

Playful machine learning: An interaction designer's journey into neural networks

Andreas Refsgaard

Abstract

In this article, interaction designer and artist Andreas Refsgaard describes how machine learning has become an integrated part of his practice. While still relying on coding for most of his work, training machine learning algorithms has changed his attitude towards a lot of problems and enabled him to see a whole new range of opportunities for projects.

Through his playful projects Andreas is actively seeking out unconventional connections between inputs and outputs using simple machine learning techniques. Games are controlled by making silly sounds, music is composed by drawing instruments on paper and algorithms are trained to decide what is funny, funky or boring. Recently Andreas has been experimenting with using machine learning to create new texts and images based on large amounts of training data and has created Booksby.ai - an online bookstore which sells science fiction novels generated by an artificial intelligence. Overall, the paper demonstrates how machine learning can be used to create playful interactions, solve problems in human-computer interaction and help us ask important questions about the future of design.

We live in a time where artificial intelligence and machine learning seems to be everywhere. Not a week goes by without stories of how algorithms and robots are going to take our jobs, drive our cars, become our lovers, replace our pets or supersede our doctors. Along with all the buzz comes a lot of misconceptions and myths. When we think of machine learning or artificial intelligence, we think of automation, not creative opportunities. Additionally, artists and designers are often scared to enter the field, imagining that they need computer science PhDs and big server parks just to get started. And that is a shame, since there is lots of low hanging fruit out there for all creatives.

Trained as an interaction designer from Copenhagen Institute of Interaction Design (CIID) I am what you might call a creative technologist working within the field of art and design using code as my main tool. But during the last five years machine learning has become an integrated part of my practice, despite the fact that I have never had any formal education within the field or academic prerequisites beyond my long forgotten high school math.

While I still write code, training machine learning algorithms has somewhat changed my attitude towards a lot of problems and enabled me to see a whole new range of opportunities for projects. But what is the real difference between the two approaches?

Coding vs. training

When programming interactive prototypes, interaction designers traditionally rely on their ability to formulate logical structures and explicit relationships between inputs and outputs through code that executes in a predictable way.

Machine learning suggests a different kind of logic. Instead of relying on explicit sets of rules to determine a system's behaviour, machine learning models learn by example, by looking for patterns within a set of examples or training data from a designer or performer, and makes the rules autonomously so as to conform to the performer's expectation. This pattern recognition process is somewhat similar to our own mental processes for learning about the world around us and provides a lot of new opportunities for interaction

designers, especially when dealing with input data too complex to account for via coding. To put some of these thoughts into context I will present a few projects built using fairly simple and accessible machine learning techniques.

Wolfenstein 3D controlled by sound

The first project is a hack of the old first-person shooter video game Wolfenstein 3D made with Lasse Korsgaard for our former studio Støj. Instead of relying on mouse and keyboard controls to play the game, we trained an algorithm to recognize and distinguish between a set of sounds and mapped each one to a different action within the game: Whistling moves the character forward, clapping opens doors, two different grunts turn you left and right and as for shooting? "Pew pew", of course!



Fig. 1. Andreas Refsgaard & Lasse Korsgaard, Wolfenstein 3D controlled by sound. 2017.

The training in itself took no more than a few minutes, and essentially just required us to record a dozen examples for each of the different sounds. Although in no way perfect, this quick and dirty approach proved accurate enough to control the game.

We initially did this project to have something fun to show to a client interested in interactive machine learning. At first glance, it may seem a bit ridiculous, and the fact that it is not exactly easier to play games in this manner goes without saying. In an

article from gaming website Kotaku covering our project, reporter Logan Booker came to the conclusion that he did not 'see this method of game-playing catching on, outside of some specific circumstances' (Booker, 2017).

Training alternative controls for a game or another interactive system, no matter how ludicrous they may seem, does however hint towards bigger goals and opportunities for interaction design and interactive art experiences. By enabling people to decide upon and train their own unique controls for a system, the creative power shifts from the designer of the system to the person interacting with it. In the case of *Wolfenstein* controlled by sound, the game simply becomes something else.

The shifting of power from the designer to the person interacting also points towards more inclusive systems, where users are free to change the systems to fit their specific preferred modes of interaction. Back in 2015 I made *Eye Conductor*, a system that helps people express themselves through music using only their eyes and facial gestures, as my final project at Copenhagen Institute of Interaction Design. Using a \$99 eye tracker and a regular webcam, *Eye Conductor* detects the gaze and selected facial movements, thereby enabling people to play any instrument, build beats, sequence melodies or trigger musical effects.

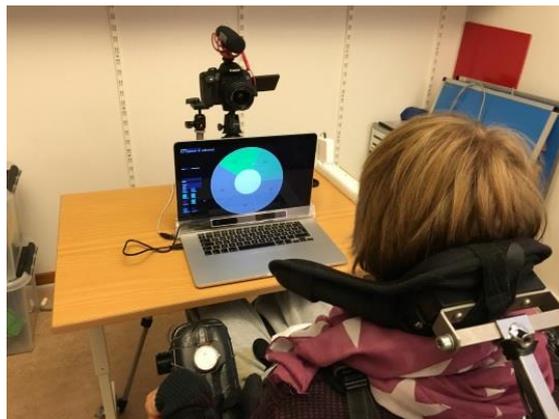


Fig. 2. Andreas Reefsgard, Eye Conductor. 2015.

The motivation for the project came from the realization that for a lot of people with physical disabilities the lack of fine motor skills excludes them from playing music using

traditional hand-held instruments. I therefore designed around eye and facial movements, assuming that all potential users would have full motor control of their facial muscles. Realizing how this was not always the case, I started wondering how a system could be flexible enough to have its core interactions be defined by the specific person using it. Instead of the designer (in this case me) deciding that raising one's eyebrows to a certain level would toggle some specific function, what if people could use unique facial expressions, which they felt comfortable performing and map those to functions inside the program? This spurred my interest in machine learning.

Doodle Tunes

My first steps into machine learning started with a short course at School of Machines, Making & Make-Believe in Berlin taught by independent artist and programmer Gene Kogan and Dr. Rebecca Fiebrink from Goldsmiths, University of London. The course introduced a range of techniques for using machine learning in a creative way and I quickly started building small projects using resources from Kogan's ml4a project (Machine Learning for Artists) as well as Fiebrink's Wekinator software (both highly recommended starting points for creatives to get started with machine learning). Among the projects I made during the course was an algorithm trained on canned laughter trying to determine how funny scenes in the sitcom Friends were. Another project was an audio visual installation trained to distinguish 'funky images' from 'boring ones' after having been trained on 15000 images of highly subjective boring (carpets, lawyers, empty offices, etc.) and funky (party, graffiti, drinks etc.) images.

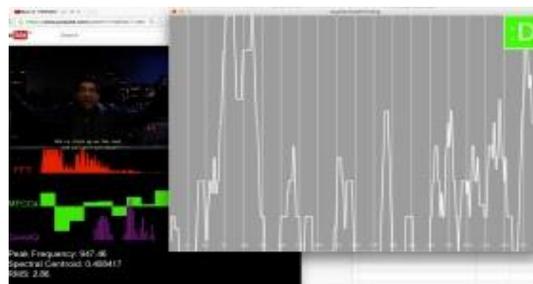


Fig. 3. Andreas Refsgaard, an algorithm watching Friends. 2016.



Fig. 4. Andreas Refsgaard, Is it funky? 2016.

Gene Kogan later invited me to participate in the Nabi AI Hackathon in Seoul, South Korea, where we made the project Doodle Tunes together.



Fig. 5. Gene Kogan & Andreas Refsgaard, Doodle Tunes. 2016.

Built in two days, Doodle Tunes is a project that lets you turn drawings of musical instruments into actual music. After drawing one or more instruments on a regular piece of paper and positioning the drawing beneath a camera, the system begins playing electronic music with any detected instruments.

Similar to the sound controlled Wolfenstein project, where new sounds are compared to the sounds made during our training phase, Doodle Tunes works by classifying new hand drawn

instruments on the basis of a model trained on other hand drawn instruments.

From an interaction design perspective both projects excel in their novel combination of inputs and outputs and ability to expand people's conception about the scope of digital interactions, notably by combining domains that are not usually related.

One of my favorite pieces of art, 'The Gift' by Man Ray has always inspired me, because its logic works in a similar way. Consisting of an iron with fourteen thumb tacks glued to its sole, The Gift is a conjunction of two seemingly alien objects and an example of how juxtaposing or combining objects from two different domains can create something new and thought provoking - in this case a paradoxical tool for ripping cloth or perhaps piercing walls.



Fig. 6. Man Ray, The Gift. 1921.

A lot of my personal work follows these lines of thought, actively seeking out unconventional mappings between inputs and outputs. What if your gaze works as an input for playing music (Eye Conductor), what if your drawings do (Doodle Tunes) or what if sounds could control video games (Wolfenstein 3D controlled by sound)?

An algorithm watching a movie trailer

An algorithm watching a movie trailer was made in collaboration with Lasse Korsgaard. We were playing around with the real-time object detection system YOLO-2 and were curious to explore how a fast paced movie trailer might look when seen through this lens.

Object detection is the process of identifying specific objects such as persons, cars and chairs in digital images or video. For most humans this task requires little effort, regardless of how the objects may vary in size, scale and rotation, or whether they are partially obstructed from view. For a long time these tasks have been difficult for computers to solve, but recent developments have shown impressive improvements in accuracy and speed, even while detecting multiple objects in the same image.

After experimenting with modern classics like *Titanic* (1997) and *The Lord of the Rings* (2001), we chose the trailer for Martin Scorsese's *The Wolf of Wall Street* (2013) because of its fast-paced cuts between scenes in relatively everyday environments like restaurants, offices, boats and houses filled with tables, glasses, computers, cars, and other objects suitable for image detection. As German artist Hito Steyerl (2014) once stated: '*The unforeseen has a hard time happening because it is not yet in the database*', and logically, the algorithm was better at detecting businessmen in suits drinking wine than the Uruk-hai and Nazgul from Tolkien's Middle-earth universe.

Instead of simply outputting the found objects on top of the original video, we made three different versions of the trailer, each using a different filter. The first video filter uses object masking, so only objects recognized by the software appear. The second version blurs all detected objects, thus acting like an automatic censoring algorithm. The final version removes the visuals entirely, essentially creating a filter of what the software 'sees' during analysis.



Fig. 7. Andreas Refsgaard & Lasse Korsgaard, An algorithm watching a movie trailer. 2017.

The project got a lot of media attention and in an article covering the project, Sidney Fussell from Gizmodo described the project in this way:

If a movie trailer distills a 2-hour film into its 3-minute essentials, what would it look like to distill a movie trailer? Strangely, it would look a lot like object recognition software. In three separate videos, we essentially see how algorithms watch movies: They label the essentials—a tie, a wine glass, a chair—but leave the specifics out. It's like visual ad-libs. (2017)

The ability to detect objects and determine their bounding boxes opens up a whole new potential for remixing and personalizing video content. Lasse and I also did a low fidelity exploration in this direction by letting visitors on a website scrub between the original version of the intro for American sitcom *Full House* or a version where all detected objects were replaced by emojis.



Fig. 8. Andreas Refsgaard & Lasse Korsgaard, Full House Emojified. 2017.

Desirable or not, prototypes like this investigate a future where software could automatically censor parts of movies based on the age and sensitivity of the viewer, or perhaps replace actors that viewers dislike with their favorite ones.

Future perspectives

Training machine learning algorithms has changed my attitude towards a lot of problems and enabled me to see a whole new range of opportunities for projects. Besides expanding our possible input-output combinations, I see big potential in making people active participants rather than passive consumers of whatever experiences and tools we build. I am interested in designing for surprises, both for the people interacting with my systems and myself, and want to create interactions that give people a pronounced sense of agency. When a tool or a piece is interactive, you do not know in advance how people will use it, which makes the experiences all the more open-ended. And with the type of machine learning I am using, people can potentially train the input methods themselves, making the interactions even more unpredictable and less determined by us, the designers.

A project that really encapsulates these thoughts is Objectifier by Bjørn Karmann, built in 2016 as his thesis project from Copenhagen Institute of Interaction Design. Objectifier empowers people to train objects in their daily environment to respond to their unique behaviors. It contains a camera, a computer, and a relay, and it lets you turn any electrical device plugged into it on or off, based on what it sees. This allows people to train the system to turn on the radio by doing a certain pose or turn off a lamp, when nobody is sitting at a desk. This provides an experience of training an artificial intelligence; a shift from being a passive consumer to becoming an active, playful director of domestic technology.



Fig. 9. Bjørn Karmann, Objectifier. 2016.

From classification to generation

Over the last few years, as I have personally moved slightly away from interaction design and more towards the digital art scene, my use of techniques has also changed. Instead of using machine learning techniques to make digital systems interpret the world by classifying inputs, I am leveraging machine learning to generate artworks by itself, most often in the form of text or images.

To unfold this change, I will highlight three text-based projects of mine: *Poems About Things* (2018), *BooksByAi* (2018) and *fAIry tales* (2019).

Poems About Things



Fig. 10. Andreas Refsgaard, Poems About Things. 2018.

Poems about things is a project that generates poetry from everyday objects around us. It consists of a mobile website that constructs quirky sentences about an object detected through the user's camera feed. As the user's camera focuses on an object, a built in machine learning model gives its best guess as to what object it is seeing. Based on this a short query is sent to Google's Suggest API, which in return sends back a list of sentences inspired by the detected object. Put together, these sentences form a poem which appears on the screen overlaying the image of the object. Hence the poem consists of a handful of sentences expressing thoughts, questions or comments related to the immediate object as well as the bigger world outside it. And sometimes the mistakes in the classification can lead to quite poetic outputs, as in the case of the hairy chest leading to questions about the realness of a fur coat.

Booksby.ai

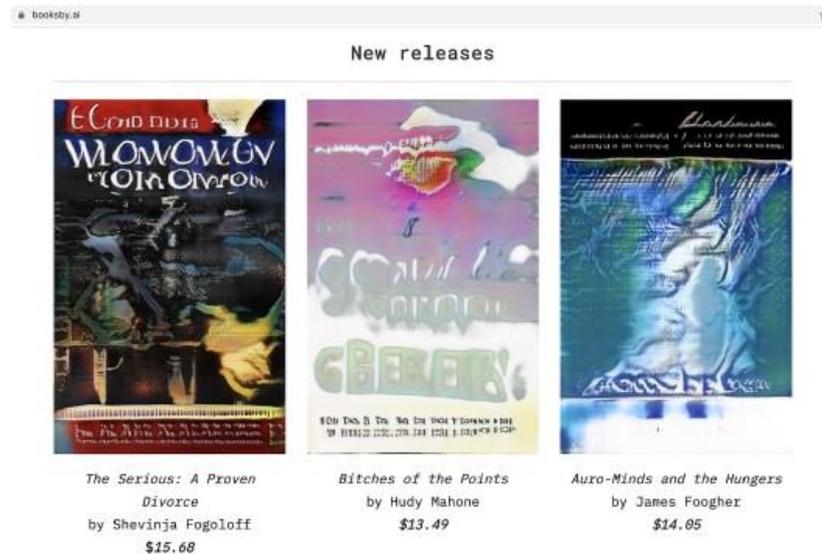


Fig. 11. Andreas Refsgaard & Mikkel Thybo Loose, Booksby.ai. 2018.

Booksby.ai is an online bookstore which sells science fiction novels written – or more correctly put generated – by an artificial intelligence. I made the project in collaboration with data scientist Mikkel Thybo Loose. Through training, we exposed the artificial intelligence to a large number of science fiction books. Hence we so to speak 'taught' the artificial intelligence to generate new science fiction books that mimic the language, style and visual appearance of the books it had read. All books on BooksBy.ai are for sale on Amazon.com and can be ordered as printed paperbacks. Judged by the text quality Booksby.ai is admittedly not the greatest example of machine-driven creativity, but the site presents an intriguing example of an entire ecosystem created by artificial intelligence. By exploring the idea of fully machine learning generated cultural products, the project invites readers to speculate about a future where human beings are no longer the sole producers of creative cultural products.

fAIry tales

Fig. 12. Andreas Refsgaard, *fAIry tales*. 2019.

fAIry tales is a project where fairy tales are automatically generated based on algorithmically detected objects in mundane images. Using images from cocodataset.org everyday objects are detected using the YOLOv3 object detection algorithm. From these objects the title and the beginning of a story is generated and used as prompts for text generation model XLNet.

Much like *Poems About Things* and *Booksby.ai* the project is an exploration of computer written fiction. As in *Poems about things* the texts produced are unique to the pictures fed into the system, but instead of single line poetry outputs the fairy tale texts are longer and form more coherent stories. Where *BooksBy.Ai* focused on creating science fiction texts, *fAIry tales* aims to generate fairy tales, and succeeds in producing more coherent plot lines and sentences than *Booksby.ai*, although surprising and absurd narratives often arise due to the attempt to create adventures about boring objects.

Closing thoughts

Artificial intelligence and machine learning hold a huge potential for creatives, and we have just started seeing the possibilities which the technology will make available to us in the future. We – coders, developers, artists, designers and people in the many other fields using machine learning – still have lots to learn and many black boxes to grasp. But by getting hands on with projects and creating the training data ourselves we get a useful feel for the creative opportunities and inevitable biases of this intriguing and powerful technology spearheading us into the future.

References

Booker, L. (2017) “Wolfenstein 3D Played With Only Claps, Whistles and Grunts”, *Kotaku*.

<https://www.kotaku.com.au/2017/03/wolfenstein-3d-played-with-only-claps-whistles-and-grunts/>

Fussel, S. (2017) “Watching a Computer Identify Objects in a Movie Trailers is Entertaining as Hell”, *Gizmodo*.

<http://sploid.gizmodo.com/watching-an-algorithm-try-to-decode-a-movie-trailer-is-1791288059>

Steyerl, H. (2014) “Politics of post-representation”, *DIS Magazine*. <http://dismagazine.com/disillusioned-2/62143/hito-steyerl-politics-of-post-representation/>