IMPLEMENTATION OF A NEURAL NETWORK TO INVESTIGATE CREATIVE CAPABILITIES OF ARTIFICIAL INTELLIGENCE

Mark Ryan

Final-Year Project – BA Digital Humanities and Information Technology

Supervisor - Dr. James Doherty
Digital Humanities
University College Cork
April 2021

Abstract:

This research topic may be abstracted to the question “Can machines think?”. However, this question is regarded by some as reductive, or at the very least simplistic, as the semantic term “think” really can only apply to human beings who have associated emotional states and conscious experiences. Rather, a better question may be: “are machines capable of assuming cognitive traits and capabilities normally associated only with people?” Examples of such capabilities include intuition, conversation, and empathy. This paper will approach this question from the perspective of creativity – can modern AI algorithms imitate a human’s ability to create?

This project will construct and implement a recurrent neural network in Python that has been designed to write musical melodies and evaluate the output. It will also investigate the philosophical implications of an artificial brain that has the attributes of a human mind.

Declaration:

Declaration of Originality

In signing this declaration, you are conforming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award. I hereby declare that:

• this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;

• with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;

• with respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Signed: . . Mark Ryan. . . . .

Date: . . . . .30/04/2021 . . . .

Introduction:

What differentiates human intelligence from Artificial Intelligence? Software can write sonatas and chatbots seem to know what we are thinking before we do – is the rise of the machines at hand? Sophisticated artificial neural networks, modelled on the human brain, can convincingly talk to us, learn, and create artwork that is comparable to the output of humans. This project will construct and implement a neural network that has been designed to write musical melodies and investigate the philosophical implications of an artificial brain that has the attributes of a human mind. We say that machines have intelligence, but we do not equate this intelligence with thinking. This paper will consider other differentiators between humans and machines - in particular, the ability to create, originate and build. I will also address some contentious debates in the area of artificial intelligence, cognitive science and the theories of the mind.

Analysis

Objectives

There are inherent difficulties in attempting to define an abstract idea like ‘intelligence’, but for the purposes of this report it is useful to attempt a working definition of what we mean by Artificial Intelligence (AI). Minsky’s Maxim suggests that AI is the science of getting machines to do things that would require intelligence if performed by humans (1). This is an incredibly broad church, ranging from logic to counting and artistic expression. There are many “things that would require intelligence if performed by humans” that machines are able to do very well without us ascribing notions of cognitive ability to them; as such, this definition requires further refinement.

When evaluating AI in respect to human intelligence we can divide it into two distinct spheres – ‘weak’ AI and ‘strong’ AI. This project aims to illustrate one of the key differentiating factors between ‘weak’ AI and ‘strong’ AI. ‘Weak’ AI is used for narrow, specific, focused applications and examples are seen in everyday life. It is what is used in movie recommendations on streaming services and the map software on your phone. It is also used in advanced modelling algorithms and sophisticated machine learning platforms. If we understand ‘Artificial Intelligence’ as the ability of software to replicate the activity of the human mind, then ‘weak’ AI is any such activity that requires a dedicated focus, like finding the quickest route through traffic. ‘Strong’ AI, on the other hand, or Artificial **General** Intelligence (AGI) is a form of AI so comprehensive that it is indistinguishable (or even interchangeable with) a human mind. Such an entity would not require the dedicated instructions of software engineers or coders to know how to behave; as a *universal* computer it would already possess all the necessary faculties to learn without observation, direction, or reinforcement (2).

‘Strong’ AI, then, remains firmly in the realm of science-fiction along with the replicants from *Blade Runner* and the droids from *Star Wars.* Even the most advanced deep learning projects that exist today, which can make incredibly accurate predictions based on the evaluation of enormous data sets would be completely bewildered by a tiny change in the query parameters. In *Star Trek* an intelligent robot posing as a human is terminated by the crew of the ship by positing to him a question involving a logical paradox. The most advanced AI operating currently would be stumped by the slightest deviation from its training parameters. It cannot infer from incomplete information or generalisations as humans can (3).

It is fair to say that the popular conception of AI as instances of cognitively aware thinking machines (who may or may not dream of electric sheep) is far removed from the contemporary reality. AI is better thought of today as a collection of enormously useful analytical tools that can be used to predict anything from weather patterns to consumer marketing preferences. The challenge in identifying which AI we are taking about can be helped by unpacking the semantic idea of ‘thinking’ that is closely enmeshed with ‘intelligence’. It is no longer sufficient to say that ‘intelligence’ means a task or activity that would require a human to perform; algorithms can perform lots of these functions and we do not consider an algorithm to be intelligent. In this regard technology has overtaken the vocabulary we use to explain it. We should instead consider elements of the human psyche that cannot easily be formalised in code: empathy, intuition, emotional states, opinions, and intentions (4).

For this project I investigated the potential for computational creativity in a relatively simple recurrent neural network. Creativity is amongst the least understood of human capabilities and yet is ultimately responsible for virtually the entirety of our civilisation. Without creativity no human would have ever picked up a stone tool or planted a seed in the ground (2). I consider this ability that humans possess; to use conjecture and criticism to arrive at previously unexplored conclusions, to be one of the key differentiators between intelligent and non-intelligent beings.

While no example of ‘strong’ AI exists today, many scientists and philosophers believe that it is a logical inevitability. It would certainly appear that no physical or scientific barrier exists to prevent the construction of an artificial brain that is functionally identical to the human mind. It therefore follows that such a development is an eventual certainty. John Haugeland writes: *“"AI wants only the genuine article: machines with minds, in the full and literal sense. This is not science fiction, but real science, based on a theoretical conception as deep as it is daring: namely, we are, at root, computers ourselves."* (5) The purpose of this project is to assess one of the key elements that contribute to the universality of the human thinking; one of the factors that make our intelligence ‘general’.

Review of Development of Thought in AI

The idea of an artificial mind has been of great interest to philosophers for centuries. Descartes considered the question in relation to his idea of cartesian dualism; the philosophical notion that mental states can exist outside of physical matter and therefore outside the body. He concluded that even the most sophisticated 'automatons' (being the most advanced machinery available at the time) could never compete intellectually with even the "dullest people" without being able to communicate as people can (6). He emphasises language as a key difference between a piece of machinery created by humans and a human mind that exists outside of the physical domain (and created by God).

Thomas Hobbes, meanwhile, maintained that 'thinking' is not at all unlike a mechanical process and is at its most clear when following a defined logical path (*"why not we say that all automata have an artificial life?"* he asks in his introduction to *Leviathan* (1651)). Thereafter the historical literature reflects a growing school of thought among scientists and philosophers regarding the relationship between human psychology and mathematics – the idea that human thought is a logical process that can be explained (and recreated) mathematically.

This idea is articulated in George Boole's *Laws of Thought* (1854), his hugely influential exposition of logical systems that laid much of the groundwork for modern computer science. He introduces his treatise by asserting that the scientific method of logic (which he seeks to construct) is an interpretation of the fundamental process of the mind, as expressed through the symbolic language of calculus. Here he is saying that the logical processes on which modern computer science is based are recreations of the thought processes in humans.

The hypothesis of this project is based on this key concept: if the logic of computer systems is fundamentally based on the logic of human thinking, then any human process can be recreated digitally. Closely linked to this idea is the concept of 'universality'. A universal system allows new novelties to emerge from within a set framework; for example, a written alphabet that replaced early pictographic representations (2). Pictograms were developed to represent specific words and specific concepts, but they could not be used to write new words or illustrate new concepts without the prior agreement of the people using them. When alphabets started appearing, by contrast, they were capable of representing every word (and every possible word) in that language.

An alphabet encodes a set of rules that, when understood, allows somebody to read words that they have never seen before. It is a *universal* system that relies on an overarching framework rather than a specific, prescribed representations. Human language is another universal system where we can easily express ideas in ways that other people can understand them. I can articulate a sentence that has never been uttered before, but as long the sentence makes grammatical and syntactical sense I will be understood.

The human mind is a clear example of a universal system. Another clear example is a modern computer (perhaps due directly to the logical ink between them). Even the earliest programmers recognised this potential: when Charles Babbage (possibly the grandfather of the computer) developed his ideas for an Analytical Engine in 1837, his colleague Ada Lovelace wrote that far from performing the role of a simple calculator, this programmable, steam-powered general-purpose computing machine that incorporated integrated memory and arithmetic logic could be applied to virtually any subject in the universe (9).

Tests For AI

While Babbage's universal machine was never built, advances in technology eventually lead to working examples of universal computers based on Boolean logic in the 20th century. Alan Turing was another who developed a hypothetical universal computer (10) and recognised the potential of the machine. He later engaged directly with the question of whether universal computers could *think* in his seminal paper (11) where he proposed the Turing Test, a 'game' designed to investigate whether a human judge could be fooled into believing they had been conversing with a human instead of a machine. Like Descartes, Turing believed that communication was a significant element in determining the presence of intelligence or 'thought'.

It should be noted that Turing regarded the question of whether computers could think as 'meaningless' and did not attempt to provide a definition of thinking. A pass or fail should not, therefore, be considered definitive evidence of the presence or absence of thought. Instead, it is a potentially good source of inductive evidence that machines are cognitively capable. The test has been criticised variously for being too easy (some rudimentary natural language processing applications can fool people) and too conservative (the test could not identify intelligence in small children or intelligent animals like chimps and dolphins). One of the most prominent critiques of the Turing Test (and of the central concept of strong AI in general) comes from John Searle and his Chinese Room thought experiment (12).

This thought experiment suggests the following premise: an 'intelligent' machine behaves as if it understands Chinese. A Chinese speaker communicates with the machine through input and output devices and is convinced that the machine displays an understanding of the conversation. This machine has passed the Turing Test. What the interviewer does not know is that inside the machine there is a man who is following a set of written instructions, identical to a computer program, that direct him to returning correct answers to the interviewer. The man does not speak Chinese but has successfully fooled an interviewer into believing that a genuine conversation has taken place. Searle argues that because there is no understanding on the part of the man in the Chinese Room then what has been observed cannot be regarded as ‘thinking’.

Searle’s paper has received an enormous amount of criticism since its publication on a wide variety of grounds, including one predicted by Turing the ‘other minds’ reply. The problem of other minds relates to the issue that a person can never know what is happening inside the mind of another human. We assume that other people have mental states and phenomenological experiences, but we have no way of verifying that this is true. In this regard, Turing would respond that the Chinese Room experiment is holding the AI to a higher standard than we would a human.

I believe an even more convincing argument against Searle’s findings is the Connectionist response. Searle asserts that because the man in the room has no understanding of Chinese, then no understanding has taken place. I would argue that the man in the room is merely one small part of a much larger system that *does* understand Chinese. The Connectionist response is related to the development of parallel distributed processing and artificial neural networks (ANN). When computer scientists or data engineers reference AI they are most likely referring to a connectionist model that is capable of machine learning. I will discuss connectionist models and ANNs further in later sections.

The Turing Test is by no means a perfect model for testing for intelligence, although, as I have said, it may provide good inductive evidence for it. Perhaps a more interesting test was proposed by mathematician Marcus du Sautoy who describes a test for computer creativity (13). In the ‘Lovelace Test’ (named for Babbage’s colleague Ada) a piece of software could be said to be intelligent if it generates something new, interesting, valuable, original, or surprising. Moreover, the coder of the algorithm should not be able to explain where the novelty originated. I will look more closely at the idea of creativity in the qualitative literature review.

Project Goals

The primary technical goal of this project was to implement a neural network that can be trained on a selection of music and produce an output that is distinct enough from the training set that it may be called ‘original’. An AGI in the future will be able to express their creativity by developing original ideas, thoughts, and opinions but in the age of ‘weak’ AI it is important to pick a narrow application and stick with it.

Music was the focus of investigation for this project for the following reasons: Firstly, it is a classic example of artistic creativity. Artistic expression is closely identified with higher human thought and consciousness, in the popular imagination at least. Secondly, music is constitutionally mathematical in form and lends itself naturally to formalisation in code. Finally, the rules of musical theory and syntax are fundamental and essential. This makes the high-level evaluation of any output created by the network much more straightforward. A cursory aural examination of the original melody will tell the listener if the chords are being formulated incorrectly, for example.

Deliverables

The deliverables for this project included:

- A fully functioning neural network that can parse notes from MIDI files
- A series of outputs or predictions that will be converted back into MIDI so they can be listened to. These outputs will then undergo human evaluation to determine whether they are novel, interesting, or valuable (this is a necessarily a subjective inference).

Requirements Analysis

Functional Requirements: This project required a reasonably advanced level of machine learning techniques to functionally perform the task that was required.

Technical Requirements: ANNs require substantial compute power to run and are often unpredictable, temperamental, and inconsistent. The network required a significant amount of time to run through enough epochs (iterations) to produce a significant output.

Transitional Requirements: I was required to learn about machine learning and recurrent neural networks effectively from scratch. I had some experience with Python before beginning this project but no experience with machine learning libraries or neural nets on a practical level.

Existing Work

To test the creative capabilities of AI it was important to assess what other examples of this investigation existed. There is a plethora of interesting implementations of this idea from academic research to commercial applications and artistic exercises. I will explore some of these in the qualitative literature review.

Specifications, Technical Decisions & Concepts

Details of the above will be covered in detail in the **Design** section.

Qualitative Literature review:

In this section I will survey other implementations of this idea and the strengths and weaknesses of their approaches. I will focus on more recent examples where possible - due to the speed at which technology progresses older solutions could appear out of date.

AIVA (Artificial Intelligence Virtual Artist) is a commercial program that composes music for applications like video games, film soundtracks and advertising (14). It uses proprietary algorithms to construct remarkably complete soundscapes in a huge variety of styles. Visitors to the AIVA website can compose their own original piece by selecting a pre-set style or uploading their own MIDI file to use as inspiration. If the results are good they can upgrade to a PRO plan to formally license the music.

To commemorate J.S. Bach's 334th birthday a team at Google installed a Doodle on the Google homepage that invited visitors to compose their own melody and have it harmonised in the style of Bach (15). Users could rate and share their compositions and Coconet, the machine learning model, received over 55 million queries in three days. This was a large-scale deployment that had significant financial and resource backing. It served as an education tool as it guided users through the concept of musical harmony. One of the main goals of this project was to encourage people to challenge themselves creatively by using digital tools to augment their creative process.

This project was designed to be used by people with little or no technical skills or knowledge of musical theory and users interacted with it through the Google homepage. Meanwhile, OpenAI's MuseNet application is capable of generating 4-minute long compositions with up to 10 different instruments and combing a wide variety of styles (16). It too was trained using examples of real music and allowed to discover musical patterns on its own. MuseNet shares a platform with the GPT-2 general-purpose learning algorithm used for text output. It uses advanced large-scale transformer models to make predictions and, like AIVA, allows users to pick their own starting points by uploading MIDI files.

The Jukebox application, also by OpenAI, is even more sophisticated, moving away from mere symbolically represented notes on a piano roll to modelling music directly as raw audio (17). This approach *"pushes the boundaries of generative models"* in the words of the authors, and needs to deal with an enormous amount of raw data (17). These examples, while undoubtedly impressive, were far too ambitious for my purposes. The OpenAI applications are extraordinarily advanced exercises in what is currently possible with neural networks. Sturm et al. (2016) describe in detail their implementation of long short-term memory (LSTM) networks trained on approximately 23,000 musical transcriptions that were grouped into genres. Here they used the more conventional and simpler symbolic process in which the sound is abstracted away and the representation of the notes are used to train the network (19). The aim of the Sturm project was to create musical outputs to assist musical composition and so the training material reflected conventional stylistic patterns of particular genres (Irish folk, etc.).

Sturm et al. (2016) analyse a large number of other music modelling and generation that utilise Recurrent Neural Networks and LSTM. They develop a number of different networks themselves that share similar architecture but operate over different vocabularies and are trained differently. Each LSTM network has three hidden layers with 512 LSTM blocks each. Sigurður Skúli took a similar approach with his LSTM network that he trained with video game music (20). These latter approaches in particular formed the basis for my own implementation.

In the next section I will assess literature that aims to engage with arguments and debates in the topic of computational creativity and what these arguments contribute to understanding of the subject:

Margaret Boden is one of the world's eminent scholars in the area of computational creativity and her paper (2009) collects and critiques many of the arguments in the area. She asserts that not only do examples of computational creativity already exist, but they can also help us to understand the ontology of human creativity. She stresses that creativity is not a rarefied gift from the gods; *"rather, it's a feature of human intelligence in general"* (21).

She differentiates between three different types of creativity and also different ways of being creative. Creativity generates something novel, but it may be historically novel (completely new in human history) or novel merely to the individual creator. If I come up with a clever joke, for example, I'm exercising my creativity, even though hundreds of other people may have arrived at the same joke before me. Examples of historically novel instances of creativity are of course extremely rare and these have the potential change paradigms and open up completely new possibilities in particular areas. For example, the development of synthesisers in the 1960s launched an entirely new model for musical experimentation with electronic music.

Boden also specifies the different ways in which creativity happens: *"Novel ideas may be produced by combination, by exploration, or by transformation"* (21). 'Combinational' creativity is often what is meant when we think of the creative act of a person; for example, a poet combining different meanings of words to come up with an elegant metaphor. ‘Exploratory’ creativity is performed within a framework or “conceptual space” (21), much like the notes in a musical scale or the grammatical rules of a language that allow us to create sentences and communicate.

The third type, ‘transformational’ creativity, is more unusual and interesting. The ideas generated by this kind of thinking can be counterintuitive or even frightening, but as they change or remove certain defining dimensions of thought the results can be truly remarkable. Interestingly Boden does not think that it is clear which of these types of creativity AI is best suited to. She thinks that an AI system will work most effectively within an ‘exploratory’ system with clear, defined rules and style but suggests that most people would assume that AI is better suited to combinational creativity. A system could indeed achieve this by using huge computational power to force ideas together, she acknowledges, but the vast majority of the output of such an approach would be gibberish.

Boden suggests that an AI using a combinational approach cannot compete with the huge accumulation of information that the human mind can intuitively curate and utilise sensibly. Any computational approach to generate creativity artificially in this way would require ‘cheating’ by building a directed system that was, for example, looking for synonyms for particular words to use in a poem.

This argument may be true of contemporary systems, but from a conceptual standpoint I do not believe that an AI in the future would be unable to extrapolate meaning from artistic or logical concepts. Humans can certainly intuit meaning from such concepts without any obvious effort, but I do not think that there is a logical impediment on machines being programmed to do the same in the future.

Boden does emphasise that there is nothing magical or spiritual about creativity and provides several examples of instances of computer creativity that she claims rival human creativity (21). Noam Chomsky, a linguist, devotes attention in his work *Cartesian Linguistics* (22) to creativity in linguistic terms, focusing on what Boden would call 'exploratory' creativity. 'Cartesian Linguistics' is his term for the universal rules of grammatical structure that are common to all languages and are based on the structure of the human mind. This example of a universal system enables the speaker of a language to know "*a great deal that he has not learned"* (22) and communicate his thoughts and ideas effectively.

With Chomsky we see again divergence between historically novel instances of creativity and instance that are novel merely to the creator. "True" (or artistic) creativity should be considered separately from the simple generation of novelties that we see in language or everyday ideas. This, more rarefied form of creativity must demonstrate *value*.

David Deutsch, a theoretical physicist, strongly emphasises in his work (2) that AGI is not only possible, but inevitable. He devotes significant attention to whether such an entity should be considered as a conscious being and therefore afforded the same rights and protections that humans receive. He considers that, under the principle of the 'universality of computation', all capabilities and mental states that exist in humans (including creativity) can be recreated in machines (2). Of creativity, he writes that it is the precursor to all of human achievement, but we do not understand it. He is very much concerned with the 'exploratory' aspect of creativity, devoting many chapters to examples of universal systems like language, numerology and even the genetic code of all extant life-forms that allow for “reach” (ideas outside what were envisaged by the system). The creation of persistent novelties by AI, Deutsch argues, can be achieved by building the framework of a universal system that facilitates it.

Deutsch does not make any specific allowance for “artistic” or “true” creativity, as Boden and Chomsky do. (Chomsky makes reference to *“appropriateness to context”* and *“unpredictability in terms of environmental circumstances”* and a distinguishing point between the two ideas (23)). Deutsch does not meaningfully interrogate the idea of creativity despite placing much emphasis on it throughout his work. He states that “conjecture” within a universal system is a key feature but does not engage with arguments for creativity by ‘weak’ AI. “Conjecture” in this context refers to the capacity of the brain arrive at conclusions in spite of incomplete information; this is the ‘first mover’ in human creativity - the spark that allows us to have ideas. He maintains that only ‘strong’ AI has the potential for creativity as any form of ‘weak’ AI is not capable of the single, unifying logic of Popperian conjecture and criticism.

Marcus Du Sautoy, a mathematician, writes (13) about creativity from a mathematical perspective, again emphasising the role of exploratory creativity in finding new mathematical proofs. He explores a plethora of examples of AI implementations that have resulted in surprising, novel or valuable results. He investigates Google’s AlphaGo program that managed to defeat the world’s best human player, an achievement that many had considered impossible. He gives particular focus to a “surprising” move that the program made, one that experts claimed no human would have made in the same scenario.

He does acknowledge that anything created by contemporary AI still pales in comparison with novelties created by humans (13). However, I believe that Boden's appeal against the "superhuman human fallacy" (21), where we expect more from AI that we do from ordinary people, is appropriate here.

Anton Oleinik argues in his paper (24) that a widespread focus on statistical regression in neural networks impedes the potential for progress in computational creativity He appeals again to the three main strands of creativity which he *calls "metaphorical thinking, social interaction and going beyond extrapolation in prediction"* (24). He argues that neural networks may be good performers but cannot authentically engage with the rules of their subject matter. He references, like Boden, the aptitude of a human mind for metaphor that a neural network could not hope to pick up on. He gives an example of a line from Austen's *Pride and Prejudice* which ostensibly refers to dancing but is strongly suggestive of a deeper sexual meaning. Thinking metaphorically, he proposes, is beyond the capacity of a network programmed to recognise patterns.

Design

Architecture (components, roles, and relationships between)

A neural network consists of layers of artificial neurons between an input layer and an output layer (18). Neural networks are trained on datasets which they can analyse through numerous hidden layers to identify patterns. They can then make predictions based on these patterns. For my project, the training data set consisted of two separate collections of MIDI songs which were converted into a list of notes and chords via the music21 python toolkit. The output layer was the completed prediction which was subsequently converted back into MIDI format.

The neural network was an appropriate choice for this project for the following reasons: Firstly, a neural network is modelled on the connections of a human brain and is thought to most closely resemble the logic of human thought (25). The structure of a neural net is based on the idea in cognitive science of a 'Connectionist' network, sometimes known as 'parallel distributed processing'. The information processing units in a neural network (nodes) process information simultaneously and the information is spread throughout many different location in the network (25). The nodes within this network communicate with each other, and the network 'learns' by adjusting the strength of these connections (26). Neural networks have been around for decades, but in recent years have been used in conjunction with deep learning techniques to achieve remarkable results.

As I have illustrated above, one of the most prominent arguments against the concept of 'strong' AI centres around the ideas that even though the illusion of understanding may be generated by a network (which appears very clever) no genuine engagement with the subject on the part of any single part of the network can be inferred by an observer (12). Connectionism is a school of thought in cognitive science that posits that something like understanding or perhaps even thought could be said to exist in the connections that are formed between neurons in the brain, or in this case in an ANN.

My intention for this project was to conduct an experiment where I would, as closely as possible, recreate elements of an artificial mind and assess the results that were generated. It made sense, therefore, to use a structure that was as close to the brain as possible. Another consideration was that many data scientists have had extremely impressive results using neural networks to generate music, as demonstrated above (Sturm et al. 2016; Payne 2019 etc). In order to pursue the best possible results for my experiment it made sense to utilize some of these ideas and implement them to suit my needs.

The neural network is structured as follows:

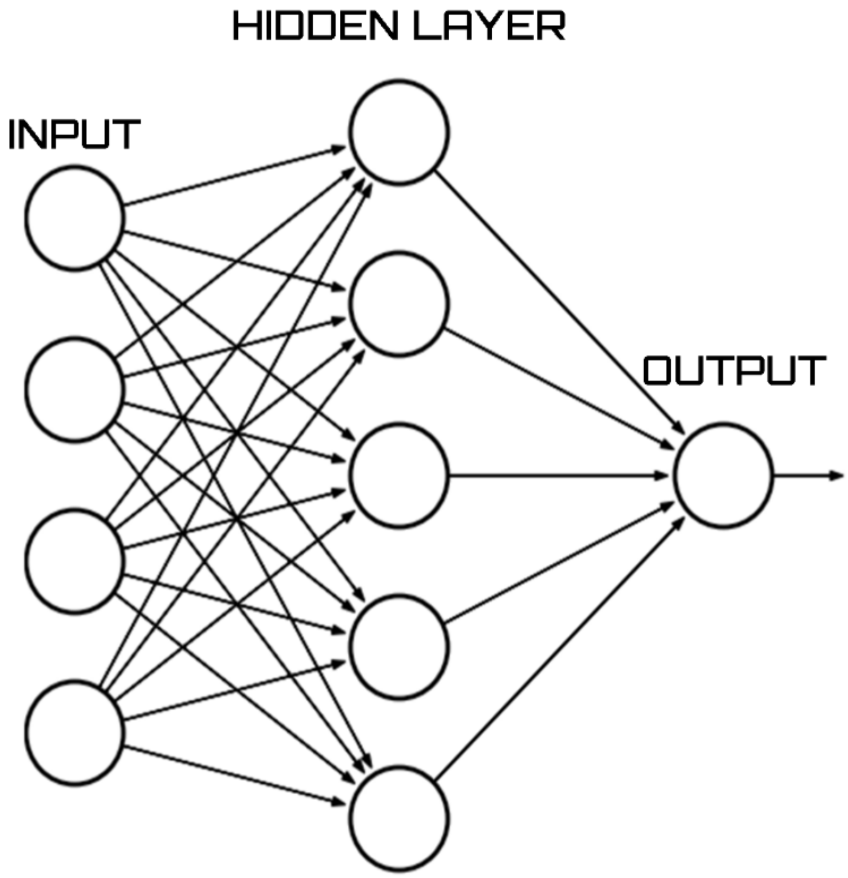

Figure 1: Structure of a neural network

Image Credit:

(27)

The elements of each input are broken down into constituent parts and fed into each neuron of the hidden layer (28). In advanced neural networks there are many hidden layers. The information is passed between neurons in these various layers through channels. Each channel is then assigned a weight value. The inputs are multiplied by the corresponding weights and their sum is sent as input to the neurons in the hidden layer.

The network assigns a 'bias' value to each neuron in the hidden layer and this is added to the input. The 'activation' layer in the network decides whether a neuron will be activated based on the sum of the bias and the weight value. When a neuron is activated, it passes its information forward to the next layer (forward propagation).

The neuron with the highest value determines the output in the output layer. Once the network has been trained it calculates the rate of error at the output layer and uses 'Backpropagation' to feed the results back through the network to compare its prediction with the training data and reduce the error in the output. The network must iterate through a set number of 'epochs' before it can make a reasonably accurate prediction.

In my network the hidden layer identifies patterns in the music training data and uses the resultant weights to make its own musical prediction. This network is a recurrent neural net, meaning that each component of the sequence undergoes the same function. This is particularly important for music as predicting what note or chord comes next in the sequence is vital.

Tools, technologies packages & libraries

Muse Score: is a notation software that allows users to write sheet music and convert it into a variety of playable formats, including MIDI. I used this program to collect community-generated notation for popular music and convert them into MIDI format. These files were used to train the neural net.

Anaconda: is a Python distribution that that simplifies package management and deployment (29). Miniconda is a light version of this software.

Pickle: is a binary serialisation format that converts objects into a byte stream (python.org). In my network this module loaded the music that trained the network and serialised it in byte form. This format was executed by the network.

NumPy: is a Python library that is used for working with arrays and matrices (numpy.org).

Music21: is a Python toolkit for computer-aided musical analysis. It can be used to compose music, analyse and edit musical notation, generate examples and study large datasets. It manipulates symbolic musical data in such way as it can be used by other python modules (web.mit.edu). In this network it creates Note and Chord objects which is necessary to generate MIDI files.

Tensorflow: is an open-source machine learning platform that can develop and train models (tensorflow.org).

Keras: is a high-level API that runs alongside Tensorflow to enable the modelling of neural networks and simplify the stacking of layers.

Implementation

Description of roles of main data structures

Sequential Model: This is one of the simplest types of machine learning model in the Keras library. It allows users to define all the layers within the structure as required (34). It is perfect for this project I am using only one type of input.

Layers:

Dense layer: is the most common deep network layer and consists of the input data, the weight data and the bias value (tutorialspoint.com).

Dropout layer: randomly generates an input value of 0 during network training to prevent overfitting. Overfitting in this case would mean the music predictions would resemble sequences from the training data too closely (Keras.io). This is undesirable as the aim for this project is to generate original music, not recreate sequences that already work musically.

LSTM layer: allows the network to store information. It consists of a set of recurrently connected memory blocks that allow it to capture long-term temporal dependencies (Brownlee 2017).

BatchNormalization: stabilises the network by normalising the inputs to each layer (38)

Activation layer: is used as a hidden layer and as the output layer. Within the hidden layer it uses the 'ReLU' function to return '0' if the input value is negative. In the output layer the activation layer uses the 'softmax' function to output a vector of values that sum to 1 (39).

Categorical cross-entropy: is a Keras function that is used to calculate the loss for each epoch.

RMSProp: is a Keras function used to optimise the network when compiling.

Importing the notes form the MIDI files: music21 provides a 'Note' object which records the **pitch** of a sound, its **octave** (effectively what key is it is) and the **offset** (where it is located in the musical piece). 'Chord' objects are collections of notes that are played together. The prediction array will need to contain every note and chord that exists in the training set. For the sake of simplicity this network ignores the offsets of intervals between notes, so the resulting musical pieces consists of notes with no intervals between them.

Each MIDI file was loaded into a music21 stream. A stream object is a fundamental container of offset-positioned notation and musical elements. It collects a list of all the notes and chords in the MIDI file and encodes them based on their pitch. This is then mapped from string data to integer-based numerical data that is used to train the network.

The next stage was creating the sequences for inputs and outputs. Each input sequence consists of 100 notes or chords, so the network can make a prediction for the next note following 100 notes. The final step used numPy to normalise the input for the LSTM network by converting the integer encoding to binary encoding (one-hot encoding) (Brownlee 2017).

The data was now formatted correctly and could be applied to the model. This sequential model contained four different types of layers: LSTM layer, dropout layer, dense layer, and the activation layer. In the first layer the shape of the data the network will be training was defined. Each layer must have a defined number of nodes where appropriate (512 for LSTM and 256 for the dense layer). The dropout layer defines the proportion of inputs that are returned as '0' (in this case .3).

Model checkpoints were used to ensure that the weights are saved after each epoch. A file-path is set to store the weights locally and the number of epochs and batch size was determined (the batch size is the number of examples from the training data set that are used to update the weights, in this case 128) (38).

Once this network was built it was time to train it. I used two distinct sets of training data in order to investigate if the network would generate dramatically different results. The first set was a collection of approximately 25 TV and movie theme songs. The second was a collection of classical and modern music. All the music files were in MIDI form with only one instrument representing the melody of the piece. The network took over 20 hours to complete each training data set to an acceptable degree.

Name	Status	#	Title
Band_of_Brothers_-_Variations_with_an_Interlude.mid			
Bates_Motel_Piano_Cover_Sheet_Music.mid			
Big_Little_Lies_-_Main_theme_Music_Piano_Version.mid			
Crocketts_Theme_-_Miami_Vice.mid			
Doctor_Who_TV_Theme_Tune_-_Piano_Version_Hard.mid			
Fargo_Piano_Cover_Sheet_Music.mid			
Game_of_Thrones_Theme_Music_-_Piano_Tutorial.mid			
Hadouken_Theme__from_Street_Fighter_Victory_TV_.mid			
His_Dark_Materials_Theme.mid			
House_Of_Cards_TV_Theme_Tune_-_Hard_Piano_Cover.mid			
MacGyver_2016_TV_series_-_Opening_Theme.mid			
Sherlock_Piano_Cover_Sheet_Music.mid			
Star_Trek_Voyager_Theme_Piano_Arrangement.mid			
Stranger_Things_Piano_Cover_Sheet_Music.mid			
The_Flash_CW_-_Blake_Neely_-_Main_Theme_-_Piano_Solo.mid			
The_Lonely_Man.mid			
The_Sixth_Sense_TV_Theme_Bells.mid			
Twin_Peaks_Piano_Cover_Sheet_Music.mid			
Westworld_Piano_Cover_Sheet_Music.mid			
X_Files_theme.mid			

Figure 2: Training data set 1

Figure 3: Training data set 2

Once the network had been trained it was possible to re-use code that prepared the data and constructed the network to generate a prediction. The network loads the weights generated by the trained network and sets a random note to generate a starting point for the prediction. It also needed to use a mapping function to convert the prediction from numerical data back into notes and chords.

This network generates 500 notes which corresponds to about 2 minutes of music. This gives the network enough time to noodle around and develop something interesting. Once the piece is in the second minute it is unlikely to break out of its pattern so longer pieces are not very useful. To generate a prediction the network needs a sequence of notes to begin. The first sequence is selected at random and the network appends a predicted note to this sequence using the highest index value. It then repeats the process with the updated sequence of notes. These notes are then collected in an array which can be decoded into Note Chord music21 objects. This is converted into a Stream object and music21 converts the stream to MIDI format, which can be played by many digital music players.

```
    return model

def train(model, network_input, network_output):
    """ train the neural network """
    filepath = "weights-improvement-{epoch:02d}-{loss:.4f}-bigger.hdf5"
    checkpoint = ModelCheckpoint(
        filepath,
        monitor='loss',
        verbose=0,
        save_best_only=True,
        mode='min'
    )
    callbacks_list = [checkpoint]

    model.fit(network_input, network_output, epochs=200, batch_size=128, callbacks=callbacks_list)

if __name__ == '__main__':
    train_network()
```

```
Parsing midi_songs\Band_of_Brothers_-_Variations_with_an_Interlude.mid
Parsing midi_songs\Bates_Motel_Piano_Cover_Sheet_Music.mid
Parsing midi_songs\Big_Little_Lies_-_Main_theme_Music_Piano_Version.mid
Parsing midi_songs\Crocketts_Theme_-_Miami_Vice.mid
Parsing midi_songs\Doctor_Who_TV_Theme_Tune_-_Piano_Version_Hard.mid
Parsing midi_songs\Fargo_Piano_Cover_Sheet_Music.mid
Parsing midi_songs\Game_of_Thrones_Theme_Music_-_Piano_Tutorial.mid
Parsing midi_songs\Hadouken_Theme__from_Street_Fighter_Victory_TV_.mid
Parsing midi_songs\His_Dark_Materials_Theme.mid
Parsing midi_songs\House_Of_Cards_TV_Theme_Tune_-_Hard_Piano_Cover.mid
Parsing midi_songs\MacGyver_2016_TV_series_-_Opening_Theme.mid
Parsing midi_songs\Sherlock_Piano_Cover_Sheet_Music.mid
Parsing midi_songs\Star_Trek_Voyager_Theme_Piano_Arrangement.mid
Parsing midi_songs\Stranger_Things_Piano_Cover_Sheet_Music.mid
Parsing midi_songs\The_Flash_CW_-_Blake_Neely_-_Main_Theme_-_Piano_Solo.mid
Parsing midi_songs\The_Lonely_Man.mid
Parsing midi_songs\The_Sixth_Sense_TV_Theme_Bells.mid
Parsing midi_songs\Twin_Peaks_Piano_Cover_Sheet_Music.mid
Parsing midi_songs\Westworld_Piano_Cover_Sheet_Music.mid
```

Figure 4: The module parses the MIDI files

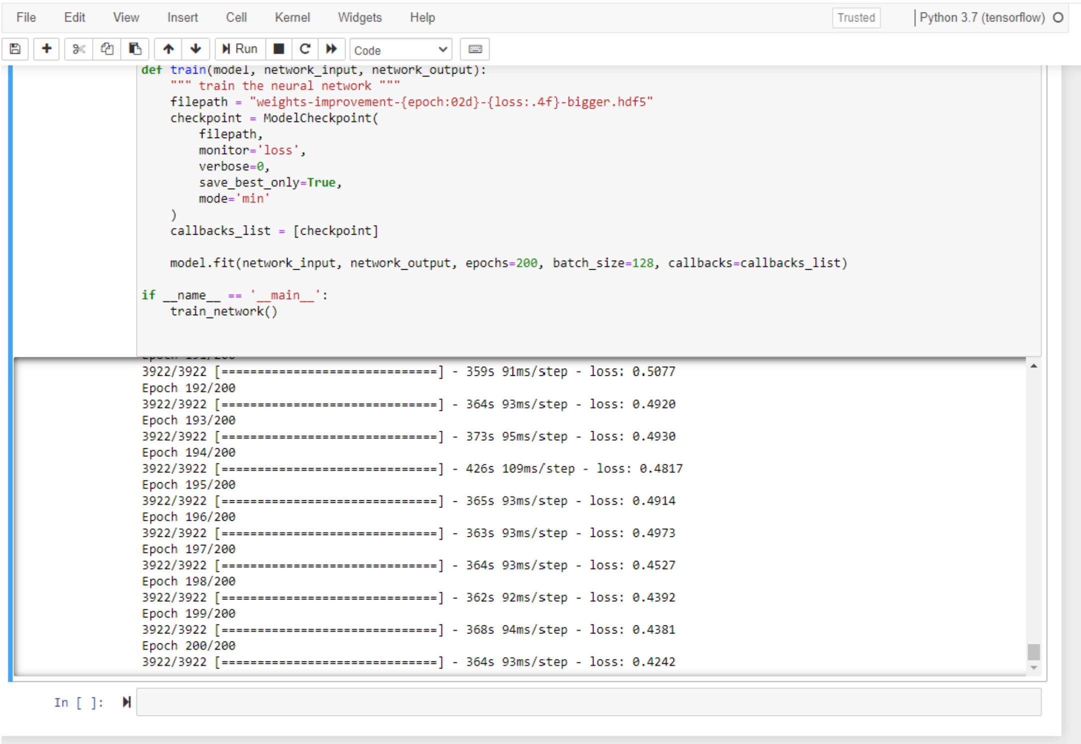

Figure 5: Final epoch complete

Technical challenges

Neural networks require a large collection of libraries, APIs, and tools to work correctly. Many of the tools are quite advanced and involve a steep learning curve. An early attempt at training the network on Bach's Goldberg variations failed when all the predictions remained stuck on the first note. This became a frequent problem which I attributed to the files in the training set being too similar.

The time it took to train the network was a significant challenge. Running on my own Windows machine using a Jupyter Notebook a single epoch took about 30 minutes to complete. 200 epochs were required to train the network to a satisfactory standard which ultimately took 20 hours. Fortunately, my use of model checkpoints meant that the weights from each epoch were saved locally in case the training needed to be interrupted.

I made an attempt to host the network on a cloud service, but an unreliable internet connection made this solution unworkable. The Oracle Cloud Infrastructure hosts a data science platform that include many libraries and tool necessary for machine learning (41). This service was available to me as part of a free trail and would have suited my needs in many regards. Unfortunately, any brief interruption to my internet connection would result in the connection to the Virtual Machine being terminated (the internet connection in my home drops regularly). All progress and weights were saved within the cloud infrastructure, but reconnection had to be performed manually. This made it impossible to train the network overnight, for example, without constant monitoring.

Training a deep learning neural network is notoriously difficult (42). They are unpredictable, temperamental, and possibly capricious. Implementing this one successfully involved stripping it back to a very basic model with a small data set and gradually adding functionality while praying that it wouldn't break.

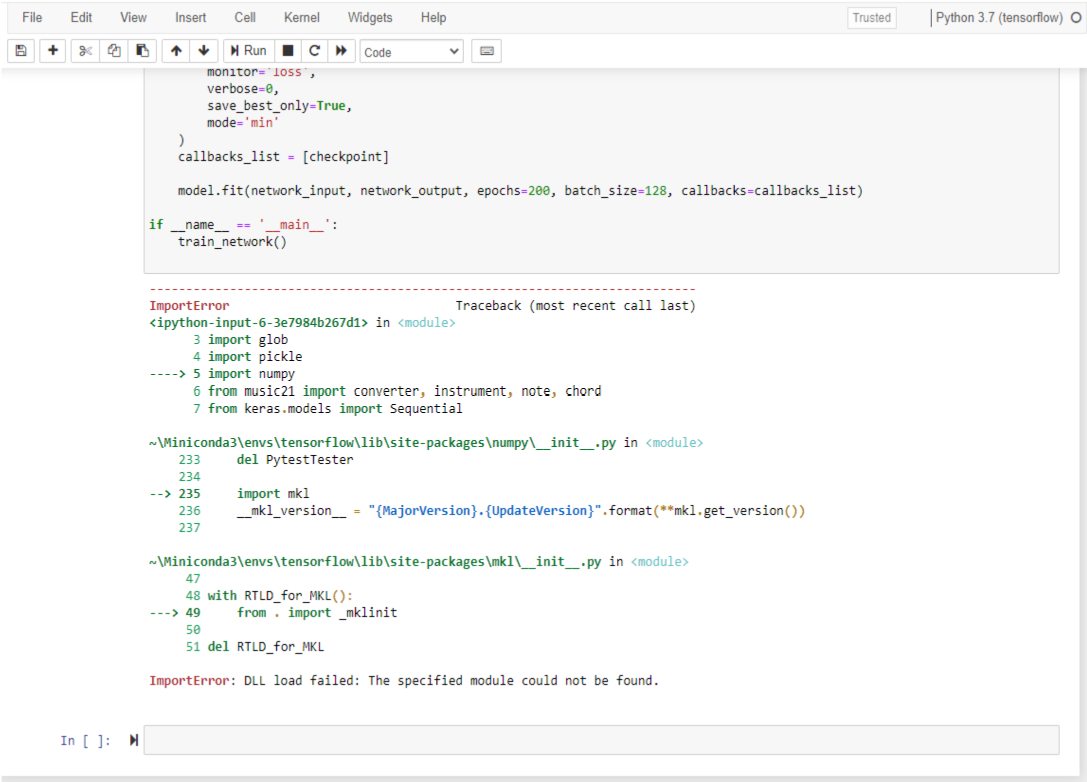

Figure 6: Failed attempt – issue was with incorrect version of libraries

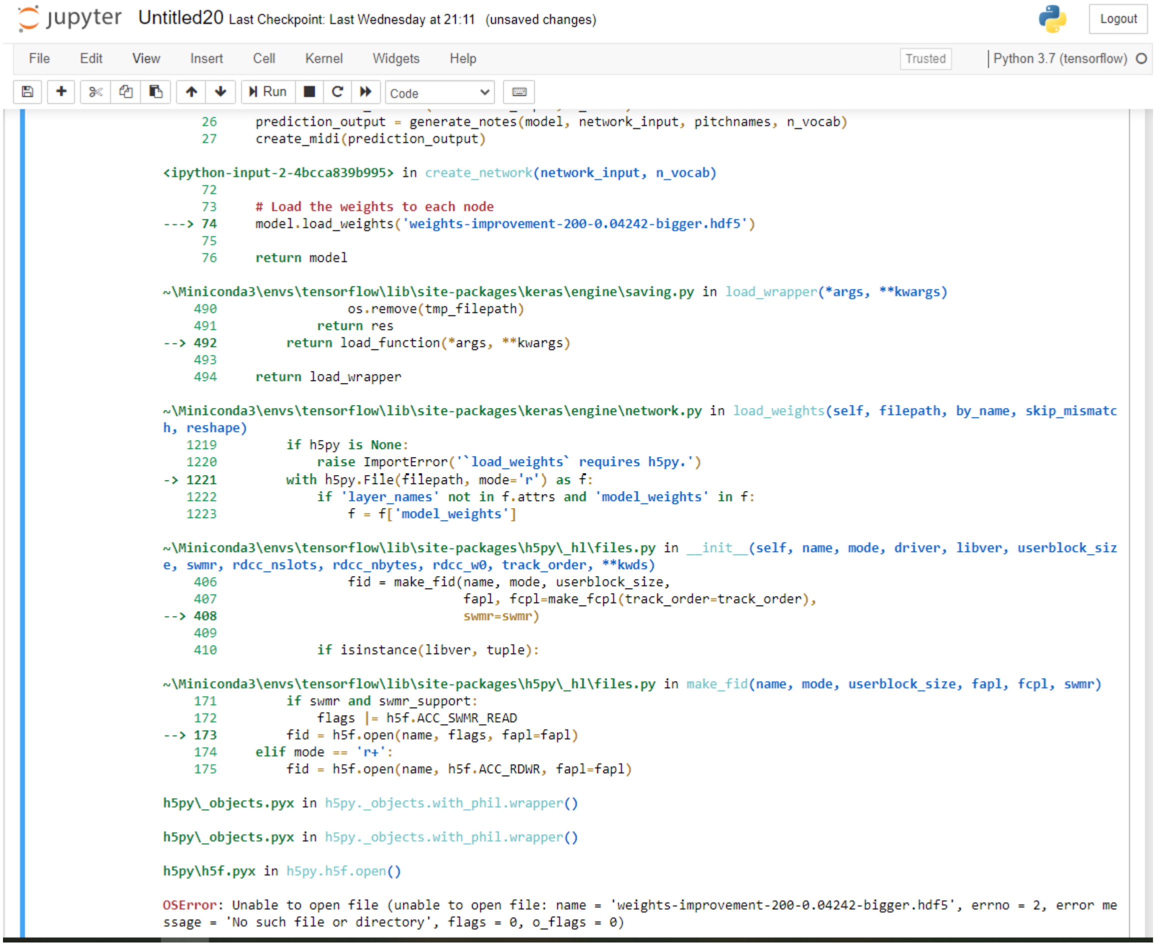

Figure 7: Typical error in generating sample (filepath of required weights is incorrect)

Installing the Tensorflow platform was a challenge in itself requiring command-line interface. The documentation provided by the publisher assumed a high level of knowledge and experience with working with platforms like this.

Software apparatus used

To install the Tensorflow platform on my Windows machine I used the Miniconda python distribution to deploy a dedicated Tensorflow environment. This was to protect the core environment in the case of a complete meltdown (not impossible). The Jupyter Notebook web application allowed me to experiment with the network without having to work within the command line interface. As mentioned earlier I also utilised the Pickle and NumPy modules.

Section 5 - Evaluation:

After many unexplained errors and re-installation of libraries this neural network was successful in generating almost 30 two-minute musical piano pieces. The majority of the samples never managed to get off the first note. Some samples simply overfitted by recreating elements from the training data and falling into a routine of repeating the same sequence for 500 notes. Some samples, however, produced interesting predictions. I will return here to Boden's (21) definition:

"Creativity can be defined as the ability to generate novel, and valuable, ideas."

I venture that some of the outputs from this trained neural network are divergent enough from the training set that inspired them to be regarded as 'novel'. I concede that the novel elements that I refer to may simply be examples of overfitting to elements of the training data that I am less familiar with, but I do not believe this to be the case. Whether any of these novelties have any 'value' or not is a matter of subjective opinion. The only way to determine whether this music is any good or not is to have people listen to it. I think for my part that some of them are interesting.

During the formal presentation of this project, I played a sample of some of the most interesting pieces and most people who heard them were very complimentary. I suspect that some of them were being polite but nonetheless I regard the human evaluation of the outputs predicted by this network to be overwhelmingly positive.

By using music of a similar genre and tone in the training data I helped to ensure that the network was not trained on a range of music that was too diverse. Using a data set that mixed the jazz of John Coltrane with the techno of Jeff Mills may have produced some 'interesting' results, but I suspect that the melody resulting from this prediction would only loosely resemble what we would call 'music'. With more time and compute power (and an endless supply of patience to manually review the results) available I believe that this network could generate some genuinely interesting and surprising outputs.

How does the project meet my objectives?

In what way do these results answer my original question? Is there now a definitive rejoinder to the problem of whether machines are capable of creativity and, therefore, thought? Unfortunately, one can only give a qualified response, and even that would be based on their own personal views on the issue. The answer could be 'Yes, if…'. It could also be 'No, but…'.

When Turing posited his Turing Test in 1950 he believed it would be beaten within 50 years. He considered the question not in terms of whether it was possible that machines can think, but *when* we will had a good enough reason to suppose that they *are* thinking (or at least doing something like thinking). I would go as far to suggest that there is a widespread, if not general, acceptance amongst both scientists and philosophers that this confidence in 'strong' AI is at least reasonable (2,4,43). There remains, however, a sizable cohert of 'scoffers' that are convinced that the AI hype-train is populated by groupies and fanatics (5). I wil not attempted to convince anyone with strong views on the subject either way, but I will make the following conclusions:

1: Machines can inspire us: We do not know precisely what is happenening inside a neural network to produce these funny sounds but we can certainly use them for our own artistic purposes. Sturm et al. (18) specifically developed their ANN to facilitate composition and experimentation for songwriters and musicians. Are these outputs the result of creativity on behalf of the coder or of the network? I don't have a definitive answer for that question, but if I was a musician and it helped me develop my craft and become a better artist I probably wouldn't think that was very important.

2: Machines can surprise us. Thinking is simply rational manipulation of symbols within the brain, and that is precisely what programmable computers do too (5). Algorithms are not surprising; they follow rules directly as they are written (unless they are written incorrectly, giving the programmer an unwelcome surprise). But when a large group of algorithms is collected together inside a connectionist neural network and allowed to mellow out for hours, days or weeks they can begin to produe surprising results. Very often the programmer cannot explain these results – the hidden layers of an ANN are a 'black box' of mystery (44). Nobody can look back on the logs after the fact to decipher why a network made a particular decision at a particular time. It is inexplicable.

Potential improvements:

As suggested earlier the availability of more compute power could have allowed more interesting experiments with larger data sets and more diverse genres of music. The OpenAI research has already demonstrated the exciting potential for generating predictions with multiple instruments directly as raw audio. From a more realistice perspective the introduction of interval offsets within this nueral network would have facilitated a more realistic timing pattern of notes, but would have made many generative examples more difficult to listen to (imagine a person with no natural sense of timing trying to keep a rhythm beat).

Conclusion:

Precisely what does this *mean?* Boden (21) is of the opinion that this is a question for the philosophers and doesn't really want to get involved herself (and I do not blame her). However, as a philosophy student it behoves me to attempt to put a slightly finer point on it.

For one to conclude that humans have abilities that cannot be recreated by machines suggests that humans are something *more* than their physically constituent parts (43). This seems to deny 'physicalism'; the predominant theory in philosophy of mind, and suggests that the mind exists outside of the physical realm. I consider this an unprovable claim and so I must conclude that machines can be creative (for what are we but thinking machines?).

As for whether *this* machine is creative, that probably depends on whether you think its outputs are any good (or if you think they could improve with time and training). If you do believe this I believe that I can say that these outputs are both novel and valuable. They are by no means transformative, but perhaps they could be said to be combinatory. They are certainly exploratory. They are certainly *not* historically significant, but we can say they are pieces of music that would require *intelligence* if they had been composed by a human.

Appendix:

Examples of musical outputs are available at: https://github.com/mark-606/didactic-octo-adventure

References:

1. Minsky M (editor). Semantic Information Processing. MIT Press; 1968.

2. Deutsch D. The Beginning of Infinity. Allan Lane; 2011.

3. Sun R. Artificial Intelligence: Connectionist and Symbolic Approaches. In: Smelser NJ, Baltes PB, editors. International Encyclopedia of the Social & Behavioral Sciences [Internet]. Oxford: Pergamon; 2001 [cited 2021 Apr 29]. p. 783–9. Available from: https://www.sciencedirect.com/science/article/pii/B0080430767005532

4. Block NJ, Fodor JA. What Psychological States are Not. The Philosophical Review. 1972;81(2):159–81.

5. Haugeland J. Artificial Intelligence: The Very Idea. MIT Press; 1985. 306 p.

6. Descartes R. Discourse on the Method of Rightly Conducting the Reason, and Seeking Truth in the Sciences. Sutherland and Knox; 1637. 136 p.

7. Hobbes T. Leviathan: Or, The Matter, Form, and Power of a Commonwealth Ecclesiastical and Civil. G. Routledge and sons; 1651. 328 p.

8. Boole G. An Investigation of the Laws of Thought on which are Founded the Mathematical Theories of Logic and Probabilities by George Boole. Walton and Maberly; 1854. 454 p.

9. Lovelace. Ada Lovelace’s Menabrea Letter Translation. 1841.

10. Turing AM. On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society. 1937 Jan 1;s2-42(1):230–65.

11. Turing AM. Computing Machinery and Intelligence. Mind. 1950 Oct 1;LIX(236):433–60.

12. Searle J. Minds, Brains, and Programs. The Behavioural and Brain Sciences [Internet]. 1980 [cited 2021 Apr 27];3. Available from: https://web.archive.org/web/20071210043312/http://members.aol.com/NeoNoetics/MindsBrainsPrograms.html

13. Sautoy M du. The Creativity Code: How AI is learning to write, paint and think. HarperCollins UK; 2019. 334 p.

14. aiva.ai. AIVA - The AI composing emotional soundtrack music [Internet]. [cited 2021 Apr 29]. Available from: https://www.aiva.ai/

15. Huang C-ZA, Hawthorne C, Roberts A, Dinculescu M, Wexler J, Hong L, et al. The Bach Doodle: Approachable music composition with machine learning at scale. arXiv:190706637 [cs, eess, stat] [Internet]. 2019 Jul 14 [cited 2021 Apr 27]; Available from: http://arxiv.org/abs/1907.06637

16. Payne C. MuseNet [Internet]. OpenAI. 2019 [cited 2021 Apr 27]. Available from: https://openai.com/blog/musenet/

17. Dhariwal P, Jun H, Payne C. Jukebox [Internet]. OpenAI. 2020 [cited 2021 Apr 27]. Available from: https://openai.com/blog/jukebox/

18. Sturm BL, Santos JF, Ben-Tal O, Korshunova I. Music transcription modelling and composition using deep learning. arXiv:160408723 [cs] [Internet]. 2016 Apr 29 [cited 2021 Apr 27]; Available from: http://arxiv.org/abs/1604.08723

19. Dieleman S. Generating music in the waveform domain [Internet]. Sander Dieleman. 2020 [cited 2021 Apr 27]. Available from: https://benanne.github.io/2020/03/24/audio-generation.html

20. Skúli S. How to Generate Music using a LSTM Neural Network in Keras [Internet]. Medium. 2017 [cited 2021 Apr 28]. Available from: https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5

21. Boden MA. Computer Models of Creativity. AI Magazine,. 2009;30(3).

22. Chomsky N. Cartesian Linguistics. Harper & Row; 1966.

23. Chomsky N. Reflections On Language. Temple Smith; 1976.

24. Oleinik A. What are neural networks not good at? On artificial creativity. Big Data & Society. 2019 Jan 1;6(1):2053951719839433.

25. McLaughlin BP. Connectionism. In: Routledge Encyclopedia of Philosophy [Internet]. 1st ed. London: Routledge; 2016 [cited 2021 Apr 26]. Available from: https://www.rep.routledge.com/articles/thematic/connectionism/v-1

26. Blais A, Mertz D. An introduction to neural networks [Internet]. IBM Developer. 2018 [cited 2021 Apr 28]. Available from: https://developer.ibm.com/technologies/artificial-intelligence/articles/l-neural/

27. Joshi P. Python Machine Learning Cookbook. Packt Publishing Ltd; 2016. 304 p.

28. SkillUP. Learn the Basics of AI [Internet]. Simplilearn.com. 2021 [cited 2021 Apr 28]. Available from: https://www.simplilearn.com/learn-ai-basics-skillup

29. anaconda.com. Anaconda | The World's Most Popular Data Science Platform [Internet]. Anaconda. [cited 2021 Apr 29]. Available from: https://www.anaconda.com/

30. python.org. pickle — Python object serialization — Python 3.9.4 documentation [Internet]. [cited 2021 Apr 28]. Available from: https://docs.python.org/3/library/pickle.html

31. numpy.org. NumPy [Internet]. [cited 2021 Apr 28]. Available from: https://numpy.org/learn/

32. web.mit.edu. music21 Documentation [Internet]. [cited 2021 Apr 28]. Available from: https://web.mit.edu/music21/doc/index.html

33. tensorflow.org. TensorFlow [Internet]. TensorFlow. [cited 2021 Apr 28]. Available from: https://www.tensorflow.org/

34. Brownlee J. How To Build Multi-Layer Perceptron Neural Network Models with Keras [Internet]. Machine Learning Mastery. 2016 [cited 2021 Apr 29]. Available from: https://machinelearningmastery.com/build-multi-layer-perceptron-neural-network-models-keras/

35. tutorialspoint.com. Keras - Dense Layer - Tutorialspoint [Internet]. [cited 2021 Apr 28]. Available from: https://www.tutorialspoint.com/keras/keras_dense_layer.htm

36. Keras.io. Keras documentation: Dropout layer [Internet]. [cited 2021 Apr 28]. Available from: https://keras.io/api/layers/regularization_layers/dropout/

37. Brownlee J. A Gentle Introduction to Long Short-Term Memory Networks by the Experts [Internet]. Machine Learning Mastery. 2017 [cited 2021 Apr 28]. Available from: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

38. Brownlee J. A Gentle Introduction to Batch Normalization for Deep Neural Networks [Internet]. Machine Learning Mastery. 2019 [cited 2021 Apr 28]. Available from: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/

39. Brownlee J. How to Choose an Activation Function for Deep Learning [Internet]. Machine Learning Mastery. 2021 [cited 2021 Apr 28]. Available from: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

40. Brownlee J. Why One-Hot Encode Data in Machine Learning? [Internet]. Machine Learning Mastery. 2017 [cited 2021 Apr 28]. Available from: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

41. oracle.com. Data Science Cloud [Internet]. [cited 2021 Apr 29]. Available from: https://www.oracle.com/ie/data-science/cloud-infrastructure-data-science.html

42. Brownlee J. Why Training a Neural Network Is Hard [Internet]. Machine Learning Mastery. 2019 [cited 2021 Apr 29]. Available from: https://machinelearningmastery.com/why-training-a-neural-network-is-hard/

43. Walmsley J. Mind and Machine [Internet]. London: Palgrave Macmillan UK; 2012 [cited 2021 Apr 27]. Available from: http://link.springer.com/10.1057/9781137283429

44. kdnuggets.com. The AI Black Box Explanation Problem [Internet]. KDnuggets. [cited 2021 Apr 30]. Available from: https://www.kdnuggets.com/the-ai-black-box-explanation-problem.html/