

Creative Machine Learning

General introduction

Pr. Philippe Esling
esling@ircam.fr



General introduction

Artificial *intelligence* ?

- Requires to first understand what intelligence is !
- Must have something to do with *thinking*
- But in a broader sense, might also be about *perception* and *action*

Why bothering at all ?

- In terms of philosophy, we would talk about problems involving these
- However, we *try to understand the mechanisms of our thinking itself*
- We seek *models* targeted at thinking, perception and actions

Our overarching goal

Construct *mathematical models reproducing human behaviors*
Hopefully helping us in understanding our own thinking process

Artificial *intelligence* ?

Hard to define intelligence and thinking

Creation ex-nihilo does not exist
(*Laplace*)

« *Man can only desire what he has already perceived* »
William Blake

— Try to invent a new animal right now —

Intelligence: *the capacity to assemble two ideas that seemed heterogeneous*
We exist (morphologically) since ~200 kYears, and around ~50 kYears

« *Take two concepts and create a third one without impairing the two first* »
Noam Chomsky

So thinking is finding similarity, discriminating and seeing common patterns

NB: Frames of Mind: the Theory of multiple intelligence – Howard Gardner
(logico-mathematic, spatial, inter-personal, corporal, linguistic, intra-personnal, musical, ecologist, existential)

To have a model, we will need a *representation* that exposes *constraints* on thinking

Artificial *intelligence* ?

POTENTIAL SLIDE HERE ON PIGEONS ?

Artificial *intelligence* ?

POTENTIAL SLIDE HERE ON SUMMARY OF QUESTION

Algorithms enabled by
Constraints exposed by
Representations supporting
Models targeted at
Thinking, perception and action

Brief history of AI

Engineer view

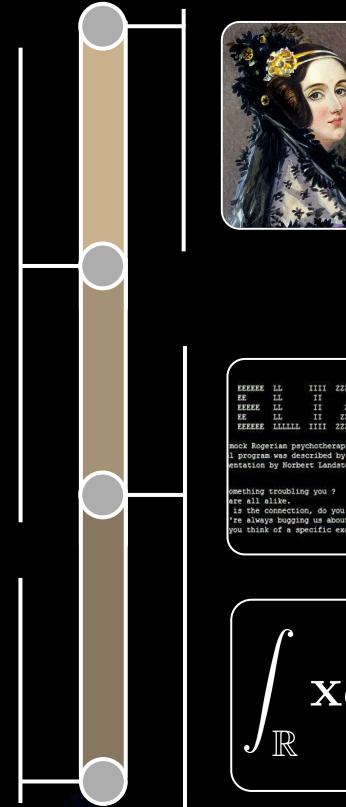
« Building smarter programs »

Scientific secret goal

« Build a computational account of intelligence »

1950 - Alan Turing

Inventor of *Bombe* and Turing test
(code-breaking machine that turned the war)



1955 - Marvin Minski

Coins the term *artificial intelligence*
“Steps towards artificial intelligence”



1966 - Shakey

First *electronic person* at Stanford
(general mobile robot with reasoning)



1970 - 1980



1st AI Winter

1842 - Lady Ada Lovelace

First-ever computer programmer
(decades before computers even existed)



$$\int_{\mathbb{R}} x dx$$

1960s - Early dawn age

1966 - ELIZA Chatbot

First-ever chatbot (*Rogerian psychotherapy*)
Beginning of Natural Language Processing (NLP)

<https://www.masswerk.at/elizabot/>

1967 - Symbolic integration

Program written by Joel Moses at MIT
Scores better than 98% of the students

Artificial *intelligence* ?

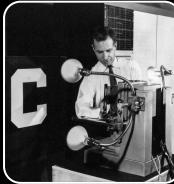
IF TIME : ZOOM IN ON ELIZA

IF TIME : ZOOM IN ON INTEGRATION

Brief history of AI

1943 - Neuron

First model by McCulloch & Pitts (purely theoretical)



1957 - Perceptron

Actual **learning machine** built by Frank Rosenblatt

Learns character recognition analogically



Lesson #2 1986 - Backpropagation

First to learn neural networks efficiently (G. Hinton)



Lesson #3 1989 - Convolutional NN

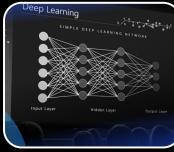
Mimicking the vision system in cats (Y. LeCun)



Lesson #4 2012 - Deep learning

Layerwise training to have deeper architectures

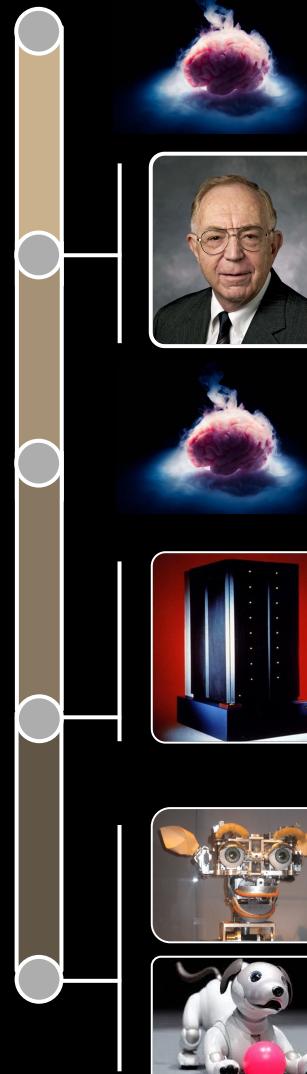
Swoop all state-of-art in classification competitions



Lesson #6 2015 - Generative model

First wave of interest in generating data

Led to current model craze (VAEs, GANs, Diffusion)



1970 - 1980 1st AI Winter

1980s - Expert systems

Emulates decisions of human experts

Rising use in healthcare (and still today)

1985 - 1995 2nd AI Winter

1997 - Deep blue

First defeat of human chess champion

Based on alpha-beta min-max algorithm

2000s - Robotic explosion

1998 - Kismet

First emotional robot

1999 - Aibo

Robotic dog

2002 - Roomba

Robotic vacuum

2011 - Siri

Personal assistant

2012 onwards Deep learning era

Artificial *intelligence* ?

POTENTIAL SLIDE HERE ON WEAK vs. STRONG AI

- + GENERAL PURPOSE vs. TASK SPECIFIC

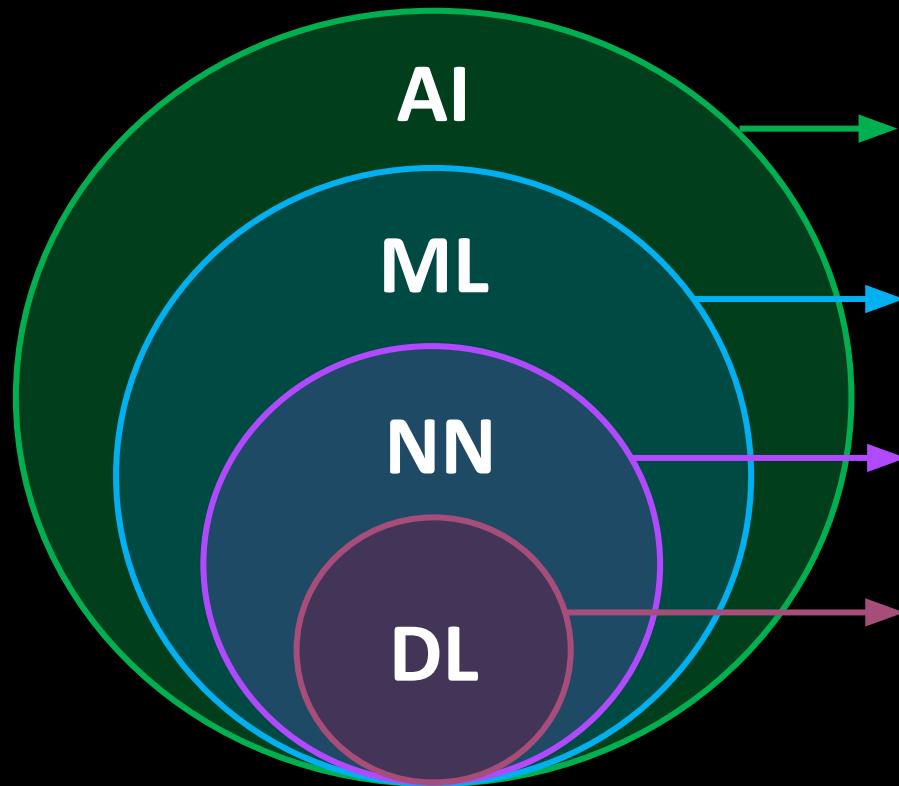
Families of Artificial Intelligence

Solving confusion around machine *intelligence*

There is wide controversies about AI (*weak vs. strong, feasible vs. fake*)

Avoiding any debate, we will only discuss here *machine learning*.

- Can we automatically learn parameters of an approximation
- The most active field recently, and also very wide topics inside it



Artificial Intelligence (AI)

Any technique allowing machines to solve human tasks

Machine Learning (ML)

Learning inference models from examples

Neural Networks (NN)

Brain-inspired ML models

Deep Learning (DL)

Building (deep) hierarchies of NN representations

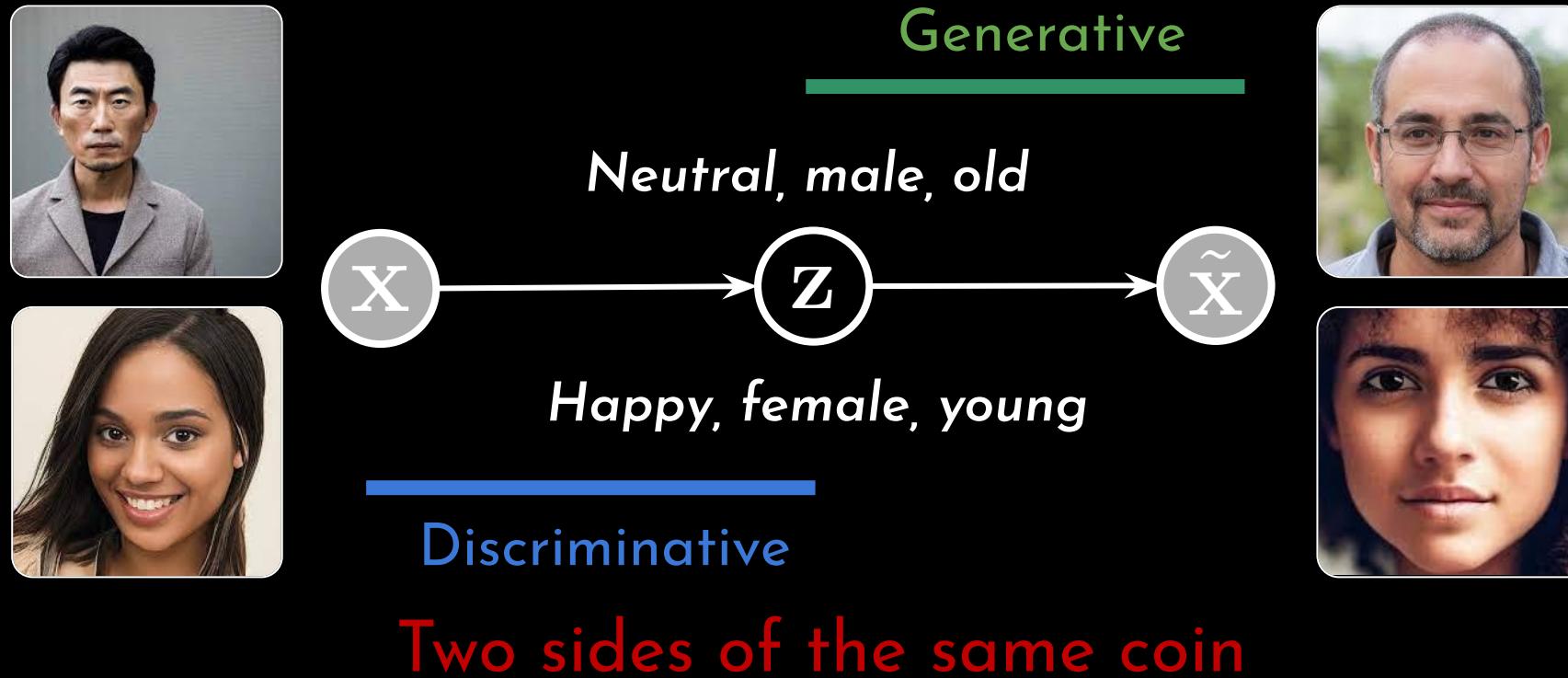
Why creative ?

3 major types of learning

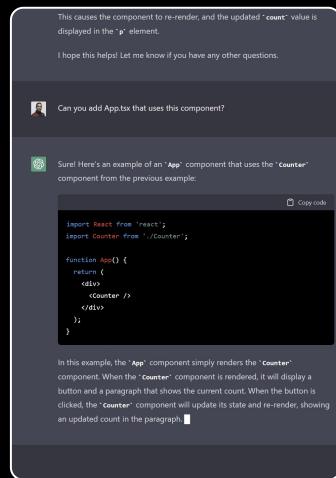
- **Supervised learning** is inferring a function from labeled training data
- **Unsupervised learning** is trying to find hidden structure in unlabeled data
- **Reinforcement learning** is acting to maximize a notion of cumulative reward

Self-supervised learning is using the data itself directly to define tasks to solve

Discriminative or generative ?



Major players in the generative field



ChatGPT

AI language model (OpenAI) based on Transformers (GPT)

Trained on vast corpus of text to generate chat answers

Impressive question answering and reasoning abilities

Lesson #5 = Transformers

Midjourney - StableDiffusion



Image-generation AI by Stability based on GANs and diffusion

Trained to generate HQ images based on prompts

Impressive capabilities used in art, entertainment, and fashion

Lesson #9 = Generative Adversarial Networks (GANs)

Lesson #10 = Diffusion models



What about musical data ?



*Creep by Radiohead
... but by The Beatles*



Generative model by OpenAI creates original music with lyrics and singing.

Combination of VQ-VAE and Transformer for raw audio at 44.1 kHz.

Can generate music in various styles, and even mimic specific artists

MusicLM



Generative model by Google to perform text-to-music

Based on AudioLM and CLIP for generative audio at 44.1 kHz.

Allows to directly prompt what music you want and generates waveform

*A fusion of reggaeton and electronic dance music, with a spacey, otherworldly sound.
Induces the experience of being lost in space [...]*

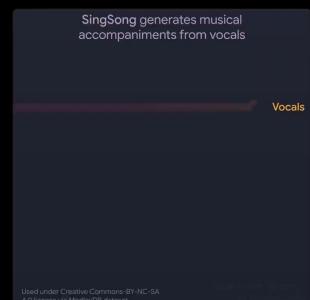


Thank you Barack !

Voice cloning

Text-to-speech allows to clone any voice

Cyberduck app



Used under Creative Commons BY-NC-SA 4.0 license via MedleyDB dataset

Reverse Karaoke

Generate music to accompany vocals

SingSong project

Course program

Goal of this course

- Ability to construct (math) and develop (code) your own *generative models*
- True and deep understanding of the mathematical theory behind those models
- Constant flow between theory, implementation and creative application

Detailed program

Discriminative

Generative

1. Introduction
2. Machine learning
3. Neural networks
4. Advanced networks
5. Deep learning
6. Probabilities and Bayesian inference
7. Latent models, EM and GMM
8. Approximate inference
9. Variational Auto-Encoders (VAEs) and flows
10. Adversarial learning (GANs)
11. Diffusion models

Why should you even listen to me



Professor Philippe Esling - esling@ircam.fr



Sorbonne université (Paris)

Teaching machine learning for 10 years



The University of Tokyo

Invited professor in ML



ircam



ACIDS

ACIDS - Artificial Creative Intelligence and Data Science

Lead of research group (in A/S team) on generative models for music



Japanese-French Laboratory for Informatics

Invited researcher for leading the Creative AI axis



Google Research (Brain)

Invited researcher at the Magenta team

Presentation of my research in the second half of this course

Guest lecturers



Taketo Akama

Industrial research
Sony CSL Tokyo
Generative models



Pr. Tatsuya Harada

Academic research
University of Tokyo
Computer vision, robotics

Course information

Basic administrative information

Type : 2-credits graduate school course

Period : April - July 2023

Span : 14 classes of 105 minutes

Date : Every thursday at 2:55 - 4:40 pm (JST)

Onsite : *Room 214, 2nd Floor, Sci. 7 Building*

Online course

[https://u-tokyo-ac-jp.zoom.us/j/
81363691008?pwd=cmxiRURMdM9udXBKbTNjQkZvbINFQT09](https://u-tokyo-ac-jp.zoom.us/j/81363691008?pwd=cmxiRURMdM9udXBKbTNjQkZvbINFQT09)



Github page

github.com/esling/creative_ml

Professor

Philippe Esling - esling@ircam.fr

Pre-requisites

Notions required but not covered in this course

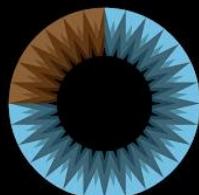
Resources to help you out if you feel behind on these topics

Calculus

Limits and infinitesimals
Differential calculus
Integral calculus

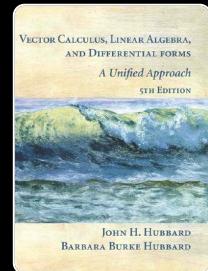
Linear algebra

Vector spaces
Matrix calculus
Geometry and analysis



3 Blue 1 Brown

[Essence of linear algebra](#)
[Essence of calculus](#)



Hubbard J. & Hubbard B.B

Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach (5th edition)

- Linear algebra [slides](#)
- Probability [revision slides](#)
- Statistics [course notes](#)
- Sampling [pages](#)
- [Matrix Cookbook](#)

Programming

Python basics
Functional notions
Object-oriented prog.

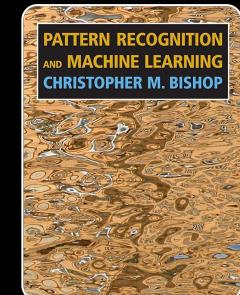
Data analysis

Data visualization
Basic filtering, processing



Machine learning bible

Christopher M. Bishop
Pattern recognition and machine learning



Course teaser

Overarching construction

Course heavily based on **interactive learning** principles

Merging **theory and application** as much as possible



#2 - Jupyter notebooks



#3 - Interactive panels

- Interactive coding environment with exercises
- Used during the class with dynamic panels
- Will also serve to fill in evaluated exercises
- Submission system impending

Fully online

Slides
slides.google.com



Colab
colab.google.com



GitHub **fork** (+ star)
Jupyterlab environment

Best solution



Course and exercises are always evolving

Help each others out and work as groups

Post issues on GitHub and Discord

Join discord

<https://discord.gg/gpF82wnh>



Understanding philosophy

General principle

We will be exploring all concepts by

1. Deriving the mathematical bases ourselves
2. Implementing low-level aspects directly as well
3. Moving up to higher-level (simpler) implementation

Essential principle for a profound understanding of concepts

ML Libraries

Concept mirrored in our choice of libraries

1. **Numpy** forces us to do mathematical derivation
2. **JAX** adds automatic differentiation over Numpy
3. **Pytorch** allows to define networks in seconds



Numpy



JAX



PyTorch

Other helpful libraries



Librosa



Scipy



Seaborn



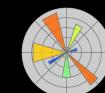
Bokeh



Scikit



Panel



Matplot



Manim

Evaluation methodology

One major evaluation and project



Continuous evaluation

Course based on interactive Jupyter **notebooks to fill and submit**

- Each notebook contains a set of exercises
- We will start exercises together but you have to finish at home
- Automatic submission system will be set in place
- Rolling deadlines of 3 sessions (~3 weeks) to submit your work

Project-based evaluation

Supplementary evaluation will be based on a large-scale project

- Several projects will be proposed
- Groups of **4 to 8 students** per project
- Projects are coded in **Python** (cf. Coding style section underneath)
- All projects should be **accompanied with a small report** in english
- The **report should be of 8 pages** following scientific papers style
- Reports are written in LaTeX with famous conferences (*ICLR, NeurIPS*) format style

Setup and course follow-up

Following the course

All lessons will follow the same structure

Mixture of animated slides and notebooks

You can follow the online or offline version

Slides

Notebooks

Online



Offline



The course is updated prior to each lesson Course not available yet



Installing the course github.com/esling/creative_ml

You need a GitHub account



1. **Star the repo** to get updates on new courses and corrections
2. **Fork the repo** to have your own copy (for exercises)
3. **Clone the repo** to your local machine

Setup your environment

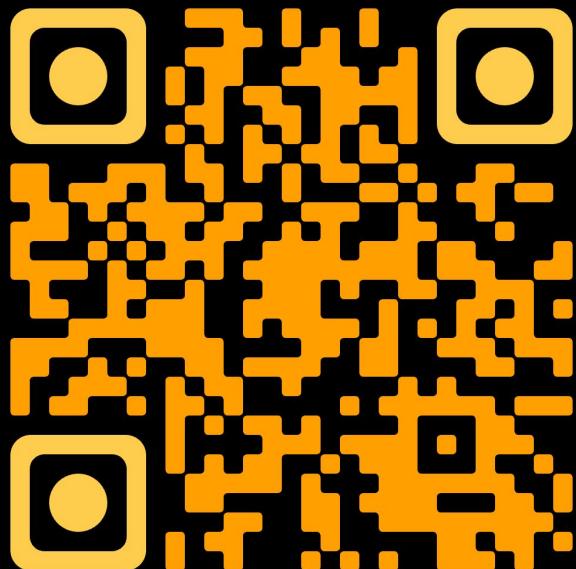


1. **Install jupyter lab** (see instructions in Setup section)
2. **Install all required libraries** (see the requirements.txt file)
3. **Test your setup** by using the 00_setup.ipynb notebook

[Setup time]

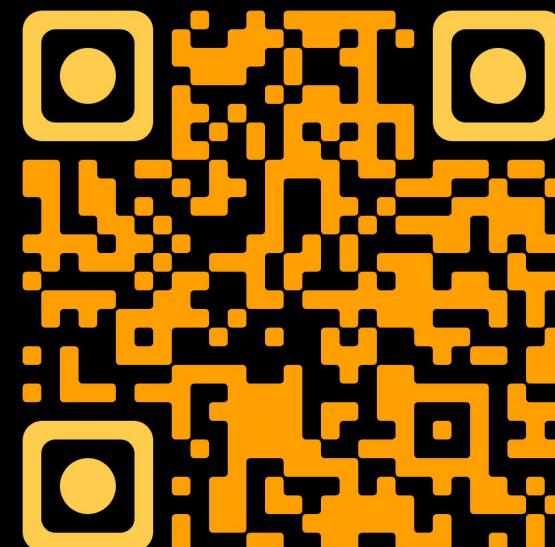
5 minutes + 5 minutes break

Github page



github.com/esling/creative_ml

Installation guide



creativecommons.org/licenses/by-sa/4.0/

Machine learning

Problem statement

We witness a phenomenon
Through a set of observations

$$(x_1, y_1) \ (x_2, y_2) \ \dots \ (x_n, y_n)$$

We know there exists a relation
between \mathcal{X} and \mathcal{Y}

But we do not know its nature

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Machine learning constructs
approximations

$$f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$$

such that

$$y_i \approx \bar{y}_i = f_\theta(x_i)$$

Regression and classification

The two most classical machine learning problems

Navigation icons: back, forward, search, etc.

Machine learning

Formal problem statement

We aim to model the relationship between \mathcal{X} and \mathcal{Y}

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

We collect a **dataset** that is representative of that relation

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_n, \mathbf{y}_n)\}$$

We will construct a **parametric approximation** $f_\theta \in \mathcal{F}_\Theta$

$$f_\theta : \mathcal{X} \rightarrow \mathcal{Y} \quad \bar{\mathbf{y}} = f_\theta(\mathbf{x})$$

We want to find the best f_θ^* so that $\bar{\mathbf{y}} \approx \mathbf{y}$

So we need to find the best parameters $\theta^* \in \Theta$

Need to compute the errors (**loss**) of our model

$$\mathcal{L}(\bar{\mathbf{y}}, \mathbf{y} \mid f_\theta, \theta)$$

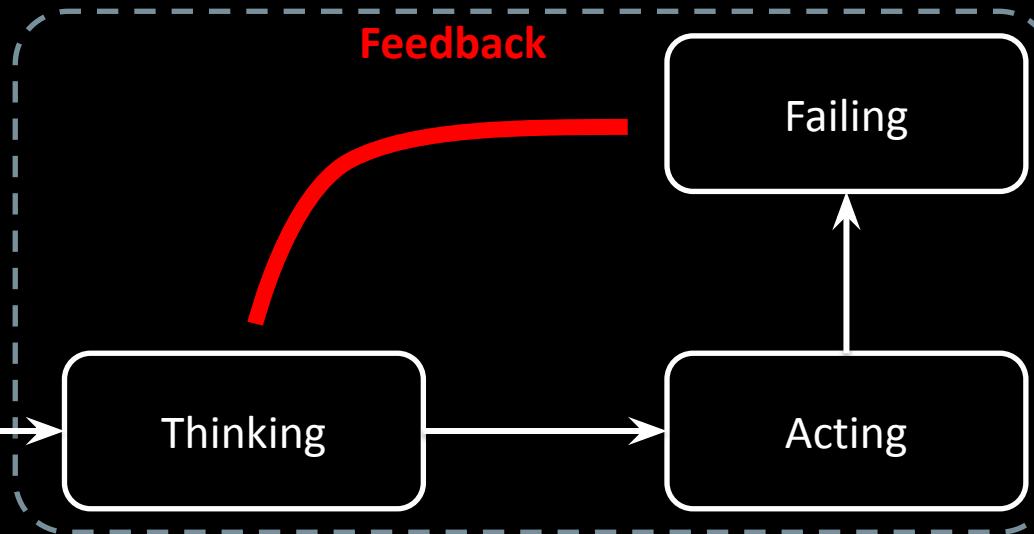
And minimize (**optimize**) this amount of errors

$$\theta^* = \operatorname{argmin}_\theta \mathcal{L}(\bar{\mathbf{y}}, \mathbf{y} \mid f_\theta, \theta)$$

Learning as error minimization

Why it makes sense in real-life

Optimization
(error-minimization)



- **Goal-directed learning** = acquiring, modifying or reinforcing knowledge
- Most problems can be summarized by a **trial-error process**
- Is a nickname for **optimization** (through **error minimization**)
 - So what can we do with ML ? Some examples from our team (ACIDS)

ACIDS Project

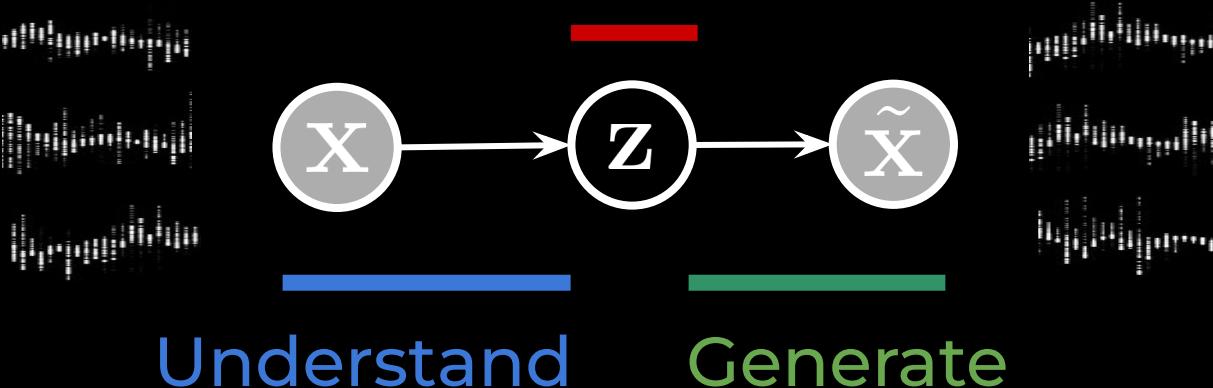


Artificial Creative Intelligence and Data Science

Research group at IRCAM, Sorbonne, Paris - Analysis / Synthesis team
Academic machine learning research since 2012

Theoretical and applied research in generative models for musical creation

Control

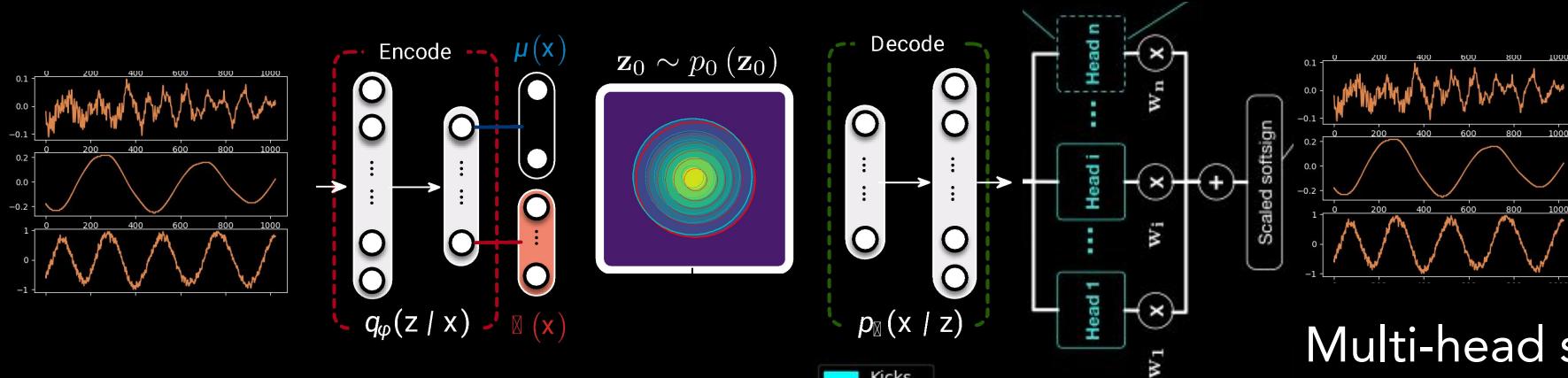


- Neural drum machine
- Timbre transfer applications
- High-level controls
- **[RAVE]** - Real-time synthesis
- **[FlowSynth]** - Synthesizer control
- **[Neurorack]** - Neural synthesizer

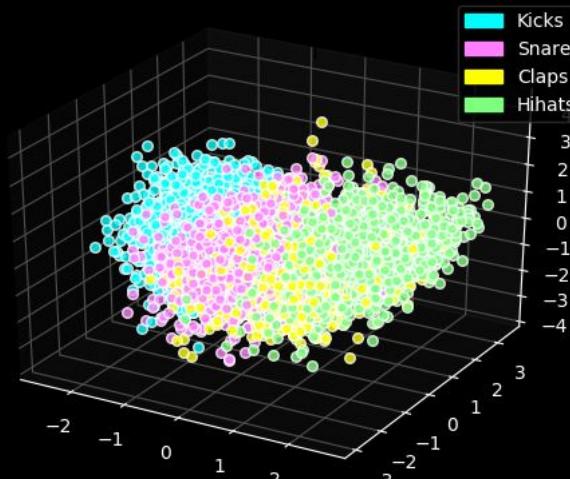
Pushing the limits of musical creativity through the use of machine learning
Creating the next generation of musical instruments

Applications - Drums generation

Drums are mostly noisy, require end-to-end training with waveforms

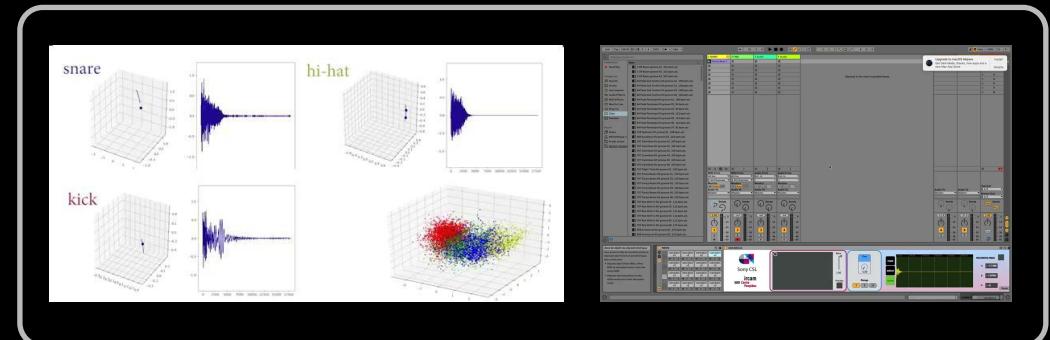


Work with Axel CRS and
Cyran Aouameur



Real-time version demonstration

Multi-head spectrum inversion
Heads unsupervisedly split into bands



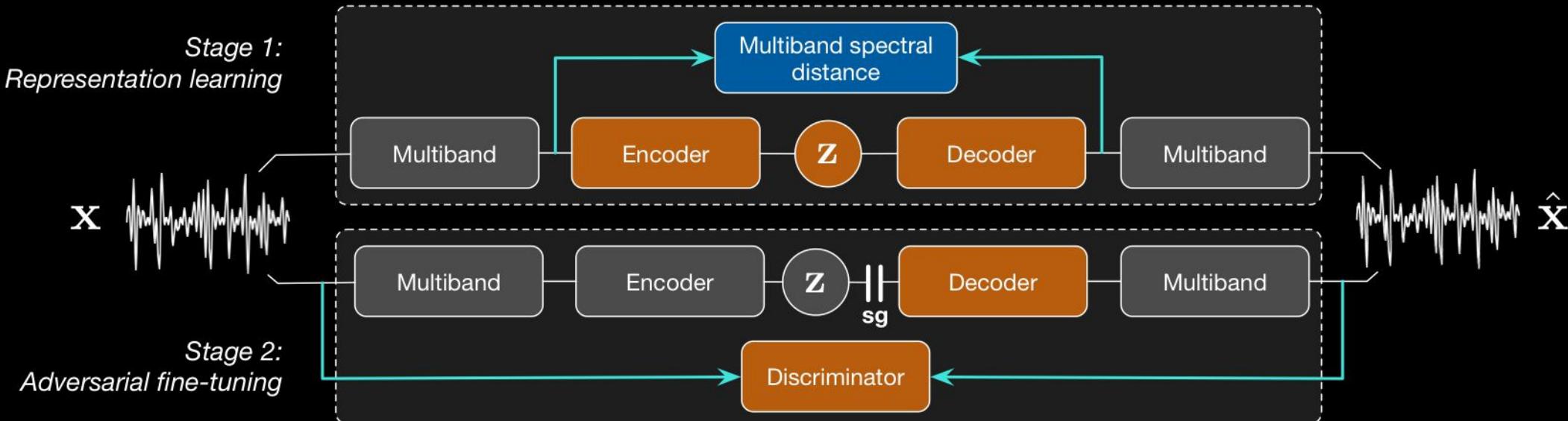
Ableton plugin and web-based interface

Aouameur, C., Esling, P. and Hadjeres G. Neural drum machine: An interactive system for real-time synthesis of drum sounds. arXiv:1907.02637, 2019.

RAVE - Realtime audio waveform VAE

Recently, we vastly improved VAE generation *Full paper submitted to ICLR 2022*

Work with Antoine Caillon



High-quality (48kHz) synthesis with 25x realtime on a laptop CPU 🔊

Perform *timbre transfer* without any specific training 🔊 → 🔊

Allows *polyphonic unconditional generation* of signals 🔊

Caillon, A. Esling, P. RAVE: A variational auto-encoder for fast and high-quality neural audio synthesis. arXiv:2111.05011, 2021.

RAVE - Multiple outputs / interactions



Realtime demo



Jetson embedding

RAVE is available as multiple types of open source possibilities

<https://github.com/acids-ircam/RAVE>

PureData / MaxMSP

https://github.com/acids-ircam/nn_tilde

VST (multi-platform)

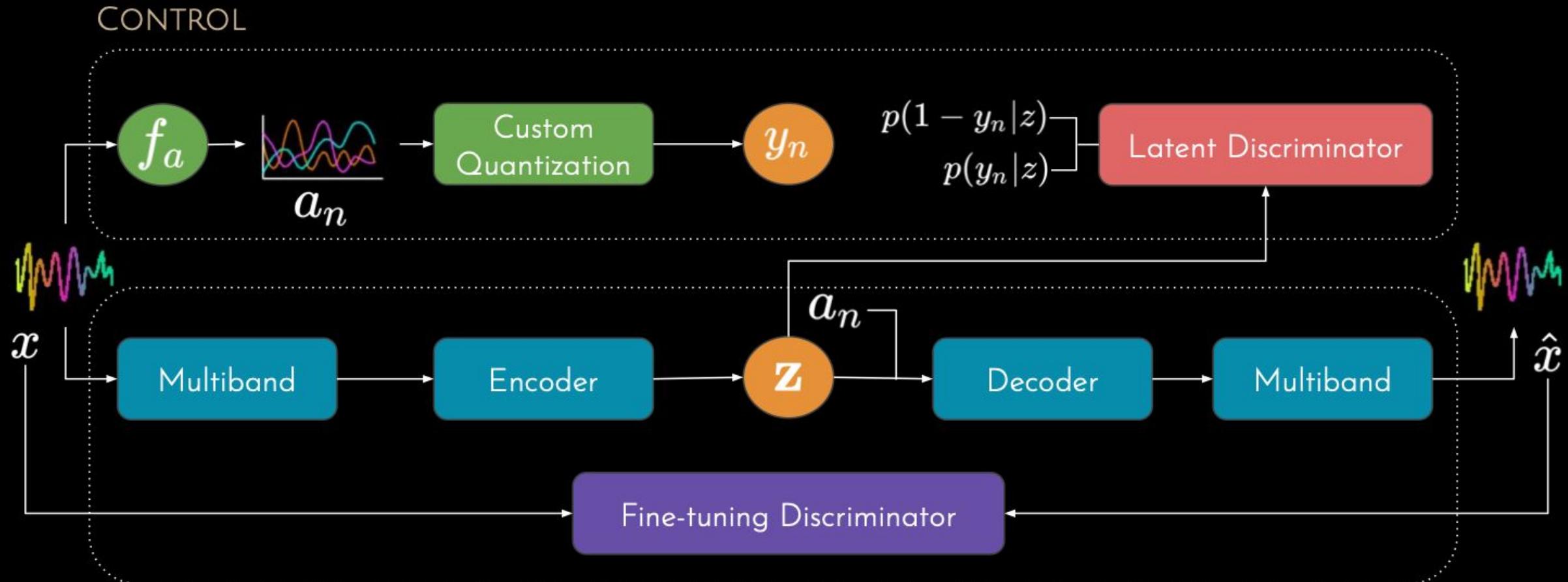
https://github.com/acids-ircam/rave_vst

Caillon, A. Esling, P. RAVE: A variational auto-encoder for fast and high-quality neural audio synthesis. arXiv:2111.05011, 2021.

Descriptor-based synthesis

Further improve the real-time generative model with high-level control

Work with Ninon Devis, Nils Demerlé, Sarah Nabi and David Genova



Devis, N. Demerle, N. Nabi, S. Genova, D. and Esling, P. Continuous descriptor-based control for deep audio synthesis. ICASSP 2023

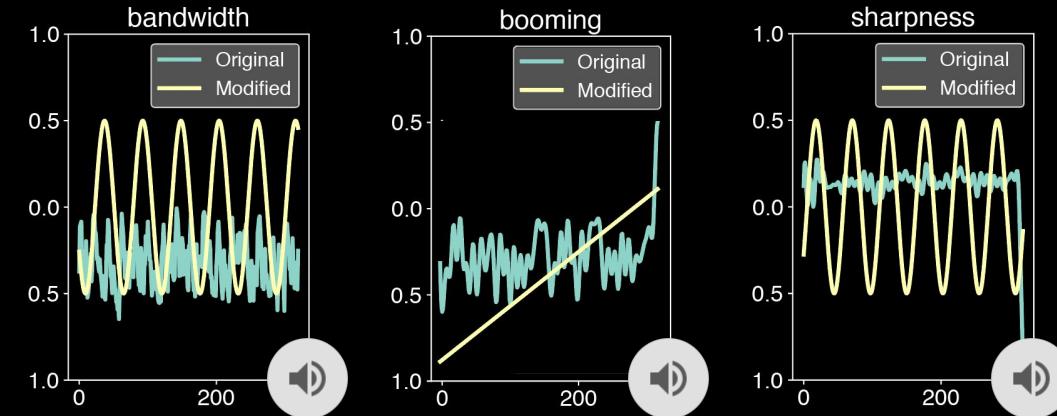
Descriptor-based synthesis

Descriptor-based synthesis (attribute transfer)

Model generation



(Derbouka)



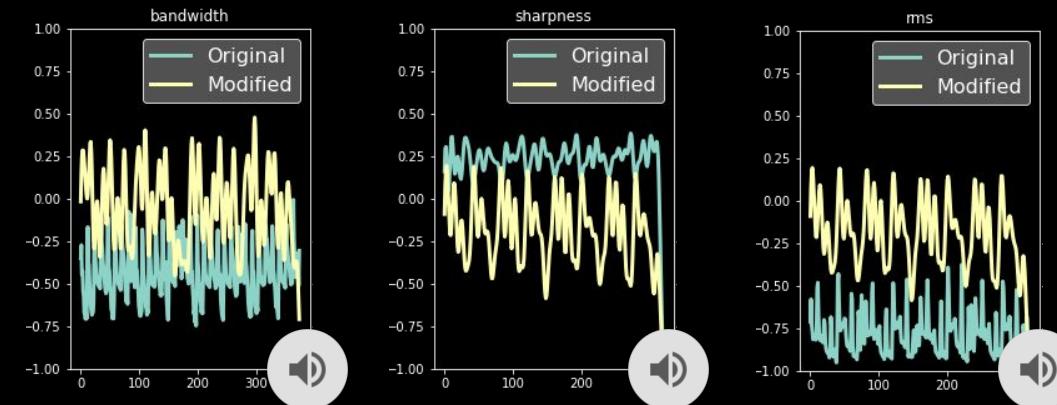
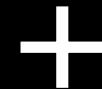
Combining timbre and attribute transfer



(Derbouka)



(Japanese)



Devis, N. Demerle, N. Nabi, S. Genova, D. and Esling, P. Continuous descriptor-based control for deep audio synthesis. ICASSP 2023

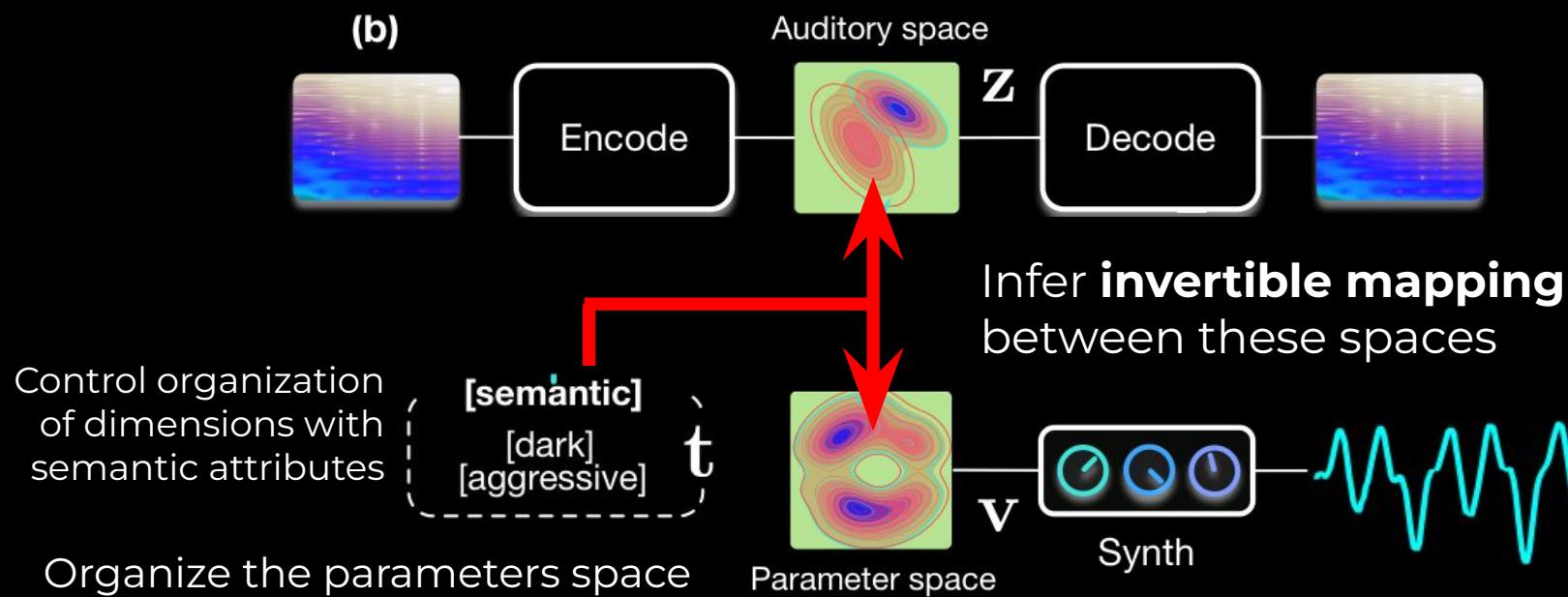
Universal synthesizer control with FlowSynth

Compared to previous approaches (limited to parameters inference)



Our proposal = Universal synthesizer control with flows

Organize the auditory space of a synthesizer audio possibilities



FlowSynth - Demo video

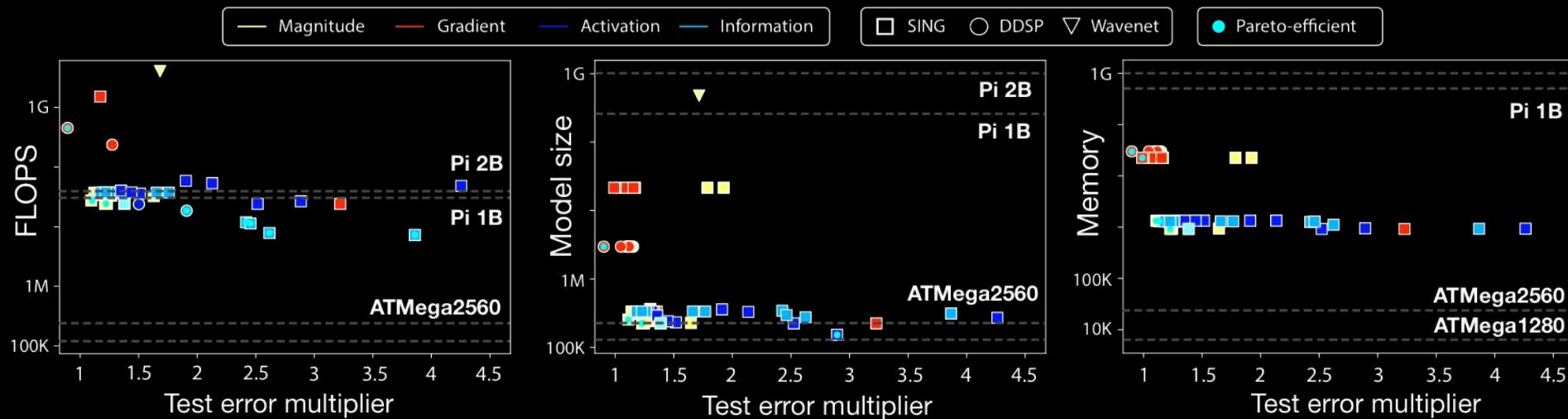
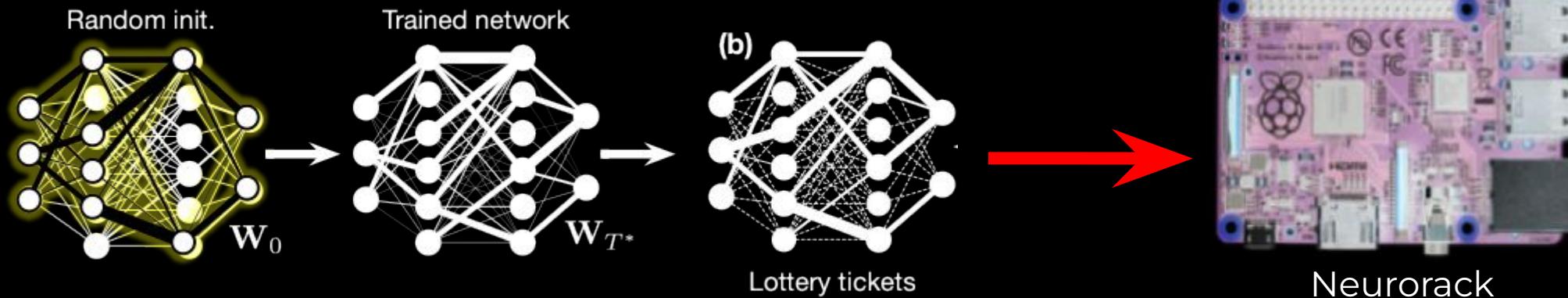


https://github.com/acids-ircam/flow_synthesizer

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

Lottery ticket hypothesis

Reducing deep models onto embedded chips

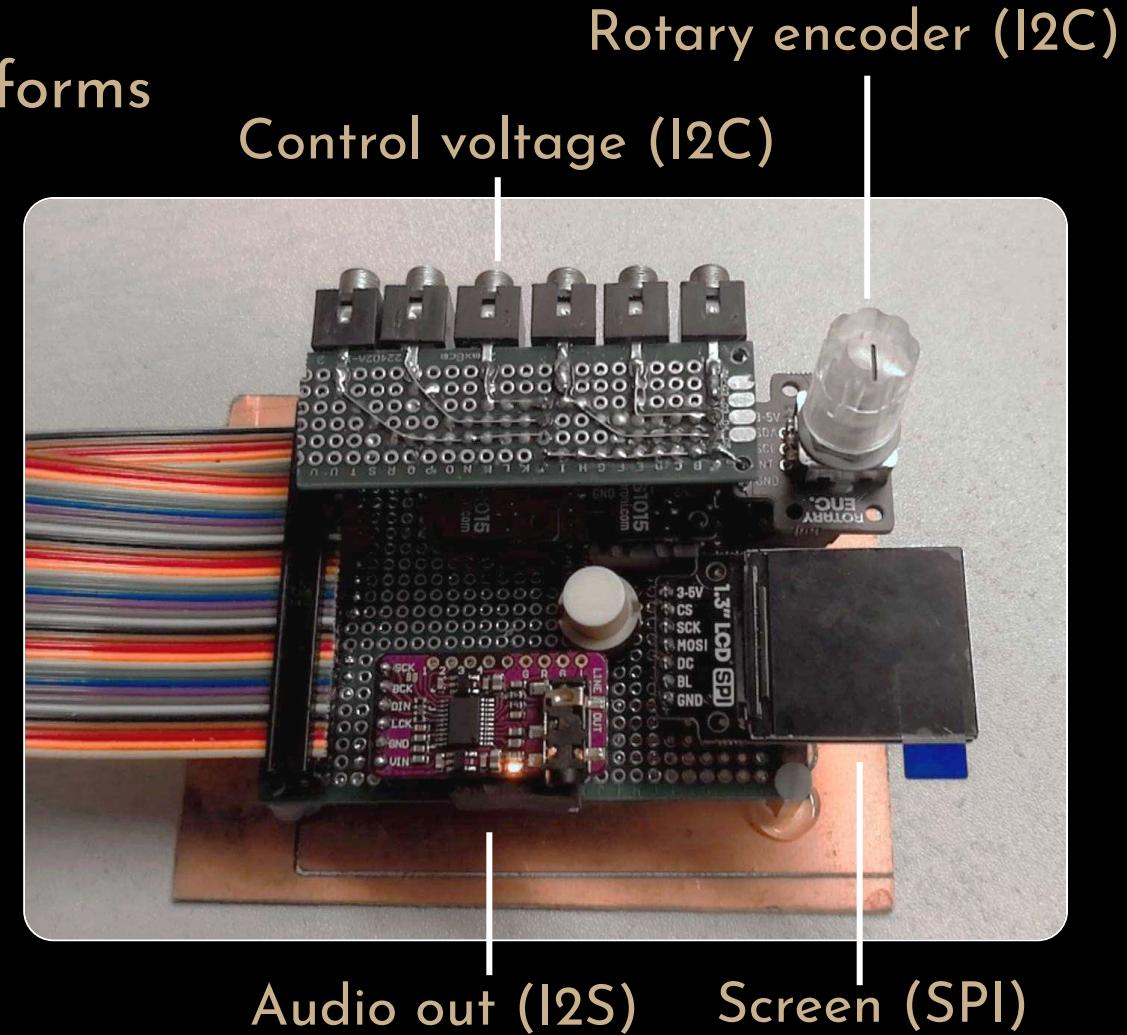
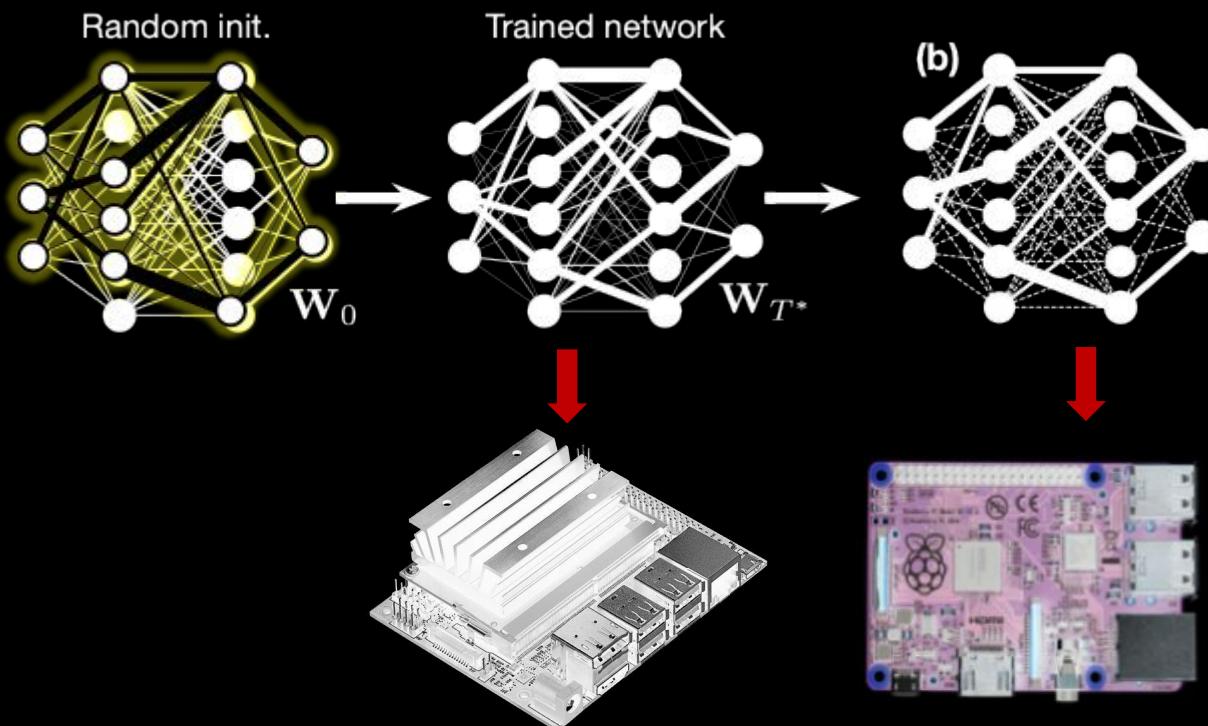


Esling, et al. *Diet deep generative audio models with structured lottery*. Proceedings of International Conference on Digital Audio Effects (DAFx), 2020

Neurorack first prototype

Deep models in embedded constrained platforms

Neurorack // Work with **Ninon Devis** and **Martin Vert**

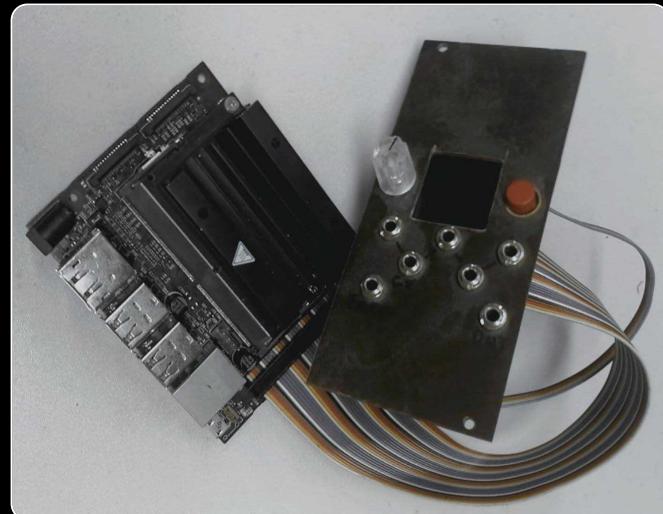


First-ever Neurorack prototype built ourselves with simple components

Devis, N. and Esling, P. *Neurorack: deep audio learning in hardware synthesizers*. EPFL PhD Conference (Lausanne), 2021

Final neurorack prototype interface

Navigate the menu Display interactive menu
Rotary **Screen**



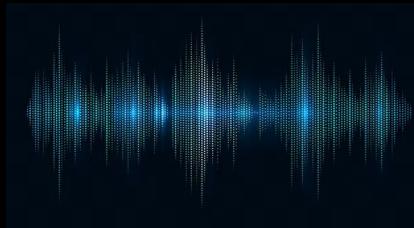
Control Voltage
Vary sound features

Gate
Trigger generation or interpolation

Button
Select and click

Audio out

Other applications (*discriminative*)



Genre: Acid techno
Tempo: 140 BPM
Key: B#
Lyrics: Avin' it



Separating individual instruments or voices from a mixed audio signal.

Usual problem of frequency coverage and bandwidth

Recent models provide stunning results

Jansonn A. et al. (2017). "Singing Voice Separation with Deep U-Net Convolutional Networks." ISMIR.

Music Information Retrieval (MIR)

Music Genre Classification

Automatically classifying music genres from audio samples.

Nam, J et a. (2018). Deep learning for audio-based music classification and tagging. IEEE signal processing magazine, 36(1), 41-51
https://e-tarjome.com/storage/panel/fileuploads/2019-07-04/1562228921_E11422-e-tarjome.pdf

Beat and Tempo Detection

Analyzing music to detect beat patterns and tempo using deep learning.

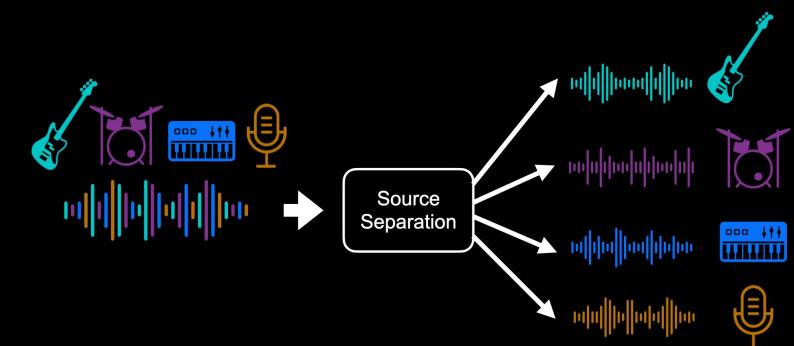
Schreiber, H., & Müller, M. (2018). "A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network." ISMIR.
https://www.tagtraum.com/download/2018_schreiber_tempo_cnn.pdf

Automatic Music Tagging

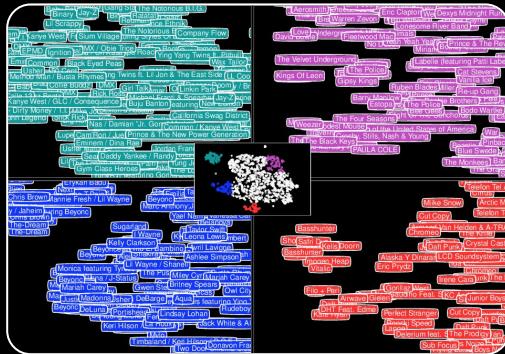
Assigning genre, mood, or other descriptive tags to songs using deep learning.

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2016). "Convolutional Recurrent Neural Networks for Music Classification." arXiv:1609.04243.
<https://arxiv.org/abs/1609.04243>

Source Separation



Other applications (*discriminative*)



Music Recommendation

Using deep learning to create personalized music recommendations

Analyze users' preference and listening history.

Well-known *cold start* problem

Aaron van den Oord et al. "Deep Content-based Music Recommendation" (2013)

<https://proceedings.neurips.cc/paper/2013/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf>

Query-by-Humming

Fingerprinting for efficient music retrieval in large databases.

Mapping the query and songs to a common reduced space

Provides efficient query-by-humming system

Mostafa, N. et al (2017). A Note Based Query By Humming System Using Convolutional Neural Network. In Interspeech
https://www.isca-speech.org/archive_v0/Interspeech_2017/pdfs/1590.PDF



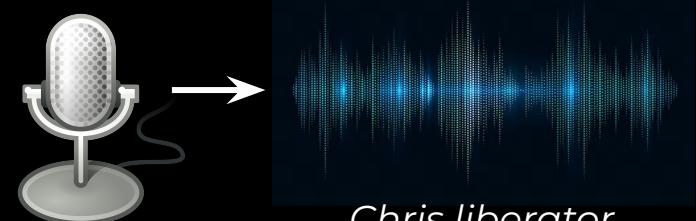
Audio-to-Score Conversion

Transcribing audio recordings into sheet music using deep models.

Non-uniqueness of the solution and polyphonic problem

Almost solved for piano but still hard for orchestra

Hawthorne et al. (2018). "Onsets and Frames: Dual-Objective Piano Transcription." ISMIR 2018.
<https://arxiv.org/abs/1710.11153>



Chris liberator

Other applications (generative)



Symbolic music generation

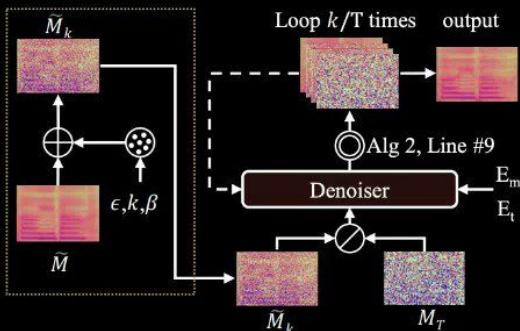
Generating symbolic music with advanced models

Handle long-range dependencies for coherent musical structures.

Huang, C. Z. A., & Du, S. (2020). "Pop Music Transformer: Generating Music with Rhythm and Harmony." *arXiv:2002.00212*.
<https://arxiv.org/abs/2002.00212>

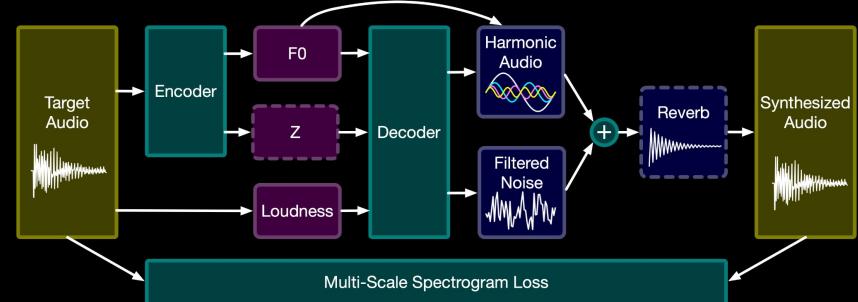
Generating novel synthesized instruments and soundscapes
Current trends increasingly mixes traditional DSP elements

Engel, J. et al. (2020). "DDSP: Differentiable Digital Signal Processing." *ICLR. 2020*
<https://openreview.net/forum?id=B1x1ma4tDr>



Analyze structure to create seamless transitions between songs
Enables automated DJing and playlist creation.

Huang, J. et al. Modeling the compatibility of stem tracks to generate music mashups. In *AAAI 2021*
<https://ojs.aaai.org/index.php/AAAI/article/view/16092/15899>



Text-to-singing

Singing voice generation based on input text

Also needs to handle prosody and emotional response

Liu, J. et al. (2022). *Diffsinger: Singing voice synthesis via shallow diffusion mechanism*. In *AAAI 2022*.
<https://ojs.aaai.org/index.php/AAAI/article/view/21350/21099>

Algorithmic DJing

