A PRACTICAL GUIDE TO CONVOLUTION NEURAL NETWORKS, DECONVOLUTION NEURAL NETWORKS, DEEP CONVOLUTION INVERSE GRAPHIC NETWORKS, AND GENERATIVE ADVERSARIAL NETWORKS.

Daniel Eldon

April 29, 2020

1 INTRODUCTION

1.1 What is artificial intelligence?

To understand what artificial intelligence is, we need to first grasp the idea of what intelligence is, in relation to human psychology. So what is intelligence?According to Essential English Dictionary,intelligence is someone's capacity to comprehend and grasp information or data or the ability to think and understand instead of acting on things by intuition or spontaneously. Let's break the definition even further by defining the word 'Think'. Thinking is the activity of using one's mind to identify and analyze problem and to create an idea for a particular purpose . Therefore, artificial intelligence is a field in computer science that deals with the study of making machines act and think rationally-that is, exhibit intelligence. One of the ways to make machines think rationally is through the concept of machine learning. According to Stuart Russel and Peter Norvig in[4] , machine learning is the means by which a computer adapts to new circumstances,detect and figure out patterns and what they mean. This means that through machine learning, a computer doesn't need to be explicitly programmed to handle future tasks that may arise or complex tasks that would be too difficult to program. There are 3 main strategies used in machine learning and they are;

- Supervised learning-'Monitored' Learning;This model is used where the algorithm used is provided with input patterns and a corresponding matching output patterns. These input-output pairs can be provided by an external 'trainer'
- Unsupervised learning or Self organization- In this model, the algorithm trains an output unit so that it can respond to bundles of patterns inside the input scope. In this model the machine is trained to analyze statistically main characteristics of the input data.
- Reinforced learning- In this learning model, the system does not need labeled data inputs in order to learn and achieve a particular task and does not need tuning to be explicitly corrected

Each of the learning approaches uses different deep learning algorithms to meet its goal . And there are so many of them. Just to mention a few, KNN (K-Nearest Neighbors), Decision Trees, Naïve Bayes, SVM, Find-S, Artificial Neural Networks etc. But out of all those, Artificial Neural Networks (ANN) have proved to be the most promising in terms of its robustness in error handling. In addition, active research is being done in order to improve

neural networks thus leading to the birth of Deep Neural Networks (DNN). These are neural networks with more than 2 hidden layers for input processing. Due to their advanced data processing methodology then can be used to solve complex task such as convolution networks are used for better image recognition. Since their birth different approaches have been proposed for deep neural network, this paper covers the different approaches they use to achieve learning.
This paper covers **Convolution Networks**;a Deep Learning algorithm that take in an input data(typically images), assign salient features (learnable weights and biases) to various aspects/objects in the data and be able to distinguish the proceesed data one from the other. It requires less computaiona power as compared to other Deep Learning algorithm [1]. **Deconvolution Networks,** just as the name suggest, this is a process of removing defects or complications from convolution networks in order to reduce error.
Deep Convolution Inverse Graphics Networks This is a variation of convolution networks aimed at relating graphics representations to images. It uses vision as its inverse graphics[2] . In simple terms its used for intelligent image processing.
Generative Adversarial Networks Introduced by Ian J. Goodfellow,these are models of machine learning algorithms that are able to generate new content from the raw input data they've been fed [3]. For instance, generating images from trained data set i.e. a frowning facial image can be the training dataset while a laughing facial image from the trained data set can be produced.

2 LITERATURE REVIEW

2.1 But What is a Neural Network(ANN)?

To understand what an ANN is. We need first to understand how biological neurons work. According to Shepherd and Koch, the brain is made up of a system of coordinated nerve cells linked together with nerve fibers. These nerve cells are referred to as neurons. These units are the simplest yet complex data processing elements in the human brain. By using multiple neurons concurrently, the brain can process information faster than modern day supercomputers available today[4]. A single neuron consists of the following parts; A soma- which is the cell body, synapse-relaying signals, Dendrites, and an axon-a long fiber for connecting to other dendrites of other neuron. In ANN, the nodes or units is modelled after the soma while the links that connects nodes of a particular layer to the next are modeled after the axon. For information to be perceived as worthy of response stimulus, the synapse of the brain releases complex chemical processes that determine whether an action potential is reached or not. If the action potential is reached, then information is carried to the next neuron hence an appropriate response is triggered. Likewise, in ANN, the chemical processes are emulated using mathematical equations called activation function. Each node has an input weight which the threshold function manipulates. Once the threshold is reached, the signal is passed to the next node. Each node does its local processing. Learning is achieved through adjustments of these numerical weights [4] .

2.2 PERCEPTRON

This is the simplest of neural networks proposed by Frank Rosenblatt in 1958. It's made up of a single neuron with two layers- that is, the input layer and the output layer. This model is also made up of a linear combiner followed by a decision maker function which is used for classification of the inputs [4]. In other words, the main goal of the perceptron is to classify bundles of data into two regions which will then be separated by a hyperplane. The hyperplane is achieved by using a linear separable function. It also has a threshold value which can be used to shift the hyperplane.

The above explanation can be summed up with the formula;
$y = 1$ if $\sum_i^n W_i * X_i \geq \theta$
$y = 0$ if $\sum_i^n W_i * X_i \leq \theta$
Where $\theta isthethreshold$

2.2.1 How the perceptron learns

The perceptron learns using two mathematical rules. That is, Perceptron rule and the Delta rule. The main objective of perceptron learning is to find the right values for the unit weights. Learning is achieved through the following steps:

- Assigning random inputs to the unit or nodes; That is X1, X2. . . Xn.
- Applying update equation to every loop of learning examples given
- If all examples are correctly clustered, terminate the process or else try again.

2.2.2 The perceptron rule

This rule requires that for every training example given say X=(X1, X2. . . Xn) Apply the update input weight using

$W_i = W_i + \triangle W_i$
$Where \triangle W_i = \eta(t - o)x_i$

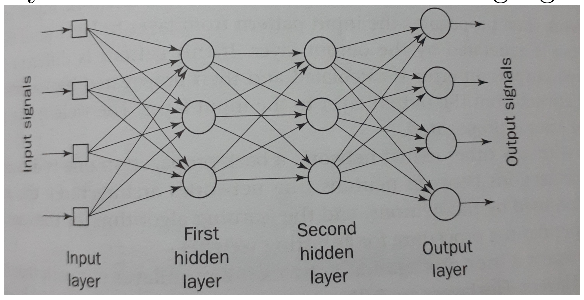
t is the expected output, o is the output produced by the perceptron an n is a small constant which is the learning rate, usually 0.1 This rule only applies when the training examples are linearly separable

2.2.3 Delta Rule

This type of rule handles data that are not linearly separable. It is geared towards establishing the closest match of the target output. It uses gradient descent which searches through the hypothesis space for the best weight vector that bests match the training data. It uses the following equations to minimize error.
$E = \sum_i^n (t - o)^2$
Where the summation is through the examples Given X= (X1, X2…Xn) update all the weight using; $W_i = W_i + \triangle W_i$
Where $\triangle W_i = -\eta E' \frac{(W)}{w_i}$ and η is the learning rate
$E' \frac{(W)}{w_i} = \sum_i^n (t - o)(-x_i)$
$W_i = \sum_i^n (t - o)(x_i)$

2.3 Multilayered Perceptron

This is a feedforward type of ANN with one or two hidden layers[4].It has 3 parts just like the perceptron. That is; input layer, hidden layer and output layer. The input layer is the source of data while the one or more hidden layer is used for processing data using mathematical functions such as the sigmoid function. The inputs are fed layer by layer in a forward direction This mechanism is known as forward propagation. Each hidden layer does its computation then directs it to the next layer. In simple terms, the output of a hidden layer serves as an input to the next hidden layer in the network. The following figure shows a multilayer perceptron:



2.3.1 How it Learns

These networks learn through back-propagation. Each neuron of a layer is connected to each other neuron in the next layer. And this is achieved through the following ways:

- Training dataset is presented to the input layer
- The network then feeds each hidden layer with the input datasets until a particular rhyme is identified by the output layer
- If the pattern identified is different from expected output, an error is calculated and then fed backwards i.e in reverse from the output layer to the input layer
- The weights of the nodes are adjusted as the errors are updated throughout the layers.

In back propagation, a sigmoid function is used which is the activation function: The derivative of this function is easy to compute and also warrants that the unit is within the range of 0 and 1 which makes decision making functions applicable.

2.4 Storkey Learning

Was introduced by Amos Storkey in 1997[5]. It's both local and incremental. It is used to train patterns in Hopfield Networks. Local to mean the rule of learning is refurbished by using the weights of neurons that the current neuron is in between them; that is the weight form either side of the associated neuron Incremental to mean No use of data from previous neurons to facilitate learning. New patterns can be learned independent of the other neurons.

2.5 Hebbian Learning

This is also a type of learning incorporated in Hopfield Networks. It uses this philosophy; neurons that fire together, link together. Neurons that don't sync, fail to link [6]. It also uses both local and incremental rules to achieve data training.

2.6 Contrastive Divergence

This is a type of training algorithm used to train Restricted Bolt Boltzmann Machines. It uses a probabilistic approach to optimize selected weights. It implements Gibbs Sampling inside a gradient descent using the following steps:

- Compute the probabilities of hidden neurons. Sample hidden activation vector(v) from the probability distribution
- Calculate Positive Gradient P which is v*h
- Perform a Gibbs sampling by sampling a reconstruction v' of the visible units from h', then resample the activation h from the result.
- Calculate the negative product N, by calculating the outer product of v' and h'
- Update the weight matrix W using (P-N) *L where L is some learning rate.

3 Convolutional Neural Network (CNN)

A Convolution is a mathematical function performed on two functions i.e f and g that results in a third function showing how the shape of one function is altered by the other[7]. Therefore, in convolutional neural networks, mathematical convolution equations are used to define the activation functions of the network in order to process the unit weights to achieve a desired result. CNN is a type of Deep Neural Networks algorithm which is commonly used to analyze computer vision imagery (processes pixel data). They are also referred to as shift invariant or space invariant artificial neural networks. Apart from visual imagery analysis, they can also be used in natural language processing and also in financial time series data.

3.1 Architecture of CNN

A CNN uses a model similar to multilayer perceptron that has been designed to lower computation power. A convolution layer has the following characteristics;

- Kernels with width and height (hyper-parameters).
- Input and output routes (hyper-parameter).

- Number of input feature map should be same as depth of the filters(input routes)

The layers of a CNN is made up of input layer, output layer and hidden layer(s); that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. Let's analyze these layers one by one:

3.1.1 Input layer (Start Point)

This is the first layer of the network which takes in a tensor with shape;number of images,and dimensions of the image such as width height and image depth. A tensor is any data type that is a vector or matrix and has particular number of dimensions. .

3.1.2 Convolution Layer

In this layer, filters are applied to scan the whole image each pixel at a time. This results into creation of a feature map

3.1.3 Pooling layer or Downsampling

can include both global and local pooling layers to narrow down to the relevant information from prior neurons. They simplify computation by reducing the number of outputs through combination of outputs of unit clusters at one layer into a single unit in the next layer for instance, breaking up the image into features and analyzing them; These features are analyzed in isolation which in turn affect the overall decision of the image. For example, if a number of pixels to be computed falls in the 3 x 3 matrix, pooling can reduce them to a 2 x 2 matrix for easier computation. Local pooling combines small clusters while global pooling works on all the neurons of the convolution layer. In other words, pooling samples the most relevant data from neurons and ignores the less useful ones.

3.1.4 Rectified Unit Layer (ReLU)

This is the layer in which the rectified linear function equation is applied to the weighted inputs. It sets negative values to zero form the activation map therefore eliminating them. This action reduces the size of the output. ReLU function preferred to other functions most of the times because it trains the neural network several times faster with minimal negative effect to generalization accuracy[7]. The equation is as follows:
$f_x(x = max(0, x)$

3.1.5 Fully Connected Layer (FCL)

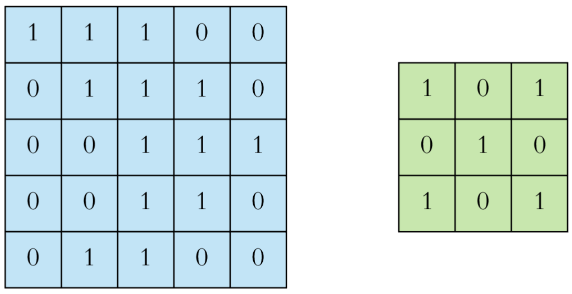
The output of the pooling Layers serve as inputs of this layer. It processes them as inputs and predicts the best labels that describe the image

3.1.6 Loss Layer

This is the final layer of CNN. It shows how far the output is from the expected results. The network can then be readjusted to reduce error rates.

3.1.7 How it works

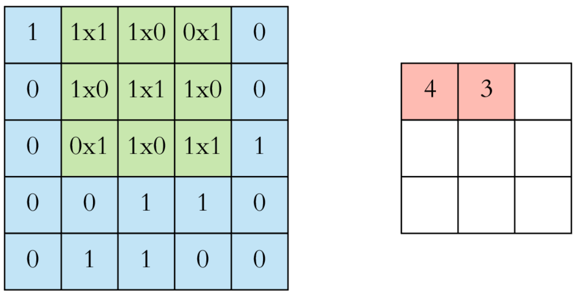
Before we analyze how the above layers process input to achieve a desired output, lets define a few terms;
Filter/ Kernels/Feature Detectors
Is a representation of vector weights with which the input is convolved with. They detect features that help define what really defines an image output or rather they show how close a patch of feature resembles an input. The filter is smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product[9]. These filters are randomly initialized at first. Then they are learned automatically during the training process. They are adjusted to produce better results in a process called fine tuning.
A dot product is the multiplication between a patch of the input and filter, the patch has the same dimension as the filter. The results are then added together resulting into a single value i.e. **Pooling or Downsampling**. This single output that it returns is called a dot product.
Filters **of small size** than the input are intentionally used as it allows multiple multiplication of input arrays to be applied each time by the same filter at different input points. Filters are designed to detect specific features in an input image i.e. where those features are or whether they are present or not. But since these filters go through the input data systematically at different input points, they can detect the features anywhere within the input data matrix. This mechanism is known as **Translation invariance.**
The resulting output formed from the multiplication of filter matrix and the input is a 2-dimension array and is called a **feature map**.
The following image shows the process of filtering which results to a feature



map. chch Input Filter / Kernel

The input and the filter is shown above

A feature map formed after filter input multiplication



chch Input x Filter Feature Map

The convolution layer is by far the most important layer in the network. It is where filter calculations are performed which eventually results into feature maps. This is where pattern detection occurs by aid of filters.

The first step in pattern recognition is determining the number of filters. Filters are first initialized randomly let's say a 3 x 3 filter matrix with random values between 0 and 1

Once the feature maps have been created they are then matched against the presumed prepared data for comparison. The ones which match the most or corresponds the most are then picked as the final output of the network

3.2 Applications of convolution neural networks

3.2.1 Image Recognition and Classification

This is by far the most notable and active application of neural networks. They are used to recognize images in various application systems such as smartphone camera apps, social media description images and face recognition in security systems. In 2012, the MINST database of trained image datasets reported an error rate of 0.23

3.2.2 Video Analysis

By treating space and time in video as identical proportion of the input, various aspects of CNNs have been explored ignored to analyze and recognized the contents of the video [7]

3.2.3 Document analysis

A good application of CNNS is also in analyzing electronic documents. A machine can scan a document and then recognize the handwriting or even the scanned letters in that document. A good example is WPSs image scanner [9]; where you can photograph a document and then convert the writings into an editable typed word document.

3.2.4 Drug Discovery

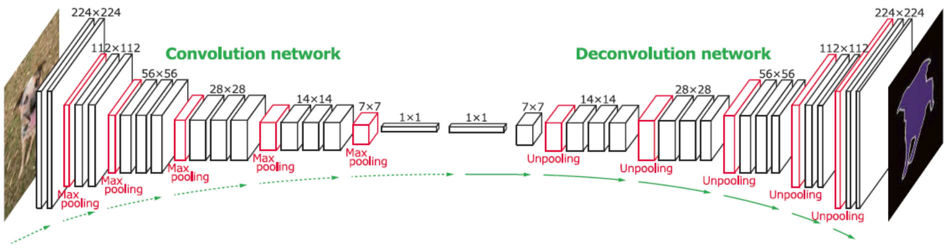
This can be done by training a CNN to predict the interactions of molecular structures of different drugs and analyze their biological proteins. This can be done by training a 3 dimensional molecular interaction of chemicals [7].

4 DECONVOLUTION NEURAL NETWORKS (DECONVNET)

This is not, by any means a different or separate network form the convolution neural network. But it is the reverse of what convolution neural network does. Its main goal is to reconstruct the original image that has been convolved and degraded by the convolution network. It can also be a layer in a convolution network. A deconvolution layer uses the same receptive fields or parameters that was used by the convolution layer. Let's say if a filter of 3 x3 was used then during the deconvolution process, the same filter would be used.

4.1 Architecture of deconvolution neural network

Architectural Image.



chch

The architecture consists of 2 main operations; Unpooling and Deconvolution.

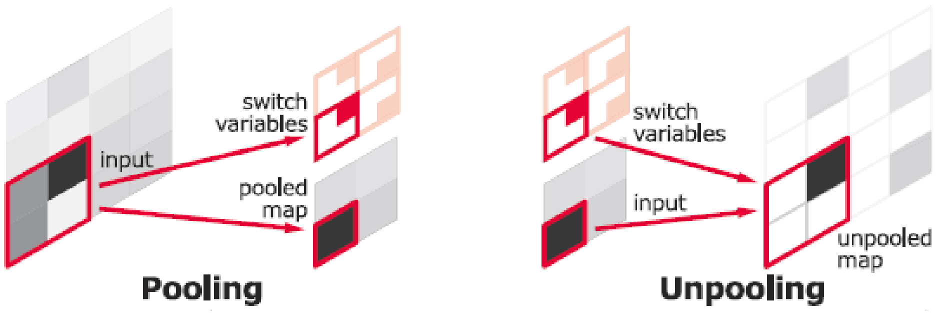
4.2 Unpooling

As earlier mentioned in convolution neural network, pooling is designed to filter out noisy activations by reducing the number of outputs into a single parameter. Whereas in order to reconstruct the original input values, unpooling is employed which reverses the output given by the convolution layer. It reconstructs the original size of the activations [10].Positions of activations during the max pooling process need to be put in check since they are the same positions needed to execute the unpooling process. This is done by applying the operation, [pl, sl] = P(zl) in the pooling layer then reapplying an unpooling operation zl = Usl.pl where s1 is the switch signifying the pooling operation while z1 represent the unpooling operation. This operation was proposed by [11]
It has 2 functions:

- Precise rectification of input through poolng when stacking up more layers
- refurbishment of switches to show the updated values of the feature map

The following image illustrates pooling and unpooling with same variables.

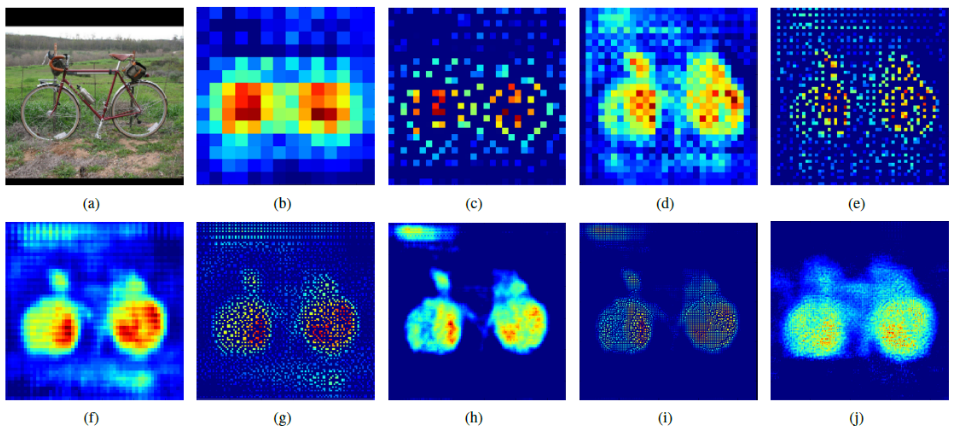
chch



4.3 Deconvolution Layer

Since the output layer of the unpooling layer is large, this layer squeezes the output together or densifies it through convolution-like layers through the application of multiple learned filters. In this layer, unlike the convolution

layer, one single activation output is associated with several outputs.
The learned filters in deconvolution layers correlate with to biases to reconstruct shape of an input object as illustrated in the image of a bicycle below [12]



chch

5 DEEP CONVOLUTION INVERSE GRAPHIC NETWORKS (DC-IGN)

There isn't many research done in incorporating the concept of inverse graphics with neural networks. Though,[13] came up with the idea of incorporating graphics network in convolution networks.
Inverse graphics is an area in computer vision which study the effects of rendering a model in order to produce varying real models of the original one without actually taking another original image of the same.
This is a type of convolution neural network that relates graphical representations to image analysis and processing using the same methods used in convolution neural network. Its composed of two main categorical layers;

- Encoder
- Decoder

In the **Encoder part**, it uses various initial layers to determine the input of the prior layers through the execution various convolution layers, utilizing max pooling then uses subsequent decoder layers to decode with unpooling. Throughout this process, the network uses special variables called scene latent variables which manipulate the image to give an output of different

perspective such as lighting and different angles. In between the encoder and the decoder is a graphics code, Q (zi —x)[13].
The **encoder** is made up of various layers of convolutions followed by max-pooling and the decoder also consists of several layers of unpooling (upsampling using nearest neighbors) prior to the next convolution layers.
[13] Suggests the following steps to be carried for image processing:

- During training, data x is passed through the encoder to produce the posterior approximation Q (zi —x), where zi consists of scene latent variables such as pose, light, texture or shape. According to (DC), in order to learn the networks parameters, gradients are back propagated using stochastic gradient descent following vibrational object function;**log(P(x—zi)) + KL (Q (zi —x) ——P(zi))** for every zi.
- During test, data x can be passed through the encoder to get latents zi. Images can be re-rendered to different viewpoints, lighting conditions, shape variations etc. by just manipulating the appropriate graphics code group (zi),

The resulting output of such a network for example if given a single face image, it should be able to give an output of the same image facing a different angle or with a different shade.

5.1 Applications of DC-IGN

This type of network is currently being researched on and applied in computer vision more so in facial recognition systems where different angles of faces can be analyzed and compared for a match

6 GENERATIVE ADVERSARIAL NETWORKS

This is another young branch of machine learning. Invented in 2014 by Ian Good Fellow. Yan LeCun; One of the lead pioneers in machine learning complimented it as the most fascinating idea in Deep learning in the last decade [14]. It consists of 2 neural networks; The **generative network and the discriminative network**
The generative network generates variables or content randomly while the discriminative network discriminates or evaluates the generated variables. It classifies the as real (from the scope) or fake (generated)This concept is then incorporated in convolution neural networks in order to generate new content such as images.

6.1 How they work

The following processes are executed for the development of a full adversarial neural network;

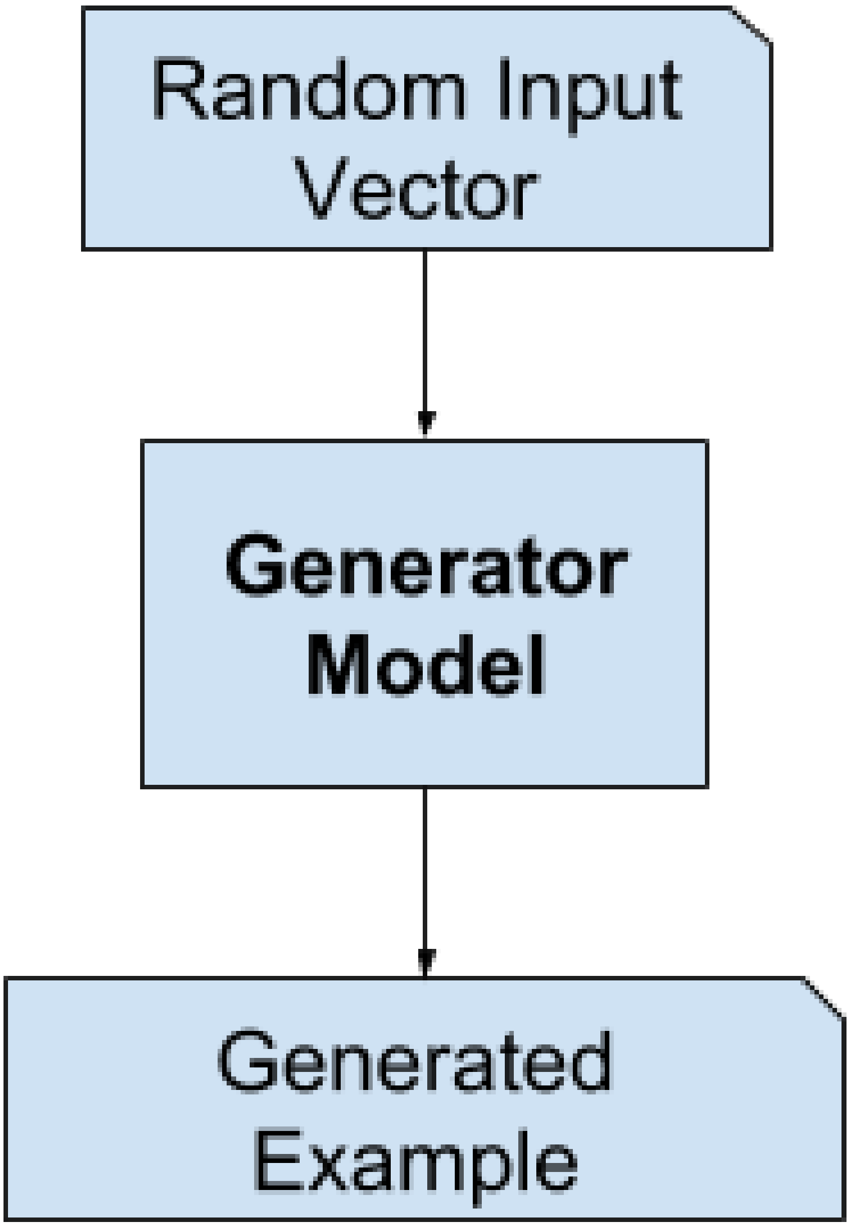
- Random variables generation
- Generator modeling
- Discriminator modeling

6.2 Random Variables Generation

This is where random number generation algorithms are used. Though number generation is not real, they are said to be pseudo random. This means that the variables are generated using a function that make them appear to be random yet they are not. The most relevant approach to generating random numbers is by using a transform function or inverse cumulative **Distribution Function** [3] which are then transformed into vectors which will serve as inputs to the models

6.3 Generator Model

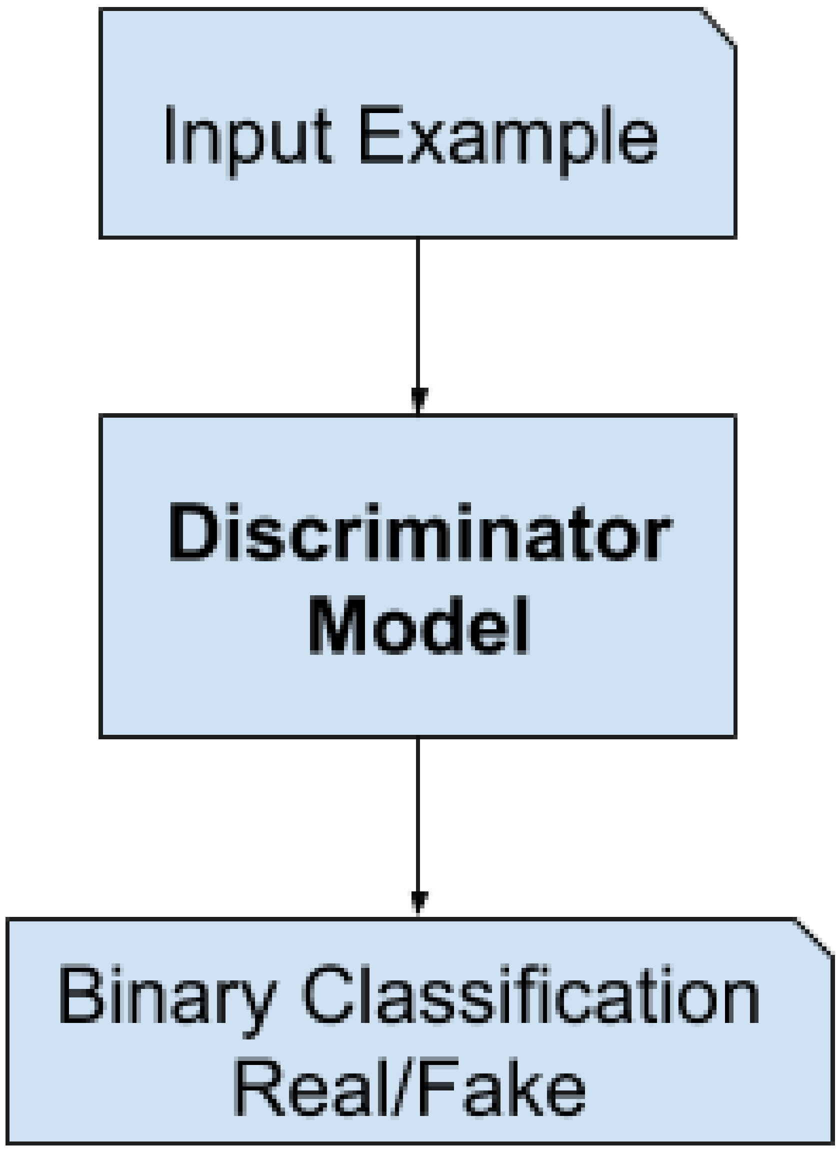
In this model, a set length of random matrix as inputs are taken. A sample is then generated in that scope [14]. The vector/matrix is then used as a base or seed for the generative process. It is then trained, then points in the multidimensional vector space correlate with points in the problem scope forming a compressed representation of the data distribution (latent Variables). Latent variables, or hidden variables, are those variables that are important for a scope but are not directly observable [14]. The generator model then takes this variables as input, process them new different outputs [14].
The figure below illustrates generator modelling



chch

6.4 Discriminator modeling

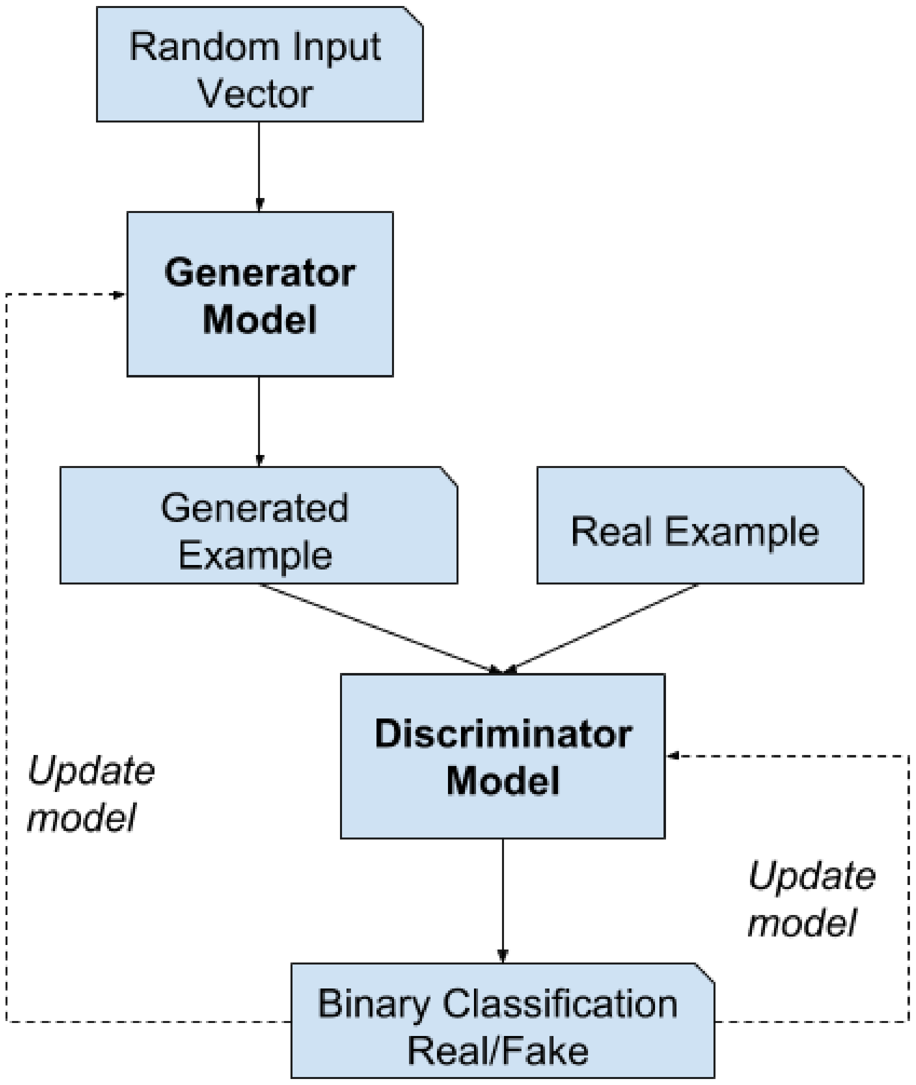
This part focuses on classifying the inputs and labels them as generated or real. It takes these inputs from the domain and then anticipate a binary class label as actual or counterfeit. Examples classified as actual are from the trained datasets while the ones classified as counterfeit are from the generator model.
The following image illustrates the working of the discriminator model



chch
Once the training has been done, constant update is done on the discriminator model in the following rounds so as to get better at classifying and labeling the models. The same is also applied to the generator model so as to get better at 'fooling' the discriminator model.

6.5 Overall Architectural representation of GANs

The image below illustrates the general architecture of the GAN



chch

6.6 GANs and convolution networks

Because GANs work with image data, they have been incorporated into CNNs where some layers can serve as generators while other layers function to discriminate outputs generated by the discriminating layers. This is due to the efficiency in the processing capability of CNNs [14]. The CNNs can be modified to produce two distinct part such that one part is to generate input variables and the other part for discrimination.

6.7 Summary of how Generative Adversarial Network Operates

- Generation of random numbers and input vector creation, i.e. image generation
- The synthesized image is the conveyed into the discriminator together with other images taken from the real life objects
- The discriminator then analyses the image and groups them according to which it perceives to be fake and which one is authentic. The classifier is binary which means a range from 0 for fakeness while 1 for authenticity

6.8 Limitations of GANs

In high resolution image processing, GANs do not preserve the distinguishing features of a person. They often produce distortions in the image making it appear funny or unreal[15]

6.9 Applications of GANs

Some of the most common application of GANs include;

6.9.1 Generation of images from descriptive text

In 2016, Han Zhang, et al. demonstrated how real life images can be generated using their stackGAN model. This model could generate simple objects such as flowers and birds[16]

6.9.2 Generation of realistic photographs

Tero Karras, et al. in their 2017 in their paper titled ‘Progressive Growing of GANs for improved Quality, Stability and Variation’ that realistic images can be generated at a very remarkable accuracy[16]. They generated a set of celebrity faces which appeared to be indistinguishable from real images

6.9.3 Face Frontal View Generation

Inverse graphics networks is not the only machine learning application of generating varying angles. GANs are also capable of doing this; and Rui Huang, et al. demonstrated this in their paper titled, ‘Beyond Face Rotation; Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis’ that images taken from an angel can be reconstructed to a frontal view and still resemble the initial original sided image [16]

7 CONCLUSION

This paper has covered the main areas involving the working of Convolution neural networks, deconvolution networks, generative adversarial networks an inverse graphics network. Their application in real life scenario has been illustrated and a comparison of how they work and are interrelated also illustrated. For instance, throughout the paper it has been shown that all the other networks; GANs, IGNs and DeCONVNET have their foundation from the convolutional networks. They all work on using the same principle as convolutions do and one can conclude that they are an innovation of convolutional networks.

References

[1] S.Saha,"A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," Towards Data Science,15 12 2018. [Online].Available: https://towardsdatascience.com/a-comprehensiveguide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.[Accessed 21 2 2020].

[2] Techopedia,"Deep Convolutional Inverse Graphics Network (DC-IGN)," Techopedia, [Online]. Available:https://www.techopedia.com/de nition/33291/deep-convolutionalinverse-graphics-network-dc-ign. [Accessed 21 2 2020].

[3] J.Rocca,"Understanding Generative Adversarial Networks(GANs)," Towards Data Science, 1 09 2019.[Online].Available:https://towardsdatascience.com/understanding-generative-adversarialnetworks-gans-cd6e4651a29.[Accessed 21 2 2020].

[4] M. Negnevitsky, "Artificial Intelligence," in Artificial Intelligence, Essex, Pearson Education limited, 2011, p. 166.

[5] Wikipedia,"Restricted Boltzmann machine," Wikipedia,[Online].Available:https://n.wikipedia.org [Accessed 20 22020]

[6] Wikipedia,"Hebbian Theory,"Wikipedia,[Online].Available:https://en.wikipedia.org/wiki/Hel

[7] Wikipedia,"Convolution," Wikipedia, [Online].Available:https://en.wikipedia.org/wiki/Convolution.[Accessed 20 3 2020]

[8] S.Saha,"A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way," Towards Data Science,15 December 2018.[Online].Available: https://towardsdatascience.com.[Accessed 20 April 2020

[9] Flat World Solutions,"7 Applications of Convolutional Neural Networks,"[Online].Available:www.atworldsolutions.com. [Accessed 20 April 2020].

[10] S.H.Tsang,"Review:DeconvNet—Unpooling Layer(Semantic Segmentation)," Towards Data Science,8 October 2018.[Online].Available:https://towardsdatascience.com/review-deconvnet-unpooling-layersemantic-segmentation.[Accessed 20 April 2020].

[11] H.Noh,S.Hong and B.Han,"Learning Deconvolution Network for Semantic Segmentation,"[Online].Available:https://arxiv.org/pdf 1505.04366.pdf.

[12] Matthew D.Zeiler,G.W.Taylor and R.Fergus,"Adaptive Deconvolutional Networks for Mid and High Level Feature Learning," [Online].Available:http://citeseerx.ist.psu.edu/viewdoc/download?oi=10.1.1.849.3679rep=rep1ty

[13] T.D.Kulkarni,W.W.P.Kohli and J.B.Tenenbaum4,"Deep Convolutional Inverse Graphics Network,"[Online].Available:https://arxiv.org/abs/1503.03167.

[14] J.Brownlee,"A Gentle Introduction to Generative Adversarial Networks(GANs),"[Online].Available: https://machinelearningmastery.com/whatare-generative-adversarial-networks-gans/.

[15] S.Arora,"Promise and Limitations of Generative Adversarial Nets (GANs)," 22 December 2017. [Online]. Available: https://simons.berkeley.edu/news/research-vignette-promise-andlimitations-generative-adversarial-nets-gans.

[16] J.Brownlee,"18 Impressive Applications of Generative Adversarial Networks(GANs),"[Online]. Available:https://machinelearningmastery.com