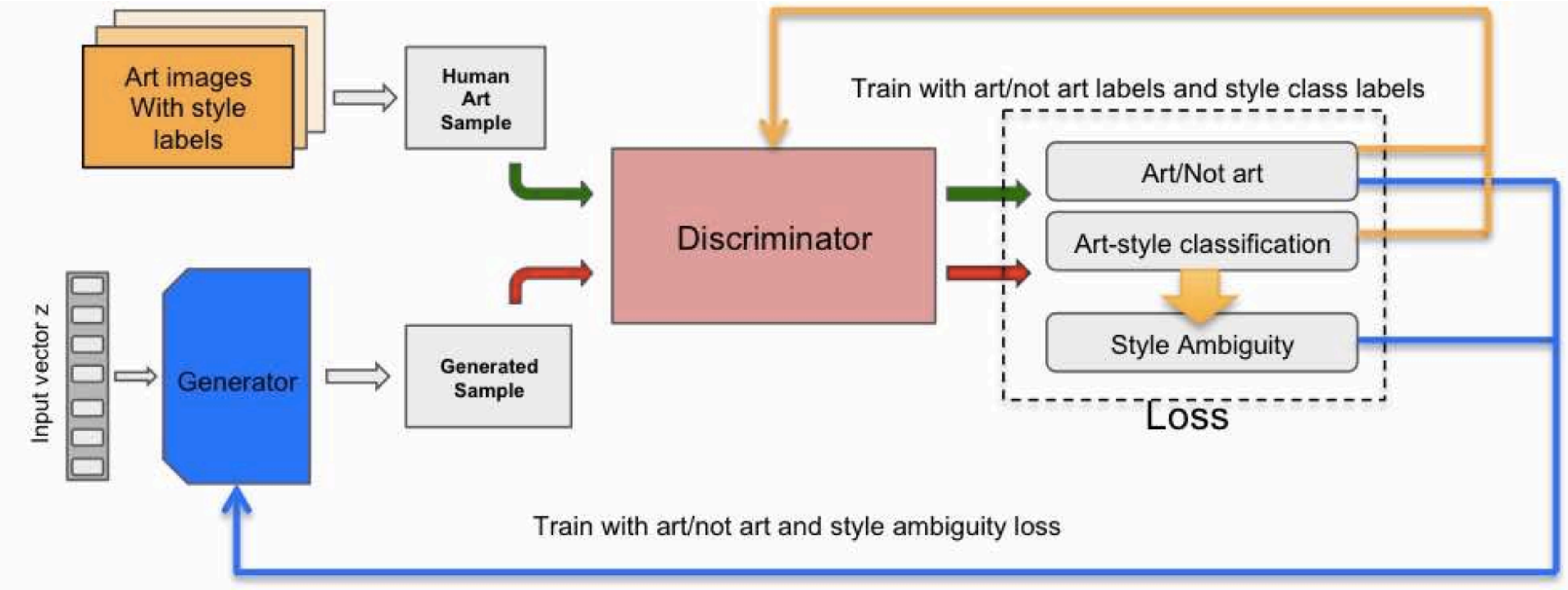
Style (learnt)

Originality



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

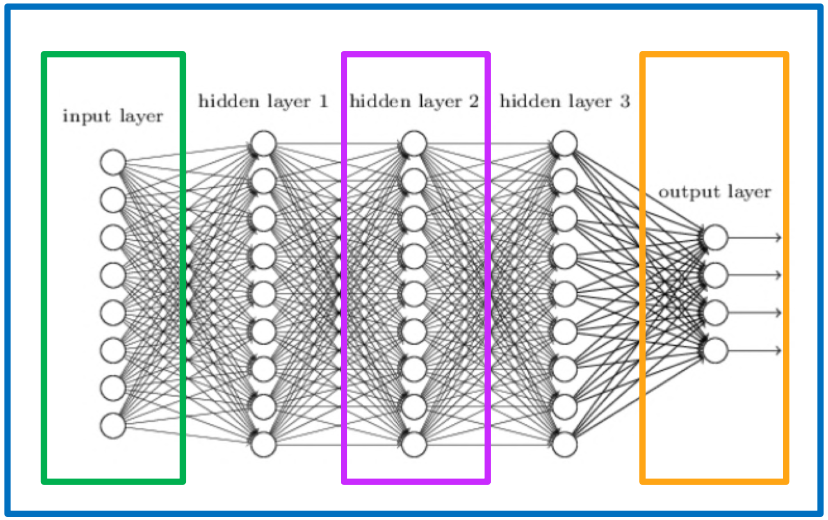
Style name	Image number	Style name	Image number
Abstract-Expressionism	2782	Mannerism-Late-Renaissance	1279
Action-Painting	98	Minimalism	
Analytical-Cubism	110	Naive Art-Primi	
Art-Nouveau-Modern	4334	New-Realism	
Baroque	4241	Northern-Renai	
Color-Field-Painting	1615	Pointillism	
Contemporary-Realism	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	

Control

- Strategies:
  - Sampling
  - Conditionning (Parametrization)
  - Input Manipulation
  - Reinforcement
  - Unit Selection

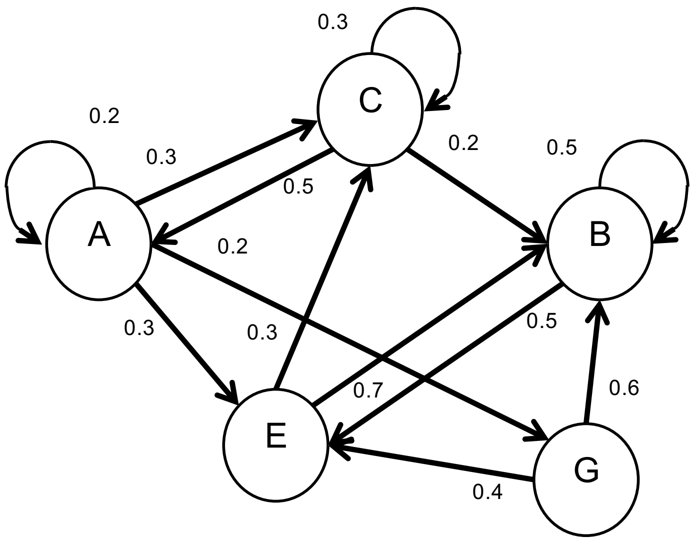
  - Bottom up (Low-level adjustment)
    - » Ex: Sampling
  - Top down (Structure imposition)
    - » Ex: Unit and Selection

- Entry points (Hooks)
  - Input
  - Hidden
  - Output
  - Encapsulation/Reformulation

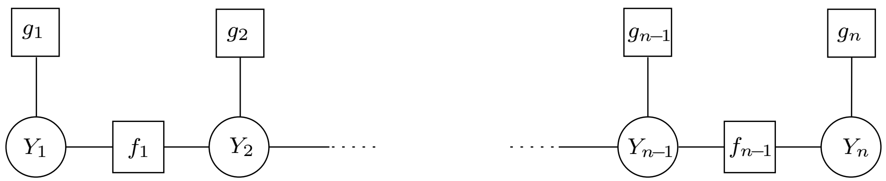


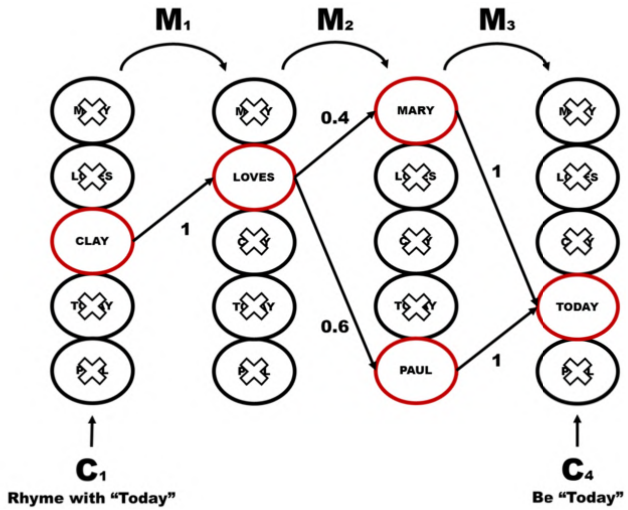
Markov Models

- Operational

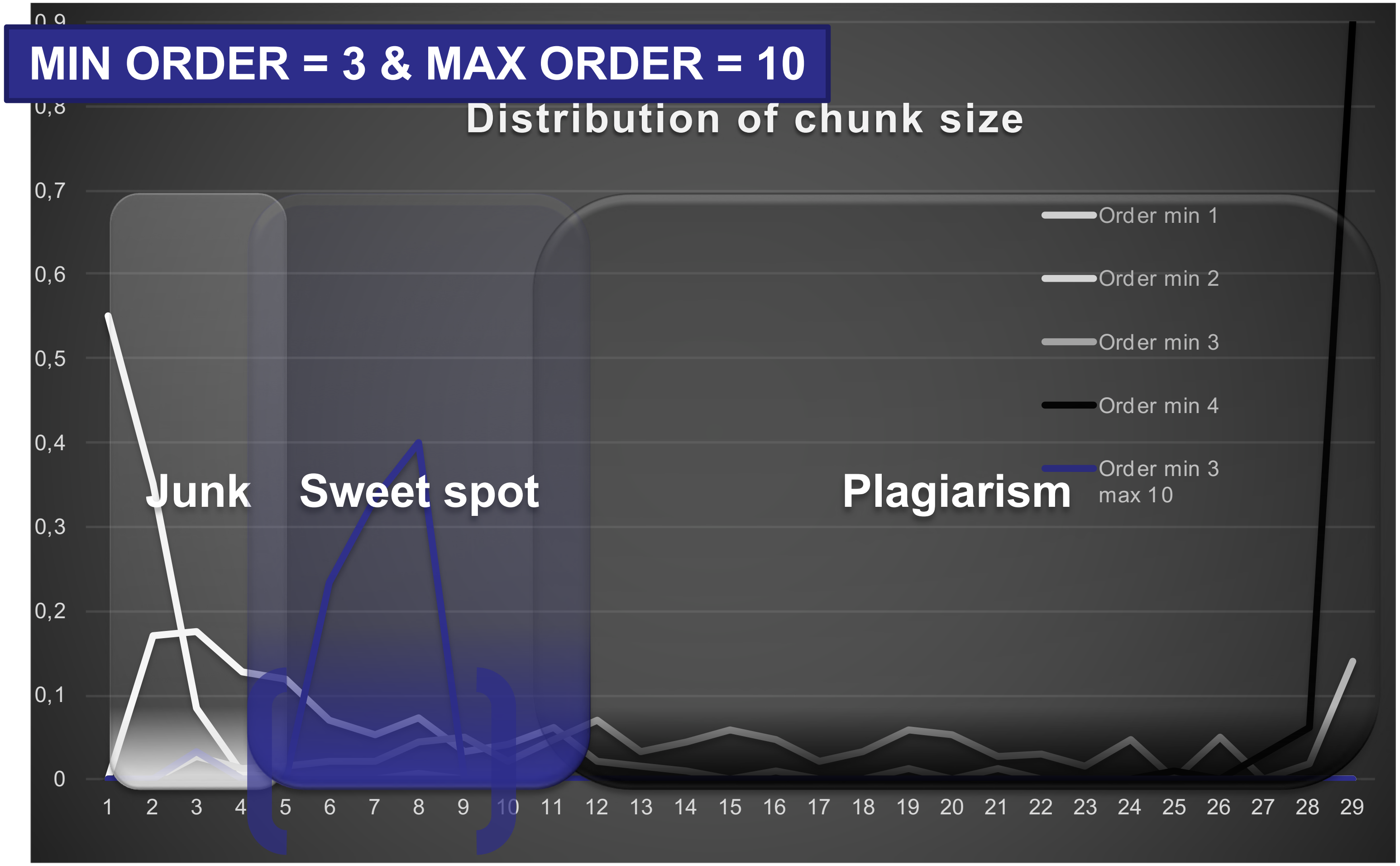


- May attach Constraints and Functions
  - Ex: Factor Graphs, Markov Constraints [Pachet & Roy, 2011]





Constrained Higher-Order Markov



MaxOrder Constraint

[Roy and Pachet, 2017]

Markov Model vs Deep Learning

+ Markov models are conceptually simple

Markov models simpler

+ Markov models have a simple implementation and a simple learning algorithm as the model is a transition probability table

-- Neural network models are conceptually simple but the optimized implementations of current deep network architecture may be complex and need a lot of tuning

-- Order 1 Markov models (that is, considering only the previous state) do not capture long-term temporal structures

-- Order n Markov models (considering n previous states) are possible but require an explosive training set size and can lead to plagiarism

+ Neural networks can capture various types of relations, contexts and regularities

+ Deep networks can learn long-term and high-order dependencies

+ Markov models can learn from a few examples

Deep learning more conformant

-- Neural networks need a lot of examples in order to be able to learn well

-- Markov models do not generalize very well

+ Neural networks generalize better through the use of distributed representations

+ Markov models are operational models (automata) on which some control on the generation could be attached

-- Deep networks are generative models with a distributed representation and therefore with no direct control to be attached

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

Autonomous Generation vs Creation Support

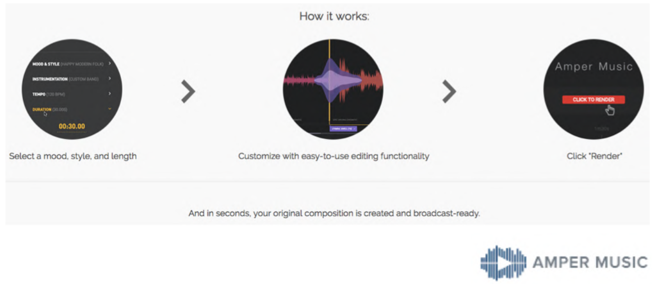
Autonomous vs Assisted Music Creation

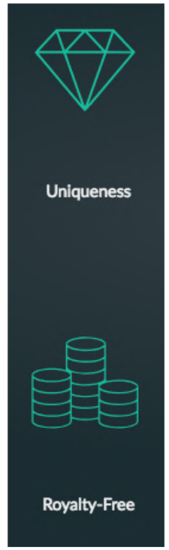
- Autonomous Generation/Interpretation
  - Turing Test
  - Symbolic or/and Audio Music Generation
  - Parametrization/User Preferences (Style, Mood, etc.)
  - For Commercials and Documentaries
  - Create Royalty-free or Copyright-buyable Music
  - Ex:

Jukedeck

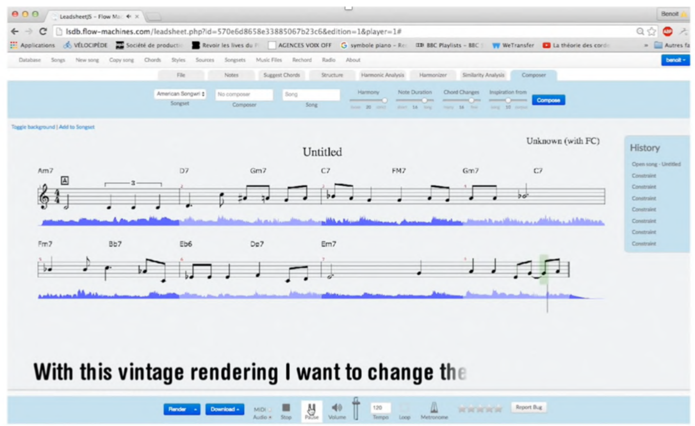
Aiva Technologies

AMPER MUSIC





- Assistance to Human Composers and Musicians
  - Propose
  - Refine
  - Analyze
  - Harmonize
  - Produce
  - Ex: FlowComposer [Pachet et al., 2014]



Autonomous Music Making

- Symbolic or/and Audio Music Generation

- For Commercials and Documentaries
- Create Royalty-free or Copyright-buyable Music
- Based on Deep learning + Samples + Sound processing techniques

\+ Business model

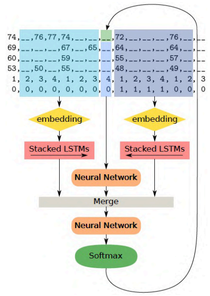
-- Musical model

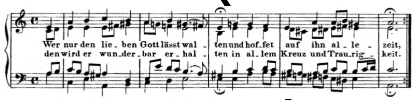
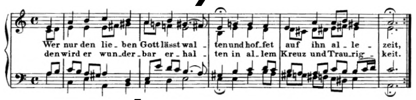
Bach Chorales Turing Test

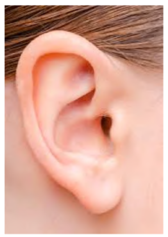
- Autonomous Artificial Musicians

- Music Composition Turing test
  - Imitation Game Scenario [Turing, 1950]
  - Designed by A. Turing to explore the question "Can Machines think?"

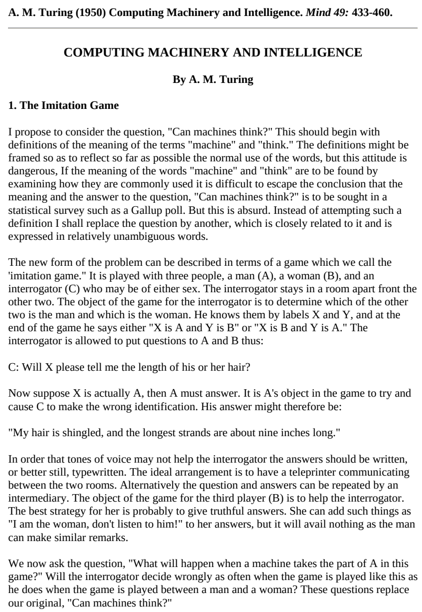
(A) J. S. Bach



(B) DeepBach [Hadjeres et al., 2017]

?



(C) Listener



A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

The new form of the problem can be described in terms of a game which we call the 'imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart front the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A." The interrogator is allowed to put questions to A and B thus:

C: Will X please tell me the length of his or her hair?

Now suppose X is actually A, then A must answer. It is A's object in the game to try and cause C to make the wrong identification. His answer might therefore be:

"My hair is shingled, and the longest strands are about nine inches long."

In order that tones of voice may not help the interrogator the answers should be written, or better still, typewritten. The ideal arrangement is to have a teleprinter communicating between the two rooms. Alternatively the question and answers can be repeated by an intermediary. The object of the game for the third player (B) is to help the interrogator. The best strategy for her is probably to give truthful answers. She can add such things as "I am the woman, don't listen to him!" to her answers, but it will avail nothing as the man can make similar remarks.

We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?"

- To evaluate artificial composers techniques
- To explore music cognition

Bach Chorales Turing Test

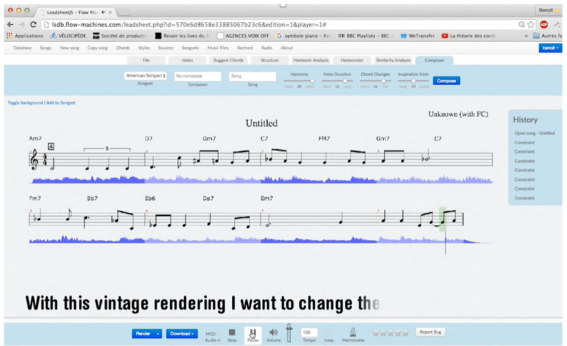
- February 2017, Dutch TV Channel
- Bach vs DeepBach Turing Test

Objective and Evaluation [Pachet, 2019]

	Current Systems Autonomous Generalization-based	**Future** Systems **Augmentation/Assistance** **Creative**-incentived
Objective	Create music	Create music **not possible otherwise**
Evaluation	Please the listener	Please the **composer**
Risk	Conventional	**Surprising** But meaningful

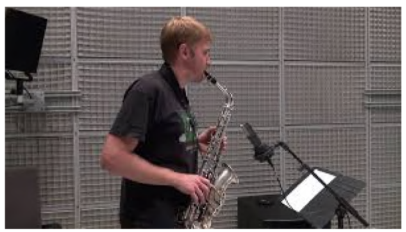
Co-Creativity

- Co-Creation by Human(s)+Machine(s)
  - Ex: FlowComposer [Pachet et al., 2014]

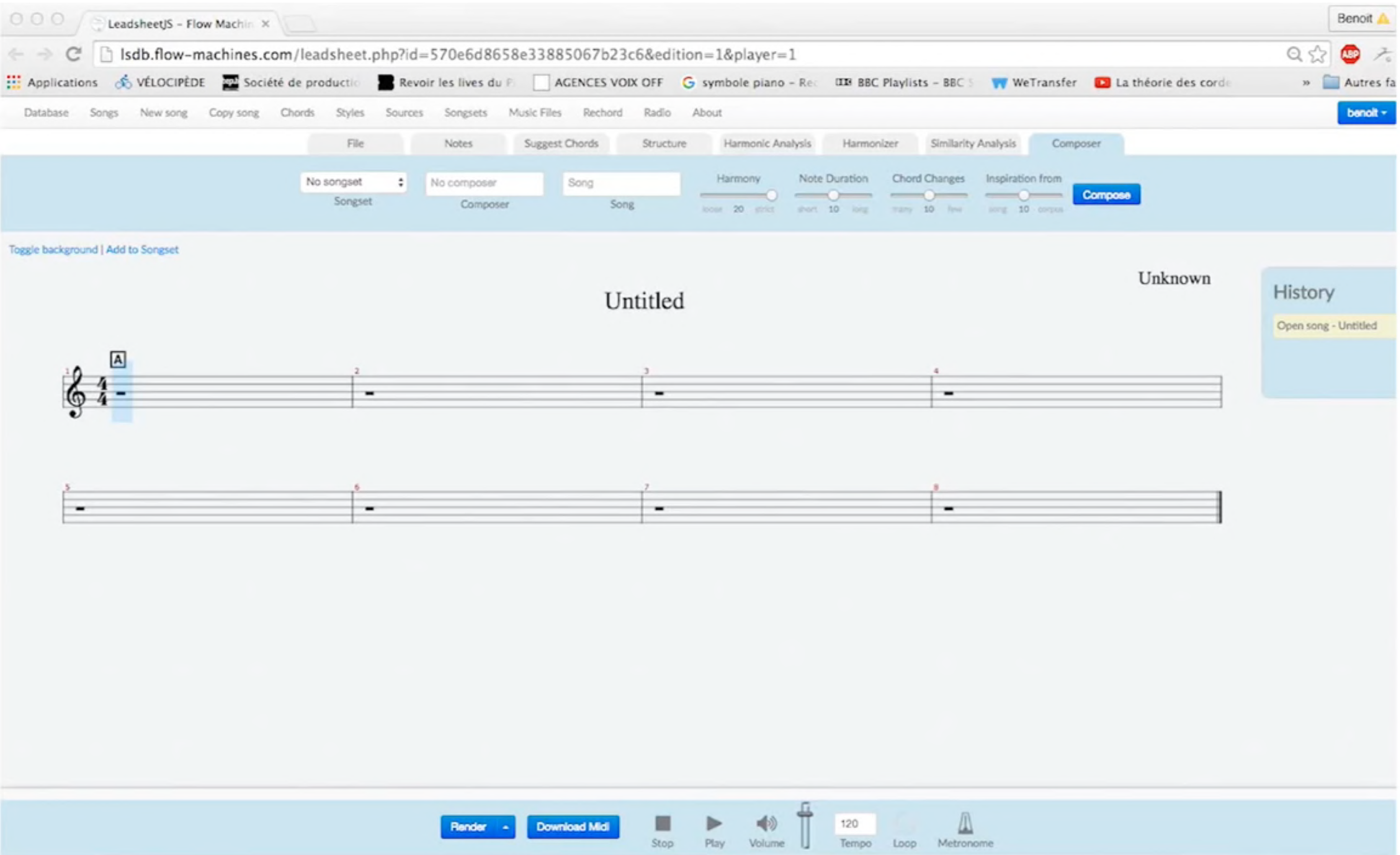


  - Continuator [Pachet, 2002]

  - Omax/DYCI2 [Assayag et al., 2003]

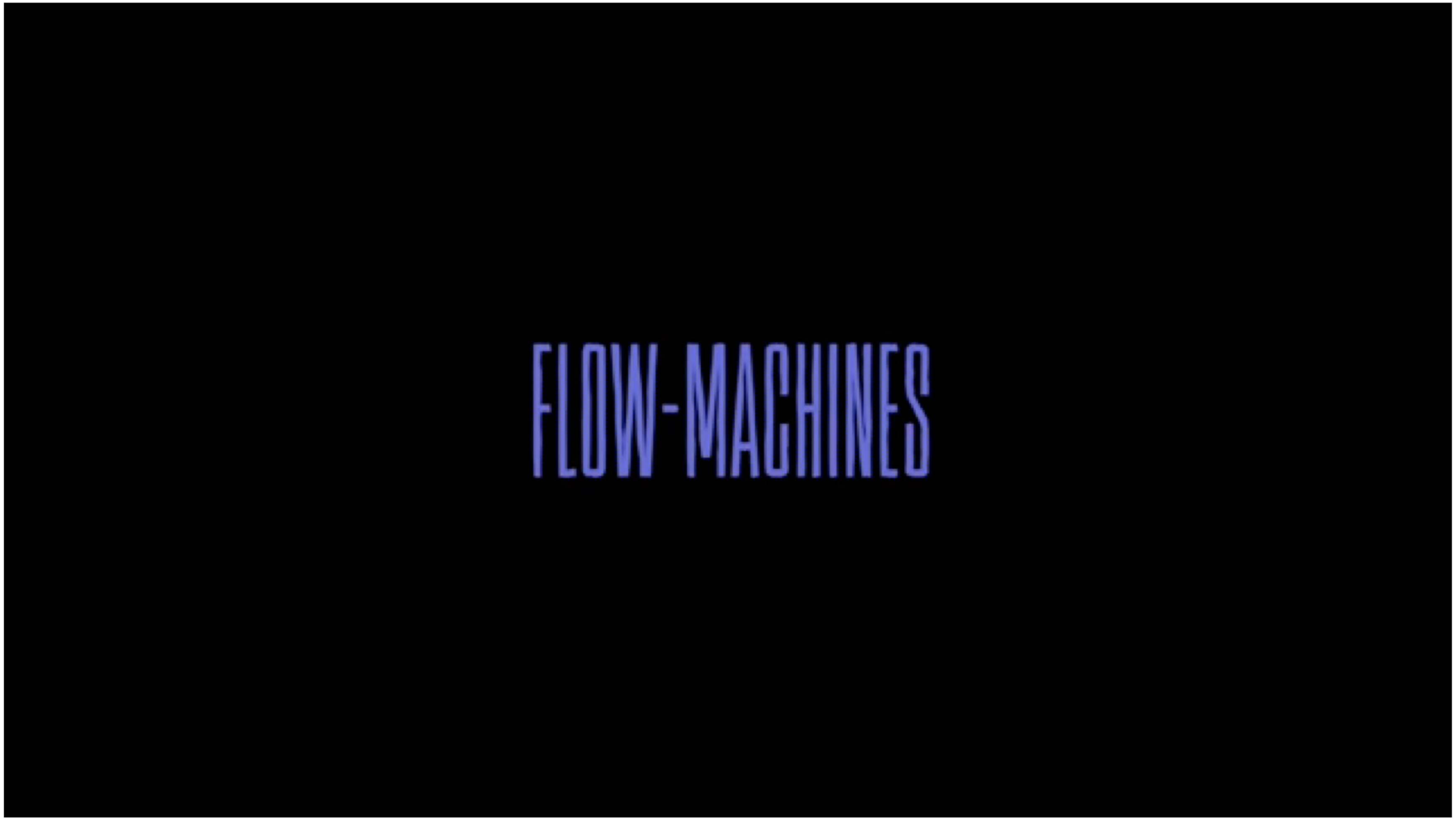


FlowComposer [Pachet et al., 2014] – Demo (B. Carré)



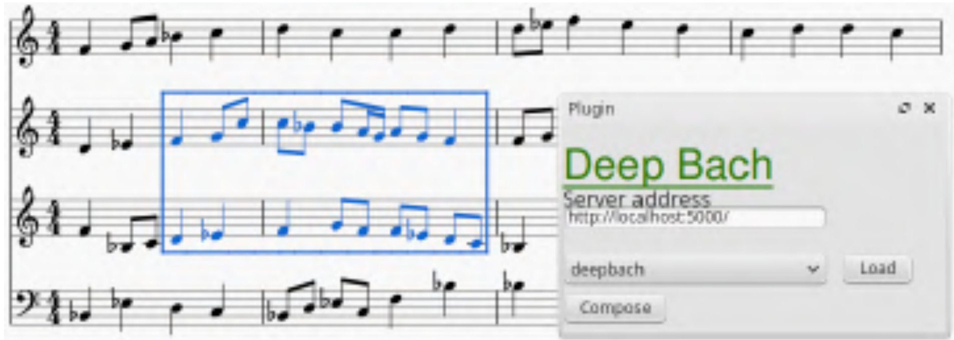
Hello World

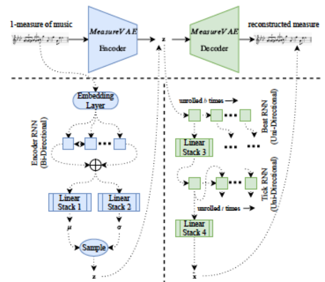
- January 2018, Hello World, Flow Records
- Making Off

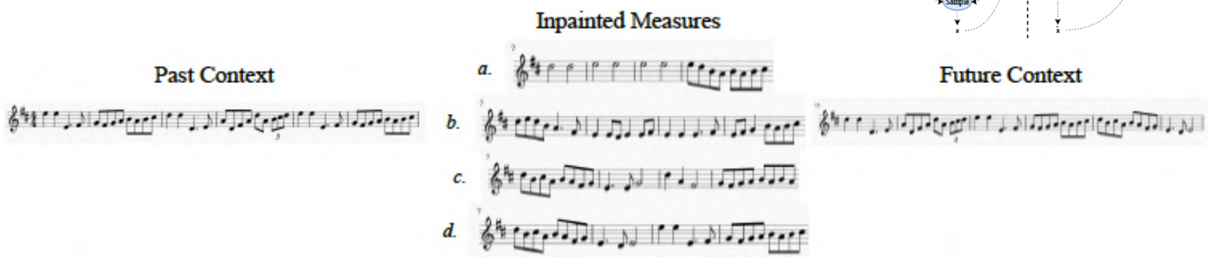


https://www.youtube.com/watch?v=yxTF-UFvoHU

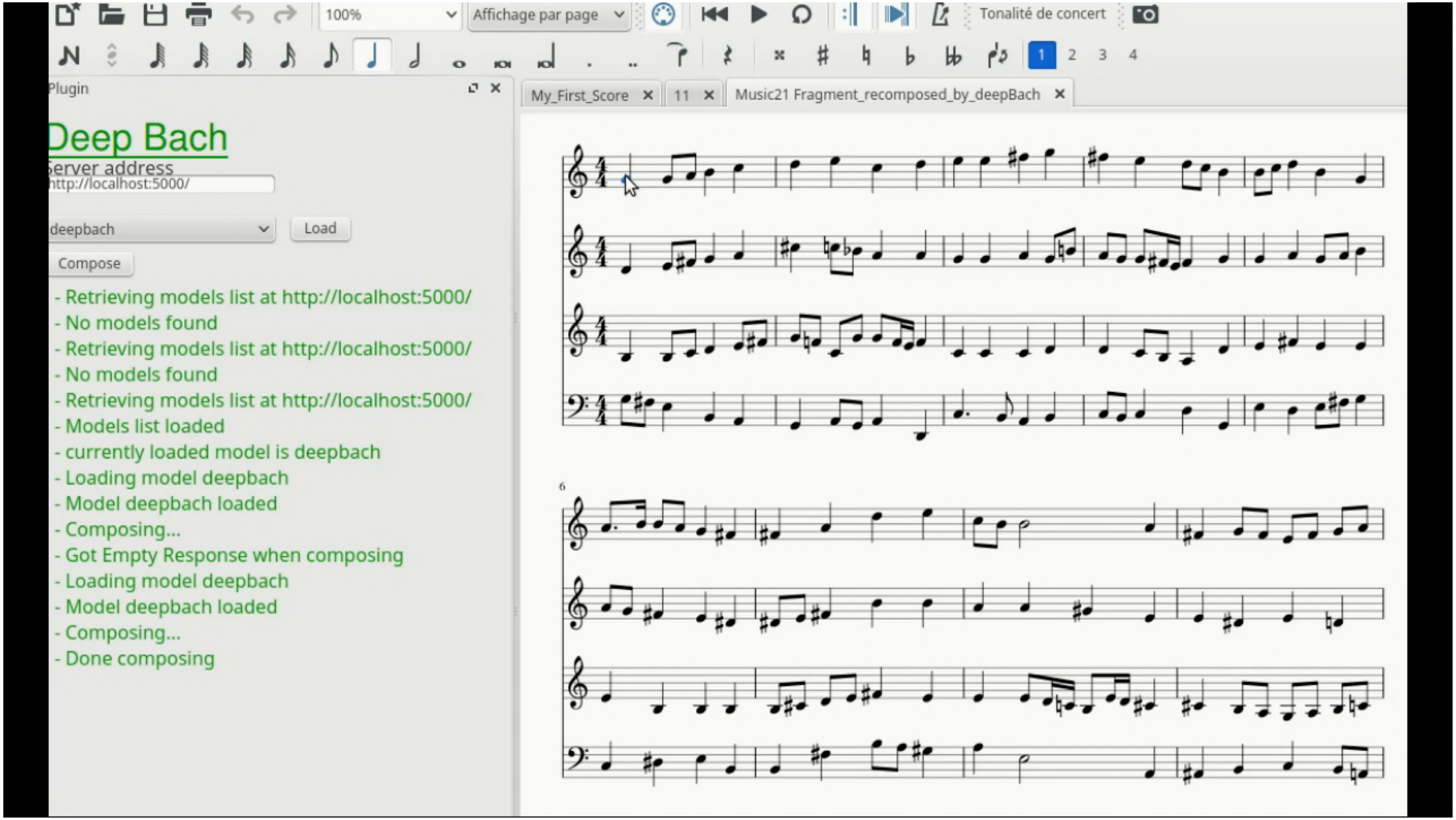
Deep Learning
Co-Creation/Assistance & Interactivity

- YΔCHT/MusicVAE [Roberts et al., 2018]
  - Non interactive Generation
  - Loops
  - Collage
- DeepBach [Hadjeres et al., 2017]
  - (Incremental Sampling)
  - Interactive/Selective Regeneration

- MeasureVAE+LatentRNN+MeasureVAE [Pati et al., 2019]
  - Inpainting
  - Previous Measure + Next measure
  - -> Latent Embeddings -> Missing Embedding
  - -> Missing Measure





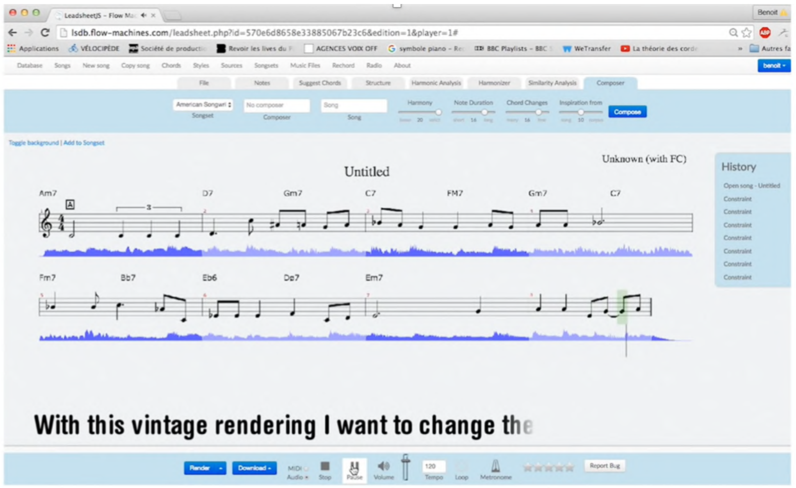
Interactivity
DeepBach [Hadjeres et al., 2017]



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

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



Conclusion/Prospects

- Deep Learning-based Music Generation
- Successes and Limits/Prospects

- Objective Loss Function Hypothesis
- Conformance Pros and Cons
- Control
- Structure
- Explication

- Markov Models (and other Models) still Interesting
- Symbolic AI (GOFAI) still Necessary
- Automated Generation vs Human-Machine Co-Creation
- New Usages

(Some) Other References

- Jordi Pons, Neural Networks For Music: A Journey Through Its History, October 2018, https://towardsdatascience.com/neural-networks-for-music-a-journey-through-its-history-91f93c3459fb
- Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2018
- Andrew Ng, Machine Learning Yearning, Deeplearning.ai
- Tom Mitchell, Machine Learning, McGraw Hill, 2017
- Pedro Domingos, The Master algorithm, Basic Books, 2015
- Judea Pearl and Dana Mackenzie, The Book of Why, Penguin Books, 2018
- Gerhard Nierhaus, Algorithmic Composition: Paradigms of Automated Music Generation, Springer, 2009
- David Cope, The Algorithmic Composer, A-R Editions, 2000
- Roger T. Dean and Alex McLean, The Oxford Handbook of Algorithmic Music, Oxford Handbooks, Oxford University Press, 2018
- Curtis Roads, The Computer Music Tutorial, MIT Press, 1996

Thank You – Questions