29-04-2020

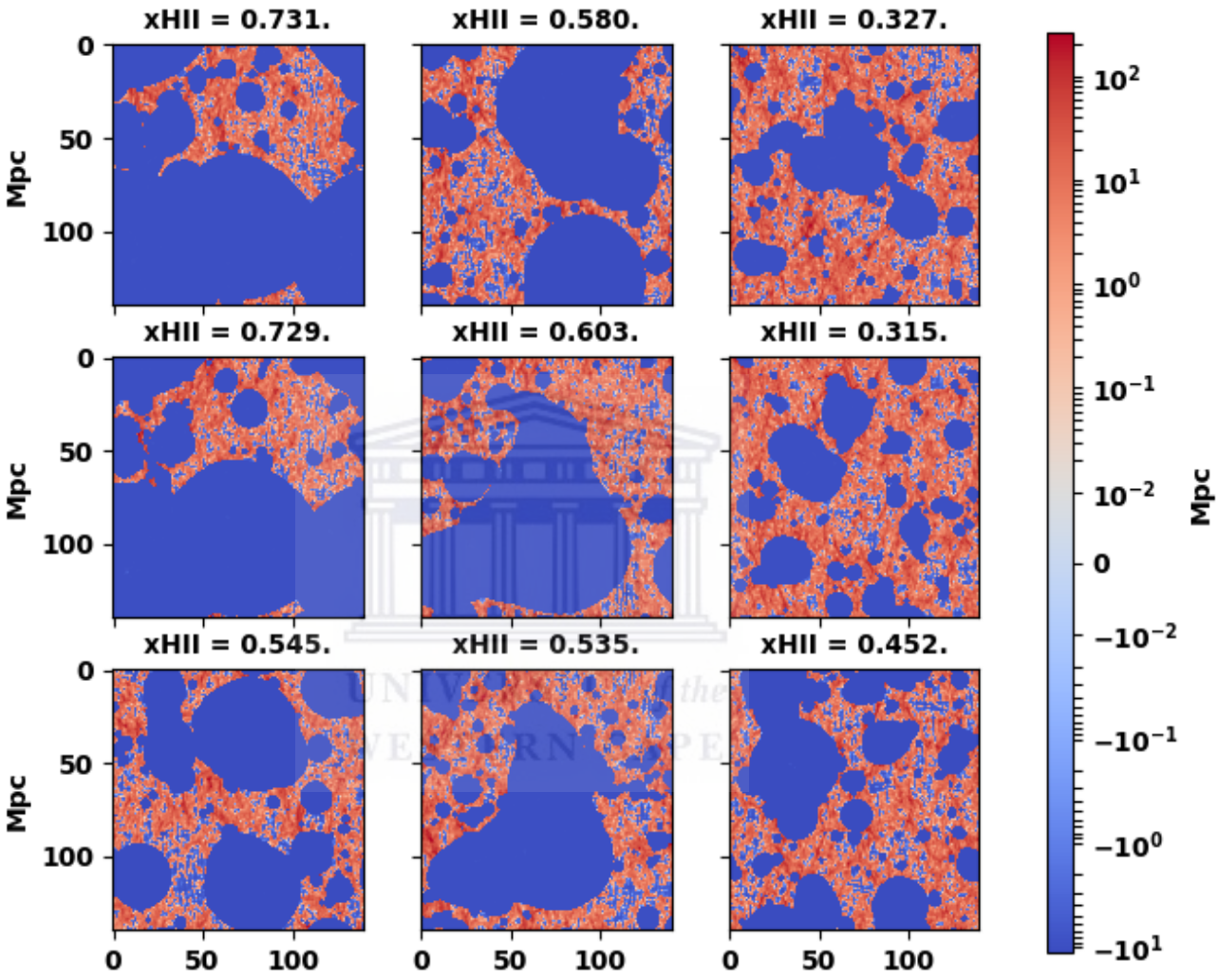

FIGURE 2.1: Randomly selected 21cm images at $z = 8$ from different simulations with their corresponding ionization fraction($x_{\rm HII}$). Ionized and neutral regions are represented by the blue and red colors on the colorbar respectively. The images with higher ionization fraction from the left going across the right with a decrease in ionization fraction for each row are presented.

2.2.1 21CM NOISE SIMULATIONS

We follow closely the recipe presented in Hassan et al. (2018), which we briefly review here, to account for various instrumental effects in order to create our mock 21cm images based on a SKA1-Low instrumental configuration. These instrumental effects involve accounting for the finite angular resolution of the experiment, the foreground cleaning and the thermal noise. The pipeline consist of three parts:

UV-SAMPLING

The baseline distribution of a given 21cm array controls its angular resolution to observe specific modes in the direction perpendicular to the observation line of sight. The comoving wavenumber in the directions perpendicular to our line of sight is defined as:

$$k_{\perp} = \frac{2\pi u_{\perp}}{D_c}, \tag{2.2}$$

where D_c is the comoving distance to the observation redshift z and $u_{\perp}$ is expressed in terms of $u - v$ baseline length units in the $u - v$ plane. Figure 2.2 shows the UV coverage which is a representation of the total number of baseline to observe a given pixel in uv plane (uv plan is described in the introduction section). We only focus on SKA1-Low array at $z = 8$ ($\nu = 157.8$ MHz) and is computed using the 21CMSENSE (Pober et al., 2013,2014)[1], for the SKA1-Low antennae distribution as summarized in Table 2.1 which then represents the total number of the baselines that observe a given uv pixel. The blue part representing zero pixel shows the signal modes that lie beyond the experiment uv-sampling.

We follow the steps below to adjust the 21cm brightness temperature boxes in order to account for an instrument's angular resolution:

- Using the antenna distribution of a 21cm experiment at a given frequency (redshift), we compute its UV coverage.

[1] a package for calculating the expected sensitivities of 21cm experiments (https://github.com/jpober/21cmSense).

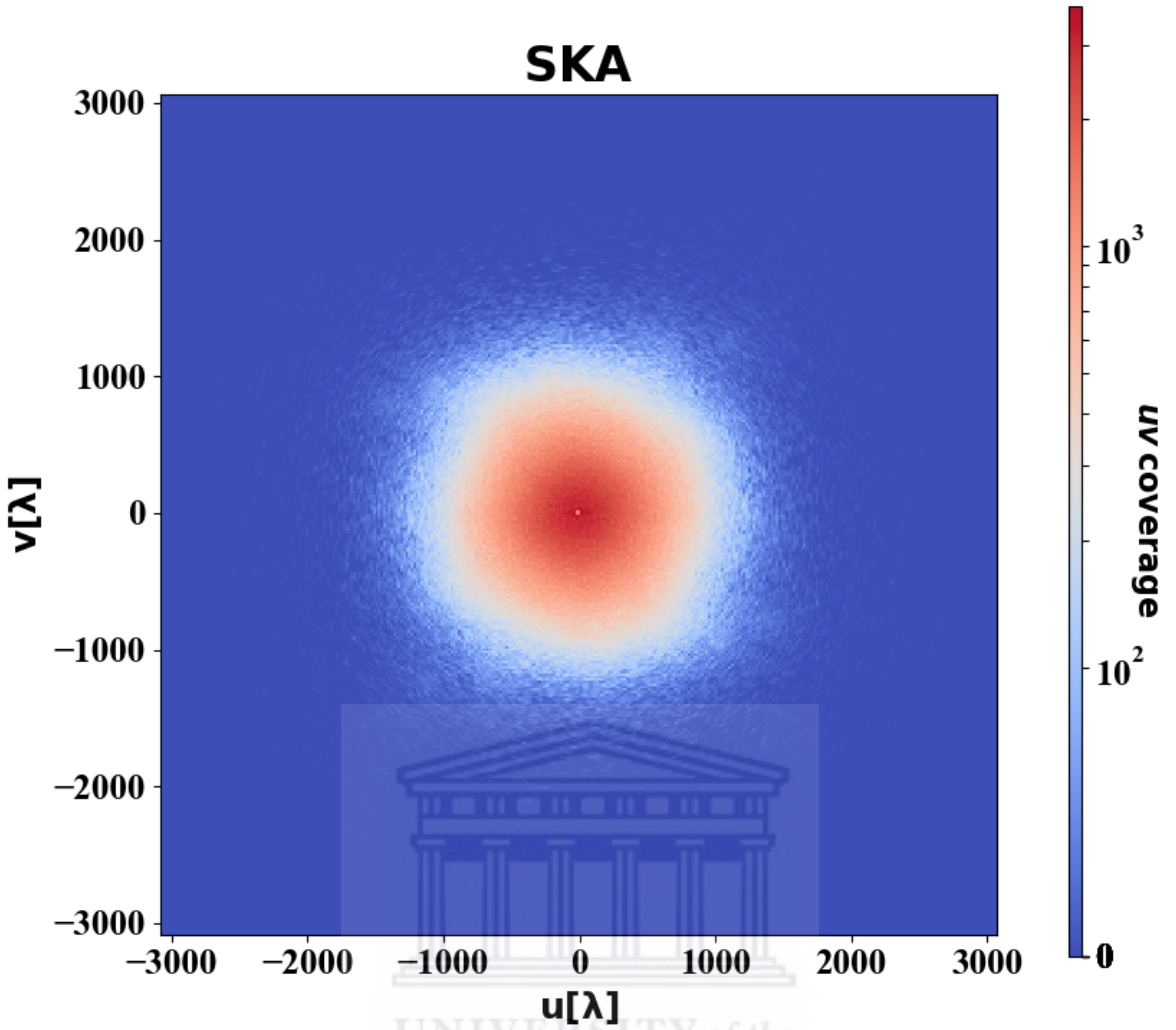

FIGURE 2.2: The uv coverage for SKA1-Low array at $z = 8$ obtained from 21CMSENSE. The blue colour represent the zero coverage which correspond to non-measured signal modes which are beyond the experiment layout angular resolution.

- The u and v coordinates are then converted into their corresponding k_x and k_y modes which are then used to calculate $k_\perp$ using the equation : $k_\perp = \sqrt{k_x^2 + k_y^2}$.

- We Fourier transform the 21cm signal simulation box and set to zero the signal for $k_\perp$ modes that correspond to zero uv-coverage (which are outside the SKA1-Low angular scales).

- We finally inverse Fourier back to the real space to obtain the angular limited 21cm simulation boxes.

Foreground cleaning

There are several ways to deal with foreground contamination, and in our work, we use a foreground avoidance approach.

To deal with foregrounds in our simulated 21cm images, we simply zero out all Fourier modes that are contaminated by foregrounds. These modes fall within the so-called foreground wedge in the $k_{\parallel}$-$k_{\perp}$ plane, with a slope defined as:

$$m = \frac{DH_0E(z)\sin\theta}{c(1+z)}, \tag{2.3}$$

where $E(z) = \sqrt{\Omega_m(1+z)^3 + \Omega_\Lambda}$ which is the field of view, H_0 is the Hubble parameter, c is the speed of light, and θ is the beam angle. The foreground wedge essentially comes from the fact that foregrounds have an anisotropic footprint between the line-of-sight and transverse directions. We quote all the wedge slope values for SKA1-Low at each redshift of interest on Table 2.1. Larger values of the slope m contaminate the signal strongly, since a greater amount of the signal is removed due to foregrounds.

Thermal Noise

Taking into account the thermal noise of the instrument into our data is the final step to complete our process of making the 21cm maps more realistic. We define the flux error by the following equation:

$$\sqrt{\langle |N|^2 \rangle} = \frac{2k_{\rm B}T_{\rm sys}}{A\sqrt{\Delta\nu t_{\rm int}}}, \tag{2.4}$$

where t_{int} here is the integration time to observe a single visibility at a frequency resolution $\Delta\nu$, and $k_{\rm B}$ is the Boltzmann constant. The total system temperature T_{sys} and other parameters are summarized in Table 2.1. This parameters are for an SKA1-Low1 like system although the numbers are not completely fixed yet. Having generated the thermal noise in 2D grid using the

Array design	866 compact core
Station diameter, D [m]	35
Station area, A [m^2]	$962\left(\frac{110}{\nu[\mathrm{MHz}]}\right)^2$
system temperature [K] (= $T_{\mathrm{sky}} + T_{\mathrm{rcvr}}$)	1.1 T_{sky} + 40
Total observation time t_{int} [h]	3500
Frequency resolution $\Delta\nu$ [kHz]	50
Redshift	10, 9, 8 ,7
Frequency [MHz]	129 , 142, 158, 178
Resolution [arcmin]	1.37, 1.24, 1.12, 0.99
Field of view θ^2 [deg^2]	14.30, 11.81, 9.57, 7.56
Default wedge slope m, Equation (2.3)	0.27, 0.23, 0.19, 0.15

TABLE 2.1: Summary of our assumed SKA1-Low design.

above equation in the Fourier space, we further suppress the noise by the amount of the uv-coverage N_{uv}. We finally inverse Fourier transform the noise grid and add it to the uv-sampled and foreground filtered 21cm signal map to form our mock 21cm map.

For a more detailed description and application of the method mentioned above, we refer the reader to Hassan et al. (2018). As an example (Figure 2.3), we show a summary of the above 21cm noise simulations at $z = 8$ for the future SKA1-Low-like observations.

Figure 2.3 shows the images before and after implementation of the steps used to account for instrumental effects of the SKA1-Low as described above. The 1st column shows the original signal (also see Figure 2.1), we notice that the bubbles are clearly visible, which gives us the impression that the network will be able to extract the ionization fraction without any difficulties. After accounting for the angular resolution where the resultant slices shown in the 2nd column, we observe that the prominent structures are still visible, this indicates the higher sensitivity of the SKA1-Low. As already discussed under the noise pipeline description, we clarify that we account for angular resolution directly to the slices that are from the 21cm boxes already foreground treated with $m = 0.19$ (wedge slope value at $z = 8$, see Table 2.1). We then observe the effect of the thermal noise on the slice treated with angular resolution and foreground avoidance. We add these slices with thermal noise of the SKA1-Low to produce our mock images. We expect the network to be still able to extract the ionization fraction but not as efficiently as with dataset before adding noise. The prominent features are still visible hence we are confident

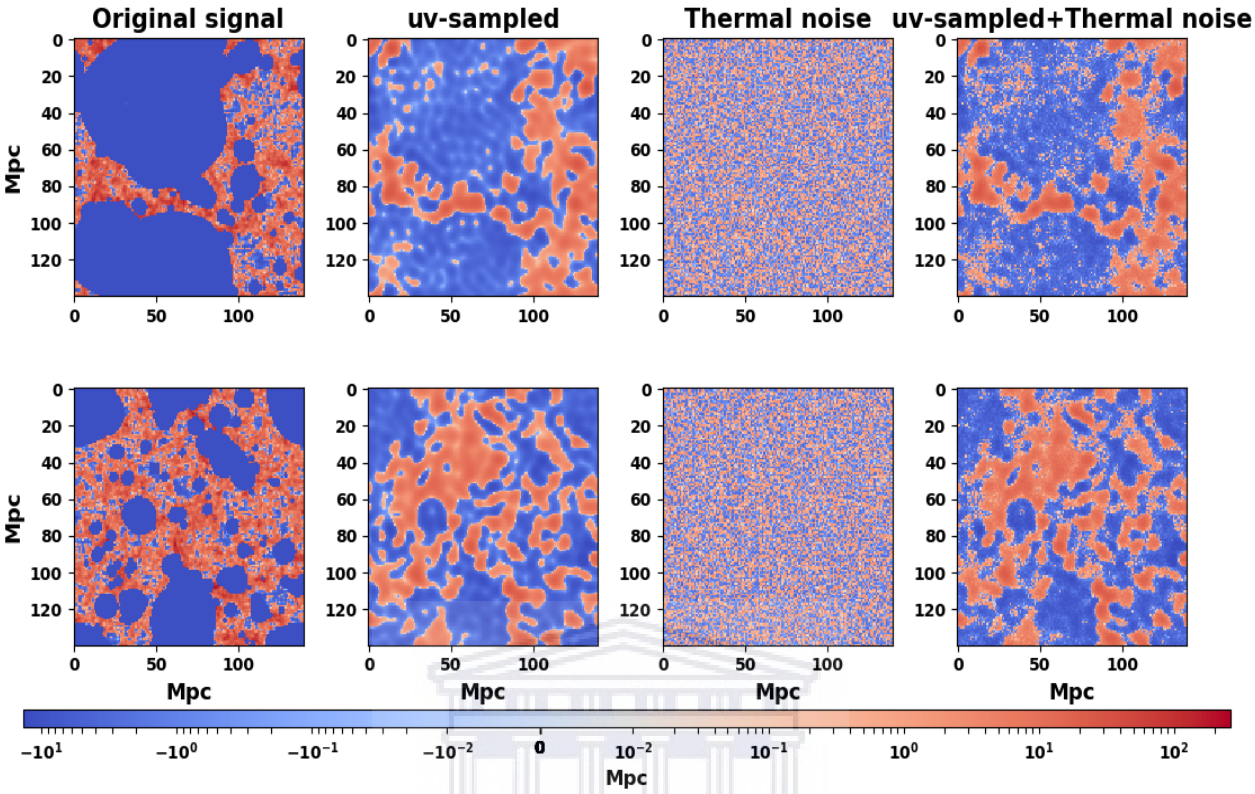

FIGURE 2.3: The 1st column represents slices of 21cm signal from the 21cm box, on the 2nd column we show slices which are angular resolution adjusted taken from a foreground treated 3D boxes, 3rd column shows the thermal noise, and the last column shows the addition of the 2nd column slices with the 3rd column slices. Note that all slices are for $z = 8$ respectively. The colorbar measurement units is Mpc.

that the network should still be able to recover the neutral fraction. This is an optimistic noise realization for future SKA1-Low observations, and we here attempt to show a proof-of-concept for using the 21cm maps to constrain the neutral fractions as an alternative approach to using the power spectrum. Our analysis can be easily refined when future 21cm dataset become available in the next decade.

UNIVERSITY of the
WESTERN CAPE

3 Machine Learning

3.1 Overview

The use of statistical techniques to give a computer the ability to learn from data without being explicitly programmed is titled as Machine learning defined by Arthur Samuel in 1959. The idea of machine learning is widely applicable in different fields including astrophysics, finances, biology, and more.

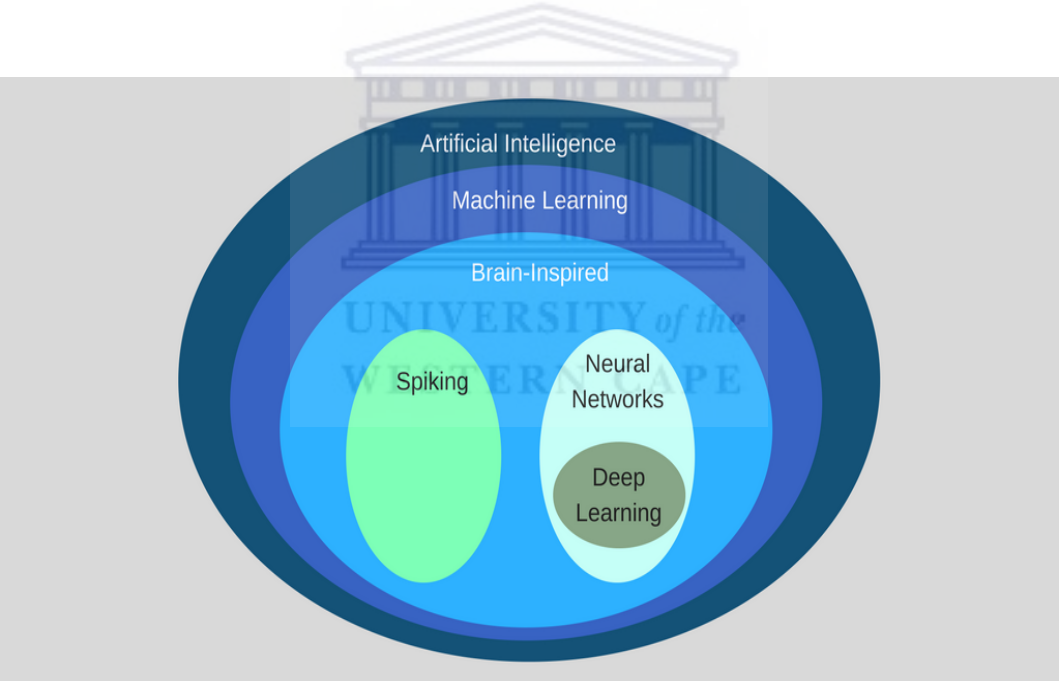

FIGURE 3.1: Relationship between Artificial Intelligence, Machine Learning and Deep Learning(Rachit Kumar Agrawal at Udacity India)

The figure above demonstrates the relationship between Artificial Intelligence (AI), Machine learning (ML) and Deep learning (DL). According to the diagram, AI is the primary field having machine learning as its large sub-field. In 1955 John McCarthy defined AI as "the science and engineering of making intelligent machines that have the ability to achieve goals as humans do". Within ML sub-field, we have an area called neural networks widely known as artificial neural

networks (ANN) which is regarded as the brain-inspired computation. A neuron is regarded as one of the vital element of our brain, so the network of neurons connected forms the basis of all the decisions made based on the various information gathered and this is precisely how ANN works. DL is an area which falls under the domain of neural networks, and it can be constructed from neural networks by designing a neural network that has more than three layers.

Unlike other traditional ML methods widely used, Neural Networks do not require explicitly constructing features from the input data but can easily and directly learn patterns from the raw input signal by using a complex combination of neurons organized in nested layers that can learn a non-linear function of the input data (Herbel et al., 2018). When a neural network is having more than three layers (input layer, at least more than one hidden layer and output layer) is then called Deep Neural Network. Unlike standard machine learning algorithms that break problems down into parts and solve them individually, deep learning solves the problem from end to end, and it draws its power of best performance in dealing with images by taking advantage of the spatial structure of the inputs (for a comprehensive review see Rawat and Wang 2017).

In the following sections we broadly introduce machine learning covering its key concepts, overview of the types of neural networks, introduction to classical neural network and convolutional neural network (CNN), the architecture of the network used in this work together with the components used to build the network. We also discuss how the training of dataset was generated, data preparation and other data manipulation methods applied with the aim to improve the results.

3.2 Introduction to Machine Learning

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E (Mitchell, 1997). Machine learning is a data-driven technological tool that is widely used to turn information into knowledge. The algorithms learn from data (experience E) in order to perform

a specific job (task T) and the more it learns how to perform a job, it will do better (performance measure P). We give the overview of the two classes of machine learning, i.e supervised and unsupervised learning. Each class has differing approaches while following the same underlying process and theory. We closed this section by highlighting some of the common challenges when using machine learning algorithms.

3.2.1 Supervised vs. Unsupervised

Supervised learning is based on training the model to be able to map input (x) data into output (Y) which can be applied to the new data with unknown outputs. The model is provided with the labelled data for training purposes and be given the new unlabelled data to predict its corresponding labels. The aim of machine learning model is to generalize from the training data to any data on the same domain as the training. Generalization refers to how well the concepts learned by a model apply to data not seen by the model during training. The two common types of supervised learning are classification and regression. In classification, the output variable can be nominal, categorical, or numerical. There are 2 types of classification, i.e binary and multi-classification. In binary classification, there are only two possible outcomes while in multi-classification there are more outcomes. In regression, the output variable is a real value, or continuous variables. The commonly used algorithms in supervised learning are nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), and Neural Networks. In our work we use CNN which falls under neural networks (Møller, 1993).

In unsupervised learning, the model is provided with unlabelled input data leaving it to the algorithm to determine the data patterns (e.g similarities and differences) on its own. The two common types of unsupervised learning are clustering and association. In clustering, the model divides the data into groups with different characteristics. In association, the model attempts to discover the new set of association rules that describe the data. This can provide some interesting relationships between variables in our dataset. This type of learning can provide an insight into the data by restructuring the data and extract new features. The commonly used

algorithms in unsupervised learning are Hierarchical clustering, K-means clustering, K-NN (k nearest neighbors), Principal Component Analysis, Singular Value Decomposition, Independent Component Analysis (Coates, Ng, and Lee, 2011).

A feature is an independent or depend measurable property or characteristic observed from a phenomenon. Features are used to train machine learning models. It is common to start with feature extraction before training the model. Feature extraction is a process of dimensionality reduction whereby a set of raw data is reduced to many manageable groups for processing. Most of supervised algorithms require features for the training purposes, while neural networks perform the feature extraction and utilization of features automatically during training. This makes CNN more convenient and powerful. Unsupervised algorithms do not require any features for the training purposes.

3.2.2 Overfitting and Imbalance

The most common challenges faced when using machine learning algorithms are over-fitting and data imbalance. Overfitting refers to when the model learns the desired details and noise of dataset too well and generalize based on both of them. This will impact the performance of the model negatively when is applied on different dataset. This leads us to also discuss underfitting which is when the model neither model training data nor generalize to new data. The above mentioned challenges can be solved as follows, k-fold cross validation where the model is trained and tested k-times using different subset of training data, and also holding back the validation dataset to evaluate the model at the end when the training is completely done.

The concept of data imbalance which often happens in classification where there are disproportionate ratio in each class, plays an important role on the performance of the model. We can deal with data imbalance by a) resampling techniques called, oversampling where one adds copies of existing data which helps when the data is not enough and undersampling where one removes some of the data from majority classes, this helps when there is more data, b) generate synthetic samples where data augmentation is used to generate more unrealistic data. Data

augmentation is a process of increasing data by transforming the existing data using rotation, cropping, padding, and horizontal flipping (Krawczyk, 2016).

3.3 Neural Networks

There are different types of Neural Networks available. For example:

- Feedforward Neural Network – Artificial Neuron: is used to approximate a given function by a flow of information taking place in the forward direction (Svozil, Kvasnicka, and Pospichal, 1997).
- Radial basis function Neural Network (RBFNs): are universal approximators and a special type of feed-forward neural networks with radial basis functions used as activation functions (Chen, Cowan, and Grant, 1991).
- Kohonen Self Organizing Neural Network: is a dimensionality reduction type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space of the training samples, called a map (Kohonen, 1990).
- Recurrent Neural Network (RNN): is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence (Pineda, 1987).
- Convolutional Neural Network (CNN): is a class of deep learning that captures local features from the input raw data through learnable kernels (LeCun and Bengio, 1995).

The above mentioned neural networks work differently and can be chosen depending on the type of problem one is trying to solve. The successful application of CNN in solving different problems encourages and motivates us to further apply CNN in solving our problem of extraction of ionization fraction from 21cm maps. The widely known deep convolutional neural network known as Visual Geometry Group Network (VGGNet) showed case a significant capability of CNN. The VGGNet was developed and trained by Oxford VGG which focused on object recognition and image classification where the network achieved outstanding performance on

the ImageNet dataset (Simonyan and Zisserman, 2014).

We further note its successful application into astronomy. The estimation of cosmological parameters demonstrated by Ravanbakhsh et al. (2016) and Gillet et al. (2018) reflect the capability and the reliability of CNN in astronomy. We also see the classification of sources responsible for reionization of the early universe presented by Hassan et al. (2018) and classification of supernova which plays a crucial role in obtaining cosmological constraints (e.g Lochner et al., 2016) being an excellent tool to classification of astronomical data. This is also driven by the big data concept arose because of the availability of large astronomical data and how relevant and helpful the classification can be in order to understand the universe. He, Wang, and Yan (2018) also demonstrates the prediction of structure formation of the universe. With more example of the successful application of machine learning into astronomy which can be found in Ntampaka et al. (2019), we confidently consider using CNN to build our ionization fraction estimator. However, the work of Shimabukuro and Semelin (2017) on the extraction of EoR parameters (i.e. the ionizing efficiency, the minimum viral temperature of halos producing ionizing photons, and the mean free path of ionizing photons through the IGM) from the 21cm power spectrum showed the capability of ANN. We then decide to attempt building our parameter estimator using classical neural networks famously referred to as ANN.

In this section we cover the overview of classical neural network, convolutional neural network, activation function used, and batch normalization technique.

3.3.1 Classical Neural Network

Classical neural network became our starting point in this work. It consists of fully connected layers where each layer is made up of a set of neurons. The term 'fully-connected' taken from the idea that each neuron in one layer is connected fully to all neurons in the previous layer.
Figure 3.2 shows an artificial neuron, it is a unit with inputs and output. Each neuron accepts the input x and computes the output y. The neuron performs a linear combination of the components x with weights and biases and passes the information to a non-linear function called activation

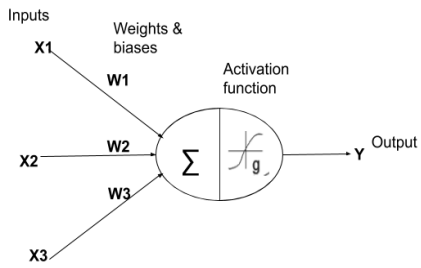

FIGURE 3.2: An example of a neuron.

function. The weights are calculated using the backpropagation method. This can be expressed mathematically by the following equation:

$$y^j = g(\sum_i^N w_i^j x_i + b^i)$$

where y^j is the j^{th} output, N is the number of inputs in each layer, w_i^j is the j^{th} weighting of the i^{th} input x_i, b^j is the j^{th} bias and 'g' represent the activation function. There are many available activation functions to choose from, and in our work, we use Rectified Linear Unit (ReLU) activation function which will be discussed in the next section. To demonstrate the use of the above equation we consider Figure 3.2 where the output will be as follows:

$$Y = g(x_1w_1 + x_2w_2 + x_3w_3)$$

Figure 3.3 shows ANN architecture made up of input layer, hidden layer and output layer. The output from one neuron become the input for next one or several other neurons. The advantage of classical neural networks is that they require less computational power compared to other neural networks. A significant challenge in using classical neural networks is that the deeper networks contain large number of parameters, which might lead to overfitting.

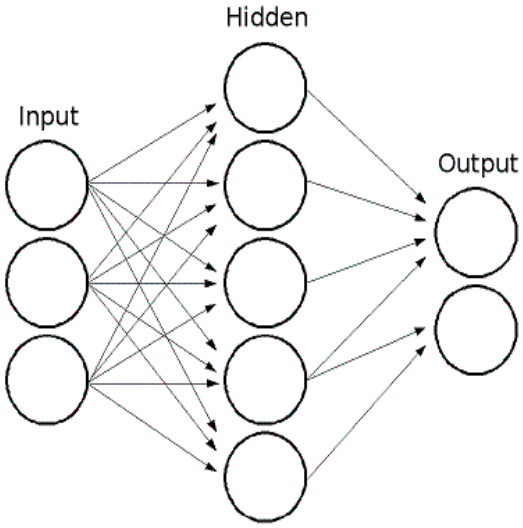

FIGURE 3.3: An example of an a simple architecture of ANN

The fully connected layers accept a vector as an input. The neurons are fully connected to the activation from the previous layer, and the training from these layers is performed through forward and backpropagation. In forward propagation, the training happens in a forward direction and compare the results obtained with the real values in order to get an error through loss function which will be covered in the next sections. During the backpropagation, the error gradient is backwards propagated to update the weight parameters. Ultimately the error is minimized as much as possible by differentiating the loss function with respect to the weights using gradient descent. This process continues until the set of weights and biases that make the error small are found.

3.3.2 CONVOLUTIONAL NEURAL NETWORK

As briefly discussed above, CNNs are a special kind of multi-layer neural networks. They are designed to be able to recognize directly from pixel images the visual patterns with very minimal pre-processing. Their ability to successfully capture the spatial and temporal dependencies in an image through the application of learned filters show their capability of dealing with complex images. In our data, the network is trained to identify the ionized bubbles and predict the number

indicating the ionized fraction of that particular image.

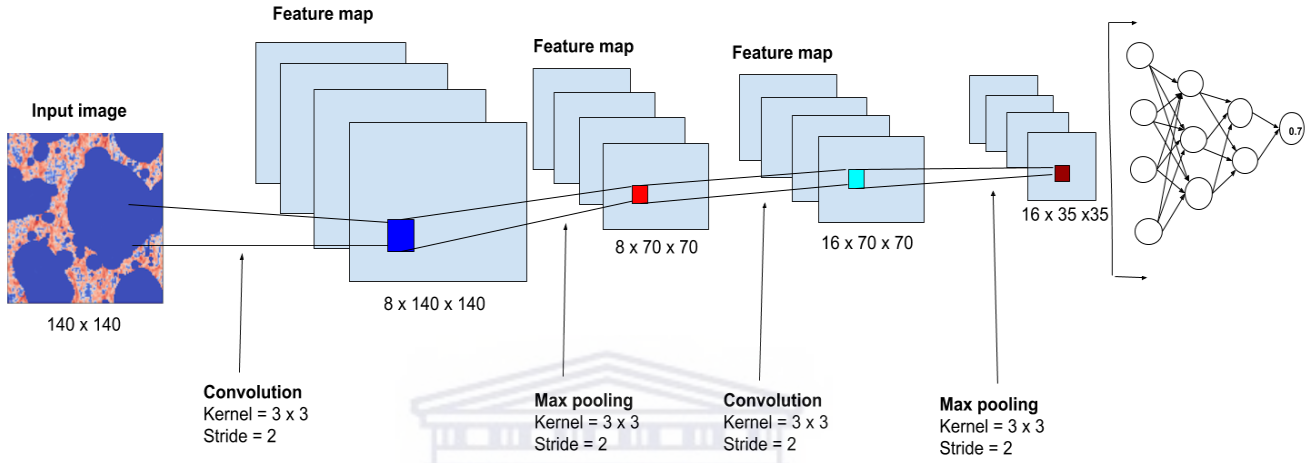

FIGURE 3.4: An example of an a simple architecture of CNN with 2 convolutional layers, and 3 fully connected layers

Figure 3.4 shows an example of a simple CNN architecture. We will use this figure for demonstration purpose on how CNN works. Unlike other neural networks, the connection of neurons in CNN from layer to the next is only to a small region of the next layer. Figure 3.4 shows the image of size 140×140 as an input image. The first convolutional layer uses multiple convolution filters or kernels that runs over the input image and compute a dot product. This is demonstrated in Figure 3.5 where the first image represent the input pixel image. The filter of size 3×3 is used to compute an element wise multiplication between the image pixel values corresponding to the filter's size and sum the answers up. This then gives us a single value for the feature map cell as shown on the right hand side. In this exaple, the convolutional filter moves two steps each time, this is called stride. This process continues until the feature map cells are all filled with the values. The resulting feature map takes the size of the filter/kernel. The output feature map passes through a ReLu activation function (following on the next section) in order to make it non-linear. From Figure 3.4 we further see that after the convolutional layer,

max pooling is applied to the feature map which takes maximum value in each window. The max-pooling will also be covered in section 3.4.

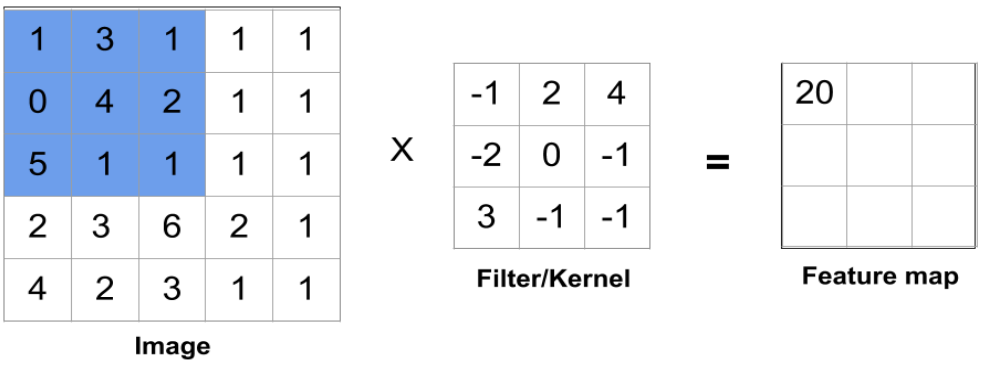

FIGURE 3.5: A dot product of a filter of 3×3 with an image of 5×5 and their feature map

Traditionally, when using CNN, the architecture ends with fully connected layers. As a results, figure 3.3 shows that the result from last convolutional layer feeds into a fully connected neural network structure that drives the final decision. In summary, the convolutional layers are serving the purpose of feature extraction and fully connected layers are then used to make a decision using feature maps received from the convolutional layers.

3.3.3 Activation Function

The following diagram shows the behaviour of our activation function known as a Rectified Linear Unit (ReLU). This graph is represented by the following function:

$$g(x) = max(0, x), \tag{3.1}$$

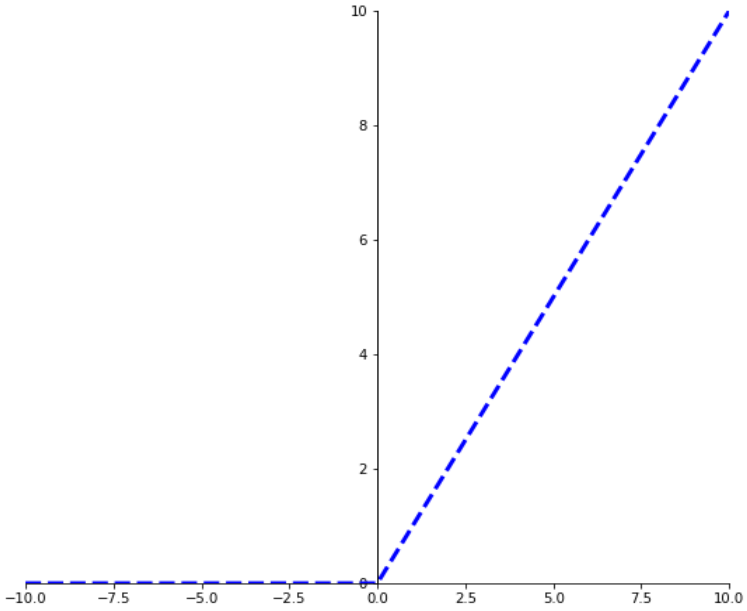

FIGURE 3.6: The graph showing behavior of activation function ReLU, this indicates that the output ranges from 0 to infinity.

The equation, as shown above, gives an output x if x is positive, otherwise the output will be 0 meaning it zero out all negative inputs since they are considered less significant. ReLU is a non-linear function in nature, and also maintains the non-linearity even when it combines with any other functions. It is less computationally expensive than other activation functions like tanh and sigmoid, and this is because of its simple mathematical operations involved. This point becomes more relevant and essential when one is building a deep neural network. It also removes the less critical features and keeps the more critical prominent features during the training of the network, which can be more useful in decision making for the output layer.

3.3.4 BATCH NORMALIZATION

During the training of a deep neural network, the input to each layer is affected by parameters in all the input layers. This means that even a small change to the network is amplified down through the network which of course will affect the performance of the network. This leads to change in the input distribution to internal layers of the deep network known as an internal covariant shift. To address the issue of an internal covariant shift, we use the method known as

Batch Normalization (BN). The effect of internal covariant shift can be counteracted by setting lower learning rate and careful parameter initialization. This idea might slow down the training of the network. Batch normalization provides any layer in a neural network with inputs that are zero mean/unit variance and allows us to use much higher learning rates and be less careful about initialization (Ioffe and Szegedy, 2015). Batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.

The following equations represent the first algorithm of batchnorm implementation. A detailed explanation can be found in (Ioffe and Szegedy, 2015).

$$\mu_\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{3.2}$$

$$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2, \tag{3.3}$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}, \tag{3.4}$$

$$y_i \leftarrow \gamma\widehat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i), \tag{3.5}$$

During the batch normalization algorithm, the input is the values of x over the mini-batch : $\beta = \{x_1 ... m\}$ and the output : $\{y_i = BN_{\gamma,\beta}(x_i)\}$ where γ and β are free parameters which scale and shift the normalized value $\hat{x}^{(k)}$ according to the following equation:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}, \tag{3.6}$$

Equation (3.2) represents the mini-batch mean (μ_β) calculated from values of x in a batch. The value m represent the number of values in a batch. The mean is then used to calculate the mini-batch variance (σ_β^2) using Eq (3.3). The normalization of values of x is performed by Eq (3.4) and the output values are obtained by Eq (3.5).

The normalization is applied to each activation independently. Some of the advantages of applying BN are as follows:

- allows use of saturating nonlinearities and higher learning rates which will speed up the learning process.
- reduces internal covariant shift.
- reduces the dependence of gradients on the scale of the parameters or their initial values.
- regularizes the model and reduces the need for dropout, photometric distortions, local response normalization and other regularization techniques.

Given the above benefits of using BN, we decided to apply it in our network. The sequence of its application is shown in the next sections on a table which shows the summary of the architecture of the network.

UNIVERSITY of the
WESTERN CAPE

4 Building the Neural Network

We explain the procedure that was followed in order to build our network used in this work. It is important to note that in any basic or complex neural network, the following components, i.e. activation function, loss function, and optimization algorithm play a vital role in efficiency and effectiveness of training a model to produce the desired results. Hence we will carefully consider explaining how and why we chose the above components of our network. The following sections outline the building method and the use of our network explicitly. We discuss how the network architecture was built in Section 3.3.1, and the discussion of the training loop, testing and evaluation of our model will be in Section 3.3.2.

4.0.1 Architecture

Building the most efficient architecture to assist in processing and extracting the relevant information from the given data is one of the challenging steps in machine learning. Architecture building depends mostly on the type of problem one is trying to solve. Here we describe two architectures, the one makes use of fully-connected layers that we refer to as “ANN” and the other is a standard CNN that uses convolutional layers that we refer to as “CNN". Our best performing ANN has the following architecture.

Classical neural network consists of three types of layers being, input layer which is the first layer of the network. Setting up the number of neurons/nodes determined by the shape of the data. On the case of our data, the input shape of our data is 2D (140×140) image which is flattened into a 1D vector array. The second layer type is hidden layers where the number of neurons varies on each layer. The final layer is the output layer which is always determined by the chosen model configuration. For our work, the network runs in a regression model where this layer returns a value which must then represent our ionization fraction. The ANN architecture in

TABLE 4.1: The summary of ANN architecture used

Layer (type)	Number of neurons
Input layer	$140 \times 140 \times 1$
Fully connected hidden layer 1	252
relu	
dropout	
Fully connected hidden layer 2	128
relu	
dropout	
Fully connected hidden layer 3	64
relu	
dropout	
Fully connected hidden layer 4	32
relu	
dropout	
Fully connected hidden layer 5	16
relu	
dropout	
Fully connected hidden layer 6	8
relu	
dropout	
Output layer	1

our work consist of input 1 input layer, 7 hidden layers and 1 output layer as indicated in Table 4.1.

We furthermore discuss how we have built a convolutional neural network which contains convolutional layers making it different from the classical neural network. Convolutional layers extract different features from the input data by convolving each input image with learned weights of two-dimensional kernels. Table 4.2 summarizes architecture of our best performing network.

The first column represents the layer type. Our network comprises of 2 convolutional layers, each convolution layer is followed by the batch normalization plus ReLu as an activation function which is discussed above. Batch normalization technique also played an essential role in improving our results.

There are two different types of poolings to can be applied into the neural network, i.e, maximum and average based pooling (Hijazi, Kumar, and Rowen, 2015). Max-pooling select maximum pixel values in a batch while average pooling select the average of all pixel values in a batch.

TABLE 4.2: The summary of CNN architecture used

Layer (type)	kernel/filter size	data dimension
Input	-	$140 \times 140 \times 1$
conv2d-1	5×5	$140 \times 140 \times 8$
batch-norm + relu		
max-pooling	5×5 /2	$70 \times 70 \times 8$
conv2d-2	5×5	$70 \times 70 \times 16$
batch-norm + relu		
max-pooling	5×5 /2	$35 \times 35 \times 16$
flattening	-	19600
fully connected	-	128
batch-norm + relu		
dropout		
fully connected	-	32
batch-norm + relu		
dropout		
Output	-	1

Pooling can be performed between layers. While max-pooling layer extracts essential features like edges, generally, the pooling layer reduces the resolution of the previous feature maps and also produces invariance to any small transformation like translation and dilation. Pooling splits the inputs into disjoint regions with a size of $(R \times R)$ to produce one output from each region (Aldhaheri and Lee, 2017). In our network we apply max-pooling after the convolutional layers has extracted features from the images and normalization of the output from the convolutional layers took place. The pooling compresses features to a lower fidelity by taking the maximum activation in a block which also reduces the size of the image into half. The size of the output image from the max-pooling layer depends on the filter size and strides. The filter size of 2 and stride of 2 is applied.

After the second max-pooling layer, we flatten the image and pass it to the dense layers with batch norm and ReLU as an activation function in each step ending by applying drop out which simply can be regarded as a temporary deactivation or ignoring of some neurons of the network depending on how useful their analysis is. This is to avoid what is called "over-fitting" which is an error that occurs when a network is too closely fit a limited set of input samples. The output layer gives out one number representing the ionization fractionx_{HII} of each input of 21 cm map.

4.0.2 Training dataset

We discuss the free parameters used in running the simulations to generate our dataset. We present in Table 4.3 the main free parameters with their range.

Table 4.3: Parameters range of values used

Parameter	Range
f_{esc}	0.01-0.5
$log10(A_{ion})$	39.0-40.0
C_{ion}	0.0-1.0

These parameters include:

- f_{esc} - is the amount of ionizing photons that can escape the galaxy or ionizing source.
- A_{ion} - is the ionizing emissivity amplitude, which scales the amount of ionizing emissivity (R_{ion}) equally across the halo mass range at a given redshift.
- C_{ion} - is ionizing emissivity-halo mass power dependence, which quantifies the $R_{ion} - M_h$ slope.

It is crucial to consider a range of values of different parameters to make sure that our data is diverse. Using SimFast21, we run 1000 reionization simulations with a box size of 70 Mpc and number of cells N=140^3, which results in a resolution of 0.5 Mpc. The number of simulation is motivated by Schmit and Pritchard (2018) where they showed that only 100 model evaluations are sufficient to learn 3 parameters. In our case we aim to learn one parameter, therefore 1000 simulations should be sufficient to ensure the learning. Each simulation is obtained from different realizations of the initial density field fluctuations through a random change of the seed number, different set of astrophysical parameters, changing the photon escape fraction $f_{\rm esc}$, A_{ion}, $C_{\rm ion}$ as indicated on Table 4.3. We also vary $R_{\rm ion}$-$\rm M_h$ and $\rm R_{ion}$ redshift evolution index $D_{\rm ion}$=(0-2), and different set of cosmological parameters, changing the matter density parameter Ω_m=(0.2-0.4), the Hubble constant H_0=(60-80), and the matter fluctuation amplitude σ_8=(0.7-0.9). The range considered for the astrophysical parameters is motivated from our previous MCMC estimates to match the simulation to several reionization observables (Hassan et al., 2017), and those of the cosmology is inspired by the recent parameters estimates from the Planck Collaboration 2018.

This ensures that our training sample contains a very different set of 21cm maps that accounts self-consistently for the cosmic variance.

TABLE 4.4: Summary of the runs from the code

Number of Simulations	1000
Number of redshift boxes per simulation	41
Number of Slices per box	50
Redshift range	7-10

After running 1000 simulations producing 41 boxes in each simulation at different range from 10 to 7, we then take 50 slices in each box to increase the dataset which is summarized in Table 4.4. We apply the cutoff of $0.05 < x_{\rm HII} < 0.95$ to our dataset in order to address the problem of imbalanced data. This then explains why the number of datasets per redshift is slightly different. Before the cutoff, we have noticed that for all redshifts, both edges, i.e. where the neutral fraction is in range 0.0 to 0.04 and also 0.95 to 1.0 the number of slices is very large, which considerably affects the performance of the network. The outcome turned to favour our desire for the excellent performance of the network.

TABLE 4.5: Dataset splits

redshift (z)	7	8	9	10
Number of training sets	10920	11280	11760	11760
Number of validation sets	1365	1410	1470	1470
Number of test sets	1365	1410	1470	1470

Table 4.5 shows the number of training, validation and testing datasets for each redshift. Deep neural networks rely on several things for its performance, and the number of datasets is one of the vital one. This means that the higher the number of training sets, the better the network performs and generalizes. From experimentation, we have found that the network requires at least 10,000 training dataset to obtain robust results. We noticed that the results remain approximately the same for datasets including more than 10,000 images, whereas lower accuracy has been found for smaller datasets. This is observed on a single redshift case only. In the following Section, we shall outline how the data was loaded and prepared to be used.

Data Optimization

Understanding of dataset is a crucial step in machine learning. As indicated in the previous section, our data is in the form of 2D images with a resolution of 0.5 Mpc. We further select an equal number of boxes in each redshift and take an equal number of slices in each bin of ionization fraction. Our ten bins range from 0.05 to 0.95, we then select roughly 1000 images per bin and end up having roughly 10 000 images in the whole range of ionization fraction for each redshift.

The slices from 21 cm boxes are 2D arrays, and their corresponding labels are the ionization fraction, which is just a single number ranging between 0 and 1. We shuffle data during training and split it into training, testing and validation.

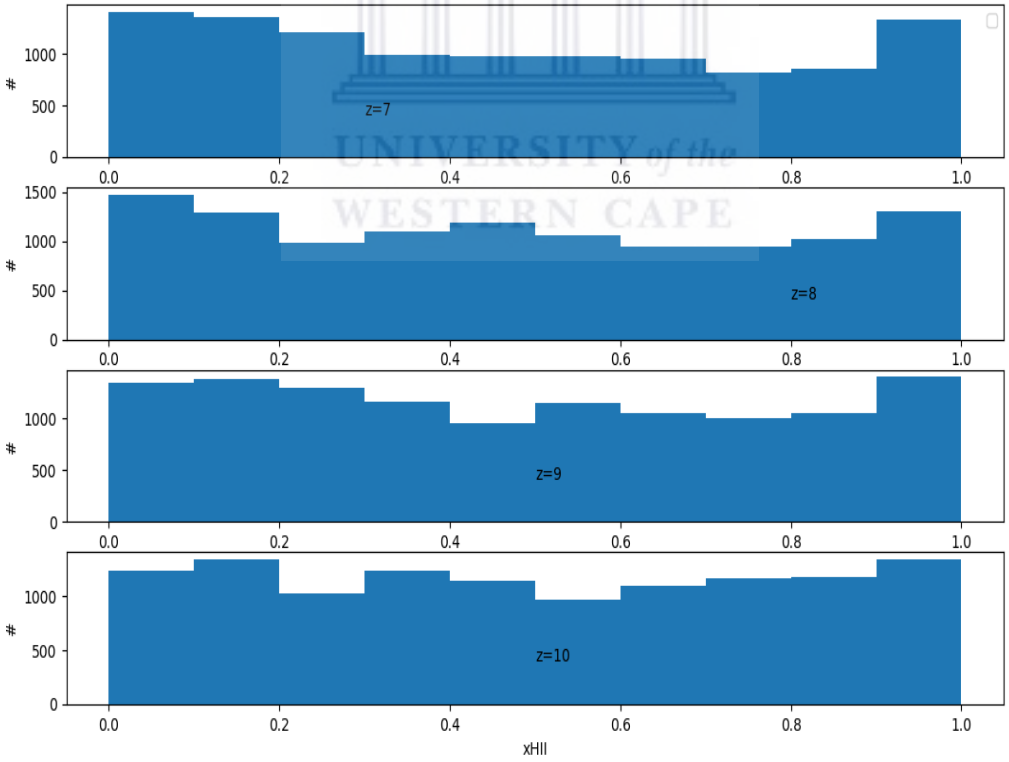

Figure 4.1: The histogram showing dataset at each redshift, the x-axis representing ionization fractions divided into 10 bins and y-axis showing the number of data points.

Figure 4.1 represents the histogram of ionization fractions at each redshift with the number of data points available. There is enough data in each bin of ionization fraction. However, Figure 4.1 also shows that the size of the dataset in each bin is not equal, but sufficient to capture variations in each neutral fraction bins.

4.0.3 TRAINING THE NETWORK

Here we describe how training for both CNN and ANN is performed. The training of the network is basically finding the set of parameters that minimizes the distance between the predictions and true labels. It is useful to construct the network which will be fast and robust in solving a problem, but it is always a challenge to achieve such a goal. However, tuning hyper-parameters has shown that it is still achievable. Hyper-parameter is a parameter that is set before the learning process begins. Since the network might have many hyper-parameters, we then apply mini-batch to allow minimal supervision on some of the hyper-parameters, in this way we can focus on less number of parameters. As helpful mini-batch can be, we also initialize the hyper-parameters as shown in the Table 4.6.

TABLE 4.6: Summary of the hyper-parameters choice for the network

Hyper-parameter	Value
Learning rate	start = 10^{-2}
Batch-size	50
Number of Epochs	25
Dropout rate	20%
Loss function	Huber Loss, Smooth Mean Absolute Error
Optimizer	AdamOp

We briefly discuss each hyper-parameter below:

- Learning rate - is the steps of learning that control how quickly or slowly a neural network model learns a problem.
- Batch-size - refers to the number of training examples in one forward/backward pass.
- Number of Epochs - is the number of times the entire training set pass through the network.

- Dropout rate - is the rate at which the randomly selected neurons are ignored or switched off during training.
- Loss function - is a function used to measure or evaluate how well the algorithm models a given data.
- Optimizer - is an algorithm used to change the network's attributes (e.g weights, learning rate, etc) in order to reduce the losses.

The **learning rate** of 10^{-2} is used as the starting value and allow it to decay exponentially where the network finds the best learning rate during the training steps on its own. The mini-batch technique allows us to use a higher learning rate, which speeds up the learning process. During the training of the network, we set **batch-size** to be 50. Batch-size is directly proportional to the learning rate, through our experimentation which suggests that if the learning rate is high, then the batch-size must be small. This obviously will depend on the number of training dataset available. We train our network for a total of 25 epochs. The **dropout rate** of 20% is also applied, implying that the network keeps 80% of the output every time the dropout is active. This method forms part of the regularization process which reduces over-fitting. In our work, we use the Huber Loss function (Huber (1992)) known as a smooth mean absolute error, and the following equation gives loss function:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{for } |y - f(x)| \leq \delta. \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{4.1}$$

where y is actual value and f(x) represents estimated values, and δ is a hyper-parameter that can be tuned. Huber loss is more robust and more sensitive to outliers than many loss functions. Moreover, it also differentiable at 0. We also note that it approaches Mean Absolute Error (MAE) when δ is too small and Mean Square Error (MSE) when δ is too large.

Figure 4.2 shows the importance of choosing the value of δ when using Huber loss. The choice of δ determines what we are willing to consider as an outlier. The different lines represent different values of δ. As it can be seen on the plot that when δ is higher, we get a smoothed

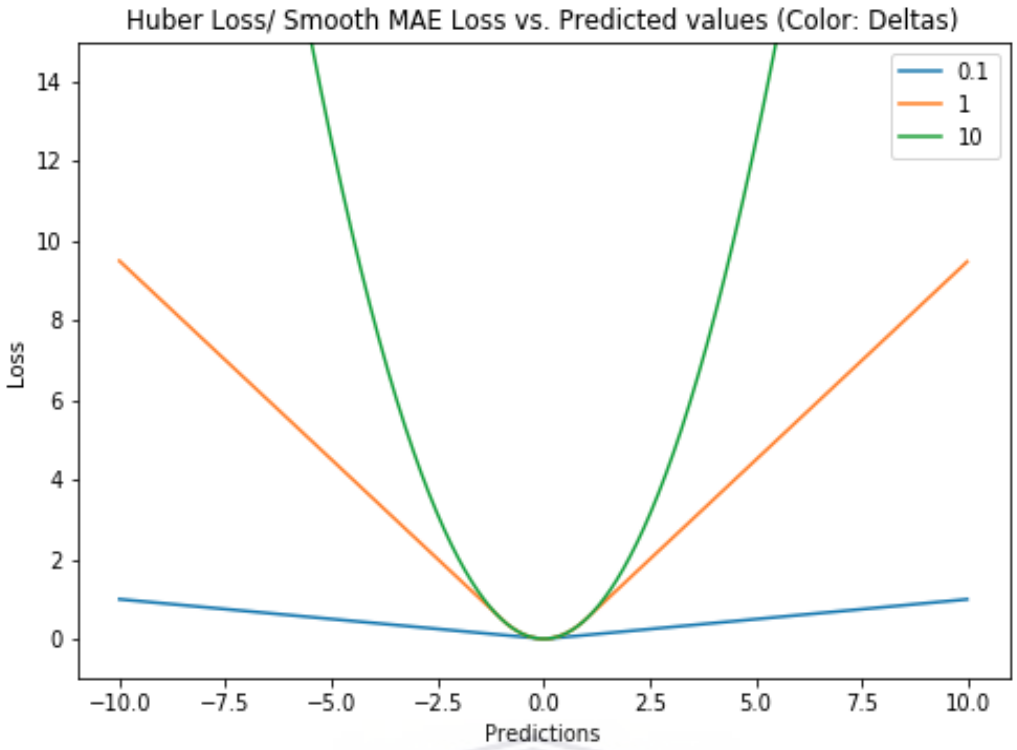

FIGURE 4.2: Plot of Huber Loss (Y-axis) vs. Predictions (X-axis). True value = 0. Rackspace, 2019

parabolic shape which indicates the amount of sensitivity to the outliers. For the case of a green curve, the Huber loss is sensitive to very few outliers compared to when δ is very low(blue curve). For our work we used $\delta = 0.1$.

The optimization functions calculate the gradient i.e. the partial derivative of loss function with respect to weights. The following equation gives an example of the calculation:

$$W^{(k+1)} = W^{(k)} - \frac{\partial L(W)}{\partial W^{(k)}} \tag{4.2}$$

Where L(W) is the loss function needed to be minimized. We further use Adam optimization algorithm (AdamOp; Kingma and Ba, 2014) to minimize our loss function. The weights are modified in the opposite direction of the calculated gradient, and the cycle repeated until the minimum loss is reached. In AdamOp, the update rule for individual weights is to scale their gradients inversely proportional to a (scaled) norm of their individual current and past gradients. Unlike with Stochastic gradient descent where the learning rate remain the same for all weights updates and throughout the training process and generally requires more of hyper-parameter

tuning during the training process, AdamOp's learning rate is maintained for each network weight and separately adaptable as learning unfolds.

The above-discussed components of the neural network come to play a role in the training of the network. Convolutional neural network layers are organized in 3 dimensions. The input layer accepts the 21cm images, and it passes the image into the hidden convolutional layers where the feature extraction is done by performing a series of convolution and pooling operation resulting in the detection of features (e.g. ionizing bubbles). The convolution operation is executed using a filter/kernel of size 2×2 which slides over our 21cm image and performs a matrix multiplication and sums the result onto the feature map. The collection of feature maps from the operation using different filters is combined and serves as a final output of convolution layer. In order to avoid the feature map from shrinking, we use what is called padding where a layer of zero-value pixels is added around the input maps to preserve the size of the maps. We add max-pooling layer after each convolution layer. This max-pooling, as discussed above. We also use strides of 2.

Since a fully connected layers accept a vector as an input, we then flatten our feature maps to pass it to the end part of our network which is the fully connected part. neurons are fully connected to the activation from the previous layer, and the training from these layers is performed through forward and backpropagation. Our output layer has one neuron which is responsible for giving the final prediction. All of the estimated ionization fraction is stored together with true ionization fraction.

5 Results and Discussion

5.1 Quantifying the results

There are other methods which can be used to quantify how well the network is able to recover the correct answers such as Welch's t-test (Welch, 1947) and Kolmogorov-Smirnov test (Kolmogorov, 1933). We here choose to quantify the results using coefficient of determination R^2. It is represented by the following equation:

$$R^2 = \frac{\sum (Y_{pred} - \overline{Y}_{true})^2}{\sum (Y_{true} - \overline{Y}_{true})^2}, \tag{5.1}$$

where the summation is performed over the entire data set, and Y_{pred}, Y_{true} and $\overline{Y}_{true}$ are predicted values from the network, the true labels and the average of all the true values of the sample. R^2 is the fraction by which the variance of the errors is less than the variance of the dependent variable. It ranges between 0 and 1, where 1 refers to a perfect inference of parameters.

5.2 Classical vs Convolutional Neural Network

We attempt to solve our problem using a classical neural network and compare the results with the convolutional neural network. This test was done to check which one can be more efficient in solving our problem. As described in chapter 4, the architecture of the network plays a role for the network's performance. The CNN contains convolutional layers which will require more time to train compared to fully connected layers, this means that CNN training will then take longer as compared to ANN training. Since our data is in the form of images, the expectations might be that we use CNN straight away, but the results show that also ANN is competent and give robust results in some instances (data without noise). The challenge we are facing is that

after taking into account the instrumental effects our ANN performs worse in extracting the ionization fraction from the 21cm images. Below we present the results before adding instrumental effect.

Figure 5.1 shows the plot of true value vs estimated values from the two networks as indicated above each plot. The results show clearly that ANN also is able to recover the parameter. However, the CNN show a much better recovery and an increase in accuracy as compared to the ANN. The results presented are for a single redshift 7, and we observe a similar trend on other redshifts not included here. ANN fails because it uses one neuron per image pixel as an input. Our noise images are 140×140 which makes 19600 pixels with very low resolution, this results in having unmanageable amount of weights and too many parameters since it is fully connected. The CNN has the ability to capture 2D information of the local bubbles on the 21cm maps and given the higher accuracy obtained, we then proceed with the CNN.

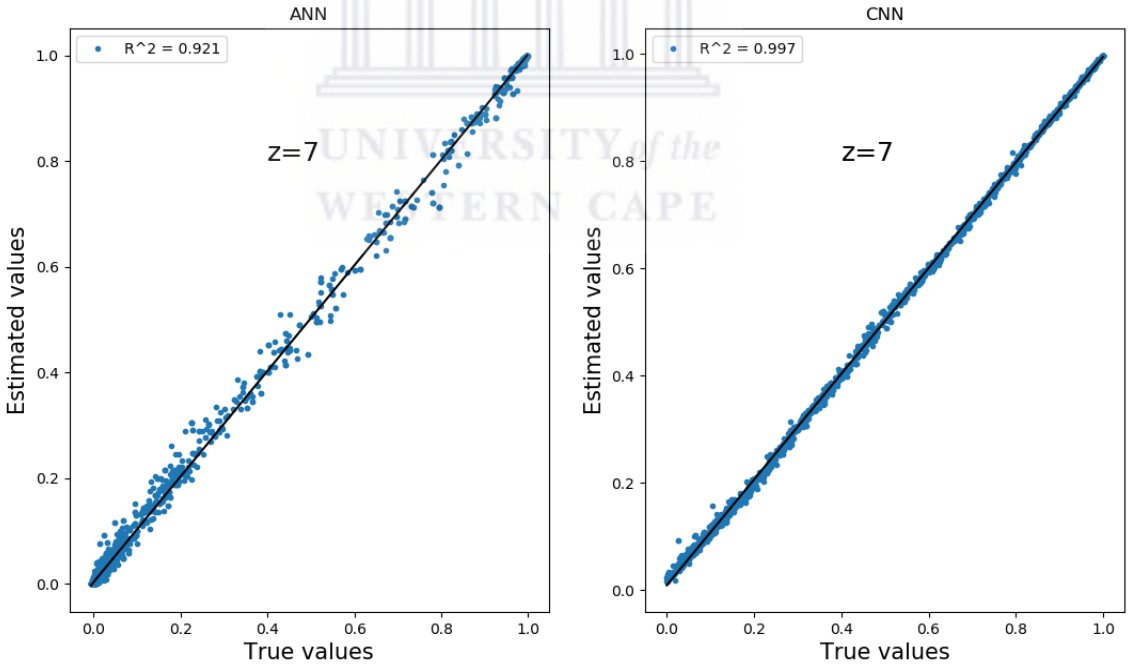

FIGURE 5.1: The plot of true values vs estimated values from ANN and CNN for $z = 7$. The blue points represent the ionization fraction of each 21cm map from the estimator and the true values directly from the ionization fraction boxes. The number of dataset is equal in both methods, i.e training = 29 000, test dataset = 1500, validation dataset = 1500

5.3 Best Network results

We now present the results for four different redshifts, i.e. $z = 7, 8, 9, 10$ obtained from our CNN and for combined redshifts. We train and test the network with the dataset for each mentioned redshifts and do the same with combined redshifts dataset. In principle, the ionization fraction ranges from 0 to 1. We show the results *without noise* and *with noise*. Finally, we combine all the redshifts for data *with noise* to make final evaluation.

5.3.1 Fitting without Noise

Single redshifts

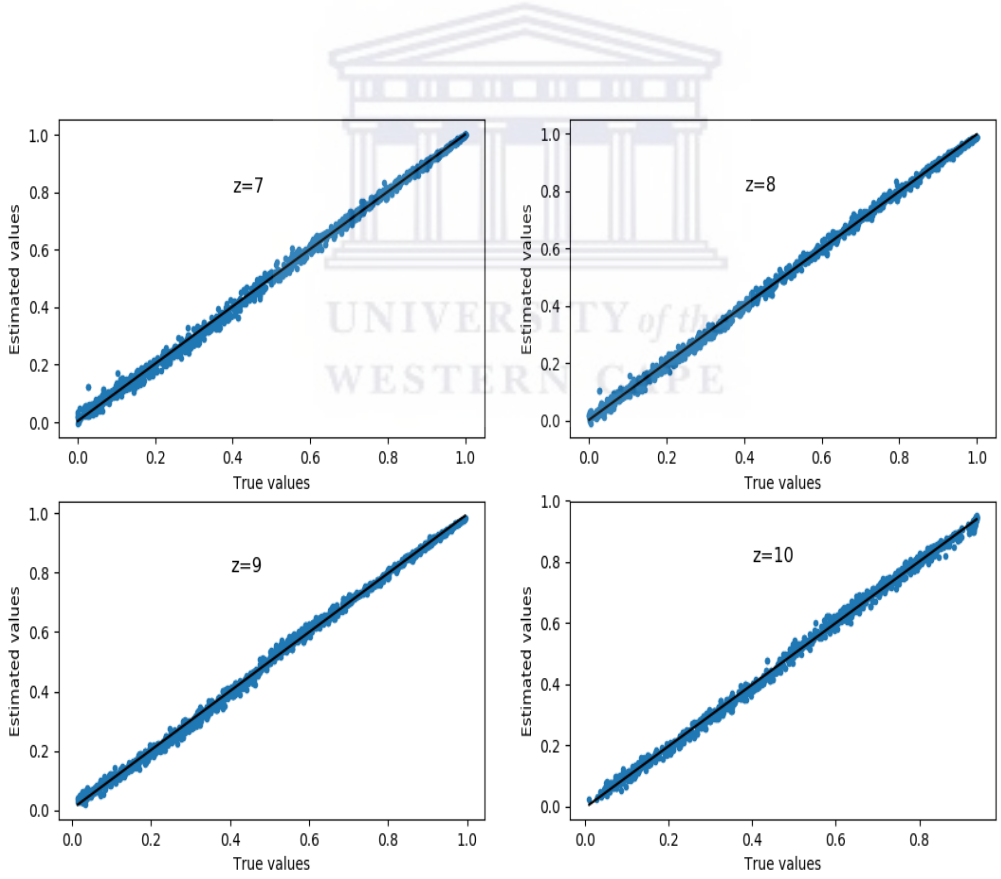

Figure 5.2: The plot of ionization fraction estimated values vs true values. The estimated values are from the CNN for different redshifts, i.e $z = 7, 8, 9, 10$ on the data before accounting for instrumental effects. Number of training data set is 10920, 11280, 11760, 11760, validation data set is 1365, 1410, 1470, 1470, and test data set is 1365, 1410, 1470, 1470 respectively.

Figure 5.2 shows the relationship between true ionization fraction of each slice and the estimated ionization fraction obtained from the network. The scatter plot is perfectly aligned with the fitted line and the points are more concentrated across the line. This is observed in all redshifts. We further calculate R^2 values for all redshifts presented in Table 5.1. All R^2 values indicate good accuracy which shows successful recovery of our parameter in all redshifts. Our results suggest that, CNN approach is a good way to recover ionization fraction from 21cm images. The ability to recover this parameter gives us an exciting start on building sophisticated methods of analyzing upcoming 21cm dataset. This means we can develop model-independent techniques to extract EoR parameters like the ionization fraction from future 21cm experiments.

It is always imperative to think of error quantification and checking whether the network is learning something during its training steps or not. There are several different methods to evaluate how well our network model the data. The loss function can be used to make these evaluations. It calculates the deviation of true value and predicted value, and if the deviation is too much, then it will output a large number. The optimization functions help the loss to learn to reduce this error.

Figure 5.3 shows the loss function for the training set. An epoch is defined as each iteration over the whole training sample. We train our network for 25 epochs, and it starts converging at around 400 iteration as can be seen on the Fig 5.3. The different redshifts are represented by different colour. The behaviour of the loss for each redshift is generally similar since it decays exponentially and converges at nearly the same epoch. The performance of the network is generally good for all redshits. This observation is in agreement with the calculated R^2 value in Table 5.1 where we observe that they all have $R^2 > 0.99$.

Combined redshifts

Considering the fact that the universe evolves with redshift we then investigate the ability of our estimator to extract the ionization fraction from the 21cm images of different redshifts passed at once. Figure 5.4 shows the result of all redshifts combined, plotting the estimated values vs the

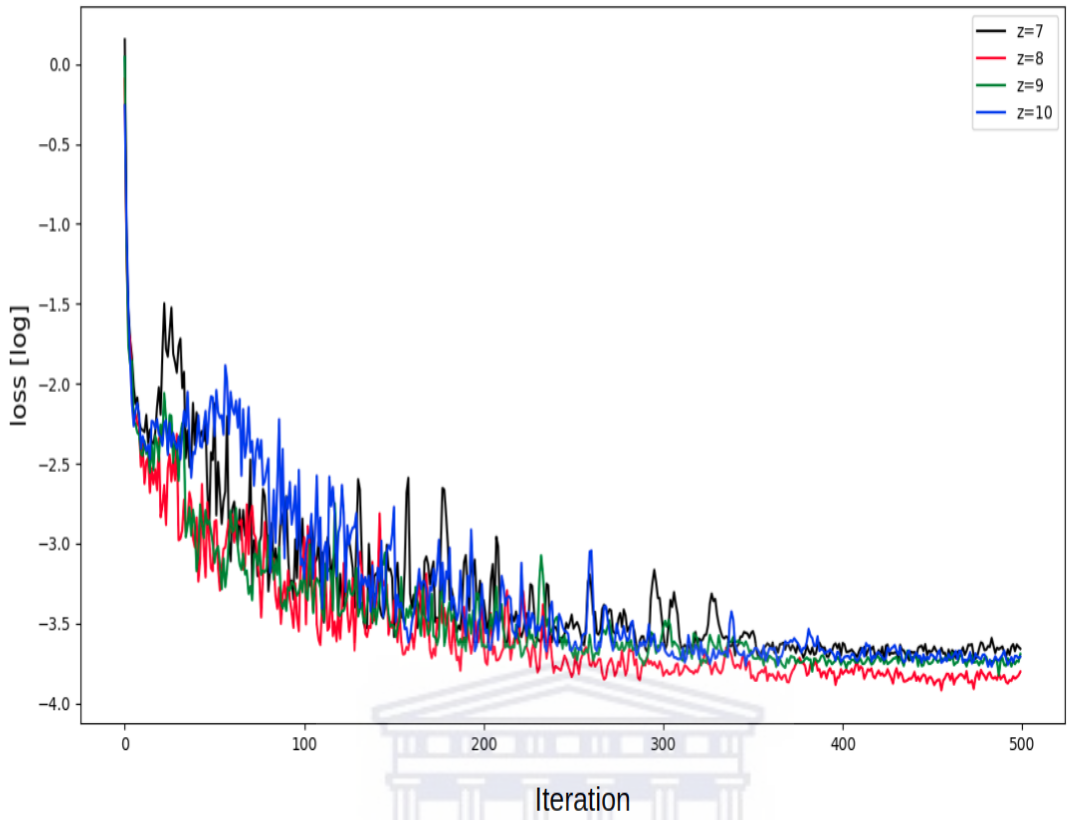

FIGURE 5.3: The plot of the loss function of training set for four different redshifts as indicated on the upper right end.

true values. The estimator is able to extract the parameter very well even after mixing up the data of different redshifts. This is impressive since the data from different redshifts is entirely different. We first examine our best performing CNN ability to recover the ionized fraction in noiseless data set. We also quote the value of R^2 calculated from this result, and it reflects a good accuracy of our result.

5.3.2 FITTING WITH NOISE

Single redshifts

Our second set of results are from the data which takes experimental effects into account. Figure 5.5 is a plot of true values of ionization fraction against the estimated ionization fraction from the CNN. The plot as compared to Figure 5.2 reflects how noise can affect the performance

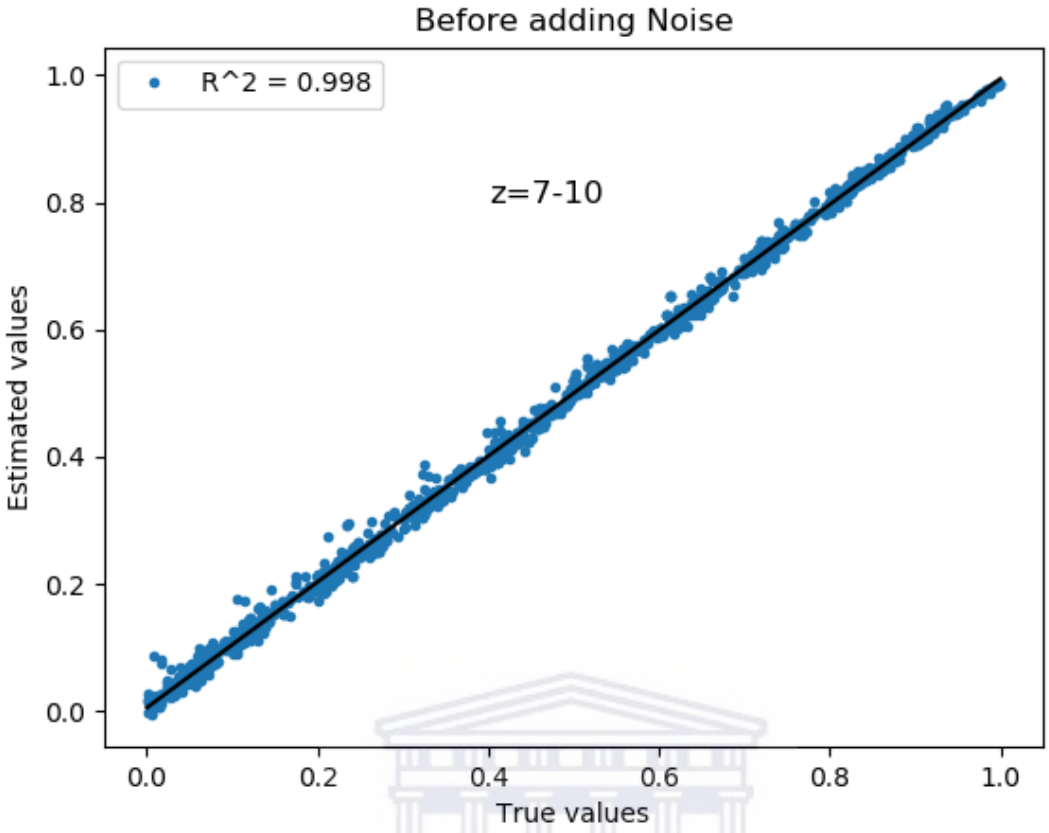

FIGURE 5.4: The plot of ionization fraction estimated values vs true values. The estimated values are from the CNN for different redshifts combined, i.e $z = 7 - 10$ on the data before accounting for instrumental effects. Number of training data set is 45720, validation data set is 5715, and test data set is 5715 respectively.

of our network. The uv-sampled data accounts for Foreground avoidance and instrumental noise. We quote the value of R^2 in Table 5.1, which indicates a decrease in accuracy when we go to higher redshift. Nevertheless, this is expected since when we go higher in redshift the level of foreground contamination increase since the wedge slope (m) is higher, allowing more background sources to be picked up.

Fundamentally, the network focuses on the structures of the image to calculate the percentage of ionized regions on that particular image, and in our case, these structures are the ionization bubbles. If the features of bubbles are destroyed or no longer clear which is probably what happens when we account for instrumental effects, then the performance of the network will be affected negatively.

A significant change is observed when accounting for foreground contamination avoidance. The wedge slope presented previously plays a significant role in the level of foreground contamination. The wedge increases with increasing redshift (see Equation (2.3), and Table 2.1), and we properly remove the wedge according to the redshift of the 21cm map. This means that our data set contains different levels of foreground removal, presenting a more challenge to ionized fraction recovery by the neural net.

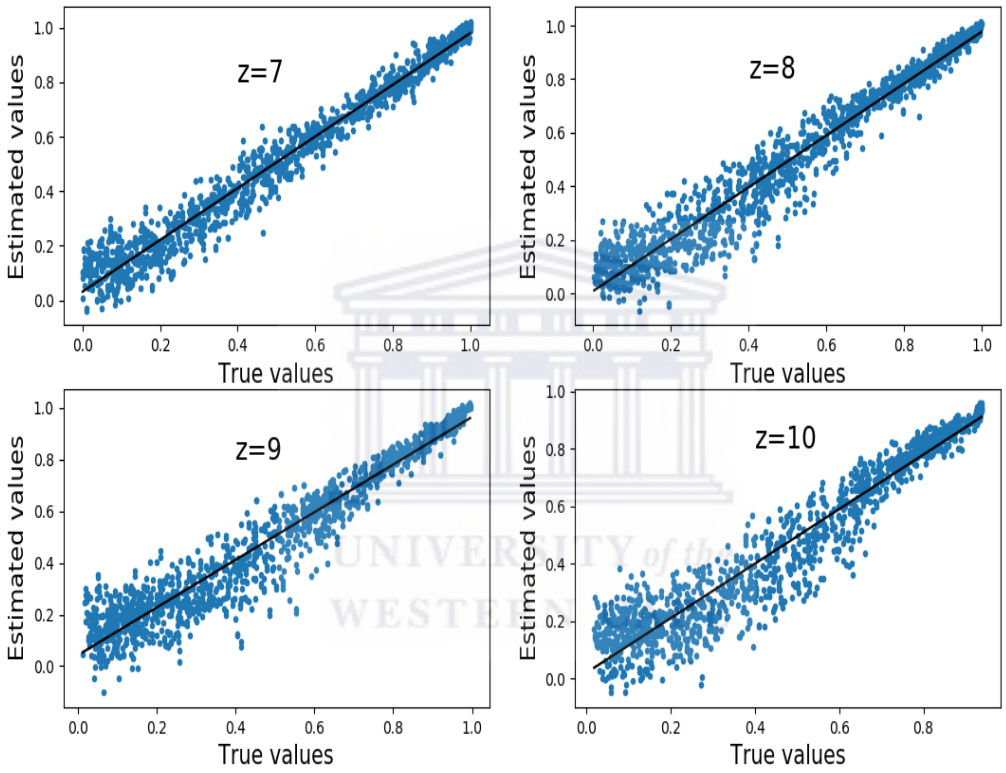

FIGURE 5.5: The plot of ionization fraction estimated values vs true values. The estimated values are from the CNN for different redshifts, i.e $z = 7, 8, 9, 10$ on the data after accounting for instrumental effects. Number of training data set is 10920, 11280, 11760, 11760, validation data set is 1365, 1410, 1470, 1470, and test data set is 1365, 1410, 1470, 1470 respectively.

Despite the effect of noise into our data, our estimator still shows a good recovery of the ionization fraction from our 21cm images with $R^2 > 0.92$. The scatter plot is aligned with the model line but not tightly concentrated across the line. This indicates the decrease in accuracy compared to the results before adding noise. We notice a good recovery in highly ionized regions than in less ionized regions at low redshifts (e.g $z = 7, 8$). This is because noise is less at highly ionized

regions. When the bubbles are small, the noise contaminate them easily, hence we get a more scattered points at less ionized regions.

Combined redshifts

We further continue to explore the capability of our network by accounting for instrumental effect, but this time around, we then combine our dataset for all redshifts, train and test the network. The role of accounting for instrumental effects become more vital when we combine the dataset from different redshifts since the level of noise is different from different redshifts.

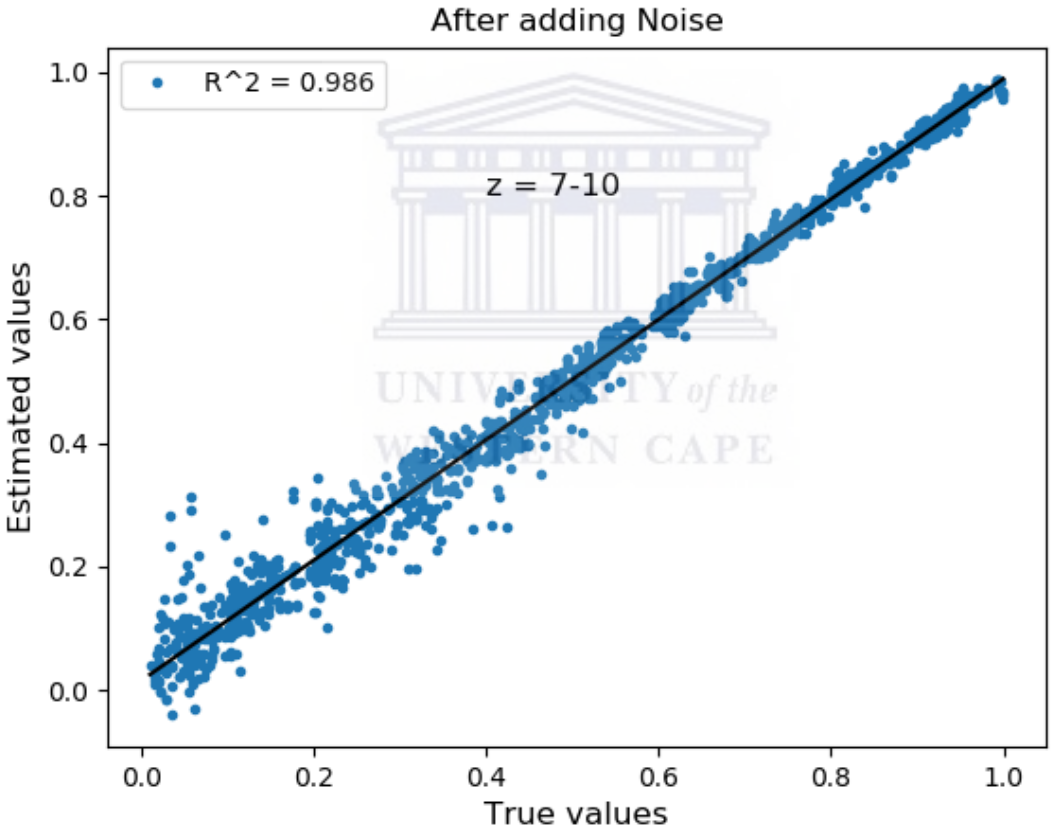

FIGURE 5.6: The plot of ionization fraction estimated values vs true values. The estimated values are from the CNN for different redshifts combined, i.e $z = 7 - 10$ on the data after accounting for instrumental effects with $m = 0.15, 0.19, 0.23, 0.27$ for each respective reshift. Number of training data set is 45720, validation data set is 5715, and test data set is 5715 respectively.

From Figure 5.6, we observe that our network was able to recover the ionization fraction from the 21cm images which were randomly taken from different redshifts($z = 7 - 10$). We select an

equal number of images from different redshifts to form part of the training, validation and testing data set. The accuracy of the result fluctuates at around 95% to 88% for single redshift and for combined redshifts we observe an increase in accuracy at 98%. This observation might be because of the higher number of training set for the combined dataset since the network is trained with 13239 images, this is 2000 more compared to other training sets presented in Table 4.5. The encoded redshift information in the maps also plays a role since adding more information essentially will help in making a good fitting. This is in agreement with the expectation when working with neural networks, more training dataset will basically improve the accuracy of the newtwork.

Table 5.1 shows the calculated R^2 values for each redshift, and it is clear to see that the network performed good at all redshifts before adding noise to the data. Generally, all four redshifts were learned well and provided good results. It is interesting to see that CNN is still competitive and give good accuracy even after accounting for instrumental effects.

TABLE 5.1: The accuracy measure(coefficient of determination)

redshift(z)	R^2 Before Noise	R^2 After Noise
7	0.9975	0.964
8	0.9983	0.941
9	0.9980	0.927
10	0.9978	0.925

5.3.3 Wedge slope effect

Another test was done in trying to investigate the effect of wedge slope during foreground avoidance technique on our images. We use a single redshift to demonstrate how the wedge slope affects the performance of our network.

From Figure 5.7 we observe that the scatter increases and the accuracy decreases at higher wedge slope values, where more modes are assumed to be foreground contaminated. The results show the significance of the value of m on the performance of the network. Nevertheless, this observation is expected and can be studied further in order to understand more about foreground contamination. The presented result is for $z = 8$, and we eliminate the results for other redshifts

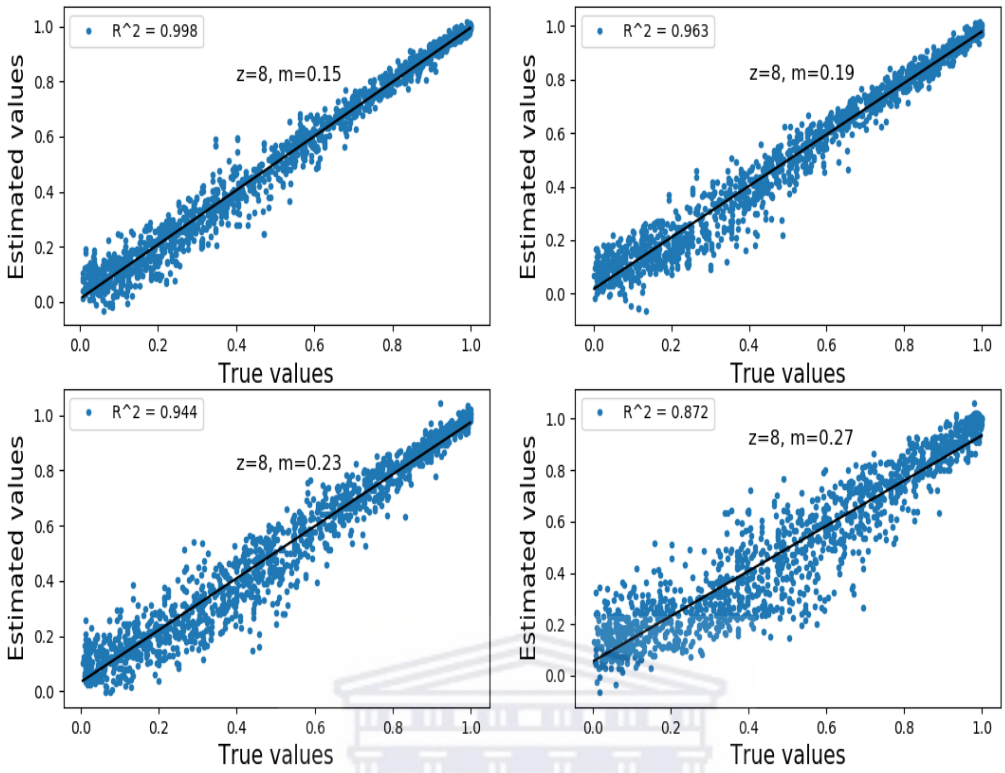

FIGURE 5.7: The plot of ionization fraction estimated values vs. true values for $z = 8$ after accounting for instrumental effects. The first plot is for $m = 0.15$, second one is for $m = 0.19$, third plot is for $m = 0.23$, and the fourth plot is for $m = 0.27$ as indicated inside each plot.

since we observe similar trend where the accuracy decreases when we go to higher values of m.

This work already has proven to be a useful indication for developing more sophisticated and improved foreground techniques. This can be helpful for us to test our network in the future when such foreground tools are available. This work also shows the effectiveness of the application of deep learning techniques into cosmology. Not limited to parameter recovering, there are a vast of number of work done recently to prove the ability of the application of machine learning in astronomy. Provided that the upcoming 21cm experiments will provide us with a tremendous amount of data, the automated systems can come to our rescue in order to assist us in analyzing this vast amount of data.

Response of Convolution

Lastly we discuss our understanding on how CNN is able to create the link between the input (21cm-image) and the output (ionization fraction). This is done by looking at the responds from the first convolutional layer. We only pull out the output from this layer at the final training step. Figure 5.8 shows the convolution of random 21cm image to a set of 8 weights from the first layer.

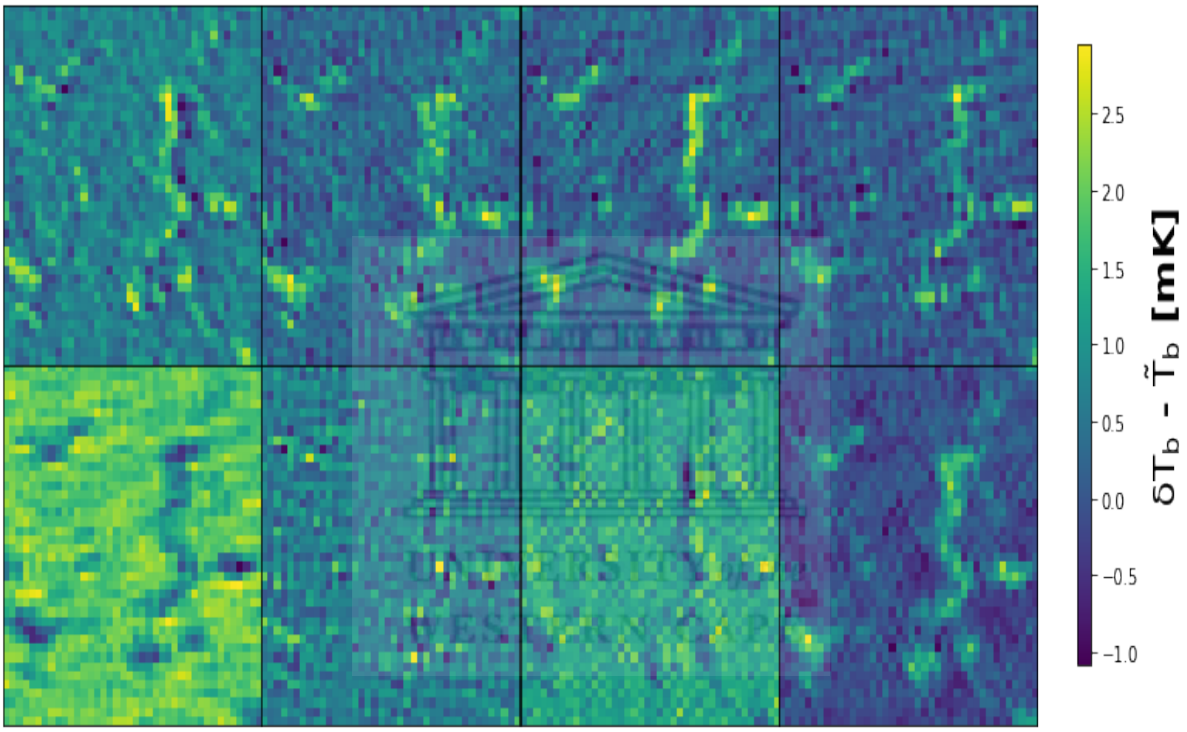

FIGURE 5.8: The response of convolving a randomly selected 21cm map with the trained set of 8 weights of the first convolutional layer before the application of the Relu. The weights activate the input map differently, particularly, the neutral region. The ionized bubble activation appears to be approximately similar, albeit fainter with some weights. These variations are used in the network to estimate the neutral fraction out of the 21cm map.

We find that the activation of the ionized bubbles is relatively similar, although the bubbles edges are somewhat fainter with some weights. However, these trained weights do activate the ionized regions of the 21cm image very differently as shown by the red color, indicating that the network is using these variations to estimate the ionization fraction out of the input 21cm map. We have decided to restrict the visualization to the first convolutional layer since with other deeper layers more features are no longer visible making it difficult to can see what the

network might be using in order to decide on what could be a corresponding ionization fraction of a given 21cm image.

6 Conclusion

In this work we have shown that machine learning methods can be used to recover ionization fraction directly from 21cm maps of the early universe. We put our focus on the ionization fraction mainly because is one of the parameters which can help us to understand the evolution of the early universe. Upcoming instruments like SKA and HERA will probe this region on large scales. Therefore, we take advantage of these observational possibilities and study the 21cm signal to build a better understanding of the expected signal from the above mentioned instruments. We use a deep learning approach known as convolutional neural network to build an estimator which inputs 21cm images and outputs the corresponding ionization fraction. Our estimator can extract the ionization fraction from 21cm images without being given any external information about the signal. The previous approaches to extract the ionisation fraction rely on the power spectrum and require the assumption of a given model for reionisation. Our technique can be more robust to the assumptions in reionisation models by including multiple simulations in the training set. The results show an excellent reconstruction at different redshifts($z = 7 - 10$). The results are quantified using coefficient of determination R^2. We obtain R^2 of $0.99 \equiv 99\%$ before and also up to $0.93 \equiv 93\%$ after taking into account the instrumental effects. The results also show a decrease in accuracy at higher redshift. This could be because of the high level of foreground contamination at higher redshifts. We combine the data from different redshifts and test the network. This also shows that the network performance is robust to redshift changes. To study the early universe using images can be a challenge since the foreground contamination worsens at higher redshift. Figure 5.7 in chapter 4 shows the impact of foreground contamination to the 21cm signal. As we increase the wedge slope m we observe a decrease in accuracy and it becomes worse at higher redshifts. This suggests that good foreground cleaning methods are needed. The results can be improved by methods like averaging of images which will increase the signal to noise ratio. We leave this for future work. We further explore the possibilities

of improving the accuracy by combining the data from $z = 7 - 10$ and Figure 5.6 shows an improvement of accuracy compared to single redshifts. The CNN strongly depends on the amount of training data and we conclude that the averaging technique can help us in improving the results when enough data is available, so this will be further tested when more data is available. While we only focused on the SKA design, our analysis can be easily extended to include other experiments such as HERA and LOFAR, although the large noise and/or low resolution might create extra challenges. We leave it for future work to present a detailed comparison between the ability of different 21cm arrays to constrain the reionization history.

Bibliography

Ade, Peter AR et al. (2016). "Planck 2015 results-xiii. cosmological parameters". In: *Astronomy & Astrophysics* 594, A13.

AIKMAN, G CHRISTOPHER L. "Smithsonian Astrophysical Observatory (SAO)". In:

Aldhaheri, Abdulrahman and Jeongkyu Lee (2017). "Event detection on large social media using temporal analysis". In: *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*. IEEE, pp. 1–6.

ASTRON (2010). *Low-Frequency Array*. http://www.lofar.org. Online; accessed 14-Nov-2020.

Barkana, Rennan and Abraham Loeb (2001). "In the beginning: the first sources of light and the reionization of the universe". In: *Physics reports* 349.2, pp. 125–238.

Becker, Robert H et al. (2001). "Evidence for Reionization at z 6: Detection of a Gunn-Peterson Trough in az= 6.28 Quasar". In: *The Astronomical Journal* 122.6, p. 2850.

Beiser, Arthur and Arthur Beiser (1969). "Perspectives of modern physics". In: *McGraw-Hill series in fundamentals of physics.*

Bennett, CL et al. (2013). "Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) observations: final maps and results". In: *The Astrophysical Journal Supplement Series* 208.2, p. 20.

Bernardi, G et al. (2016). "Bayesian constraints on the global 21-cm signal from the Cosmic Dawn". In: *Monthly Notices of the Royal Astronomical Society* 461.3, pp. 2847–2855.

Bolton, James S and Martin G Haehnelt (2007). "The observed ionization rate of the intergalactic medium and the ionizing emissivity at z 5: evidence for a photon-starved and extended epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 382.1, pp. 325–341.

Bond, JR et al. (1991). "Excursion set mass functions for hierarchical Gaussian fluctuations". In: *The Astrophysical Journal* 379, pp. 440–460.

Bowman, Judd D et al. (2013). "Science with the murchison widefield array". In: *Publications of the Astronomical Society of Australia* 30.

CARINA CHENG (2020). *Epoch of Rionization*. http://w.astro.berkeley.edu/~carinacheng/research.html. Online; accessed 14-Jan-2020.

CASPER (2015). *Collaboration For Astronomy Signal Processing And Electronics Research*. https://casper.ssl.berkeley.edu/astrobaki/images/b/b1/Hyperfinetrans_GSnotes.png. Online; accessed 9-Jan-2018.

Chen, Sheng, Colin FN Cowan, and Peter M Grant (1991). "Orthogonal least squares learning algorithm for radial basis function networks". In: *IEEE Transactions on neural networks* 2.2, pp. 302–309.

Ciardi, Benedetta and Andrea Ferrara (2005). "The first cosmic structures and their effects". In: *Space Science Reviews* 116.3-4, pp. 625–705.

Coates, Adam, Andrew Ng, and Honglak Lee (2011). "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223.

Cooray, A., E. Komatsu, and A. Melchiorri (2015). *New Horizons for Observational Cosmology*. EBL-Schweitzer. IOS Press. ISBN: 9781614994763. URL: https://books.google.co.za/books?id=IEAoBgAAQBAJ.

Cooray, Asantha and Steven R Furlanetto (2004). "Free-free emission at low radio frequencies". In: *The Astrophysical Journal Letters* 606.1, p. L5.

Curtin Institute of Radio Astronomy (CIRA) (2009). *The Murchison Widefield Array*. http://www.mwatelescope.org. Online; accessed 14-Nov-2019.

Datta, A, JD Bowman, and CL Carilli (2010). "Bright source subtraction requirements for redshifted 21 cm measurements". In: *The Astrophysical Journal* 724.1, p. 526.

Davé, Romeel et al. (2013). "The neutral hydrogen content of galaxies in cosmological hydrodynamic simulations". In: *Monthly Notices of the Royal Astronomical Society* 434.3, pp. 2645–2663.

DeBoer, David R et al. (2017). "Hydrogen epoch of reionization array (HERA)". In: *Publications of the Astronomical Society of the Pacific* 129.974, p. 045001.

Di Matteo, Tiziana et al. (2002). “Radio foregrounds for the 21 centimeter tomography of the neutral intergalactic medium at high redshifts”. In: *The Astrophysical Journal* 564.2, p. 576.

Dunkley, J et al. (2009). “Five-Year Wilkinson Microwave Anisotropy Probe* Observations: Likelihoods and Parameters from The Wmap Data”. In: *The Astrophysical Journal Supplement Series* 180.2, p. 306.

Fan, Xiaohui, CL Carilli, and B Keating (2006). “Observational constraints on cosmic reionization”. In: *Annu. Rev. Astron. Astrophys.* 44, pp. 415–462.

Finlator, Kristian et al. (2015). “The reionization of carbon”. In: *Monthly Notices of the Royal Astronomical Society* 447.3, pp. 2526–2539.

Finlator, Kristian et al. (2009). “A new moment method for continuum radiative transfer in cosmological re-ionization”. In: *Monthly Notices of the Royal Astronomical Society* 393.4, pp. 1090–1106.

Furlanetto, Steven R, Matias Zaldarriaga, and Lars Hernquist (2004). “The growth of H II regions during reionization”. In: *The Astrophysical Journal* 613.1, p. 1.

Gehlot, BK et al. (2019). “The first power spectrum limit on the 21-cm signal of neutral hydrogen during the Cosmic Dawn at z= 20–25 from LOFAR”. In: *Monthly Notices of the Royal Astronomical Society* 488.3, pp. 4271–4287.

Gillet, Nicolas et al. (2018). “Deep learning from 21-cm images of the Cosmic Dawn”. In: *arXiv preprint arXiv:1805.02699*.

Gnedin, Nickolay Y and Peter A Shaver (2004). “Redshifted 21 centimeter emission from the pre-reionization era. I. Mean signal and linear fluctuations”. In: *The Astrophysical Journal* 608.2, p. 611.

Greig, Bradley and Andrei Mesinger (2015). “21CMMC: an MCMC analysis tool enabling astrophysical parameter studies of the cosmic 21 cm signal”. In: *Monthly Notices of the Royal Astronomical Society* 449.4, pp. 4246–4263.

Gunn, James E and Bruce A Peterson (1965). “On the Density of Neutral Hydrogen in Intergalactic Space.” In: *The Astrophysical Journal* 142, pp. 1633–1641.

Haarlem, M áP van et al. (2013). “LOFAR: The low-frequency array”. In: *Astronomy & astrophysics* 556, A2.

Harker, Geraint et al. (2009). “Non-parametric foreground subtraction for 21-cm epoch of reionization experiments”. In: *Monthly Notices of the Royal Astronomical Society* 397.2, pp. 1138–1152.

Hassan, Sultan et al. (2016). “Simulating the 21 cm signal from reionization including non-linear ionizations and inhomogeneous recombinations”. In: *Monthly Notices of the Royal Astronomical Society* 457.2, pp. 1550–1567.

— (2017). “Epoch of reionization 21 cm forecasting from MCMC-constrained semi-numerical models”. In: *Monthly Notices of the Royal Astronomical Society* 468.1, pp. 122–139.

Hassan, Sultan et al. (2018). “Identifying Reionization Sources from 21cm Maps using Convolutional Neural Networks”. In: *arXiv preprint arXiv:1807.03317*.

Hawking, Stephen W and George FR Ellis (1968). “The cosmic black-body radiation and the existence of singularities in our universe”. In: *The Astrophysical Journal* 152, p. 25.

He, Sheng-wu, Jia-gang Wang, and Jia-an Yan (2018). *Semimartingale theory and stochastic calculus*. Routledge.

Herbel, Jörg et al. (2018). “Fast Point Spread Function Modeling with Deep Learning”. In: *arXiv preprint arXiv:1801.07615*.

Hijazi, Samer, Rishi Kumar, and Chris Rowen (2015). “Using convolutional neural networks for image recognition”. In: *Cadence Design Systems Inc.: San Jose, CA, USA*.

Hinshaw, Gary et al. (2013). “Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) observations: cosmological parameter results”. In: *The Astrophysical Journal Supplement Series* 208.2, p. 19.

Hubble, Edwin (1929). “A relation between distance and radial velocity among extra-galactic nebulae”. In: *Proceedings of the National Academy of Sciences* 15.3, pp. 168–173.

Huber, Peter J (1992). “Robust estimation of a location parameter”. In: *Breakthroughs in statistics*. Springer, pp. 492–518.

Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167*.

Kashikawa, Nobunari et al. (2006). “The end of the reionization epoch probed by Lyα emitters at z= 6.5 in the Subaru Deep Field”. In: *The Astrophysical Journal* 648.1, p. 7.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kohonen, Teuvo (1990). "The self-organizing map". In: *Proceedings of the IEEE* 78.9, pp. 1464–1480.

Kolmogorov, Andrey (1933). "On the empirical determination of a distribution law". In: *Inst. Ital. Attuari, Giorn.* 4, pp. 83–91.

Koopmans, LVE et al. (2015). "The cosmic dawn and epoch of reionization with the square kilometre array". In: *arXiv preprint arXiv:1505.07568*.

Krawczyk, Bartosz (2016). "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4, pp. 221–232.

Kubota, Kenji et al. (2016). "Expected constraints on models of the epoch of reionization with the variance and skewness in redshifted 21 cm-line fluctuations". In: *Publications of the Astronomical Society of Japan* 68.4, p. 61.

La Plante, Paul and Michelle Ntampaka (2018). "Machine Learning Applied to the Reionization History of the Universe". In: *arXiv preprint arXiv:1810.08211*.

LeCun, Yann, Yoshua Bengio, et al. (1995). "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10, p. 1995.

Lima, Clausius G de and EL Wehry (1986). "The Shopl'skil Effect as an Analytical Tool". In: *CRC Critical Reviews in Analytical Chemistry* 16.3, pp. 177–221.

Lochner, Michelle et al. (2016). "Photometric supernova classification with machine learning". In: *The Astrophysical Journal Supplement Series* 225.2, p. 31.

Loeb, Abraham and Rennan Barkana (2001). "The reionization of the universe by the first stars and quasars". In: *Annual review of astronomy and astrophysics* 39.1, pp. 19–66.

Majumdar, Suman et al. (2014). "On the use of seminumerical simulations in predicting the 21-cm signal from the epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 443.4, pp. 2843–2861.

McQuinn, Matthew et al. (2007). "The morphology of H II regions during reionization". In: *Monthly Notices of the Royal Astronomical Society* 377.3, pp. 1043–1063.

Meiksin, Avery and Piero Madau (1993). “On the photoionization of the intergalactic medium by quasars at high redshift”. In: *The Astrophysical Journal* 412, pp. 34–55.

Mellema, Garrelt et al. (2006). “Simulating cosmic reionization at large scales–II. The 21-cm emission features and statistical signals”. In: *Monthly notices of the royal astronomical society* 372.2, pp. 679–692.

Mesinger, Andrei and Steven Furlanetto (2007). “Efficient simulations of early structure formation and reionization”. In: *The Astrophysical Journal* 669.2, p. 663.

Mitchell, Thomas M et al. (1997). *Machine learning*.

Mitra, Sourav et al. (2011). “Reionization constraints using principal component analysis”. In: *Monthly Notices of the Royal Astronomical Society* 413.3, pp. 1569–1580.

Molaro, Margherita et al. (2019). “ARTIST: Fast radiative transfer for large-scale simulations of the epoch of reionisation”. In: *arXiv preprint arXiv:1901.03340*.

Møller, Martin Fodslette (1993). “A scaled conjugate gradient algorithm for fast supervised learning”. In: *Neural networks* 6.4, pp. 525–533.

Morales, Miguel F and Jacqueline Hewitt (2004). “Toward epoch of reionization measurements with wide-field radio observations”. In: *The Astrophysical Journal* 615.1, p. 7.

Morales, Miguel F and J Stuart B Wyithe (2010). “Reionization and Cosmology with 21-cm Fluctuations”. In: *Annual review of astronomy and astrophysics* 48, pp. 127–171.

NASA LAMBDA Archive Team (2006). *Optical Depth to Reionization*. https://lambda.gsfc.nasa.gov/education/graphic_history/taureionzation.cfm. Online; accessed 05-Nov-2019.

Ntampaka, Michelle et al. (2019). “The Role of Machine Learning in the Next Decade of Cosmology”. In: *arXiv preprint arXiv:1902.10159*.

Oh, Siang Peng (1999). “Observational signatures of the first luminous objects”. In: *The Astrophysical Journal* 527.1, p. 16.

Paciga, Gregory et al. (2011). “The GMRT Epoch of Reionization experiment: a new upper limit on the neutral hydrogen power spectrum at z 8.6”. In: *Monthly Notices of the Royal Astronomical Society* 413.2, pp. 1174–1183.

PAPER Team (2012). *Precision Array for Probing the Epoch of Reionization*. http://eor.berkeley.edu. Online; accessed 14-Nov-2018.

Parsons, Aaron R et al. (2010). "The precision array for probing the epoch of re-ionization: Eight station results". In: *The Astronomical Journal* 139.4, p. 1468.

Parsons, Aaron R et al. (2012). "A per-baseline, delay-spectrum technique for accessing the 21 cm cosmic reionization signature". In: *The Astrophysical Journal* 756.2, p. 165.

Parsons, Aaron R et al. (2014). "New limits on 21 cm epoch of reionization from paper-32 consistent with an x-ray heated intergalactic medium at z= 7.7". In: *The Astrophysical Journal* 788.2, p. 106.

Peebles, PJE (1968). "Recombination of the primeval plasma". In: *The Astrophysical Journal* 153, p. 1.

Peng Oh, S and Katherine J Mack (2003). "Foregrounds for 21-cm observations of neutral gas at high redshift". In: *Monthly Notices of the Royal Astronomical Society* 346.3, pp. 871–877.

Penzias, Arno A and Robert Woodrow Wilson (1965). "A measurement of excess antenna temperature at 4080 Mc/s." In: *The Astrophysical Journal* 142, pp. 419–421.

Pineda, Fernando J (1987). "Generalization of back-propagation to recurrent neural networks". In: *Physical review letters* 59.19, p. 2229.

Pober, Jonathan C et al. (2013,2014). "Opening the 21 cm epoch of reionization window: Measurements of foreground isolation with paper". In: *The Astrophysical Journal Letters* 768.2, p. L36.

Pober, Jonathan C et al. (2014). "What next-generation 21 cm power spectrum measurements can teach us about the epoch of reionization". In: *The Astrophysical Journal* 782.2, p. 66.

Prochaska, J Xavier, Gabor Worseck, and John M O'Meara (2009). "A Direct Measurement of the Intergalactic Medium Opacity to HI Ionizing Photons". In: *The Astrophysical Journal Letters* 705.2, p. L113.

Rackspace (2019). *Smooth Mean Absolute Error/ Huber Loss*. https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb. Online; accessed 14-Feb-2019.

Ravanbakhsh, Siamak et al. (2016). "Estimating Cosmological Parameters from the Dark Matter Distribution." In: *ICML*, pp. 2407–2416.

Rawat, Waseem and Zenghui Wang (2017). "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9, pp. 2352–2449.

Santos, Mário G, Asantha Cooray, and Lloyd Knox (2005). "Multifrequency analysis of 21 centimeter fluctuations from the era of reionization". In: *The Astrophysical Journal* 625.2, p. 575.

Santos, Mário G et al. (2007). "Cosmic Reionization and the 21cm signal: simulations and analytical models". In: *arXiv preprint arXiv:0708.2424*.

Santos, MG et al. (2010). "Fast large volume simulations of the 21-cm signal from the reionization and pre-reionization epochs". In: *Monthly Notices of the Royal Astronomical Society* 406.4, pp. 2421–2432.

Schmit, Claude J and Jonathan R Pritchard (2018). "Emulation of reionization simulations for Bayesian inference of astrophysics parameters using neural networks". In: *Monthly Notices of the Royal Astronomical Society* 475.1, pp. 1213–1223.

Shaver, PA et al. (1999). "Can the reionization epoch be detected as a global signature in the cosmic background?" In: *arXiv preprint astro-ph/9901320*.

Shimabukuro, Hayato and Benoit Semelin (2017). "Analysing the 21 cm signal from the epoch of reionization with artificial neural networks". In: *Monthly Notices of the Royal Astronomical Society* 468.4, pp. 3869–3877.

Shimabukuro, Hayato et al. (2016). "Constraining the EoR model parameters with the 21cm bispectrum". In: *arXiv preprint arXiv:1608.00372*.

Simonyan, Karen and Andrew Zisserman (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV].

SKA South Africa (2016). *Hydrogen Epoch of Reionization Array*. https://www.sarao.ac.za/gallery/hera/. Online; accessed 14-Feb-2016.

Songaila, Antoinette (2004). "The evolution of the intergalactic medium transmission to redshift 6". In: *The Astronomical Journal* 127.5, p. 2598.

Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal (1997). "Introduction to multi-layer feed-forward neural networks". In: *Chemometrics and intelligent laboratory systems* 39.1, pp. 43–62.

Tata Institute of Fundamental Research (2001). *Giant Metrewave Radio Telescope*. http://www.gmrt.ncra.tifr.res.in. Online; accessed 1-Jan-2020.

Trott, Cathryn M, Randall B Wayth, and Steven J Tingay (2012). "The impact of point-source subtraction residuals on 21 cm epoch of reionization estimation". In: *The Astrophysical Journal* 757.1, p. 101.

Vedantham, Harish, N Udaya Shankar, and Ravi Subrahmanyan (2012). "Imaging the epoch of reionization: Limitations from foreground confusion and imaging algorithms". In: *The Astrophysical Journal* 745.2, p. 176.

Wagoner, Robert V (1973). "Big-bang nucleosynthesis revisited". In: *The Astrophysical Journal* 179, pp. 343–360.

Wang, X (2006). "X. Wang, M. Tegmark, MG Santos, and L. Knox, Astrophys. J. 650, 529 (2006)." In: *Astrophys. J.* 650, p. 529.

Welch, Bernard L (1947). "The generalization ofstudent's' problem when several different population variances are involved". In: *Biometrika* 34.1/2, pp. 28–35.

Zahn, Oliver et al. (2007). "Simulations and analytic calculations of bubble growth during hydrogen reionization". In: *The Astrophysical Journal* 654.1, p. 12.

Zaldarriaga, Matias, Steven R Furlanetto, and Lars Hernquist (2004). "21 Centimeter fluctuations from cosmic gas at high redshifts". In: *The Astrophysical Journal* 608.2, p. 622.

Zaroubi, Saleem (2013). "The epoch of reionization". In: *The First Galaxies*. Springer, pp. 45–101.

Zel'dovich, Ya B (1970). "Gravitational instability: An approximate theory for large density perturbations." In: *Astronomy and astrophysics* 5, pp. 84–89.