TARP DIGITAL ASSIGNEMNT 3

Name: Om Ashish Mishra

Registration Number: 16BCE0789

Slot: TF2

3. Explain the Software/Hardware technologies selected for the identified for the real world problem?

We mainly used the following software technologies for our problem:

GANS

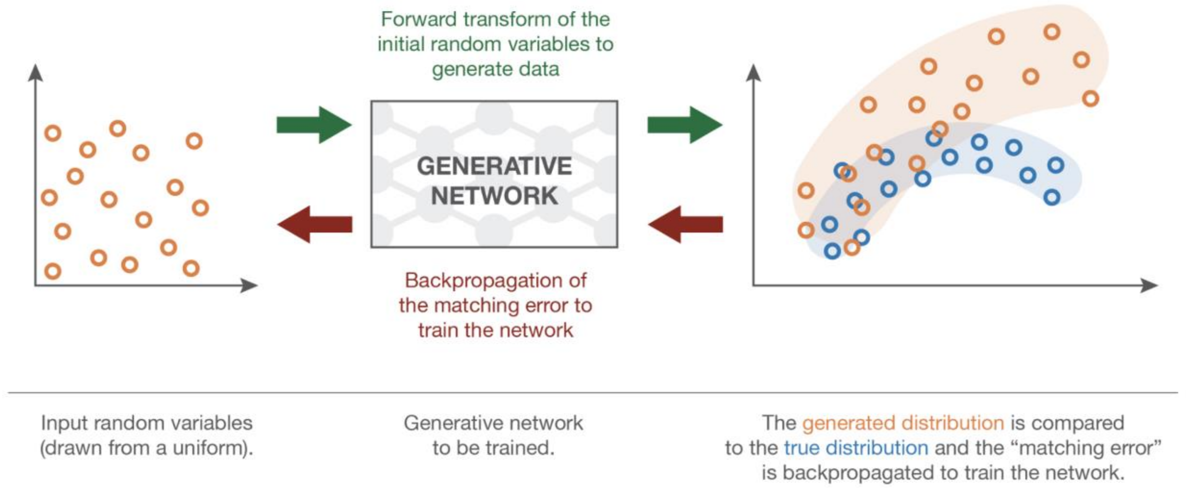
A **generative adversarial network** (**GAN**) is a class of machine learning systems. Two neural networks contest with each other in a zero-sum gameframework. This technique can generate photographs that look at least superficially authentic to human observers, having many realistic characteristics. It is a form of unsupervised learning.

Generative Adversarial Networks belong to the set of generative models. It means that they are able to produce / to generate new content. To illustrate this notion of “generative models”, we can take a look at some well known examples of results obtained with GANs.

Illustration of GANs abilities by Ian Goodfellow and co-authors. These are samples generated by Generative Adversarial Networks after training on two datasets: MNIST and TFD. For both, the rightmost column contains true data that are the nearest from the direct neighboring generated samples. This shows us that the produced data are really generated and not only memorised by the network.

he *generative* network generates candidates while the *discriminative* network evaluates them. The contest operates in terms of data distributions. Typically, the generative network learns to map from a latent space to a data distribution of interest, while the discriminative network distinguishes candidates produced by the generator from the true data distribution. The generative network's training objective is to increase the error rate of the discriminative network (i.e., "fool" the discriminator network by producing novel candidates that the discriminator thinks are not synthesized (are part of the true data distribution).

A known dataset serves as the initial training data for the discriminator. Training it involves presenting it with samples from the training dataset, until it achieves acceptable accuracy. The generator trains based on whether it succeeds in fooling the discriminator. Typically the generator is seeded with randomized input that is sampled from a predefined latent space (e.g. a multivariate normal distribution). Thereafter, candidates synthesized by the generator are evaluated by the discriminator. Backpropagation is applied in both networks[5] so that the generator produces better images, while the discriminator becomes more skilled at flagging synthetic images. The generator is typically a deconvolutional neural network, and the discriminator is a convolutional neural network.



Generative Matching Networks take simple random inputs, generate new data, directly compare the distribution of the generated data to the distribution of the true data and backpropagate the matching error to train the network.

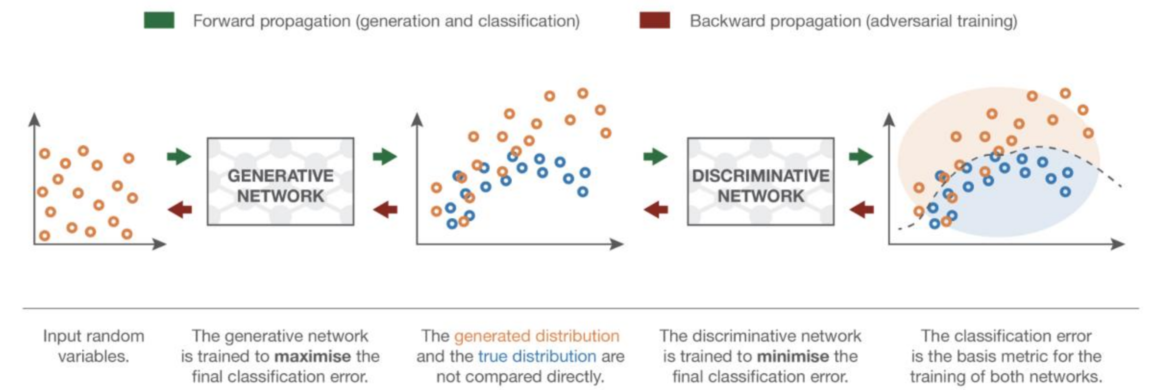
Applications of GANS

GAN applications have increased rapidly. GANs that produce photorealistic images can be used to visualize interior/industrial design, shoes, bags and clothing items or items for computer games' scenes. Such networks were reported to be used by Facebook.

GANs can model patterns of motion in video, reconstruct 3D models of objects from images and improve astronomical images.

GANs can be used to age face photographs to show how an individual's appearance might change with age.

In 2018, GANs reached the video game modding community, as a method of up-scaling low resolution 2D textures in old video games by recreating them in 4k or higher resolutions via image training, and then down-sampling them to fit the game's native resolution (with results resembling the "Super Sampling" anti-aliasing method). With proper training, GANs provide a clearer and sharper 2D texture image magnitudes higher in quality than the original, while fully retaining the original's level of details, colors, etc. Known examples of extensive GAN usage include Final Fantasy VIII, Final Fantasy IX, Resident Evil REmake HD Remaster, Max Payne. GAN 2D texture modding can be applied only to PC game releases.



Generative Adversarial Networks representation. The generator takes simple random variables as inputs and generate new data. The discriminator takes "true" and "generated" data and try to discriminate them, building a classifier. The goal of the generator is to fool the discriminator (increase the classification error by mixing up as much as possible generated data with true data) and the goal of the discriminator is to distinguish between true and generated data.

Convolutional Neural Networks (CNN)

In deep learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural networks, most commonly applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

Design

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

Description of the process as a convolution in neural networks is by convention. Mathematically it is a cross-correlation rather than a convolution (although cross-correlation is a related operation). This only has significance for the indices in the matrix, and thus which weights are placed at which index.

Convolutional

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.

Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for *each* neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation.

Pooling

Convolutional networks may include local or global pooling layers. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. M*ax pooling* uses the maximum value from each of a cluster of neurons at the prior layer. A*verage pooling* uses the average value from each of a cluster of neurons at the prior layer.

Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

Receptive field

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its *receptive field*. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

Weights

Each neuron in a neural network computes an output value by applying some function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is specified by a vector of weights and a bias (typically real numbers). Learning in a neural network progresses by making incremental adjustments to the biases and weights. The vector of weights and the bias are called a *filter* and represents some feature of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons share the same filter. This reduces memory footprint because a single bias and a single vector of weights is used across all receptive fields sharing that filter, rather than each receptive field having its own bias and vector of weights.

Applications of CNN

Image recognition

CNNs are often used in image recognition systems. In 2012 an error rate of 0.23 percent on the MNIST database was reported. Another paper on using CNN for image classification reported that the learning process was "surprisingly fast"; in the same paper, the best published results as of 2011 were achieved in the MNIST database and the NORB database. Subsequently, a similar CNN called AlexNet won the ImageNet Large Scale Visual Recognition Challenge 2012.

When applied to facial recognition, CNNs achieved a large decrease in error rate. Another paper reported a 97.6 percent recognition rate on "5,600 still images of more than 10 subjects".CNNs were used to assess video quality in an objective way after manual training; the resulting system had a very low root mean square error.

Video analysis

Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream. Long short-term memory (LSTM) recurrent units are typically incorporated after the CNN to account for inter-frame or inter-clip dependencies. Unsupervised learning schemes for training spatio-temporal features have been introduced, based on Convolutional Gated Restricted Boltzmann Machines and Independent Subspace Analysis.

Natural language processing

CNNs have also been explored for natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

Drug discovery

CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments. In 2015, Atomwise introduced AtomNet, the first deep learning neural network for structure-based rational drug design. The system trains directly on 3-dimensional representations of chemical interactions. Similar to how image recognition networks learn to compose smaller, spatially proximate features into larger, complex structures, AtomNet discovers chemical features, such as aromaticity, sp^3 carbons and hydrogen bonding. Subsequently, AtomNet was used to predict novel candidate biomolecules for multiple disease targets, most notably treatments for the Ebola virus and multiple sclerosis.

Histogram Equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensitiescan be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

In scientific imaging where spatial correlation is more important than intensity of signal (such as separating DNA fragments of quantized length), the small signal to noise ratio usually hampers visual detection.

Histogram equalization often produces unrealistic effects in photographs; however it is very useful for scientific images like thermal, satellite or x-ray images, often the same class of images to which one would apply false-color. Also histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color depth. For example, if applied to 8-bit image displayed with 8-bit gray-scale palette it will further reduce color depth (number of unique shades of gray) of the image. Histogram equalization will work the best when applied to images with much higher color depth than palette size, like continuous data or 16-bit gray-scale images.

Histogram Thresholding

In image processing, the **balanced histogram thresholding method** (BHT),is a very simple method used for automatic image thresholding. Like Otsu's Method and the **Iterative Selection Thresholding Method**, this is a histogram based thresholding method. This approach assumes that the image is divided in two main classes: The **background** and the **foreground**. The **BHT** method tries to find the optimum threshold level that divides the histogram in two classes.

This method *weighs* the histogram, checks which of the two sides is heavier, and removes weight from the heavier side until it becomes the lighter. It repeats the same operation until the edges of the weighing scale meet.

Given its simplicity, this method is a good choice as a first approach when presenting the subject of *automatic image thresholding*.