

Contents

1	Introduction	2
1.1	Methodology	2
1.2	Tiered Learning Resources	2
1.3	Music-Forward Resources	3
1.4	Active Learning	3
2	Ecosystem of Learning Materials	4
2.1	Creative Coding Environment Materials	4
2.2	Web Reference	4
2.3	Learn Articles	5
2.4	Explore Articles	5
2.4.1	Interviews	5
2.4.2	Made With	5
2.5	Discourse	5
2.6	Inter-connectivity of Resources	5
3	“101” Tutorials	5
3.1	MLP Regressor & Classifier	5
3.2	2D Sound Browser	5
4	Common Learning Challenges & Strategies	5
4.1	New Ways of Using Buffers	5
4.2	Advanced Neural Network Tutorials	6
4.3	Stateful Objects	6
4.4	Threading	6
4.4.1	Max	6
4.4.2	SC	6
4.5	Dealing with Time	6
4.6	Navigating Human & Machine Assumptions	6
4.7	De-Myth-ifying Machine Learning	6
5	Multi-Day Workshops	6
6	Suggested Semester-Long Syllabus	6
7	Relevance to Contemporary Society	6
8	More Information	6

1 Introduction

The FluCoMa Toolkit is not limited to the actual code objects that use the “Fluid” prefix. FluCoMa is also a collection of learning resources, code examples, commissioned artworks, musicological articles, interviews, podcasts, a philosophy about interface design for creative coding, a conversation about the future of computer music, a curriculum of machine listening and machine learning topics, a community of users around the world, and more. The materials included here extend this ecosystem to encompass resources for pedagogues that might be used to teach FluCoMa in various settings. While some of the ideas and resources presented below are FluCoMa-specific, many of them are toolkit-agnostic and we hope that they can be used by anyone looking to teach machine listening and machine learning for creative music making.

The extent of the FluCoMa ecosystem supports our belief that “providing the tools” is not enough to achieve FluCoMa’s mission: *enable techno-fluent musicians to use machine listening and machine learning in their creative practices*. To truly enable these artists we also must provide the motivation, inspiration, and knowledge, all in a way that is learnable for our main user base: computer musicians. Topics in machine listening, machine learning, computational thinking, and data science are often not included in the training of electronic musicians and therefore our task is to build bridges of understanding from the knowledge of computer music training to a degree of fluency with these topics that enables creative music making.

1.1 Methodology

The learning materials included have been created through participatory, iterative, and interactive design. The first draft of learning materials were developed based on feedback from the [composers commissioned](#) to create music with early versions of the toolkit. Their feedback on what was confusing about the initial versions of the tools, resources, examples, etc. led to a second draft of learning materials which were then used in over thirty workshops around the world with computer musicians from various backgrounds. The feedback from these workshop participants further refined the learning materials and have led to the broad ecosystem of learning resources now found as part of the FluCoMa Toolkit.

1.2 Tiered Learning Resources

Because different learners will desire different degrees of fluency with these topics, we have tried to tier the learning resources accordingly. The most proximal resources (such as [help files](#)) provide a working understanding of what an algorithm does alongside a musical example of how it might be used. Additional resources (such as the [internet resources](#)) provide a deeper understanding that might satisfy one’s curiosity and/or build an intuition of what is happening “under the hood”, both of which can enable a more informed manipulation of the tools. When appropriate, we also link to the white paper that describes the algorithm from an engineering perspective, should the learner wish to understand with that amount of technical detail.

In some cases the path of pursuing additional resources is more extensive and less linear. This is especially true for the more complex tools in FluCoMa, such as with the neural network objects which have multiple web pages dedicated to them with varying degrees of technical information, any one of which might come after an initial introduction, but when taken all together encompass the degree of fluency we suggest for our learners and users. Learners who are eager to jump to manipulating the many neural network parameters might jump to [MLP Parameters](#), while a user who needs a little more time absorbing how a neural network works might opt for [MLP Training](#). (Also see [MLPRegressor](#), [MLPClassifier](#), and [Training-Testing Split](#))

Tiered learning resources allow the learner to pursue knowledge as far as they deem appropriate to feed their creative practice in a given moment. Providing the learner what the *need* to know *when* they *need* to know it enables them to stay focused on a creative idea and not become overwhelmed by what could be a very large body of knowledge with a steep learning curve. This sensitivity to the relationship between creative pursuits and technical knowledge reflects earlier findings of FluCoMa outlined [here](#) as “‘Techno-Fluency’ and ‘Divergence’”. By offering links to further resources the user knows where to keep learning if they need it at the moment or if in the future they decide to continue exploring.

1.3 Music-Forward Resources

A unique aspect of the learning resources developed is the pair of beginning and destination points identified: our target learner is a creative musician with fluency in a creative coding environment (Max, SuperCollider, or Pure Data) and our goal is to provide them the knowledge needed to expand their creative practice with machine listening and machine learning algorithms. Keeping this in mind, we have always tried to keep our learning materials and examples musically oriented (as can be seen below). We aim to have the help files and example code make sound in a creative way. When possible we always offer test files and thought experiments that include materials familiar and relevant to our learner such as instrument samples, drum loops, synthesizer settings, measures of pitch and loudness, etc. We hope this not only explains an object and its interface, but also provides some copy-and-paste code to get started quickly, and generally gets the creative juices flowing while a user is engaging in the learning process.

1.4 Active Learning

Bringing their own corpus, datasets, synthesizers Using creativity as part of the assignments Cross-environment peer learning Problem Solving

2 Ecosystem of Learning Materials

2.1 Creative Coding Environment Materials

Each creative coding environment (CCE) supported by FluCoMa (Max, SuperCollider, and Pure Data) has a native system for offering reference materials. In Max and Pure Data a “help file” provides annotated examples, while a “reference” offers additional description and detail about parameters. In SuperCollider all of this information is contained in one “help file” document. Despite this difference in interface, we have strived to keep the CCE-based FluCoMa materials similar across all three environments. This is enabled, in part, by the [shared documents used to render the reference materials](#) for all three CCEs.

The FluCoMa materials provided natively in the CCE is often the learner’s first engagement with our supporting materials. Therefore, the information provided is intended to provide the learner/user a working understanding of what an object does, how it might be used musically through a sound-making example, and if appropriate, how it interfaces with other FluCoMa objects. This information will hopefully provide a learner some motivation for exploring an object and the amount of knowledge necessary to do so. Each resource in the CCE contains a link to the corresponding [web reference](#) if the learner wishes to pursue a deeper understanding of the object.

The example code provided in the CCEs has been created to be as-similar-as is possible and idiomatic across the three coding environments. One goal of this is to enable cross-environment communication and knowledge sharing. Users in one environment may be able to describe or explain an example to users in a different coding environment. It is also possible that this allows referencing help files and example code to a classroom containing a diversity of coding environment users. Lastly, this keeps all three CCEs on an equal status, preventing any potential inference of CCE preference within the FluCoMa Toolkit.

2.2 Web Reference

Every object in FluCoMa (except CCE-specific objects) has a web reference found at learn.flucoma.org. Because the materials that appear natively in the CCEs link to the web, the web references are considered to be a secondary resource. The goal of the web references are to offer more detailed descriptions of how an algorithm is working “under the hood”. This may be useful for satisfying curious learners, or it may build a better intuition of what a particular FluCoMa object might be used for musically. In either case, it should lead to a more informed, and therefore potentially more advanced, use of the object.

Many of the web references have interactive explanations that allow a learner to “use” the algorithm in the browser. We have found that these are very useful for explanations in the course of teaching. For example, when creating a KDTree for the first time during a code-along class, using the [interactive page](#) helps give learners a visual sense of what is happening (especially if the KDTree is about to be used with the `plotter`). The [MFCC reference](#) has an [interactive explanation](#) that invites a learner (or a teacher

as a demonstration) to step through a series of interactions that build intuition about MFCCs.

We imagine the web references to be used as solo-learning resources, in parallel with class assignments, teaching demonstrations, and useful reminders.

2.3 Learn Articles

FluCoMa’s learning website (learn.flucoma.org) also contains many articles about

2.4 Explore Articles

2.4.1 Interviews

2.4.2 Made With

2.5 Discourse

2.6 Inter-connectivity of Resources

As described above, the FluCoMa learning resources are generally tiered to offer learners the degree of detail needed at a given moment to pursue a creative idea. One way in which our tiered approach is executed is cross referencing the different learning resources. The help files in each creative coding environment (CCE) link to their respective web reference

Fuzzy Search on Website

3 “101” Tutorials

3.1 MLP Regressor & Classifier

3.2 2D Sound Browser

4 Common Learning Challenges & Strategies

4.1 New Ways of Using Buffers

Buffer interface

Time series of audio

Time series of not audio

Arrays (the idea that it would be not a time series)

Reasoning for why buffers are used

Specific complaints from users
Strategies for responding to these challenges
Changes in how buffer interfaces work in CCEs evolving

4.2 Advanced Neural Network Tutorials

4.3 Stateful Objects

Stateful objects
Added names to objects (max & pd) (sc already has variable names)

4.4 Threading

4.4.1 Max

4.4.2 SC

4.5 Dealing with Time

4.6 Navigating Human & Machine Assumptions

Know Your Data
Nearest Neighbor Scaling & Distances Stats
Histograms Clustering as an example

4.7 De-Myth-ifying Machine Learning

5 Multi-Day Workshops

6 Suggested Semester-Long Syllabus

7 Relevance to Contemporary Society

8 More Information