Nadezhda Atanasova

Methods of Artificial Intelligence Creativity

Methods of Artificial Intelligence Creativity

Nadezhda Atanasova
Bachelor's Thesis
Term Spring year 2021
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Option of Internet Services

Author(s): Nadezhda Atanasova
Title of the bachelor’s thesis: Methods of Artificial Intelligence Creativity
Supervisor(s): Anne Keskitalo
Term and year of completion: Spring 2021 Number of pages: 51

The objective of this thesis was to study the concept and methods of artificial intelligence creativity. The methodology consisted of several steps. Firstly, the basic structure of artificial neural networks was examined and compared to that of biological neural networks. The neuroscience of creativity was researched, and connections were drawn to the respective structures of different types of artificial neural networks. It was concluded that are the neuroscientific construction of creativity in the brain is most closely resembled by the recurrent neural network and the generative adversarial neural network. Examples of visual and literary artificial intelligence creativity were researched and presented.

The executional goal of the thesis was implementing different neural networks and observing how they compare to one another when given a creative task. The two types of models were a long short-term memory recurrent neural network and a generative adversarial neural network. They were fed a dataset of fictional book summaries and instructed to output sentences. Overall, the LSTM RNN performed better in terms of coherence and structure of the sentences. Both networks produced vivid imagery through their words on several occasions.

Keywords: artificial intelligence, machine learning, neural network, creativity

PREFACE

I would like to thank my thesis supervisor Anne Keskitalo for her patience and support.

Oulu, May 25, 2021
Nadezhda Atanasova

CONTENTS

1 INTRODUCTION 7
2 ARTIFICIAL INTELLIGENCE 9
  2.1 Types of machine learning 9
    2.1.1 Supervised machine learning 9
    2.1.2 Unsupervised machine learning 10
  2.2 Deep learning 12
3 NEURAL NETWORKS 15
  3.1 Biological neural networks 15
  3.2 Artificial neural networks 16
    3.2.1 Structure of an artificial neural network 17
    3.2.2 Types of artificial neural networks 19
4 CREATIVITY 22
  4.1 The neuroscience of creativity 22
  4.2 Creativity in artificial neural networks 24
    4.2.1 Visual creativity 24
    4.2.2 Linguistic creativity 26
    4.2.3 Overview of ANN creativity 29
5 TRAINING A NEURAL NETWORK 31
  5.1 Data preparation 31
  5.2 LSTM RNN 32
    5.2.1 Results 33
    5.2.2 Discussion 34
  5.3 GAN 35
    5.3.1 Architecture 1 37
    5.3.2 Architecture 2 38
    5.3.3 Comparison of GAN architectures 1 and 2 42
6 CONCLUSION 44
REFERENCES 45

ABBREVIATIONS

AI – Artificial Intelligence

ANN – Artificial Neural Network

CNN – Convolutional Neural Network

FNN – Feedforward Neural Network

GAN – Generative Adversarial Neural Network

RNN – Recurrent Neural Network

1 INTRODUCTION

The ever-increasing presence of artificial intelligence and automation in modern life is giving rise to an important question – is there anything AI will not be able to do? Machine learning relies on detecting patterns since patterns can be presented mathematically, and AI, just like any other computer program only speaks the language of mathematics. For this reason, it can quickly master skills that have a strictly defined set of rules, such as assembly, manufacturing, cleaning and even driving. The learning process, however, is not that easy to define for concepts that do not have a strict definition in the first place. One such concept is creativity.

Creativity is a contradictory notion. While a work must break free from the pre-established pattern in order to be considered creative it also has to have enough of a pattern to not be complete nonsense. It is difficult to invent a mathematical solution for creativity, especially considering the neuroscience behind it has not yet come to a definite conclusion about the brain processes responsible for it. This thesis aims to provide insight on the topic of artificial intelligence creativity while also conducting an in-depth research on both neural networks and creativity.

The thesis consists of two parts: research and execution. The first part includes investigating the basis and types of neural networks in detail and comparing their structure to that of biological neural networks. The concept of creativity and the brain processes behind it are also investigated. Examples of neural network creativity are explored and presented. The objective of the research is determining which types of neural networks work best for different creative purposes, drawing parallels between the creative mechanisms of the brain and artificial neural network architectures and exploring the different ways AI creativity has been demonstrated so far.

The second part utilizes the knowledge gathered during the research by creating several types of neural networks and testing their performance. The task which is given to the networks is writing creative sentences after being fed a dataset of fictional book summaries. The objective of the execution is comparing how different neural networks with different adjustments perform with the task of creative writing and how does their performance live up to the expectations set in the research part of the thesis.

2 ARTIFICIAL INTELLIGENCE

Artificial intelligence is the ability of a computer program to mimic human thought. This can be exhibited through object and language recognition as well as decision making and problem-solving capabilities (IBM Cloud Education 2020. Cited March 14, 2021). Machine learning is a subset of artificial intelligence. What makes machine learning differ from general AI is its approach to achieve intelligence through data science. It does so through advance algorithms that parse data and discover patterns in it (Goyal 2020. Cited March 14, 2021). These algorithms learn a function that best maps input variables (X) to an output variable (Y) (Brownlee 2016. Cited March 14, 2021). The two main types of datasets used in ML are labelled and unlabelled. Labelled datasets have both input and output parameters, while unlabelled only have an input parameter (Gupta 2020. Cited March 21, 2021).

2.1 Types of machine learning

There are two main types of machine learning in terms of the datasets used to teach the algorithm – supervised and unsupervised. Supervised ML uses labelled datasets where both the input and the output variables are known, and it is only the function itself that is discovered by the algorithm. An individual adjusts or supervises the model until it returns the desired output. In unsupervised ML the output is unknown, and the model works on its own to discover unknown patterns in the data set. Unsupervised ML is usually used with unlabelled data (Guru 99 2021A. Cited March 15, 2021). There is also reinforcement machine learning where no datasets are used – the algorithm is modelled so that it interacts with the environment. It is then respectively given either positive or negative feedback based on whether it succeeds or fails to do what it was intended to (Kumar 2020. March 31, 2021).

2.1.1 Supervised machine learning

The two types of supervised machine learning are classification and regression. Classification is a learning task where the output has defined labels. The first dataset in figure 1 determines whether an individual will make a purchase based on several factors and the two possible outcomes are 1 or 0 – a purchase or the lack of a purchase. In regression the desired output has a continuous value in a particular range and the goal is that the function gets as close as possible to the actual output value, while evaluated through calculating the error value. In the second dataset in figure 1 the regression function is used to calculate the continuous values of the wind speed based on several parameters (Gupta 2020. Cited March 21, 2021).

User ID	Gender	Age	Salary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	1
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	1
15728773	Male	27	58000	1
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	1
15727311	Female	35	65000	0
15570769	Female	26	80000	1
15606274	Female	26	52000	0
15746139	Male	20	86000	1
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1

Figure A: CLASSIFICATION

Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
10.69261758	986.882019	54.19337313	195.7150879	3.278597116
13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
24.23155922	988.796875	19.74790765	318.3214111	0.329656571

Figure B: REGRESSION

FIGURE 1. Classification vs. regression (Gupta 2020. Cited March 21, 2021)

2.1.2 Unsupervised machine learning

There are two types of unsupervised machine learning – clustering and association. Clustering is a way of finding a pattern or a structure in an unlabelled dataset. (Guru 99 2021B. Cited April 5, 2021). The pattern is determined through finding similarities and dissimilarities between certain points and grouping them into clusters based on these findings, as shown in figure 2 (Priy 2020. Cited April 5, 2021).

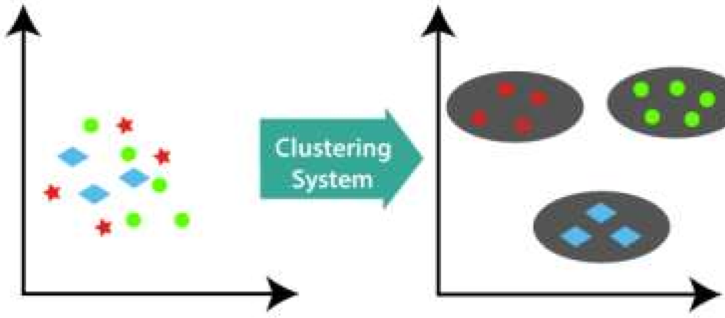

FIGURE 2. Clustering in machine learning (Laptrinhx 2019. Cited April 15, 2021)

The simplest and most commonly used clustering algorithm is the K-means algorithm. It operates by grouping similar data points and discovering underlying patterns. The K in K-means stands for a fixed number (k) of clusters which is predefined when writing the algorithm. The algorithm then knows it is looking for this precise number of centroids – a centroid being an imaginary or real location representing the centre of a cluster. In the beginning the centroids are randomly selected. Their positions are adjusted through each iteration of the algorithm, until the optimized positions are calculated (Garbade 2018. Cited April 16, 2021).

Association is a type of learning that discovers relations between variables in large datasets and identifies strong rules within them by counting their frequency of complimentary occurrences. The idea is that associations that take place together often are not randomly distributed and there is a reason behind it. One of the most common practical uses of association learning is basket data analysis – finding out which products are frequently bought together in stores and placing them together so that customers notice them. (DeepAI 2021. Cited April 17, 2021).

There are several ways to measure association. The three most common ones are support, confidence and lift. Support measures the proportion of the transactions in which a set of items appears. Confidence calculates how likely it is for an item Y to be purchased if an item X is purchased. Lift determines how likely is it for an item Y to be purchased when item X is purchased while also acknowledging the popularity of item Y by itself. It them returns a value which indicated whether there is an association between the items and furthermore, is it positive or negative, in other words, are customer more likely or less likely to purchase Y if they are purchasing X. A diagram of the associations between item selections can be seen in figure 3 (Ng 2016. Cited April 17, 2021).

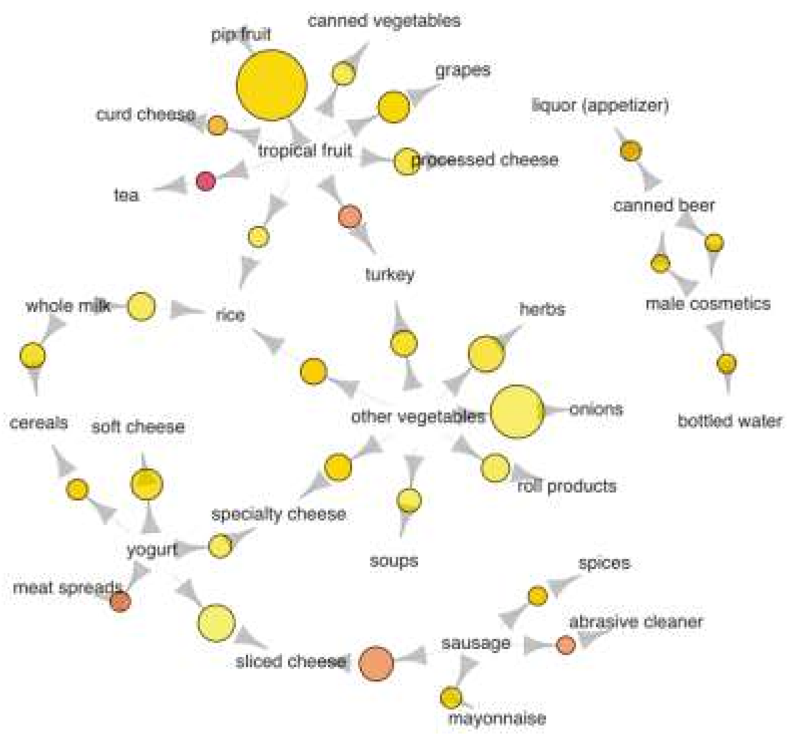

FIGURE 3. Association rules network graph (Ng 2016. Cited April 17, 2021)

2.2 Deep learning

Deep learning is a subset of machine learning that is conducted through artificial neural networks. What makes deep learning differ from general machine learning is its ability to learn and make intelligent decisions through its own computational methods, as if it has a brain of its own. The logic structure an artificial neural network has when analysing data is similar to that of the human mind (Grossfeld 2021. Cited April 18, 2021).

In deep learning no human operator is needed because the program learns from its own experience – defining complex concepts based on smaller ones. This conceptual hierarchy represents the depth in deep learning (Goodfellow et al 2016. Cited April 19, 2021).

There are two types of learning models – parametric and nonparametric. Parametric algorithms already have a function with a pre-set number of parameters in the very beginning. When they process the training data, they essentially learn the coefficients for that function, but the function or the number of parameters do not change. Nonparametric algorithms, on the other hand, do not have any assumptions about what the function might be in the beginning. This way they are free to learn any functional form from the training data, and they seek the one that fits it best. This provides them with better flexibility and more accurate prediction results. However, they require a lot more training data than parametric models and thus they are slower to train (Brownlee 2020. Cited April 20, 2021).

It is difficult to determine whether deep learning models are parametric or nonparametric, as they belong to what could be referred to as a grey area. They do have a pre-set count of parameters and a neural network that has a particular function, which means they could be considered parametric. However, neural networks are typically constructed of an incredibly large number of parameters. Each parameter has a coefficient that switches and adjusts as the algorithm is trained. The complexity of the network allows the model to behave almost as if it is nonparametric, giving it a considerable degree of freedom. This freedom is what is usually considered an advantage that deep learning holds over general machine learning.

A category where general machine learning falls short is its flexibility when it comes to decision boundary. Decision boundary is what determines which class a data point belongs to in a classification algorithm. The first graph on figure 4 shows a logistic regression algorithm, which is a type of classification. The data is linear, and the algorithm learns the linear decision boundary. However, it cannot learn decision boundaries for nonlinear datasets such as the one on the second graph in figure 4 (Pai 2020. Cited April 19, 2021). This occurs because a typically parametric model like this one stays true to its initial function and it can only do so much with its limited number of parameters.

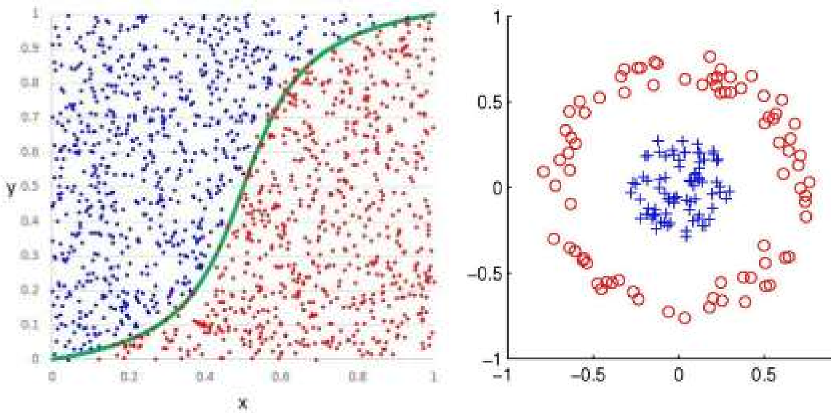

FIGURE 4. Linear vs. nonlinear data (Pai 2020. Cited 04.19.2021)

3 NEURAL NETWORKS

Nowadays the term neural network is usually used when referring to artificial neural networks. However, the design of these algorithms is partially inspired from the biological neural networks present in the human nervous system (Schiappa & Rudd 2017. Cited February 5, 2021).

3.1 Biological neural networks

The basic unit of a biological neural network is the neuron. The different functions of the neuron are carried out by its different parts, the main ones being dendrites, a soma, an axon and axon terminals.

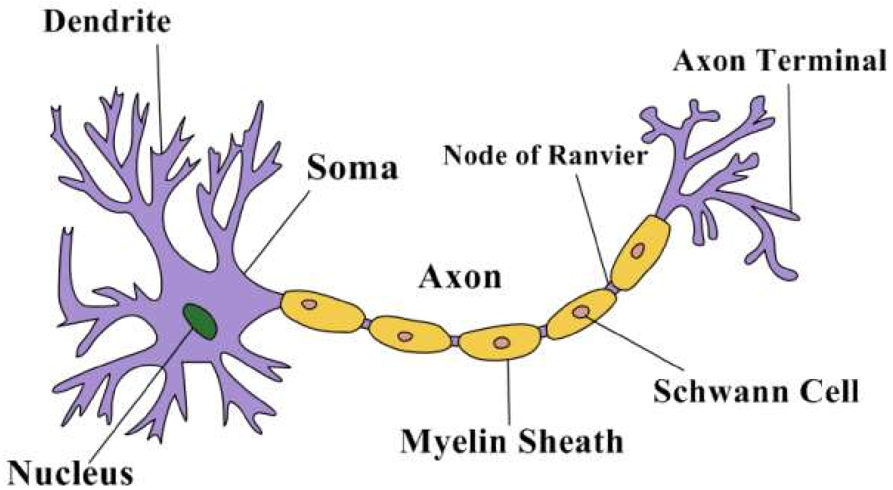

FIGURE 5. The structure of a neuron (NIDA 2021. Cited February 5, 2021)

The dendrites receive information from other neurons in the form of a chemical called a neurotransmitter. Examples of neurotransmitters are dopamine and serotonin. Attached to them are molecules called receivers. Each neurotransmitter is bound to a specific receiver like a key to a lock. Once received, the information travels within the neuron itself in the form of an electrical impulse (NIDA 2021. Cited February 5, 2021). The cell body of the neuron is called a soma. There the signals from the different dendrites are joined together. It is then determined whether the electrical signal in the neuron will fire or not. If the stimuli are strong enough then the total sum of the electrical signals coming from the dendrites exceeds the threshold limit of the axon hillock, causing the voltage gated sodium channels in the membrane of the axon to open and the neuron to fire. The axon then transmits the information to the other neurons through the terminal buttons located at its end. The connections between the dendrites of one neuron and the axon of another neuron are called synapses (Cherry 2020. Cited February 5, 2021).

Biological neural networks learn through neuroplasticity. Neuroplasticity is the ability of the brain to re-write itself upon learning new skills. Whenever something is experienced and learned new connections between the neurons responsible for the learning are formed. The paths of connected neurons are strengthened whenever the new skill is practiced (CQUniversity 2020. Cited February 7, 2021). This strengthening happens because the number of dendrites belonging to a neuron and connecting it to other neuron increases with the frequency of a behaviour. This enables the neurotransmitters to travel faster (Fit4D 2017. Cited February 7, 2021). Neurotransmitters are also sped up by certain cells belonging to the glia, which consists of the non-neuronal cells in the brain and essentially holds them together, participating in maintaining the neural working environment (The University of Queensland 2021. Cited February 7, 2021).

Another important part of learning is the ability to unlearn unused skills. This phenomenon is known as synaptic pruning. It is essentially the practice of breaking down old neural connections so that new ones can be formed instead. The most important role in synaptic pruning belongs to a subsection of glial cells called microglial cells. When a synaptic connection is not used it gets market with a specific protein. Upon detecting this protein microglial cells bound to it and destroy the synapse (Reeves 2018. Cited February 7, 2021).

3.2 Artificial neural networks

Artificial neural networks are based on biological neural networks, however, they are purely computational. There are different types of artificial neural networks that are trained for different uses.

3.2.1 Structure of an artificial neural network

An artificial neural network is a system of computational nodes organized in layers which processes information based on a response to external inputs. Similarly to biological neural networks, the building blocks of artificial neural networks are also called neurons. They receive signals either from a data set or from other neurons, depending on the network layer they belong to. A neural network consists of three types of layers, respectively named input, hidden and output layers. Each network has a single input and output layer but can have multiple hidden layers (Inzaugarat 2018. Cited February 15, 2021). Some simple neural networks do not have any hidden layers.

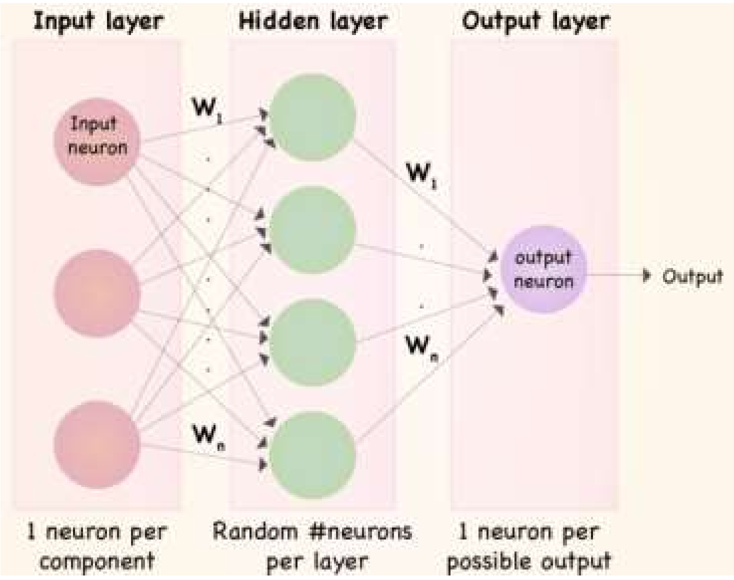

FIGURE 6. The structure of an artificial neural network (Inzaugarat 2018. Cited February 15, 2021)

A neuron takes one or multiple inputs, depending on its position in the network and releases a single output. Each input is multiplied by a weight – for example, if a neuron is connected to four different neurons from the previous layer, each of these connections had a weight. Weights are crucial to machine learning as they are one of the learnable parameters in an ANN – they are updated with each iteration so that the best output values are generated as the network learns (Ahirwar 2017. Cited April 16, 2021). A weight controls the strength of the connection between the neurons and determines the influence a certain input will have on the output (AI Wiki 2020. Cited February 17, 2021).

The other learnable parameter in an ANN is the bias. The bias measures how likely it is that a neuron will fire. While the weights determine the form of the function curve, the bias shifts the entire function along the x axis (Figure 7) in a way that it fits the data best. It is a constant in the whole network, meaning different neurons can have different weights, but they all have the same bias. A bias can be positive, negative or 0 – in case of a 0 bias the function depends on the weights and input values alone. The activation function (Formula 1) which is based on the input value, weight and bias determines whether the neuron fires or not (Collis 2017. Cited April 27, 2021).

*f(x) = w*x + b*
x = input value *FORMULA 1*
w = weight
b = bias

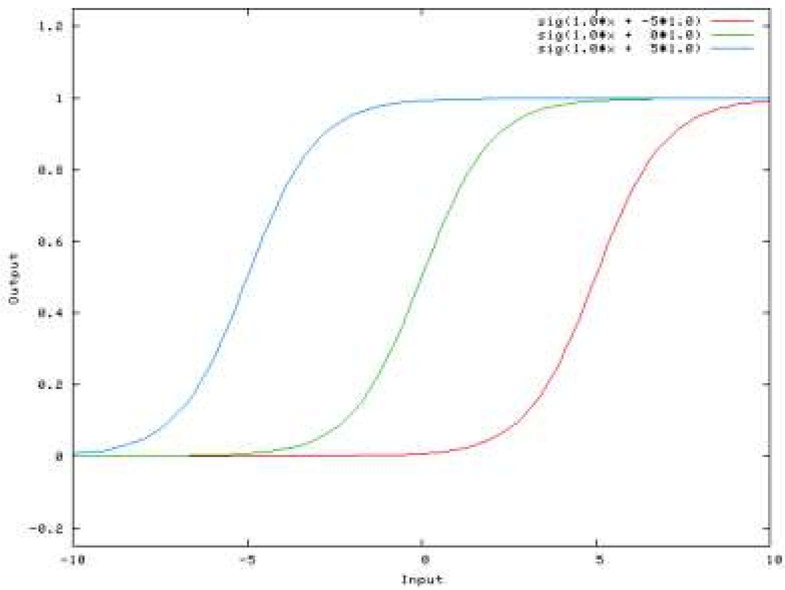

FIGURE 7. Three sigmoid curves with the same input data but with different biases (Collis 2017. Cited April 27, 2021)

The difference between the predicted value and the actual value is measured with a cost function. After each iteration the weights and biases are updated and adjusted in a way that promotes better accuracy. This precise adjustment that the parameters must undergo is calculated by an algorithm called backpropagation. Backpropagation calculates the gradient of the cost function with respect to the weights and biases (Hansen 2018. Cited April 28, 2021).

3.2.2 Types of artificial neural networks

There are several different types of artificial neural networks based on the set of parameters and mathematical operation they use. There are three types of neural networks that are commonly used for creative purposes – the Feedforward Neural Network, the Recurrent Neural Network, the Convolutional Neural Network and the Generative Adversarial Network.

3.2.2.1 Feedforward Neural Network

The Feedforward Neural Network (FNN) is the most simplified form of an ANN. The connections between the different units of an FNN do not form a cycle and the information only travels forward. Due to its simplicity the FNN is often combined with other networks in constructing multi-network learning structures rather than used alone. (Goyal 2019. Cited April 28, 2021).

3.2.2.2 Recurrent Neural Network

The Recurrent Neural Network (RNN) is essentially the opposite of an FNN as it works based on a loop – the output of each layer is sent back to the input. This is useful because it allows information to be retained and used later in order to increase the accuracy of the back-propagation prediction and self-learn when the outcome is wrong (Goyal 2019. Cited April 28, 2021). RNN is most commonly used for natural language processing since it closely resembles the way the human mind processes text or speech – when reading something, the brain interprets it while also having in mind what has been previously written in the text. A particular kind of an RNN called a Long Short Term Memory network (LSTM) specializes in learning long-term dependencies – this is useful in word prediction when processing long pieces of text (Olah 2015. Cited April 28, 2021).

3.2.2.3 Convolutional Neural Network

The Convolutional neural network (CNN) is a type of network which contains a specific type of layers called convolutional layers. The neurons in these layers receive input, then transform it and hand the transformed version of the input to the neurons in the next layer. The layers have filters which specialize in detecting patterns – for that reason CNN are commonly used in image processing. Mathematically, a filter is a small matrix which is given a number of rows and columns. Each number in this matrix corresponds to a unit of the information that is processed – for example a 3 x 3 convolutional filter processes 3 x 3 blocks of pixels in an image. The filter then slides across the whole image until it has processed every possible 3 x 3 block – the action of sliding and processing is called convolving (Figure 8).

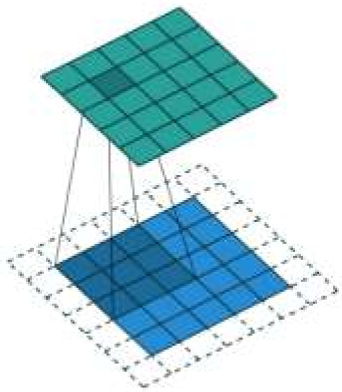

FIGURE 8. A convolutional processing operation (Deeplizard 2019. Cited April 28, 2021)

After all the filters in the layer have convolved the entire image, the output of each block is stored and sent to the next layer as a new pixel. The deeper a layer is within the network, the more specific the patterns it detects are. The first few layers, for example only detect geometric shapes while the deeper ones detect features such as eyes, feathers, fur. The deepest layers are capable of detecting full objects such as animals, plants and everyday items (Deeplizard 2019. Cited April 28, 2021).

3.2.2.4 Generative Adversarial Network

The Generative Adversarial Network (GAN) is an architecture which uses two different neural networks – a discriminator and a generator and respectively, two different algorithms – discriminative and generative. Discriminative algorithms classify data. They learn how to do so through a training data set which has features linked to labels – an example of features would be emails and an example of labels – span or not spam. After learning the data set the model knows which words indicate spam, hence it would be able to classify emails as spam or not spam based on what it has learned. Generative algorithms, on the other hand model the probability of features based on how likely they would be to appear in a certain label (Nicholson 2021. Cited May 4, 2021).

The basic mechanism behind GANs is simple – the generator generates new instances while the discriminator evaluates them. The discriminator is fed both the instances that the generator is learning from and the instances the generator is generating. The discriminator must learn to differentiate the instances made by the generator and mark them as false so that the generator can receive the feedback and learn how to better mimic the true features (Nicholson 2021. Cited May 4, 2021). A scheme of a GAN can be observed on figure 9.

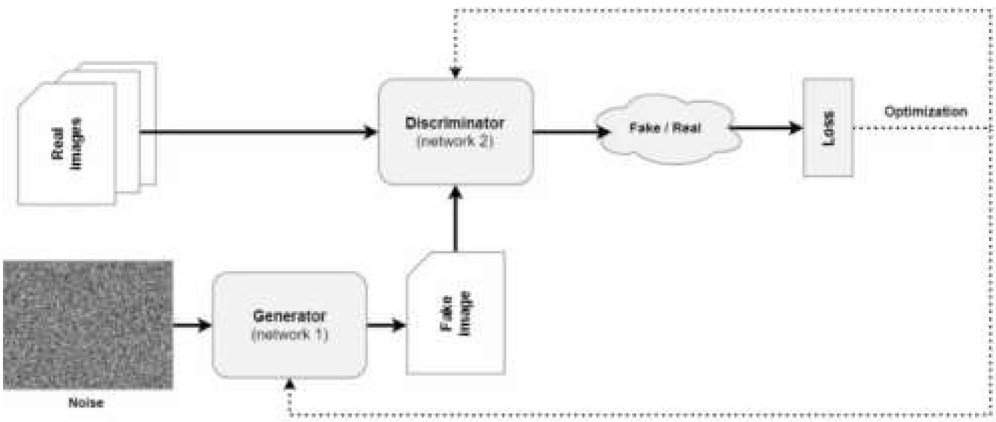

FIGURE 9. GAN workflow scheme (Versloot 2018. Cited May 4, 2021*)*

4 CREATIVITY

The concept of creativity is commonly reduced to the field of the arts but its relevance can be seen within numerous disciplines. Creativity is defined by the ability of the brain to seek innovative solutions that are also useful and relevant (Beaty 2020).

4.1 The neuroscience of creativity

Creative thinking is closely tied to the general ability to imagine the future. MRI scans have shown that both imagining the future and recalling the past are executed by the same brain region – the hippocampus. This observation supports the constructive episodic simulation hypothesis – a theory which suggests that both memory and imaginations are present when recalling past memories or imagining future events. Remembering an event requires the brain to reconstruct it by piecing together the people, objects and places that comprised it, while an imagined event is constructed by combining knowledge of the past (Beaty 2020).

A procedure called episodic specificity induction has proven that individuals become more creative after they have been trained to recall episodic memories in detail. This study contradicts the romanticized notion that creativity is completely spontaneous and suggests that there is cognitive effort behind it. Upon examining the brain connectivity pattern of the participants, it is proven that highly creative people have stronger functional connections between the default and control network, as well as the salience network, which is the one responsible for switching between the two. The default network is the part of the brain that is activated when one does not have a cognitive task to do. It engages in memory and imagination and thus it allows ideas to float through the mind in the form of episodic simulation. The control network is responsible for focus and critical evaluation. The default and control networks tend to work complimentary – when one is activated the other is deactivated. This suggests that creative people have the ability to switch them swiftly, allowing their brain to generate ideas and evaluate them immediately after that. Creativity also requires cognitively overcoming the stickiness of prior knowledge in order to find new pathways (Beaty 2020).

There are two different types of neural systems which underpin the mechanisms by which the creative process emerges. These two neural systems are both situated in the prefrontal cortex (PFC). One of these systems it is connected to four posterior neural regions (hippocampal, temporal, parietal and occipital cortexes) responsible for long term memory storage. It is believed that this system is responsible for maintaining the 'working memory buffer' (Dietrich 2004) by making query like searches of these long-term memory stores and effectively staging data to partake in the formation of novel associations.

The other, contrasting neural system is connected to a hierarchy of emotional processing centres beginning with several limbic system structures and ending in the cingulate cortex (Damasio 1994). It is involved in little database searching behaviour as emotional signals are staged in a primarily unconscious fashion owing to there being many neurons coming from the emotional centres to the PFC region and very few being directed back toward the emotional centres. This emotion focused PFC system affectively attaches a value to stimuli based on its biological significance while the cognitive PFC system processes all information neutrally without adding significance on certain events more than others.

While these two systems have separate limbic structures and each one of them keeps their own distinct memory (emotional and conceptual), it should be noted that there is a high degree of interconnectivity between the two regions (Fuster 2000).

Building on this, it has been proposed by Arne Dietrich (2004) that the compartmentalisation of the decision-making PFC described above leads to 4 non-exclusive modes of creativity with distinct neural pathways. The four modes can be summarized through their derivation: there are two centres of processing (emotional and cognitive) and two methods by which long-term memory data is staged in the working memory buffer (spontaneous and consciously searched). Creativity is proposed to be the result of a combination of these 4 modes operating within a processing system which may be limited to manipulating a single quaternary relation in parallel (Halford et al. 1998).

4.2 Creativity in artificial neural networks

The two-neural-systems theory proposed by Dietrich is closest to a GAN than to any other known artificial neural network. In both cases there are two systems, one of which acts on creating associations from memory while the other one acts as an evaluator. A pattern of similarity can also be drawn to RNNs, since imagination has to do with the ability to reflect the past and RNNs essentially act as memory, processing the information in relation to what they have previously processed.

4.2.1 Visual creativity

Artificial neural networks are often utilized for visual purposes such as object detection and object segmentation. Usually, visual tasks are performed by CNNs. Some experimentation has led to visual neural network skills also being utilized for creative tasks.

4.2.1.1 Google DeepDream

An example of visual ANN creativity is Google DeepDream – a learning model which alters images by detecting and enhancing patterns. DeepDream was created by Alexander Mordvintsev in 2015 (Wikipedia 2021. Cited March 10, 2021). It is a way of running a network rather than a particular algorithm and similar results can be achieved with any previously trained deep convolutional neural network. The specific network used in DeepDream is called GoogLeNet and it is a classification model which determines what an image contains by detecting patterns. Such models are usually trained to detect particular patterns, which makes them highly biased – for instance, if the model was initially trained to detect dogs it will find dogs in any shape or texture that might slightly resemble the pattern of a dog. DeepDream works as a feedback loop which modifies the image with the patterns it has detected as it makes them visible and enhances them as shown on figure 10 (Computerphile 2016. Cited April 22, 2021).

FIGURE 10. An original photo and its DeepDream pattern enhanced version

4.2.1.2 StyleGAN

Visual ANN creativity is also expressed through GANs. What makes them effective in creative tasks is their ability to perfect a skill through the constant optimization of the result due to the information exchange between the generator and the discriminator. StyleGAN is a type of GAN that specializes in generating realistic images of human faces. It was created in 2018 by Nvidia researchers Tero Karras, Samuli Laine and Timo Aila. StyleGAN is an unsupervised learning model which has a unique style-based generator. The style-based generator differs from regular generators in its lack of an input layer. Instead, it begins the generative process from a learned constant. After that the style of the image is further adjusted with each convolutional layer. Another reason for StyleGANs exceptional accuracy is the high quality and extensiveness of the dataset it has been trained with (Figure 11).

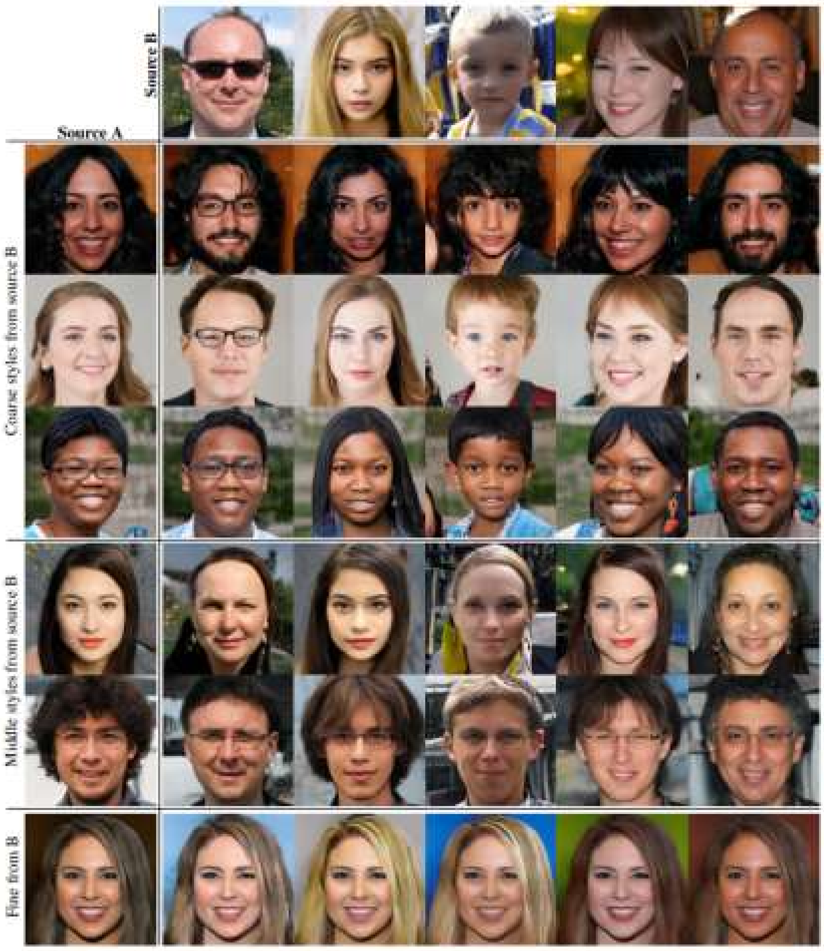

FIGURE 11. Faces generated by StyleGAN (Karras et al 2018)

4.2.2 Linguistic creativity

In recent years neural networks are often utilized in next work prediction algorithms. These models are most commonly LSTM RNNs and they are used in message and texting applications.

4.2.2.1 Fiction

In 2016 the artificial intelligence writing platform Literai (O’Neill et al) was created. Even though the writings generated by Literai have little coherence it is evident that the network manages to grasp the patterns of the writing style it is being fed (Figure 12).

"The moment what the villas which had been of present. The second strength of the house of the white would rather the first devil in the while and two threatened to the last business. The facts was too and an hour who is the strange hands to his eye and side. I have been of a face and twenty and down the description of an end and shall get for a little from his whole instantly desire, seek of which I am in the man about a gentleman langer.

"I was about in the street."

"No, no, when you have the papers?"

FIGURE 12. A segment Literai has generated after learning from several Sherlock Holmes novels (Sahay 2018. Cited May 25, 2021)

4.2.2.2 Poetry

ANN creativity has also been observed in poetry. Deep-speare (Lau et al 2018) is a joint architecture which specializes in writing poems by focusing on the stress and rhyme aspects. Its learning dataset is divided into two parts – one is made out of all kinds of poems and serves for pre-training and word embedding purposes, while the other consists solely of sonnets. A sonnet is a type of poem which has a particular structure – 3 quatrains consisting of 4 lines and a couplet that consists of 2 lines. It also follows specific aesthetic forms such as stress and rhyme. This consistency of patterns makes sonnets a perfect training example for neural networks. Deep-speare is based on 3 different LSTM models – a language model, a pentameter model and a rhyme model. The language model specializes in predicting the next word. It trains the architecture in general language processing since the sonnet data is too small to do that. The pentameter model captures the stress patterns. Each sonnet line has 10 binary stress symbols which are either positive or negative (Figure 13).

S^- S^+ S^- S^+ S^- S^+ S^- S^+ S^- S^+

Shall I compare thee to a summer's day?

FIGURE 13. Stress pattern in a sonnet line (Lau et al 2018)

Based on these symbols the attention weights of each sonnet line are mapped (Figure 14). When a line is produced it mimics these attention weights and checks whether the stress pattern match the actual stresses of the words in a pronunciation dictionary. If the stresses do not match the words are discarded and the line is re-written until there is a match.

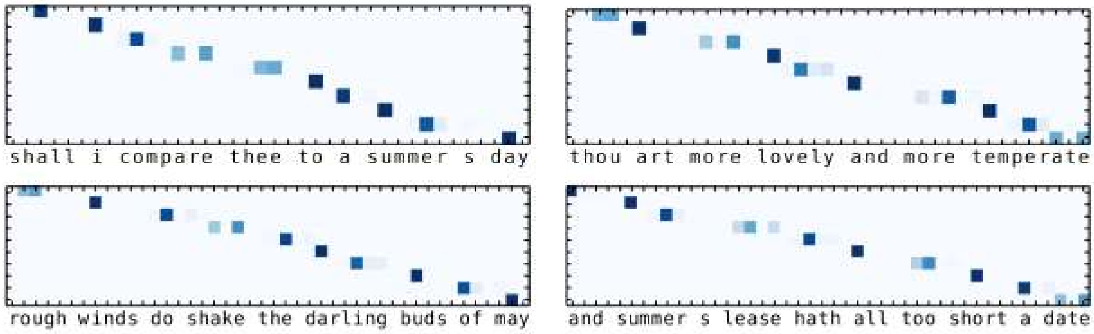

FIGURE 14. Attention weights in Shakespeare's Sonnet (Lau et al 2018)

The rhyme model is trained in an unsupervised manner in order to be more flexible when learning poetry in other languages. Similarly to the pentameter model it is also trained by being fed rhyming words that are evaluated based on the pronunciation dictionary.

Upon evaluation from English literature experts it was concluded that even though Deep-speare has higher accuracy in terms of following the exact rules of poetry compared to human poets it cannot replicate the emotional impact and readability human-written poems stand out with. Furthermore it was noted that poets often break the rules on purpose in order to create other effects (Lau et al 2018).

4.2.2.3 Screenwriting

In 2016 the sci-fi short film Sunspring (Sharp & Goodwin 2016) was shot based on an AI-written screenplay. The network that produced it - an LSTM RNN was trained on various screenplays, mainly ones of sci-fi films from the 80s and 90s (Wikipedia 2019. Cited April 29, 2021). There is no information available on the exact execution of the training model, however the screenplay and the film itself can be viewed online. Jonathan Cohn (2021) argues that the screenplay has no coherence and actual artistic value as a storytelling artifact. He insists that in the actual film the director and the actors try their best to present the lines and events as if there could be meaning behind them, although no such meaning can be recognized in the script itself. Upon reading the screenplay (Figure 15) it is evident that even though it is correctly structured and contains complex sentences it clearly lacks signs of cohesive storytelling.

```
INT. SHIP

We see H pull a book from a shelf, flip through it while
speaking, and then put it back.

                    H
          In a future with mass unemployment,
          young people are forced to sell
          blood. That's the first thing I can
          do.

                    H2
          You should see the boys and shut
          up. I was the one who was going to
          be a hundred years old.

                    H
          I saw him again. The way you were
          sent to me... that was a big honest
          idea. I am not a bright light.

                    C
          Well, I have to go to the skull.  I
          don't know.
```

FIGURE 15. A snippet of the screenplay of Sunspring (Sharp & Goodwin 2016)

4.2.3 Overview of ANN creativity

Regardless of these examples it is still debatable whether truly creative neural networks have been developed. Arthur Juliani (2016) proposes that rather than being creative, current neural networks are simply impressive imitators. While there have been paintings, images, music, poetry and even screenplays generated by artificial intelligence algorithms it is still argued whether those artifacts can fit into the strong definition of creativity. Deep learning algorithms create after learning the pattern of what has been previously created. For instance, in order to produce an image, they learn the pattern of other images. All the grand movements in art have proclaimed themselves as meaningful by inventing something completely new. Painters such as Leonardo da Vinci, Picasso and Salvador Dali are not considered great just because of their skills, but rather because of their ability to break out of the established pattern of the past and create something so innovative that it changes the whole course of art. This phenomenon is called alterity. Alterity means something new that is not simply a novel combination of pre-existing elements, but rather a fundamentally different concept that has never been seen before (Juliani 2016. Cited April 21, 2021).

5 TRAINING A NEURAL NETWORK

This chapter explores different approaches to training neural networks to generate creative text. Several different artificial neural networks were implemented with the machine learning library Tensorflow. All the networks were fed the same dataset consisting of fictional book summaries. The task of the neural networks was generating creative and cohesive sentences. The chapter documents the differences in their architecture, output and success with the given task.

The minimal requirement for training a neural network is to have a set of ‘features’ and ‘labels.’ In our case the features are the sentences (processed into sequences of integers) and the labels are some judgements about a property that sentence could have, for example, whether or not a sentence is grammatically correct or incorrect. The network then makes predictions about the label for a sentence and then it is adjusted according to how close it was to the actual label for that sentence. In this investigation the high level Keras API inside of Tensorflow 2.0 was used to construct and train all the ANN architectures.

5.1 Data preparation

The first step in training a text generating neural network is to source a corpus of human created examples. Since the focus of this investigation is on generative creative writing, the following experiments all make use of the CMU Book Summary Dataset (Bamman & Smith 2013) corpus which consists of a collection of book summaries labels with the genre of the text along with other categorical groupings (only genre was made use of in this investigation). This corpus of text was then processed into a form that could be used to efficiently and accurately prototype a range of architectures.

Firstly, the summary paragraphs were first broken into individual sentence based upon the occurrences of a full-stop character. Then the words were tokenised (associating a single integer value to a word instead of a string of character values) and had a series of regex replace functions applied to isolate the individual words from non-word items such as brackets, underscores and quotation marks (Figure 16).

```python
def cleanUpsentence(sentence):
    sentence = sentence.lower() #make all letters lowercase
    sentence = re.sub(r"\(.+\)", "", sentence.strip()) #remove bracketed content
    sentence = re.sub(r"\n", "", sentence)  #remove new line characters
    sentence = re.sub(r"[\d_@+\\\/\,'\"]", "", sentence) #remove numbers and non word characters
    sentence = re.sub(r" {2,}", " ", sentence) #remove excess white spaces
    squished_sentence = re.sub(r" ", "", sentence)
    for i, character in enumerate(squished_sentence):
        char_code = ord(character)
        if(char_code < 97 or char_code > 122):
            return False #remove all sentence with non latin letters
    if(len(sentence) < 1):
        return False #remove all empty strings
    return sentence
```

FIGURE 16. Cleaning the sentences of irrelevant characters

Since in most cases the input layer of a neural network has a constant shape, for most of the networks it was required that the input sequences were of equal lengths. Hence, once the sentences were tokenised, the longest sentence (by token count) was taken as the definition for the NN’s input shape, and all other sequences were padded with zeros in order to have the same shape as this maximum sequence (Figure 17).

```python
def getRealTokenLists():
    paragraphs = getRawParagraphs() #extract paragraphs from CMU book summaries dataset
    training_sentences = getSentences(paragraphs, 500000, 25) #seperate and process sentences

    tokenizer = Tokenizer(oov_token="<OOV>") #use out of vocabulary token to handle unkown words
    tokenizer.fit_on_texts(training_sentences)
    word_index = tokenizer.word_index  #associate token index with each word ie. cat => 29

    training_sequences = tokenizer.texts_to_sequences(training_sentences)
    #create sequences of tokens ie. the black cat => [5, 32, 29]
    training_padded = pad_sequences(training_sequences, padding='post')
    #make all sequences the same length by padding smaller sentences ie. [5, 32, 29] -> [5, 32, 29, 0, 0]
    return (training_padded, tokenizer)
```

FIGURE 17. Cleaning the sentences of irrelevant characters

5.2 LSTM RNN

The first network type of network that was implemented was an LSTM RNN. Four different models were implemented and compared. This network takes a sequence of words and outputs a prediction for the next word in the sequence. The data required an extra step of processing whereby each sequence was used to create a collection of smaller sequences with the final word removed and used as the label for the sequence (Figure 18).

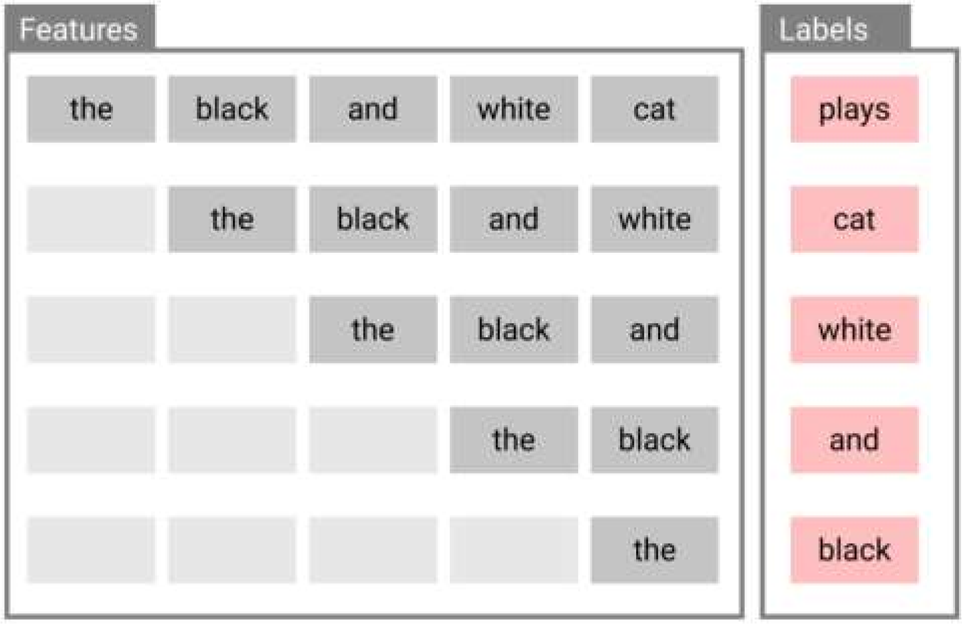

Figure 18. The separation of features and labels from padded word sequence N-grams

5.2.1 Results

Several arrangements of the aforementioned architectures were trialled to see how each of the elements influenced the ability of the network to produce an output that resembled. All the models were trained on the same corpus of processed word sequences and were used to generate 12-word sentences from the seed word sequence: “there is”. A sample of the results can be observed in table 1.

Model number	Sequence length	dimensions		Epochs	Output sentence
		Embedding	LSTM		
1	20	280	150	5	There is the arrive agent dedlock stanford loan

					loan bullied temporary voldemort.
2	10	100	100	5	There is a letter race of the novel painting gear kilimanjaro criminal.
				30	There is a ghost among porcelain statues dance back home mrs harassment.
3	10	20	50	30	There is a young woman in the mantelpiece and agrees lakes again.
4	10	1	50	30	There is are the only thing of the black week of the.

Table 1. Results of the different LSTM models

5.2.2 Discussion

From comparing Model-1 and Model-2, it appears that the length of the sentence is correlated with quality and coherence of the generated output. Model-1 is trained on sequences that are a maximum of 20 tokens long while Model-2 is trained on sequences are a maximum length of 10 tokens. The extra information provided to Model-1 seems to be detrimental to its performance as even though it is a larger network both in terms of the embedding and LSTM dimensions, it produces a lower quality generation compared to MODEL-1 event after the same amount of training (5 epochs).Arguably, the difference is marginal, however given the presence of repeated words in the Model-1 output, it seems that training on longer sentences can provide a range of variables that such small networks cannot handle.

As expected, it appears that the larger the network, the better the quality of the generated output. This is exemplified in the comparison of Models 2,3 and 4. The displayed output for each of these models was generated following a common 30 epoch training procedure involving around 20,000 sequences at 10 tokens in length. Interestingly, the accuracy of the model predictions for the training sequences is confidently around 30% after 30 epochs. However, the output of the models degrades noticeably as we reduce the number of embedding and LSTM layer dimensions. Despite the apparent need of some punctuation, the generated sentence after 30 epochs from Model-1 quite closely resembles a poetic extract. Model-2, similarly, appears to be in need of punctuation, but with a fifth as many embedding dimensions and half as many LSTM dimensions, we start to see some non-sensical word combinations. The output of Model-3 makes even less sense than that of Model-2, given that the embedding layer only has one dimension in this case (the words are positioned along a one-dimension line) while the LSTM dimensions are the same as Model-3, we can see that more embedding dimensions seem to yield better output quality.

5.3 GAN

As previously discussed, a GAN involves two distinct networks, a “generator” which produces a fake product and a “discriminator” which judges whether or not the fake product looks like a real product. In the case of this project, the product produced by the generator is a sequence of words: a sentence.

The discriminator network is trained by providing it with fake sentences and real sentences, its loss is determined based upon how well it predicts whether a sentence is real or fake. The generator network on the other hand is trained based upon how poorly the discriminator predicts that the generated sentence is “fake”. If the discriminator correctly predicts that the generated sentences are fake, the generator network will incur a greater loss for that training step.

In this project I created two different GAN architectures and made deductions about the effectiveness of their architectural elements based upon a qualitative comparison of the generated sentences. The two architectures differ in terms of the methods used for vectorising the text sentences, the training regime for the discriminator and the nature of the generator input, while sharing the same generator (Figure 19).

```
generator = tf.keras.Sequential([
  tf.keras.layers.Dense(10, activation='relu'),
  tf.keras.layers.Dense(100, activation='sigmoid')
  tf.keras.layers.Dense(50, activation='relu'),
  tf.keras.layers.Dense(30, activation='relu'),
  FloatToToken(10) #custom built layer for turning floats into integer word tokens
])
```

Figure 19. The generator shared by all the GAN models

There were three distinct architectures of discriminators used throughout the GAN models. These included a standard FNN discriminator (Figure 10), a CNN discriminator (Figure 21) and an LSTM discriminator (Figure 22).

```
feed_forward_disriminator = tf.keras.Sequential([
  tf.keras.layers.Dense(100, activation='sigmoid')
  tf.keras.layers.Dense(50, activation='relu'),
  tf.keras.layers.Dense(30, activation='relu'),
  tf.keras.layers.Dense(10, activation='relu'),
  tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figure 20. FNN discriminator

```
convolution_discriminator = tf.keras.Sequential([
  tf.keras.layers.Conv1D(32, 5, activation='relu', input_shape=(max_length, 0, 0)),
  tf.keras.layers.GlobalAveragePooling1D(),
  tf.keras.layers.Dense(50, activation='relu'),
  tf.keras.layers.Dense(30, activation='relu'),
  tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figure 21. CNN discriminator

```
lstm_discriminator = tf.keras.Sequential([
  tf.keras.layers.Reshape((max_length, 1)),
  tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(100)),
  tf.keras.layers.Dense(50, activation='relu'),
  tf.keras.layers.Dense(30, activation='relu'),
  tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figure 22. LSTM discriminator

5.3.1 Architecture 1

In this architecture the discriminator network was pre-trained CNN (FIGURE___) using shuffled (labelled as fake) and un-shuffled real sentences (labelled as real) sourced from the CMU book summary dataset. The sentences were vectorised as sequences of integer tokens. The discriminator was trained on 40,000 sentences with a maximum length of 10 words for 10 epochs with the standard Keras binary cross-entropy loss function. During training the accuracy of the network (the probability that it correctly predicts whether a sentence is shuffled or not) steadily reached a maximum accuracy at around 94%.

The generator in this architecture took a real sentence as an input and returned a new generated sentence in the form of an array of integer tokens. These token sequences were passed as input into the pertained discriminator which returned a prediction for whether or not the sequence was fake (a shuffled sentence) or real (an un-shuffled sentence). The loss for the generator was also calculated using the Keras binary cross-entropy loss function with higher loss being incurred when the discriminator predicted that the output was fake. The network was trained with 20,000 input sentences of 10 words in length, within one epoch it reached an accuracy of 100% meaning that the discriminator consistently judged that the generated output was not jumbled.

The main idea behind this architecture is to have the discriminator judge whether or not the generated output resembled meaningful text sourced from the corpus or meaningless shuffled text. By training the generator with the discriminator’s predictions, we should be able to create a generator that outputs coherent sentences instead of jumbled word sequences.

The model came up with the following outputs:

fairly mastermind chillingworth interviewed lane campbell introducing rents rejoin combined

relief manifestations blunt salesman kuhn encourage attractive nd fouquet campbell

costume upcoming investment fairly april wanda improves attractive introducing ambiguous

5.3.2 Architecture 2

For this architecture the discriminator and generator were trained simultaneously. Instead of providing the discriminator with shuffled sentences (as was the case in the last architecture), the fake input consisted of sentences produced by the generator network during the training step. Similarly to the previous architecture, the generator was trained within the same training step based upon the output of the discriminator. The GANs were trained on 10 epochs with 20,000 10 word long sentences, custom loss functions were used for the discriminator and generator (Figure 23). Five different versions of this architecture were trialled throughout the process of experimentation.

```
def discriminator_loss(real_output, fake_output):
  real_loss = cross_entropy(tf.ones_like(real_output), real_output)
  fake_loss = cross_entropy(tf.zeros_like(fake_output), fake_output)
  total_loss = real_loss + fake_loss
  return total_loss

def generator_loss(fake_output):
  return cross_entropy(tf.ones_like(fake_output), fake_output)
```

Figure 23. Custom loss function

5.3.2.1 Version 1

In the first version I experimented with using 10 word long real sentences as the input for the generator (as was the case in the previous architecture) and random noise (a 10-element sequence of random floats). The discriminator was a simple feed forward network (Figure 20). When the generator input was real sentences the loss for the generator quickly become arbitrarily small (less than 1 x 10^-5) within the first epoch during which time the discriminator loss reached the maximum possible value suggesting that all its predictions were close to random.

The model came up with the following outputs:

forsythe peewee fennec inquire woodhouse rutherfords gentlemans stirring rousillon steerforth

forsythe disown cuthbert ramoth raych tuckborough galaxias iunio inquire guermantes

eurydice insolvency gollums ulam nuptials robe sultry jondrette birlings kolenkhov

5.3.2.2 Version 2

For version 2 the input for the generator was random noise, and like in version 1, the discriminator was a feed forward network (figure). In the case this version, loss for both the generator and discriminator fluctuated with each epoch of training (same regime as version 1). This pattern of loss was more promising in this version compared to that of version1 as it suggests that both the discriminator and the generator are simultaneously and competitively improving. On the contrary, the rates of change seen in the loss of version 1 indicated that the training of the discriminator failed which in turn resulted in an invalid training of the generator.

The model came up with the following output:

. sees . . humans protagonist . . three duel

. each . . dying wounded . . dies angel

. kills . . president show . . find court

virginia . decides . . ghost incident alone . farm

blind . help . . political taking attack . poi-rot location . st . . students ships if . accident

dangerous . man . . c happened like . turned

5.3.2.3 Version 3

For this version I kept the random noise as generator input and experimented with using a CNN as the discriminator network architecture (figure). From experimenting with the first GAN architecture (with a pre-trained discriminator) I found that implementing convolutional layer effectively improved the accuracy with which the network classified sequences. When the more discriminator network was used, the GAN consistently underwent mode collapse within 5 epochs whereby the generator output the same result (despite the input being randomly generated).

5.3.2.4 Version 4

In this version I experimented with using an LSTM layer (figure___) instead of a convolutional layer to see if this style of network would prevent mode collapse from occurring. However, the same trend of mode collapse occurring within 5 epochs in version 3 was observed in this version.

5.3.2.5 Version 5

It was considered that the cause of the modal collapse was due to an unideal method of vectorising the sentences. Hence another approach besides tokenisation was attempted whereby the embedding layer of a separately trained CNN classifier (Figure 24) was used to convert each word token into a point within a 29 dimension space.

```
model = tf.keras.Sequential([
  tf.keras.layers.Embedding(len(token_index), 29, input_length=max_length),
  tf.keras.layers.Conv1D(5, 5, activation='relu', input_shape=(max_length, 0, 0)),
  tf.keras.layers.GlobalAveragePooling1D(),
  tf.keras.layers.Dense(29, activation='relu'),
  tf.keras.layers.Dense(len(genre_tokens), activation='softmax'),
  #output genere is taken as the corresponding node with the highest value
])
```

Figure 24. Embedding classifier

This classifier involved a convolution layer and was trained to predict the literary genre of the text that a sentence was sourced from. The network reached an accuracy of 30% for predicting the genre of a sentence out of 125 possible genres. The input for the CNN discriminator (figure) and the output of the GAN was shaped to match that of an input sequence consisting of 29 dimensional vectors. The idea was that the genre classifier would have spatially mapped the words in a linguistically meaningful way that could give the discriminator a head-start on distinguishing real and fake sequences thereby preventing mode collapse. However, these hopes were not realised, and this architecture also underwent mode collapse at the same rate that was seen in versions 3 and 4.

5.3.2.6 Comparison of versions 1-5

In comparing the different versions of architecture 2 we can see that when using a more complex discriminator, known to be suited for classifying text, the generator is more likely to undergo mode collapse. This was exemplified in versions 3 to 5 across which a range of discriminator network types and vectorisation methods were found to all result in a mode collapse mostly consisting of a sequence of spacer tokens (the tokens used to pad incomplete sentences). Since the same models used as discriminators were found to be effective as standalone classifiers, the resultant modal collapses indicates that the generator did not have a sufficient complexity/effectiveness to match the discriminator which is an essential feature in successful GAN networks.

Comparing versions 1 and 2 we can see that using real sentences or random noise as generator input has a distinct effect on the style and quality of the output. When the input is real sentences, there are no spacer tokens in the output and full 10-word sequences of words are produced. However, the sequence is by no means coherent and the words present do not seem to have any logical association with each other. This is in slight contrast to output from version 2 (using random noise as input) where the sequences did include spacer tokens but the words that were present in the sequence appeared to have at least some logical association. An exemplar of this trend is seen in the output “. sees . . humans protagonist . . three duel”. The observation (sees) of a hero archetype (human protagonist) often involves an event where multiple parties partake in some epic conflict (three duel). It is important to note that the generator of version 2, unlike that of version 1, was not directly exposed to any real sentences. This suggests that overexposing the generator to real sentences can hinder its ability to produce novel and meaningful constructions.

5.3.3 Comparison of GAN architectures 1 and 2

The output from architecture 1 in many regards is objectively of a higher quality than that of architecture 2, however, the creativity of architecture 1 should be called into question. Only version 2 of architecture 2 will be used in this comparison as the other versions were considered completely incoherent.

The sentences from architecture 1, unlike those of architecture 2, did not contain any spacer tokens so we can observe the networks ability to produce entire sentences. Although the grammatical coherence of the sentences are quite lacking, many of the outputs do seem to follow some kind of storyline as evidenced by the output “fairly mastermind chillingworth interviewed lane campbell introducing rents rejoin combined.” In this sentence we are provided with some imagery involving an interaction between two characters and we have some vague description about their qualities and intentions.

On the contrary, there is very little explicitness in the imagery provided by the outputs for architecture 2. An example of this is seen in the output “. each . . dying wounded . . dies angel”. Indeed this output is compelling and it is likely to conjure some kind of abstract imagery in many responders, however, the variation of interpretation is much broader for this input compared to that of architecture 1. The relative vagueness of the output of architecture 2 could easily be attributed to the fact that it involves less words due to the presence of spacer tokens.

From this comparison it appears that pretraining the discriminator improves the generators ability to produce complete sentences with explicit imagery subject to limited variation in interpretation. However, the validity of the creative element for architecture 1 is questionable. Since this architecture received real sentences as input it could have copied the real sentences to some extent thereby making the output more akin to roe regurgitation than true creative writing.

6 CONCLUSION

Artificial intelligence creativity is a complex topic which depends on various perceptual and computational factors. Machine learning has managed to capture and recreate the patterns of many different creative fields, such as visual art and writing. However, it is arguable whether this can be considered actual creativity – AI generated works are always lacking in one way or another. Visual styles are easily replicated by AI since they are heavily reliant on patterns, however machine learning algorithms have not come up with a new artistic movement hence it is difficult to consider their visual art to be actual creativity. Literary AI creativity manages to capture a desired style, such as type of poem or a writing genre. While its outputs are most often grammatically correct, they lack coherence. The two neural networks which are closest in their structure to the system of creativity in the brain are LSTM RNN and GAN.

The main factors in the performance of the LSTM RNN network are the length of the sentences in the training data and the amount of LSTM layers in the network. The bigger the sentences are, the more incoherent the output is. The quality of the output increases at the same rate as the increase of the number of LSTM layers.

A GAN network with a pre-trained discriminator provides more coherence than one where the discriminator is trained simultaneously with the generator. However, its creativity is questionable since it might have copied the structure of the sentences rather than focusing on the content. A GAN where the generator and the discriminator are trained at once produces poorly structured sentences, however the motives in them occasionally evoke vivid imagery.

REFERENCES

1. IBM 2020. Artificial Intelligence (AI). Date of retrieval March 15, 2021. https://www.ibm.com/cloud/learn/what-is-artificial-intelligence
2. Goyal 2020. Machine Learning vs Neural Networks. What is the Difference? Date of retrieval March 15, 2021. https://www.upgrad.com/blog/machine-learning-vs-neural-networks/
3. Brownlee J. 2016. How machine learning algorithms work? Date of retrieval March 15, 2021. https://machinelearningmastery.com/how-machine-learning-algorithms-work/
4. Guru 99. 2021A. Supervised vs Unsupervised Machine Learning: Key Differences. Date of retrieval March 15, 2021. https://www.guru99.com/supervised-vs-unsupervised-learning.html
5. Guru 99. 2021B. Unsupervised Machine Learning: What is, Algorithms, Example. Date of retrieval April 15, 2021. https://www.guru99.com/unsupervised-machine-learning.html
6. Priy S. 2020. Clustering in Machine Learning. Date of retrieval April 15, 2021. https://www.geeksforgeeks.org/clustering-in-machine-learning/
7. Laptrinhx. 2019. Machine Learning Clustering Algorithm. Date of retrieval April 15, 2021. https://laptrinhx.com/machine-learning-clustering-algorithm-2431293740/
8. Garbade M. 2018. Understanding K-means Clustering in Machine Learning. Date of retrieval April 16, 2021. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
9. DeepAI. 2021. Association Learning. Date of retrieval April 16, 2021. https://deepai.org/machine-learning-glossary-and-terms/association-learning

10. Ng A. 2016. Association Rules and the Apriori Algorithm: A Tutorial. Date of retrieval April 17, 2021. https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

11. Grossfeld B. 2021. Deep Learning vs Machine Learning: A Simple Way to Learn the Difference. Date of retrieval April 18, 2021. https://www.zendesk.com/blog/machine-learning-and-deep-learning/

12. Goodfellow I., Bengio Y, Courville A. 2016. Deep Learning. Boston: MIT Press. pp. 164-165. https://www.deeplearningbook.org

13. Brownlee J. 2020. Parametric and Nonparametric Machine Learning Algorithms. Date of retrieval April 20, 2021. https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/

14. Pai A. 2020. CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning. Date of retrieval April 20, 2021. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

15. Gupta M. 2020. Types of Learning – Supervised Learning. Date of retrieval March 20, 2021. https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/

16. Kumar A. 2020. What is Supervised Learning and its different types? Date of retrieval March 20, 2021. https://www.edureka.co/blog/supervised-learning/

17. Schiappa, M. & Rudd, E. 2017. Man vs machine: comparing artificial and biological neural networks. Date of retrieval February 5, 2021. https://news.sophos.com/en-us/2017/09/21/man-vs-machine-comparing-artificial-and-biological-neural-networks/

18. Cherry, K. 2020. An Overview of the Different Parts of a Neuron. Date of retrieval February 5, 2021. https://www.verywellmind.com/structure-of-a-neuron-2794896

19. Srinivasan, B. 2019. Biological neuron. Date of retrieval February 5, 2021. https://www.youtube.com/watch?v=PQEpTu47fS4

20. NIDA Junior Scientists Program 2021. Sending and receiving messages. Date of retrieval February 5, 2021. https://www.drugabuse.gov/sites/default/files/mod3.pdf

21. CQUniversity 2020. Neuroplasticity explained. Date of retrieval February 7, 2021. https://www.futurelearn.com/info/courses/neuroplasticians-and-neuromyths/0/steps/112300

22. Reeves C. 2018. Neurons that fire together wire together. Date of retrieval February 7, 2021. https://themindisthemap.com/neurons-that-fire-together-wire-together/?fbclid=IwAR3yaK_4WAhbfJVIqKzEXptRVRQovVfNGreA6nFB42mzJ1e9V1zaU2TeWNo#:~:text=There's%20an%20old%20saying%20in,the%20stronger%20that%20circuit%20becomes.&text=The%20more%20you%20practice%20piano,have%20known%20this%20for%20years

23. Fit4D. 2017. The neuroscience of behavior change. Date of retrieval February 7, 2021. https://healthtransformer.co/the-neuroscience-of-behavior-change-bcb567fa83c1

24. The University of Queensland. 2021. Types of glia. Date of retrieval February 7, 2021. https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/types-glia

25. Inzaugarat E. 2018. Understanding Neural Networks. What, How and Why? Date of retrieval February 15, 2021. https://towardsdatascience.com/understanding-neural-networks-what-how-and-why-18ec703ebd31

26. Nielsen M. 2015. Neural Networks and Deep Learning. New York: Determination Press. Date of retrieval February 15, 2021. http://neuralnetworksanddeeplearning.com/chap1.html

27. Zhou V. 2019. Machine Learning for Beginners. An Introduction to Neural Networks. Date of retrieval February 15, 2021. https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9

28. McCullum N. 2020. Deep Learning Neural Networks Explained in Plain English. Date of retrieval February 15, 2021. https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/

29. AI Wiki. 2020. Weights and Biases. Date of retrieval February 15, 2021. https://docs.paperspace.com/machine-learning/wiki/weights-and-biases

30. Ahirwar K. 2017. .Everything You Need to Know about Neural Networks. Date of retrieval April 26, 2021. https://hackernoon.com/everything-you-need-to-know-about-neural-networks-8988c3ee4491

31. Collis J. 2017. Glossary of Deep Learning Bias. Date of retrieval April 27, 2021. https://medium.com/deeper-learning/glossary-of-deep-learning-bias-cf49d9c895e2

32. Beaty R. 2020. The Creative Brain. Cerebrum 30(3), 10-13. Date of retrieval April 27, 2021. https://online.flippingbook.com/view/617165/10/

33. Dietrich A. 2004. The cognitive neuroscience of creativity. Psychonomic Bulletin & Review. 11 (6), 1011-1026

34. Halford, G. S., Wilson, W. H., & Phillips, S. 1998. Processing ca- pacity defined by relational complexity: Implications for compara- tive, developmental, and cognitive psychology. Behavioral & Brain Sciences, 21, 723-802.

35. Fuster, J. M. 2000. The prefrontal cortex of the primate: A synopsis. Psychobiology, 28, 125-131.

36. Damasio, A. R. 1994. Descartes' error: Emotion, reason, and the human brain. New York: Putnam.

37. Oleinik A. 2019. What are neural networks not good at? On artificial creativity. Big Data & Society. January–June 2019, 1–13 https://journals.sagepub.com/doi/pdf/10.1177/2053951719839433

38. Wikipedia. 2021a. DeepDream. Date of retrieval March 10, 2021. https://en.wikipedia.org/wiki/DeepDream

39. Computerphile 2016. Deep Dream (Google) – Computerphile. Date of retrieval March 10, 2021. https://www.youtube.com/watch?v=BsSmBPmPeYQ

40. Juliani A. 2016. Are Neural Networks Truly Creative? Date of retrieval April 21, 2021. https://awjuliani.medium.com/are-neural-networks-truly-creative-e713ac963f05

41. Goyal K. 2019. The 7 Types of Artificial Neural Networks ML Engineers Need to Know. Date of retrieval April 28, 2021. https://www.upgrad.com/blog/types-artificial-neural-networks-in-machine-language/

42. Olah C. 2015. Understanding LSTM Networks. Date of retrieval April 28, 2021. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

43. Hansen 2019. Neural Networks: Feedforward and Backpropagation Explained & Optimization. Date of retrieval April 28, 2021. https://mlfromscratch.com/neural-networks-explained/#/

44. Sharp O. and Goodwin R. 2016. Sunspring. Date of retrieval April 29, 2021. https://www.docdroid.net/lCZ2fPA/sunspring-final-pdf#page=2 https://www.youtube.com/watch?v=LY7x2Ihqjmc

45. Wikipedia. 2019b. Sunspring. Date of retrieval April 29, 2021. https://en.wikipedia.org/wiki/Sunspring

46. Cohn J. 2021. The Scientiest of the Holy Ghost. JCMS: Journal of Cinema and Media Studies 60(5) 1-21. Date of retrieval April 29, 2021. https://muse.jhu.edu/article/788639

47. Nicholson C. 2021. A Beginner's Guide to Generative Adversarial Networks (GANs). Date of retrieval May 4, 2021. https://wiki.pathmind.com/generative-adversarial-network-gan

48. Lau J. H., Cohn T., Baldwin T., Brooke J., Hammond A. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. Date of retrieval May 17, 2021. https://www.aclweb.org/anthology/P18-1181.pdf

49. Bamman D., Smith N. 2013. CMU Book Summary Dataset. Date of retrieval May 17, 2021. http://www.cs.cmu.edu/~dbamman/booksummaries.html?fbclid=IwAR1WzoONGCSJhx8nre5KcEu8RMkEVp7mL6S0ghfMlkQnpDiaLq0UYTgXdtU

50. O'Neill M., Voutas A., Lamego I., Zou A. 2016. Literai. Date of retrieval May 25, 2021. https://www.literai.com/

51. Sahay S. 2017. The Adventures of Cyborg Holmes. Date of retrieval May 25, 2021 https://www.literai.com/story/99504aa3f9