

2021-08-01

Modeling The Spatiotemporal Dynamics Of Active Regions On The Sun Using Deep Neural Networks

Godwill Amankwa
University of Texas at El Paso

Follow this and additional works at: https://scholarworks.utep.edu/open_etd



Part of the [Astrophysics and Astronomy Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Amankwa, Godwill, "Modeling The Spatiotemporal Dynamics Of Active Regions On The Sun Using Deep Neural Networks" (2021). *Open Access Theses & Dissertations*. 3213.
https://scholarworks.utep.edu/open_etd/3213

This is brought to you for free and open access by ScholarWorks@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.

MODELING THE SPATIOTEMPORAL DYNAMICS OF ACTIVE REGIONS ON THE
SUN USING DEEP NEURAL NETWORKS

GODWILL AMANKWA ASARE MENSAH

Master's Program in Computational Science

APPROVED:

Olac Fuentes, Ph.D., Chair

Amy Wagler, Ph.D.

Felicia Manciu, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

©Copyright

by

Godwill Amankwa Asare Mensah

2021

Dedication

to my

MOTHER and FATHER

with love

MODELING THE SPATIOTEMPORAL DYNAMICS OF ACTIVE REGIONS ON THE
SUN USING DEEP NEURAL NETWORKS

by

GODWILL AMANKWA ASARE MENSAH

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

AUGUST 2021

Acknowledgements

I would like to express my deep-felt gratitude to my advisor, Dr. Olac Fuentes of the Computer Science Department at the University of Texas at El Paso, for his advice, encouragement, enduring patience and constant support. He was never ceasing in his belief in me. I also wish to thank the other members of my committee, Dr. Amy Wagler of the Mathematics Department and Dr. Felicia Manciu of the Physics Department, both at The University of Texas at El Paso. Their suggestions, comments and additional guidance were invaluable to the completion of this work.

Additionally, I want to thank the Computational Science Program for the support given me in completing this thesis.

I am also grateful to my loved ones, family and research group for their support and contribution to this work.

Abstract

Solar active regions are areas on the Sun’s surface that have especially strong magnetic fields. Several phenomena that can have significant negative effects on technology and subsequently on human life, such as solar flares and coronal mass ejections (CMEs), are often associated with active regions. Since the physical phenomena underlying the evolution of active regions are still poorly understood, the accurate prediction of solar flares and coronal mass ejections remains an open problem.

Extracting insights from the available datasets of solar activity that can lead to a better understanding of solar active regions has been an important research goal at the intersection of artificial intelligence and solar physics. With the advancement in artificial intelligence, some machine learning models have been applied to predict solar flares from a 6-hour to 48-hour window. Support Vector Machine (SVM) [6, 42], K-Nearest-Neighbor (KNN) [29], Extremely Randomized Trees (ERT) [37], and deep neural network [36] are some of the machine learning models that have been used in predicting solar flare but results are not good. This can be attributed to the fact that the models are trained using a selection of Active regions parameters and an imbalance data (few positive flare examples).

As a result, there is a need to understand space weather and the basis by which these events occur. In this study, we applied a deep learning architecture originally designed for video prediction to predict the changes happening on the Sun in continuous time by using time series Helioseismic and Magnetic Imager data captured by Solar Dynamics Observatory (SDO) and compared it against a no-change baseline and a regression baseline.

These three approaches were compared against one another based on their mean squared error (MSE) and structural similarity index measure (SSIM) and it was found out that the regression model outperforms the others in MSE whilst the deep learning model outperforms the rest in SSIM.

From this, we seek to continue our work by adapting deep learning models used in solving

image-to-image translation problems to produce high-quality synthetic data to solve the class imbalance data problem and incorporate other time-series data of the Sun to improve upon the predictions of the spatiotemporal changes of active regions on the Sun.

Table of Contents

	Page
Acknowledgements	v
Abstract	vi
Table of Contents	viii
List of Tables	x
List of Figures	xi
Chapter	
1 Introduction	1
1.1 Active Regions	1
1.1.1 What Is An Active Region	1
1.1.2 What Is A Sunspot	2
1.2 Solar Flares	3
1.2.1 What Are Solar Flares	3
1.2.2 Classification Of Solar Flares	3
1.2.3 Prominent Solar Flare Events	3
1.2.4 Goal	6
1.2.5 Significance of The Goal	6
1.3 Thesis Statement	6
1.4 Research Questions	6
1.5 Outline	7
2 Background	8
2.1 Convolutional Neural Network (CNN)	10
2.1.1 Convolution Operation	10
2.1.2 Pooling	11
2.1.3 Activation	13

2.1.4	Loss Function	17
2.2	Recurrent Neural Network	18
2.2.1	LSTM	20
3	Related Work	25
3.1	Active region and Solar Flare Prediction	25
3.2	Next Frame Prediction	28
4	Data and Methodologies	30
4.1	Data	30
4.1.1	HMI Instrument	31
4.2	Data Processing	32
4.3	Models	41
4.3.1	Memory In Memory Network (MIM)	41
4.3.2	Baseline	45
4.3.3	Regression	46
5	Preliminary Results	47
6	Research Plan	52
6.1	Proposed Work	52
6.2	Timeline	54
	References	56
	Curriculum Vitae	64

List of Tables

1.1	Solar Flare Classes	4
4.1	Number of Sequences	40
5.1	One Hour Input to Predict the Next One Hour	50
5.2	Three Hour Input to Predict the Next Three Hour	50
5.3	Six Hour Input to Predict the Next Six Hour	51
6.1	Timeline	55

List of Figures

1.1	Sunspots (left) and Coronal loops (Right). Credit: Royal Swedish Academy of Sciences - Göran Scharmer and Mats Löfdahl (sunspots) and NASA/SDO and the AIA, EVE, and HMI science teams (coronal loops)	2
2.1	Neural Network Representation	9
2.2	An example of 2-D convolution	11
2.3	Max pooling vs Average Pooling	12
2.4	Sigmoid	14
2.5	Hyperbolic tangent	15
2.6	Rectified Linear Units	16
2.7	ReLU vs Leaky Reaky ReLU	16
2.8	Recurrent Neural Network - Type 1	18
2.9	Recurrent Neural Network - Type 2	20
2.10	Standard RNN	20
2.11	LSTM	21
2.12	Cell of LSTM	21
2.13	Structure of gate of LSTM	22
2.14	Forget gate of LSTM	22
2.15	Input gate of LSTM	23
2.16	Updated state of LSTM	24
4.1	Graph of the number of sunspots for Solar Cycles ranging from March 1911 to March 2021	33
4.2	Region cutout shown with the red rectangle	34

4.3	Plot of distribution of pixel values of image from solar maximum after transformation using k ranging from 0.01 – 0.07.	36
4.4	Plot of distribution of pixel values of image from solar minimum after transformation using k ranging from 0.01 – 0.07.	37
4.5	Plot of distribution of pixel values of image from solar maximum after transformation using k ranging from 0.001 – 0.007.	38
4.6	Plot of distribution of pixel values of image from solar minimum after transformation using k ranging from 0.001 – 0.007.	39
4.7	Contiguous Sampling from 2011 to 2013	41
4.8	Schematic of MIM-N [56]	43
4.9	Schematic of MIM-S [56]	44
4.10	A MIM network with two MIMs and one ST-LSTM. [56]	45
5.1	Sequence tested at different iteration step	47
5.2	MIM: Min-Max Norm - Input: 6 Hour Output: 6 Hour	51
5.3	MIM: Sig Norm - Input: 6 Hour Output: 6 Hour	51
5.4	Regression: Sig Norm - Input: 6 Hour Output: 6 Hour	51

Chapter 1

Introduction

The dynamic conditions and events on the Sun in near-Earth space and in our upper atmosphere have a significant role in both human lives and innovation. This phenomenon is depicted by space weather. Solar activities give rise to different kinds of space weather. These activities are solar flares, coronal mass ejections, high-speed solar wind, and solar energetic particles. The solar magnetic field is the definitive wellspring of all of these activities.

1.1 Active Regions

1.1.1 What Is An Active Region

An Active Region is an area on the Sun with especially strong magnetic field. It is known that most solar storms -solar flares and coronal mass ejections(CME) erupt from active regions. Active Regions are most common during the peak of the sunspot or solar cycle when the Sun's magnetic field is highly disturbed. Active Regions can be seen in X-ray and ultraviolet images of the Sun as bright areas. The powerful magnetic fields around active regions release intense bursts of energy, which often take the form of high-energy X-ray and UV photons. Sunspots are visual indicators of active regions but not all active regions have sunspots. Also other solar features such as solar prominences and coronal loops sometimes do appear around Active regions [48].

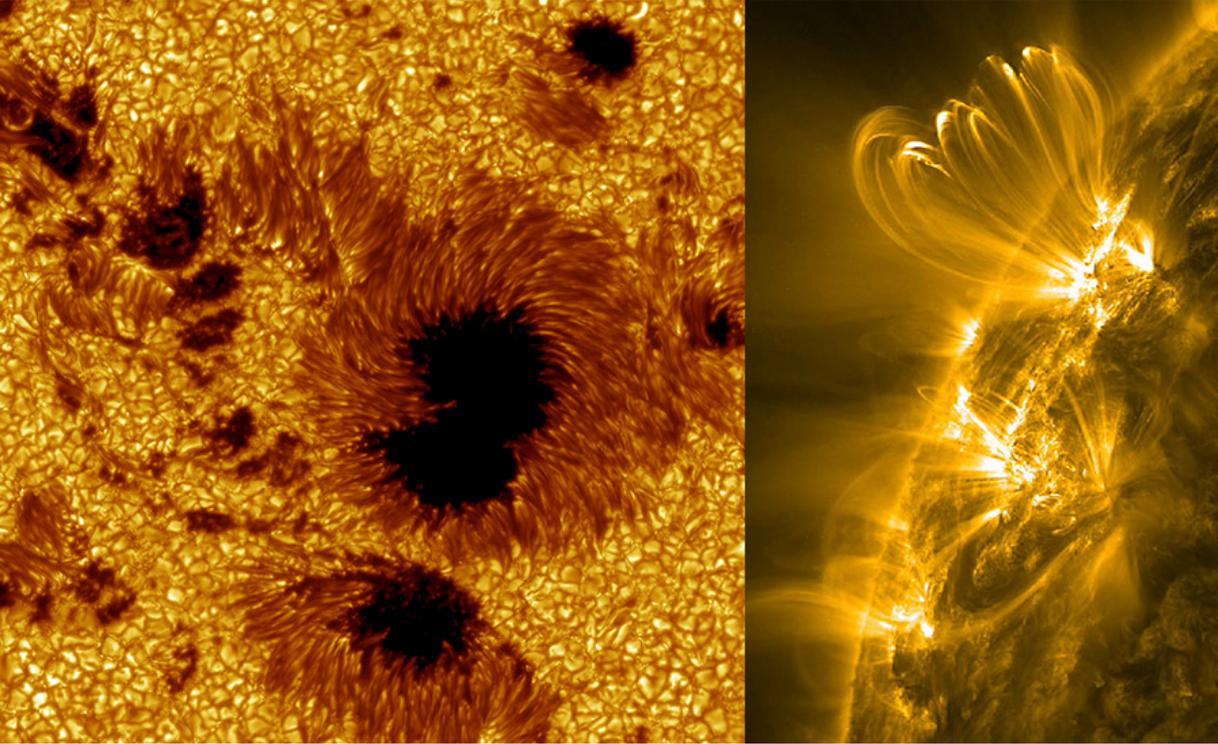


Figure 1.1: Sunspots (left) and Coronal loops (Right).

Credit: Royal Swedish Academy of Sciences - Göran Scharmer and Mats Löfdahl (sunspots) and NASA/SDO and the AIA, EVE, and HMI science teams (coronal loops)

1.1.2 What Is A Sunspot

Sunspots are regions on the surface of the Sun found in the photosphere which appear darker and also cooler compared to other areas on the Sun. Their dark characteristic is due to the fact that whilst other areas have average temperature of 5,800 degrees Kelvin, sunspots have average temperature of 3,800 degrees Kelvin. Though Sunspots appear with small in solar images, their diameter can be as large as 50,000 kilometers. Formation of Sunspots is not fully understood but it is known to be formed in areas on the Sun with intense magnetic activity [51] (Figure 1.1).

1.2 Solar Flares

1.2.1 What Are Solar Flares

A sudden release of distorted magnetic fields that produces a huge amount of energy and drives that energy into space creates a sudden flash of light known as a solar flare. Solar flares are considered as the largest explosive events of our solar system. Flares can last from minutes to hours. This electromagnetic emission travels at the speed of light. Sometimes the released energy accelerates very high energy particles such as protons and electrons. It takes about tens of minutes for these high energy particles to reach the Earth [35].

The energy emitted by a solar flare is more than a million times greater than the energy from a volcanic explosion on Earth. Although solar flares can be visible in white light, they are often more readily noticed via their bright X-ray and ultraviolet emissions.

1.2.2 Classification Of Solar Flares

The eruption and the associated storms with solar flare can carry huge radiation to the earth and/or travel long distances in space. Due to this, the National Oceanic and Atmospheric Administration (NOAA) has devised categories to classify flares and the various storms. The letters A, B, C, M and X are used in the classification of solar flares with the biggest flares classified as "X-class flares" and the smallest ones as "A-class flares". Just like the Richter scale for earthquake, each letter represents a 10-fold increase in energy output. A, B and C are the weak ones having less noticeable effect on the Earth. The "M-class flares" can cause brief radio blackouts at the poles and minor radiation storms that might endanger astronauts. Table 1.1 shows the various categories and their respective measurements.

1.2.3 Prominent Solar Flare Events

Solar flares strongly influence the local space weather in the vicinity of the Earth. They can produce streams of highly energetic particles in the solar wind or stellar wind, known as a

Table 1.1: Solar Flare Classes

Class	W/m^2 between 1 & 8 Ångströms
A	$< 10^{-7}$
B	$\geq 10^{-7} < 10^{-6}$
C	$\geq 10^{-6} < 10^{-5}$
M	$\geq 10^{-5} < 10^{-4}$
X	$\geq 10^{-4}$

solar proton event. These particles can impact the Earth's magnetosphere. Also, massive solar flares are sometimes accompanied by coronal mass ejections (CMEs) which can trigger geomagnetic storms that have been known to disable satellites and knock out terrestrial electric power grids for extended periods of time. These events have been recorded for the past one hundred and fifty years. Below are some of the serious solar flare events that was felt on earth or by satellite in space:

1. One of the most powerful solar flare events happened on September 1, 1859. This event is known as the "Carrington event". The event was first reported by one of England's foremost solar astronomers named Richard Carrington. Because of that event, high energy particles entered into Earth's atmosphere and overpowered the Earth's protective magnetic field which resulted in a huge destruction on the ground. Telegraph system disrupted worldwide. Telegraph offices were ignited by the spark. Colorful aurora were seen at near tropical latitudes over Cuba, the Bahamas, Jamaica, El Salvador, and Hawaii. Even when batteries were disconnected, the messages were transmitted because of electric currents in the wires induced by Aurora [10].
2. One of the historical powerful series of solar storms happened in August 1972. The solar storms were associated with solar flare, geomagnetic storm and high energy particles. This storm set off sea mines in Vietnam. This event disrupted satellite,

hindered communication-grid and electric service. This event occurred between the Apollo 16 and Apollo 17 lunar missions. If this would happen during one of the mission, the particles could have hit astronauts outside of Earth's protective magnetic field and the result could have been life-threatening.

3. Another example of space weather events is the collapse of the Hydro-Québec power network on March 13, 1989 due to geomagnetically induced currents (GICs). Caused by a transformer failure, this event led to a general blackout that lasted more than 9 hours and affected over 6 million people. The geomagnetic storm causing this event was itself the result of a CME ejected from the sun on March 9, 1989. [35]
4. From mid-October to early November 2003, one of the largest solar flare events occurred. This is actually the largest solar flare event that has been recorded by Geostationary Operational Environmental Satellite (GOES) system. This event saturated the GOES detectors, and because of this its classification is only approximate. The Sun's magnetic field lines were stretched because of the flare and then all of a sudden the magnetic field lines stretched beyond their limit. Due to this, a giant explosion occurred on the Sun's surface leading to coronal mass ejections(CME). CMEs are capable of exploding billions of tons of electrified gas and subatomic particles into space at a speed of five million miles per hour. This event disrupted communication, satellite-based system. A 90-minute blackout in Sweden was caused because of this event. Aircraft controllers changed their route to avoid high altitudes near the polar regions.
5. The solar storm of 2012 was an unusually large and strong coronal mass ejection (CME) event that occurred on July 23 that year. It missed the Earth with a margin of approximately nine days, as the equator of the Sun rotates around its own axis with a period of about 25 days. The region that produced the outburst was thus not pointed directly towards the Earth at that time. The strength of the eruption was comparable to the 1859 Carrington event that caused damage to electric equipment

worldwide, which at that time consisted mostly of telegraph stations.

1.2.4 Goal

In this research work, we seek to predict the changes in Active region by employing deep learning methods created for video predictions. For this purpose, we will utilize Solar Dynamics Observatory (SDO) images produced from Helioseismic and Magnetic Imager Instrument with a cadence of 12 minutes.

1.2.5 Significance of The Goal

The physical phenomenon underlying the evolution of active regions is still an open problem and a lot of work has been done to model and predict the changes in active regions. Researchers such as Bobra et al. [6], Qahwaji et al. [42], Li et al. [29] and Nishizuka et al. [36] have been pioneering works in feature selection of parameters of active regions and applying machine learning models to predict solar flares. Also Rempel et al. [45] have investigated the use of numerical simulations in modeling active region scale flux emergence. Despite all these advancements, there is a poor understanding of the phenomenon surrounding active regions. In this project, we seek to predict the changes of active regions to help improve the understanding of active regions and the prediction of solar flare.

1.3 Thesis Statement

Deep learning models that have been created for video prediction can be adapted to the problem of predicting the evolution of active solar regions, providing a valuable source of information towards better space weather understanding and forecasting.

1.4 Research Questions

In this research, we are looking for the answers to the following questions:

1. Could deep learning models designed for solving next frame prediction be adapted to predict the changes in active regions and produce practical results better than conventional baseline approaches?
2. Could we solve the data sparsity problem associated with solar flare prediction by generating high-quality synthetic data to improve results in solar flare prediction?
3. Could we incorporate the knowledge of the 3D viewing geometry and the Sun's rotation to simplify the learning process of predicting the changes in Active regions?
4. Could we improve the predictions of the changes in active regions by modifying the architecture of deep learning models made for video prediction to exploit information from other data sources such as x-ray time series?
5. Could deep learning methods developed to predict changes in active regions in Magnetograms from Helioseismic and Magnetic Imager (HMI) be successfully applied to images in other wavelengths from Atmospheric Imaging Assembly (AIA)?

1.5 Outline

This write-up is structured as follows:

- Chapter 2 describes the background of this proposal.
- Chapter 3 presents some related works to this research.
- Chapter 4 describes the data used and methodologies that are implemented to achieve the expectations of this work.
- Chapter 5 discusses some groundwork done and the results obtained.
- Finally, my work plan and the timeline are shown in chapter 6

Chapter 2

Background

Neural network is one of the important components in Artificial Intelligence (AI). A neural network model (or neural model) as that term is used refers to a connectionist model that simulates the biophysical information processing occurring in the nervous system. A preliminary definition of neural network is given by Kevin Gurney in his course package (1999) as follows:

A Neural Network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

Generally, a neural network consists of three layers - input layer, hidden layers and output layer. Input layer receives input of various formats, for example, number, text, audio, image pixels and so on. Hidden layer does data manipulation, mathematical calculation, feature extraction and so on based on the information it receives from input layer. Output layer receives information from hidden layer and based on that it generates the desired output.

Given input values x_1, \dots, x_p , where x_1, \dots, x_p represents an external input, a processing unit computes its output as $y = f(z)$. Here, f is called activation function and z is obtained by applying a propagation rule(activation function and any intermediate function) which maps all the inputs to the intermediate input z . Neural networks can be visualized in the means of a directed graph called network graph. Each unit is represented by a node labeled according to its output and the units are interconnected by directed edges. This is illustrated in (Figure 2.1). An input unit computes the output $y := x$ where x is the

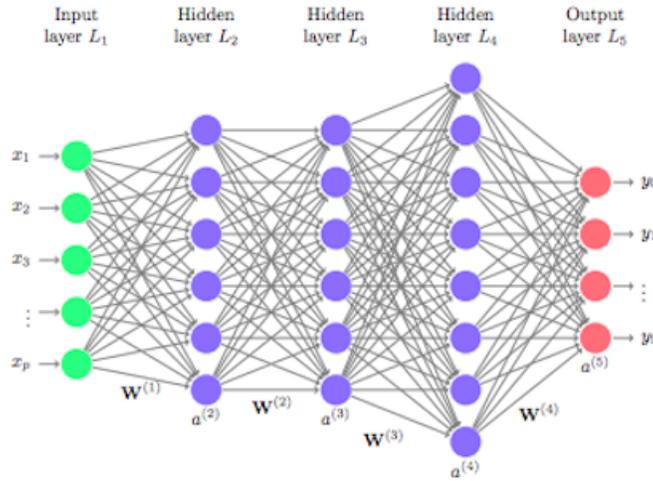


Figure 2.1: Neural Network Representation

single input value of the unit whilst output units might accept an arbitrary number of input values. The network represents a function $y(x)$ whose dimensions are fixed by the number of input and output units.

Each node in the network carries some weights to the next layer and there is an assigned bias to each layer. Result of the weighted sum of inputs with an addition of bias is passed to activation functions and based on that the output layer generates the final output. Then a cost function calculates the error. To minimize the error, weights need to adjust. To do this, we backpropagate the error through the network.

Since the inception of neural networks, many models have been created based on the task at hand using the underlying computations explained above. Out of them Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) stand out and serves as basis for most.

2.1 Convolutional Neural Network (CNN)

2.1.1 Convolution Operation

The convolution operation is usually denoted by asterisk. The convolution of f and g is defined as follows

$$s(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(a)g(t - a)da \quad (2.1)$$

and described as weighted average of $f(a)$ at the moment t where the weighting is given by $g(-a)$ with a shift of amount t . The weighting function emphasizes different parts of the input function based on the value of t .

The function f is known as the input, the function g is called the kernel and the output is referred to as the feature map in convolutional network terminology.

t will be discretized while working with data and then t can only take on integer values. In that case the discrete convolution can be defined as:

$$s(t) = (f * g)(t) = \sum_{a=-\infty}^{\infty} f(a)g(t - a) \quad (2.2)$$

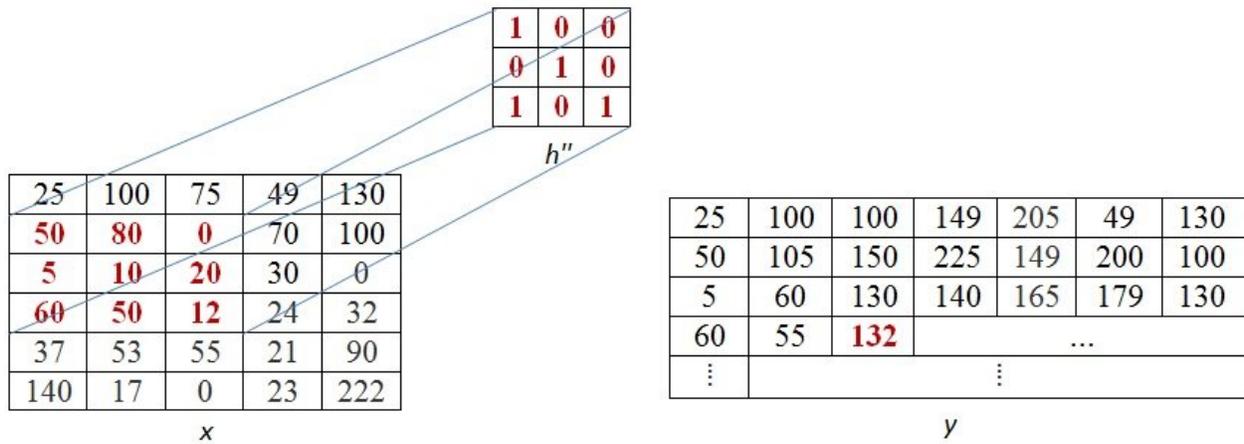
When the input is a multidimensional array, then the kernel is also a multidimensional array. These multidimensional arrays are referred to tensors. Both input and kernel elements are stored separately. It is assumed that both input and kernel functions are zero everywhere but the finite set of points for which the values are stored. Consequently, the infinite summation can be expressed as a summation over a finite number of array elements. In this case convolution is defined as

$$S(i, j) = (F * G)(i, j) = \sum_m \sum_n F(m, n)G(i - m, j - n) \quad (2.3)$$

One of the nice property of convolution is commutative property. So the previous equation is equivalent to (2.4).

$$S(i, j) = (G * F)(i, j) = \sum_m \sum_n F(i - m, j - n)G(m, n) \quad (2.4)$$

The following figure shows how 2-D convolution works where x is the input and y is the feature map produced with h'' as the kernel.



$$y(4,3) = 50 \times 1 + 80 \times 0 + 0 \times 0 + 5 \times 0 + 10 \times 1 + 20 \times 0 + 60 \times 1 + 50 \times 0 + 12 \times 1$$

$$= 50 + 0 + 0 + 0 + 10 + 0 + 60 + 0 + 12 = 132$$

Figure 2.2: An example of 2-D convolution

2.1.2 Pooling

After performing several convolutions in parallel to produce a set of linear activations, each linear activation is run through a nonlinear activation function, for example, the rectified linear unit. This is called detector stage. Then pooling function is used to replace the output of the network at a certain location with a summary statistic of the nearby outputs. To implement pooling there are several non-linear functions - max pooling, average pooling, L2 norm, weighted average. Max pooling uses the maximum value of a rectangular neighborhood and average pooling takes the average of a rectangular neighborhood.

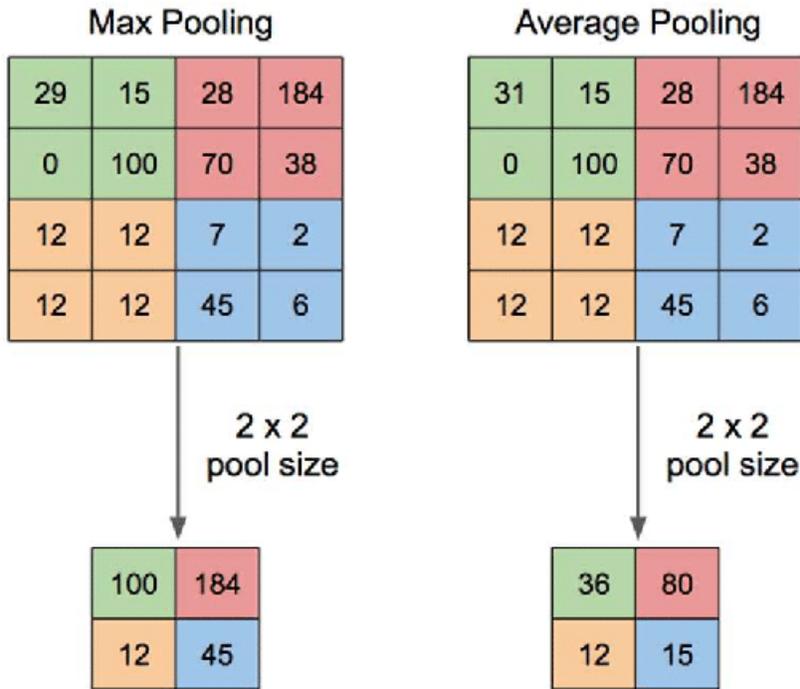


Figure 2.3: Max pooling vs Average Pooling

It is assumed that the presence of some feature is more important than its exact location relative to other features. In this case translation invariance property is very useful. And pooling helps to make the representation approximately translation invariant.

Pooling over spatial regions produces invariance to translation, but pooling over the outputs helps the features to learn which transformations to become invariant to.

Usually pooling units are fewer than detector units because pooling uses the summary statistics for pooling regions spaced k pixels apart rather than 1 pixel apart. That means the next layer has approximately k times fewer inputs to process which is computationally efficient as well as statistically efficient. This also reduce the memory requirements to store the bottlenecks.

2.1.3 Activation

Sigmoid or Logistic Activation Function

The mathematical form of sigmoid function is

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.5)$$

This activation function is very popular among deep learning researchers and engineers because of its advantages -

- By nature sigmoid function is nonlinear, so the linear combination of this function is also nonlinear. This function has smooth gradient also.
- This function takes any real value as input and output is always in the range $(0, 1)$, while the range for linear function is $(-\infty, \infty)$.
- From the figure 2.4 of the sigmoid function, it is clear that the function is very steep in the range $(-2, 2)$. That means in this range the function will change significantly for a small change in z . Consequently, there will be gradient which is greater than zero. It is also clear that the output of this function is between 0 and 1, which is good for a classifiers as this function is making a clear distinctions on prediction.

The main drawback of this activation function is outside of the range $(-2, 2)$, the function changes very little for any amount of change in z . That means the gradient will be close to zero. This problem is called "vanishing gradients" problem. This is an undesirable property. In this case the gradient becomes zero and there will be no flow of signal to its weights and recursively to its data.

Hyperbolic tangent

Mathematically hyperbolic tangent is expressed as

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.6)$$

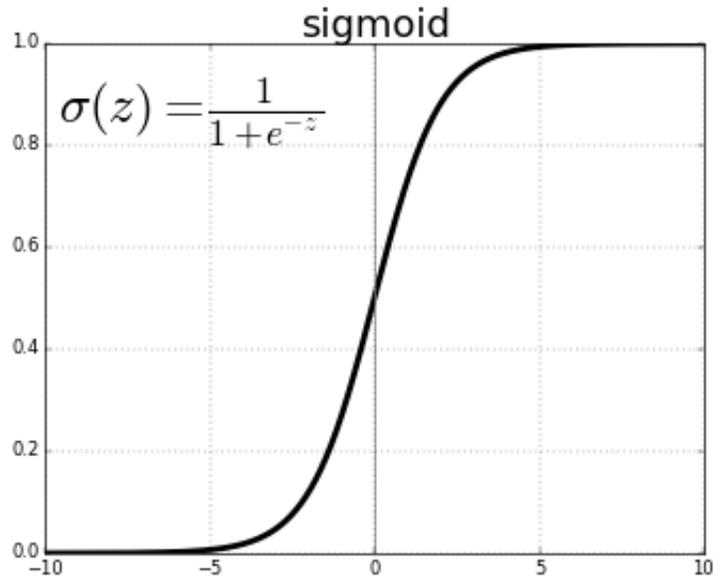


Figure 2.4: Sigmoid

which can be written as

$$\tanh(x) = 2\sigma(2x) - 1 \quad (2.7)$$

That means \tanh is scaled sigmoid function. The characteristics of this activation function is similar to sigmoid function. This function also takes any real value and the output range is $(-1, 1)$. The gradient of this function is stronger than the sigmoid function. In addition to this, \tanh is zero centered unlike sigmoid function.

Rectified Linear Units (ReLU)

ReLU is the most popular and widely used activation function. This function is defined as the positive part of its argument, $R(x) = \max(0, x)$. That means this function takes real-valued number and the output is x if x is greater than or equal to zero and zero if x is less than zero.

Sigmoid or \tanh activation is dense meaning that to describe the output of a network almost all the activations are processed, which is costly. But ReLU makes the network

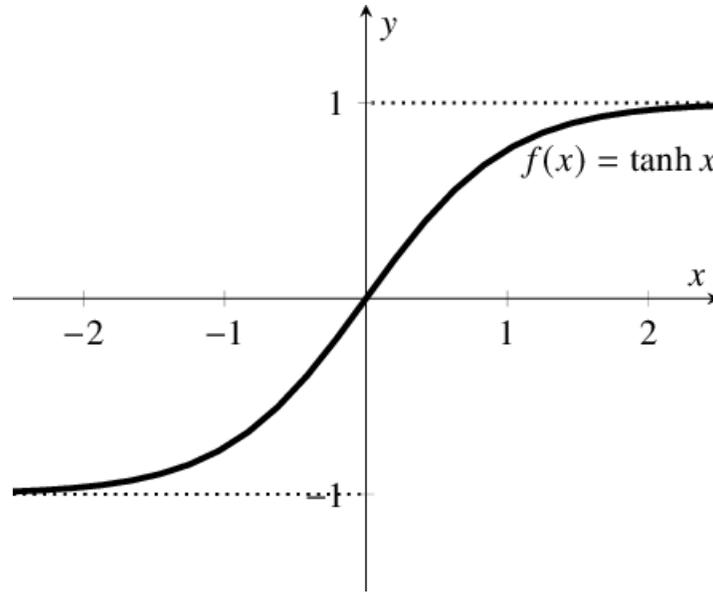


Figure 2.5: Hyperbolic tangent

lighter and efficient by not activating some neurons in the network.

The characteristic of the ReLU is that for negative values of x , this function is zero, which means the gradient zero. This will result in not adjusting the weights of the neurons during backpropagation. So these neurons will never activate again. This gives rise to "dying ReLU" problem.

Leaky ReLU

This function was proposed to solve the "dying ReLU" problem. To solve this, leaky ReLU is designed to have a small negative slope at the point where in ReLU will evaluate to zero, that is, at $x < 0$. The range of the Leaky ReLU is $(-\infty, \infty)$. Usually the slope is set to be at 0.01. If the slope is not at 0.01, then it is called Randomized ReLU. This leak as can be seen in figure 2.7 helps to increase the range of the ReLU function. Also the slope can be made into a parameter of each neuron as seen in Parametric Rectified Linear Unit (PReLU).

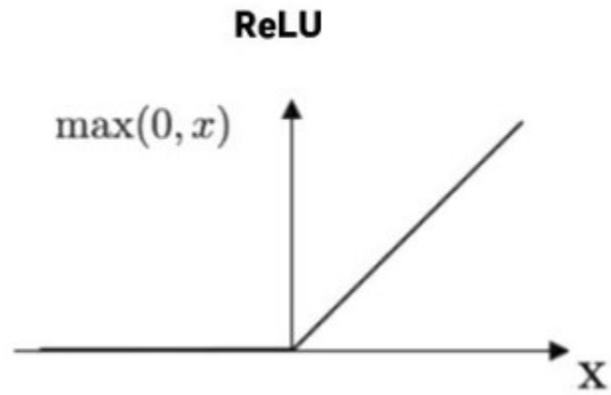


Figure 2.6: Rectified Linear Units

Leaky ReLU is defined as

$$\begin{cases} ax & \text{if } x < 0 \text{ where } a \text{ is } 0.01 \\ x & \text{if } x \geq 0 \end{cases} \quad (2.8)$$

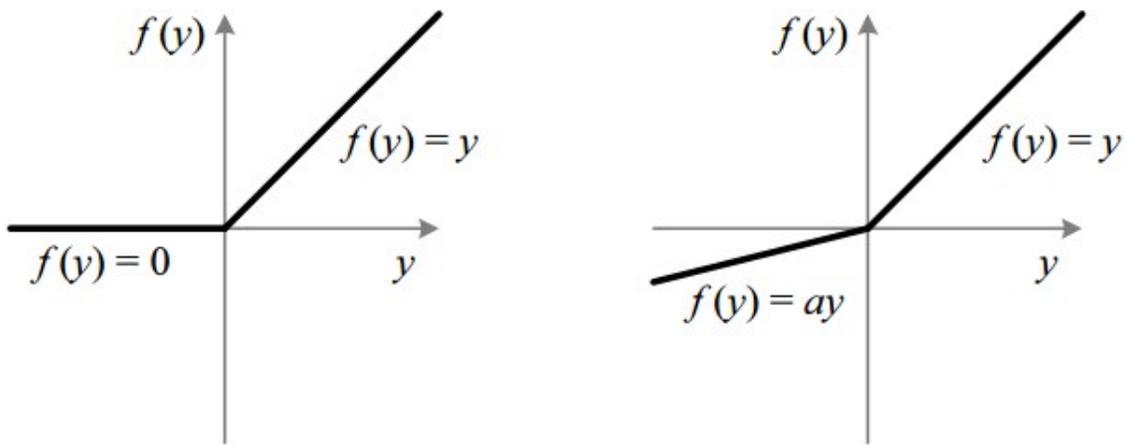


Figure 2.7: ReLU vs Leaky ReLU

2.1.4 Loss Function

Loss function measures the performance of the predicted result. Our objective is to minimize the loss function. The loss is used to calculate the gradients and the gradients are used to update the weights of the neural network. Through this update the neural network is trained.

Mean Squared Error (MSE) is used as the loss function to measure the distance between the predicted result and true result. It is mostly used in regression task. MSE is given by,

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.9)$$

where y_i 's are the true values and \hat{y}_i 's are the predicted values.

Another loss function is **Mean Absolute Error (MAE)**, which is given by,

$$MAE = |y_i - \hat{y}_i|, \quad (2.10)$$

where y_i 's and \hat{y}_i 's are the true and predicted values respectively.

Binary Crossentropy loss is used for Binary Classification tasks. This loss is implemented using the *sigmoid* activation function.

Categorical Crossentropy is also used in multi-class classification tasks. The final layer output is passed through a *softmax* activation so that each node output a probability value between (0-1). The softmax is represented as

$$\sigma(\vec{z}_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \text{ where } z \text{ is the final layer} \quad (2.11)$$

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN), a specialized kind of neural network, is used in processing a sequence of values $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$. This has the ability to process sequences of variable length.

The key idea of recurrent neural networks is to share the parameters across different parts of the model. This sharing plays a crucial role when there is a correlation of data appearing at multiple places in a sequence.

The output of a recurrent network is a sequence, where each member of the output is a function of previous members of the output and the output is obtained from using the same rule applied to the previous outputs.

There are various kinds of recurrent neural network. For examples:

Example - 1 : Recurrent networks with recurrent connections between hidden units produce an output at each time step (Figure 2.8). This maps an input sequence of x values to a corresponding sequence of output o values. Loss L is the measurement of the variation of the output o values with the target values y .

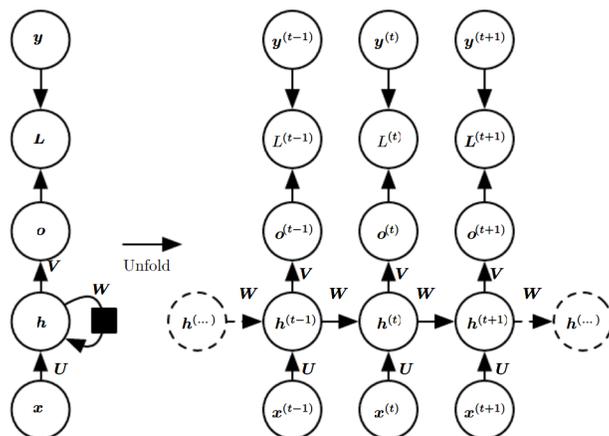


Figure 2.8: Recurrent Neural Network - Type 1

In the figure $\mathbf{x}^{(t)}$ represents the input, $\mathbf{h}^{(t)}$ represents the hidden layer activations, $\mathbf{o}^{(t)}$

represents the output, $\mathbf{y}^{(t)}$ represents the target and $\mathbf{L}^{(t)}$ represents the loss at each time step t .

To develop the forward propagation equations for this recurrent network, we assume the hyperbolic tangent activation function as the activation function for the hidden units. We also assume the output is discrete. The output is the unnormalized log probabilities of each possible targets. Then the softmax operation is applied to obtain a vector $\hat{\mathbf{y}}$ of normalized probabilities over the output. $\mathbf{h}^{(0)}$ is the initial state. And the updated equations are the following:

$$\mathbf{a}^{(t)} = \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \quad (2.12)$$

$$\mathbf{h}^{(t)} = \tanh(\mathbf{a}^{(t)}) \quad (2.13)$$

$$\mathbf{o}^{(t)} = \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \quad (2.14)$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{o}^{(t)}) \quad (2.15)$$

Here, \mathbf{U} , \mathbf{V} and \mathbf{W} represent the input-to-hidden, hidden-to-output and hidden-to-hidden connections respectively. And \mathbf{b} and \mathbf{c} are the bias vector along with the weight matrices.

Example - 2 : Recurrent networks having recurrent connections only from the output at one time step to the hidden units at the next time step to produce an output at each time step (Figure 2.9).

This recurrent network is trained to put a specific output into \mathbf{o} and only the information in \mathbf{o} is allowed to send to the future. The previous \mathbf{h} is connected to the current \mathbf{h} through the output produced by the previous \mathbf{h} . There is a lack of important information from the past unless \mathbf{o} is very high-dimensional. It may be easier to train but this makes the recurrent network less powerful [24].

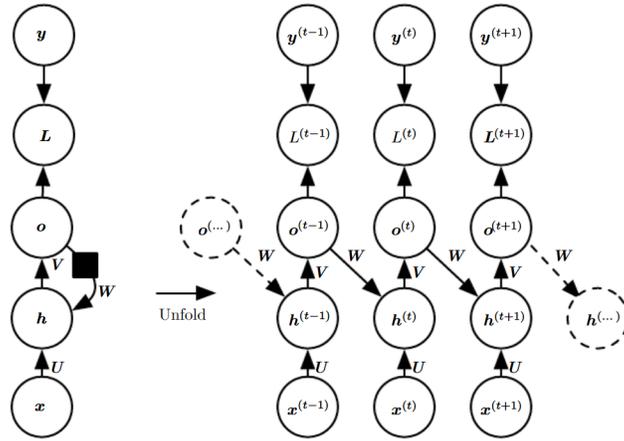


Figure 2.9: Recurrent Neural Network - Type 2

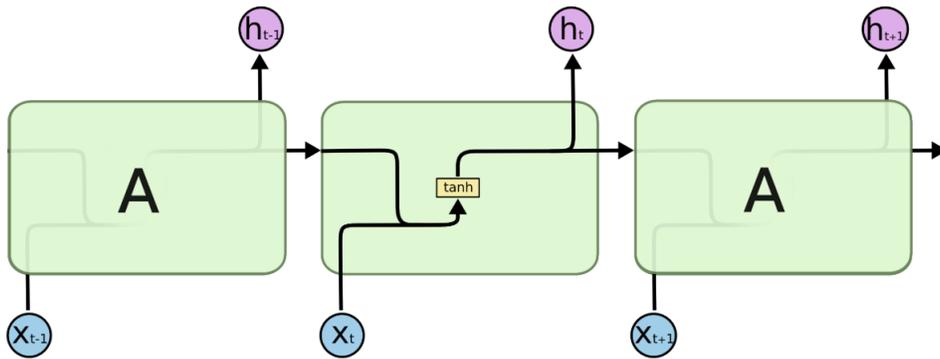


Figure 2.10: Standard RNN

2.2.1 LSTM

Generally, recurrent neural networks have a chain form of repeating modules of neural network [38]. In standard RNNs, the structure of this repeating module is very simple, such as a single hyperbolic tangent layer (Figure 2.10).

But in LSTM, this repeating module has four layers and they are interacting in a very special way (Figure 2.11).

The first step of LSTM is to decide what information it needs to throw away from the cell state. This cell state is the key to LSTM (Figure 2.12). The cell state runs straight down the entire chain, with very little linear interactions. So the information flows along

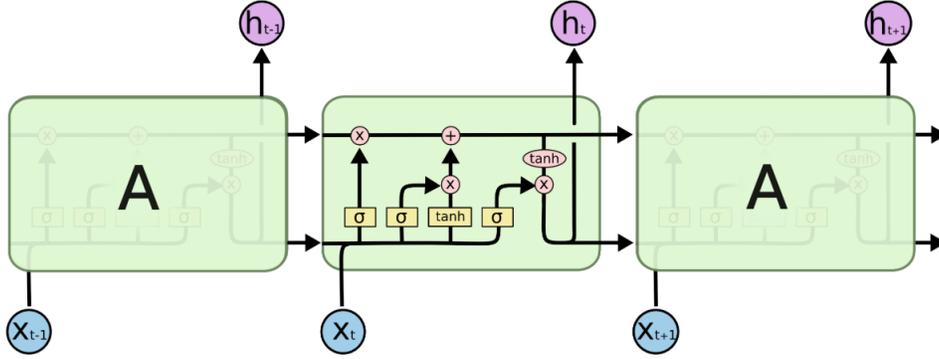


Figure 2.11: LSTM

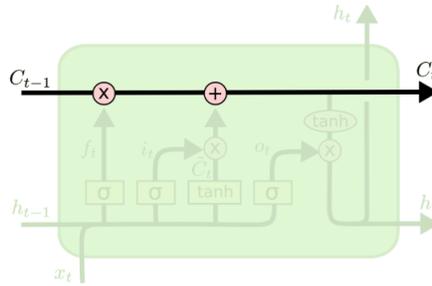


Figure 2.12: Cell of LSTM

it without any change. One of the structure of LSTM, called gates, controls the process of removing or adding information in the cell states. Gates are consisting of a sigmoid layer and a pointwise multiplication operation (Figure 2.13). The sigmoid layer decides how much of each component should go through. The output of sigmoid layer is between 0 and 1, where 0 means no information is going through and 1 means all information are going through. In the first step forget gate layer (Figure 2.14) decides what information needs to be remove from the cell state.

$$f_t^j = \sigma \left(b_f^j + \sum_k U_f^{j,k} x_t^k + \sum_k W_f^{j,k} h_{t-1}^k \right), \quad (2.16)$$

where b_f , U_f and W_f represents the biases, input weights and recurrent weight for the forget gates respectively, x_t is the current input vector and h_t is the current hidden layer vector that contains the outputs of all the LSTM cells.

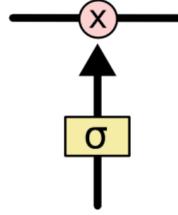


Figure 2.13: Structure of gate of LSTM

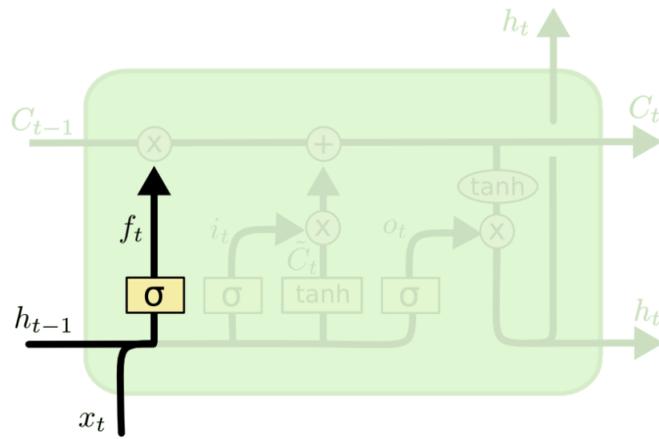


Figure 2.14: Forget gate of LSTM

In the next step, LSTM decides what new information needs to store in the cell state. This is done in two parts - input gate layer decides what values need to update (Figure 2.15) and tanh layer creates a vector of new candidates value, \tilde{C}_t , to update.

$$i_t^j = \sigma \left(b_i^j + \sum_k U_i^{j,k} x_t^k + \sum_k W_i^{j,k} h_{t-1}^k \right), \quad (2.17)$$

$$\tilde{C}_t^j = \tanh \left(b_C^j + \sum_k U_C^{j,k} x_t^k + \sum_k W_C^{j,k} h_{t-1}^k \right), \quad (2.18)$$

where b_i , U_i and W_i represents the biases, input weights and recurrent weight for the input gates respectively. b_C , U_C and W_C are the biases, input weights and recurrent weights for the hyperbolic tangent layer to create the candidate values.

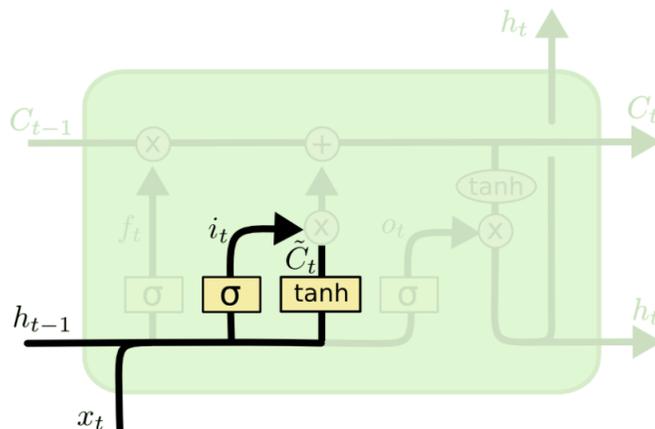


Figure 2.15: Input gate of LSTM

With all these information cell state will be updated (Figure 2.16).

$$C_t^j = \sigma \left(f_t^j * C_{t-1}^j + i_t^j * \tilde{C}_t^j \right), \quad (2.19)$$

The output gate o_t^j that uses a sigmoid unit for gating stops the output h_t^j of the LSTM cell:

$$h_t^j = \tanh \left(C_t^j \right) o_t^j \quad (2.20)$$

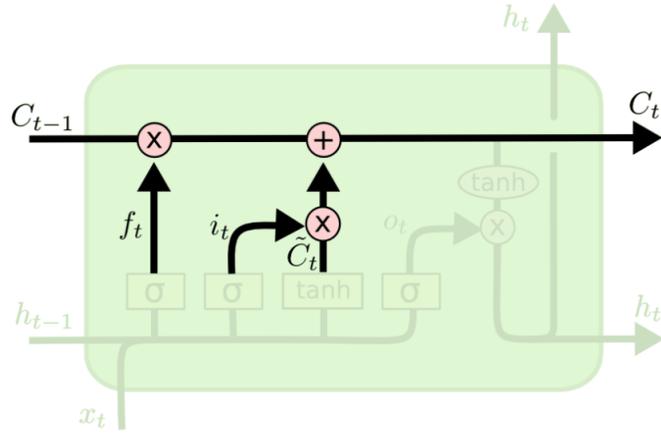


Figure 2.16: Updated state of LSTM

$$o_t^j = \sigma \left(b_o^j + \sum_k U_o^{j,k} x_t^j + \sum_k W_o^{j,k} h_{t-1}^k \right), \quad (2.21)$$

where b_o , U_o and W_o represents the biases, input weights and recurrent weight for the output gates respectively.

Chapter 3

Related Work

3.1 Active region and Solar Flare Prediction

Solar flares associated with energetic charged particles and electromagnetic radiation could affect radio communication, the precision of Global Positioning Systems, and the safety of satellites and astronauts within minutes. Solar flares are of economic importance, therefore steps have been taken to reduce or prevent damages by them. The steps taken include research to understand the origin, processes, and effects of solar flares and development of solar flare forecasting technology.

Research on solar flares and the energies release mechanisms associated with them shows that a high correlation between them and magnetic activity exists [41]. Thus, in understanding and predicting solar flares it is crucial to study the magnetic field configuration of the solar atmosphere. Though there is no direct mapping of the magnetic field in the corona, the magnetic field in photosphere can be mapped. For sometime, most photospheric magnetic field data contained only the line-of-sight component of the magnetic field and those that contained full vector components either contained data with gaps in time or captured only part of the solar disk. The Helioseismic and Magnetic Imager (HMI) onboard the Solar Dynamics Observatory (SDO), which began its mission in 2010, has continuously map the full-disk photospheric vector magnetic field [46]. From the start of its operation in May, 2010, HMI has mapped the vector magnetic field every 12 minutes 98.44% of the time [18].

Most studies done so far in solar flare prediction use photospheric magnetic field data to parameterize active regions (ARs). In the parameterization, some use the energy, helicity,

currents, and shear angles [16,23,28,34], while some also use the magnetic field topology [47] and others use the integrated Lorentz force exerted by an AR [13]. The ultimate goal of these parameterization is to find a connection between the behavior of the photospheric magnetic field and solar activity, which usually take place in the chromosphere, and the transition region of the solar corona.

The relationship that exist between the photospheric and coronal magnetic field when a solar flare happens is not fully understood. Due to this, flare prediction has been based on statistical and machine learning approaches to find relationships rather than on theoretical models from which causal relationships could be obtained [6].

Most machine learning approaches have posed the problem as a binary classification problem, where an active region belongs to the positive class if it leads to one or more flares in a given time frame or it belongs to the negative class otherwise. Li et al. [30], Qahwaji et al. [42], Song et al. [50], Yu et al. [59], Yuan et al. [60], Ahmed et al. [1] used nonlinear Machine Learning (ML) algorithms to forecast solar flares. These studies were made using line-of-sight magnetic field data, solar radio flux or metadata (eg., McIntosh class, sunspot number) to characterize their features. Leka et al. [27] started the use of vector magnetic field data for flare prediction by using vector magnetic field data from the Mees Solar Observatory Imaging Vector Magnetograph with a prediction technique known as discriminant analysis, which is a linear classifier. Other prediction models used so far include the superposed epoch analysis [32], statistical analyses [5, 12] and support vector machine (SVM) [6, 30, 37].

These forecasting methods usually utilize morphological or physical parameters extracted from active regions. Research has been ongoing to find out the best and effective predicting parameters to be used in training statistical or machine learning models. The idea of sunspots classification as proposed by McIntosh et al. [33] led Lee et al. [26] to examine the relationship between classifications of sunspots and solar flares. From the findings obtained so far, it is known that the physical parameters such as the length of the neutral line [11], the gradient of the magnetic field [8], the highly stressed longitudinal magnetic

field [19], the distance between active regions and predicted active longitudes [19], and the Zernike moment of magnetograms [43], provide better predictive capabilities when used in predictive models.

Moreover, the changes in the physical parameters in active regions were analyzed by Yu et al. [59], and Huang et al. [20]. Apart from that, Wheatland et al. [58] proposed that the history of solar flares is a significant parameter in predicting subsequent flares. Despite the use of all these parameters, most of the information obtained from active regions show that they were not much different from each other and therefore they did not perform better than the others when used in solar flare predicting models [4, 28]. The extraction of the effective predicting parameters from ARs has become a bottleneck to improving the accuracy of solar flare predicting models.

The physical and morphological parameters show a weak relationship with solar flare. However, one field in machine learning, which has gain ground due to the increase in computational resources, has created the opportunity to fish out effective predicting parameters from data with little to no human intervention. This field is known as deep learning [2, 17, 24]. Therefore Nishizuka et al. [36] proposed a deep neural network for solar flare prediction which they called Deep Flare Net (DeFN). This architecture utilize manually calculated features selected from automatically detected sunspots. The detection of sunspots were done by setting the threshold value to 140 G in addition to rules outlined by [37]. They evaluated their results using the operational setting, that is, the chronological splitting of the dataset into training and testing. They achieved a better TSS (True skill statistic) as compared to other machine learning models like SVM, KNN (k-nearest-neighbor) and ERT (extremely randomized trees).

Active regions change from their time of emergence to disappearance. They disappear either by dying out or moving out of the face of the visible Sun disk. Appearance changes are complex spatiotemporal non-stationary processes, since the joint distribution of adjacent pixel values changes in both space and time [40]. These processes can be broken down into deterministic and stochastic components. Research works in neural networks have explored

spatiotemporal prediction from these two aspects within the context of video classification and next frame prediction. Modeling such features has been done using Convolutional Neural Networks (CNN) [22] and Recurrent Neural Networks (RNN) [53].

3.2 Next Frame Prediction

Next frame prediction combines two deep learning techniques, namely *predictive learning* and *image generation*. In *predictive learning*, we seek to model the different possibilities of the future by using information from the past. By using recurrent networks, patterns in sequence data are captured and used as basis for predicting the future. In *image generation*, new images are generated by extracting attributes from an existing dataset to create new images. The two most commonly used networks in image generation are autoencoders and Generative Adversarial Networks (GANs) [15]. The autoencoder is made up of the encoder part and the decoder part. The encoder takes the image and encodes it into a latent variable, whilst the decoder takes the latent variable to recreate the original image. GANs are composed of two parts, the generator model and the discriminator model. The generator model generates new images that look real, whilst the discriminator classifies the generated images and the samples from the generator model as either real or fake.

Ranzato et al. [44] proposed using a recurrent model to predict frames in a discrete space of patch clusters. Then, following the success of sequence-to-sequence mapping in language modeling using Long Short-Term Memory (LSTM) networks, Srivastava et al. [52] adapted the approach to video prediction. Shi et al. [49] introduced the integration of the convolution operator into recurrent state transition functions and proposed the Convolutional LSTM. With Convolutional LSTM as the base idea, Villegas et al. [54] and Patraucean et al. [39] developed recurrent models to leverage optical flow guided features. Kalchbrenner et al. [21] presented the idea of encoding the time, space, and color structures of videos. The proposed network called Video Pixel Network (VPN) produced good predictions but had a high computational complexity. The PredNet model proposed by Lotter et al. [31] used

Convolutional LSTM units in a top-down and bottom-up approach to create next frame prediction of a video sequence. Wang et al. [55] added the zig-zag memory flow to the Convolutional LSTM, which increased modeling capability for short-term video dynamics.

These deep learning models stated above developed for next frame prediction can be divided into two architectures based on how prediction of future images or frames is done; they are sequence-to-one and sequence-to-sequence architecture. The first group, takes a sequence of images, say from time step t to $t + h$, and makes a frame/image prediction of the next time step, $t + h + 1$. This architecture primarily focuses on the spatial structure of the input frames. In the second architecture group, sequence-to-sequence architecture, temporal frames are inputted into the model separately where frame at time step t is fed into the network and prediction is made for time step $t + 1$. This is continued until the frame at a specified time step is achieved. This architecture focus primarily on temporal sequence. Autoencoders are widely used in these two types of architectures.

The recurrent model developed by Villegas et al. [54] is an example of sequence-to-one architecture which was able to predict up to 128 frames into the future. PredNet [31] is an example of sequence-to-sequence architecture which can make a prediction of up to 5 frames into the future.

Chapter 4

Data and Methodologies

4.1 Data

In this work, we utilize the data produced by the Helioseismic and Magnetic Imager (HMI) on board NASA's Solar Dynamics Observatory (SDO). The NASA's SDO is a satellite launched in 2010 under Living With a Star (LWS) program to observe and record the Sun's activities. The three instruments on board this satellite used in monitoring the Sun are:

- The Atmospheric Imaging Assembly takes pictures of the Sun at the following wavelengths 94, 131, 171, 193, 211, 304, 335, 1600, 1700 and 4500 Å at a resolution of 4096×4096 pixels (approximately 1 arcsec).
- The Helioseismic and Magnetic Imager (HMI) which captures the oscillations and magnetic field of the Sun with a resolution of 4096×4096 pixels (pixel size of 0.5 arcsec) [46].
- The Extreme Ultraviolet Variability Experiment (EVE) measures the Sun's extreme ultraviolet spectral irradiance from 1 to 1050 Å.

Images produced by the HMI, AIA, and EVE instruments can be accessed at the Joint Science Operations Center (JSOC) at Stanford University, Lockheed Martin Solar & Astrophysics Laboratory (LMSAL) and the Laboratory for Atmospheric and Space Physics (LASP) at the University of Colorado, Boulder.

As of late 2020, there were over four thousand eight hundred refereed scientific publications using data from SDO. This great scientific contribution can be attributed to the reliability of the spacecraft and its instruments since it launched over ten years ago, the consistency and quality of the data captured, the mission’s open data policy, and the tools readily available by the various science data centers and science institutions to researchers and anyone interested in studying the sun [14].

4.1.1 HMI Instrument

SDO/HMI is a space-based instrument designed to succeed the Michelson Doppler Imager on the Solar and Heliospheric Observatory. This instrument measures the oscillations and magnetic field of the Sun. Both measurements are taken from the photosphere providing a full-disk photospheric vector magnetic field data. Processing of the data from HMI is done at JSOC. Documentations, tools and APIs needed to access and work on the data stored at JSOC are available to aid people in their research. APIs such as Astropy and Sunpy designed for the python programming environment are examples of the resources to help researchers work with the data from the HMI instrument.

Two camera setups in the HMI instrument capture full-disk images of the Sun after the images have moved through a series of bandpass filters. A 12-filtergram is created every 45 seconds from the two cameras. One camera measures the six polarization states whilst the other measures the right and left circular polarization for six different wavelengths.

Before the filtergrams get to JSOC, they are first sent to a ground station at White Sands, New Mexico after the data has been processed at the satellite level. At White Sands, further processing is done before the data is forwarded to JSOC. When the data arrives at JSOC, it is combined with flight dynamics data of the satellite to generate calibrated Level 1 filtergrams. From this set of filtergrams, four data products are produced, namely, continuum filtergrams, dopplergram, Line-of-Sight(LoS), and vector magnetograms. Using the LoS observables code, HMI observables such as the Doppler velocity and LoS magnetic field strength are calculated. Couvidat et al [7] talks of a MDI-like algorithm for computing

the LoS observables. The Line-of-Sight magnetograms has a cadence of 12 minutes.

In a magnetogram, grey areas indicate that there is no/weak magnetic field, while black and white areas indicate regions where there is a strong magnetic field. The grey-black-white is a convention normally used by astrophysicists. The darkest areas are regions of "south" magnetic polarity (inward directed, moving toward the center of the Sun, negative) and the whiter regions "north" (outward directed, moving toward us, positive) polarity.

From the time of the launch of the SDO mission, the HMI instrument has not stopped working, producing approximately one terabyte of data per day. The times during which flow of data has paused includes the eclipse of the Sun by the Earth, poor weather conditions at the ground station, ground equipment failures, and maintenance.

4.2 Data Processing

In order to carryout the experiments, we curated a dataset which ranges from May 2010 to December 2018 from the dataset that was preprocessed and made available by Galvez et al [14]. Preprocessing included elimination of corrupt frames, correction for orbital variations and sensor degradation, and downsampling from the original image size of 4096×4096 to 512×512 . The date range was chosen because it covered almost one complete Solar Cycle which is Cycle 24 as shown in figure 4.1. We used the LoS magnetograms portion, that is, the data obtained from Helioseismic Magnetic Imager (HMI). Regions of size 96×128 centered above the equator were cut out of the magnetograms figure 4.2.

Therefore in creating a sequence of say 10 magnetograms, a missing magnetogram makes that sequence invalid. Since the size of the sequence is arbitrary, the number chosen determines the number of sequences you can generate from the dataset. In our experiments, we use a sequence of length 10. We used two different scaling functions along with three different time interval when processing the data. Therefore for each time interval, there were two different scaling used, namely:

International Sunspot Number

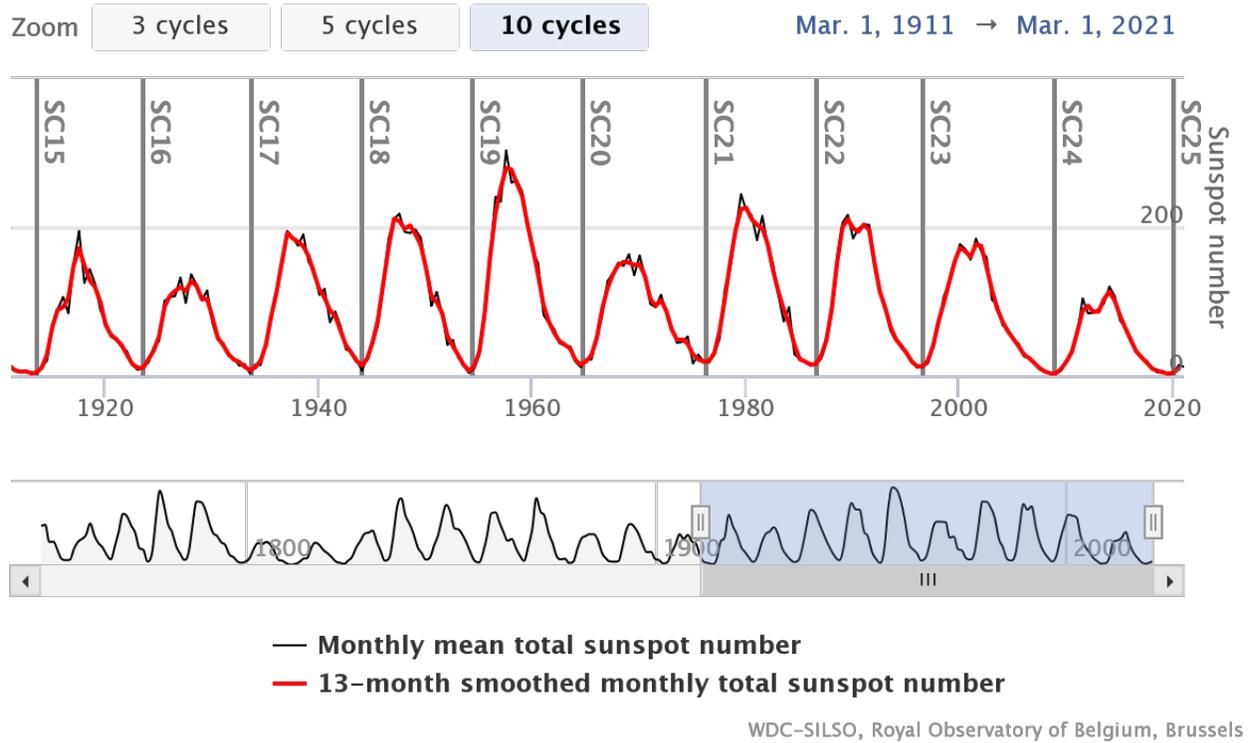


Figure 4.1: Graph of the number of sunspots for Solar Cycles ranging from March 1911 to March 2021

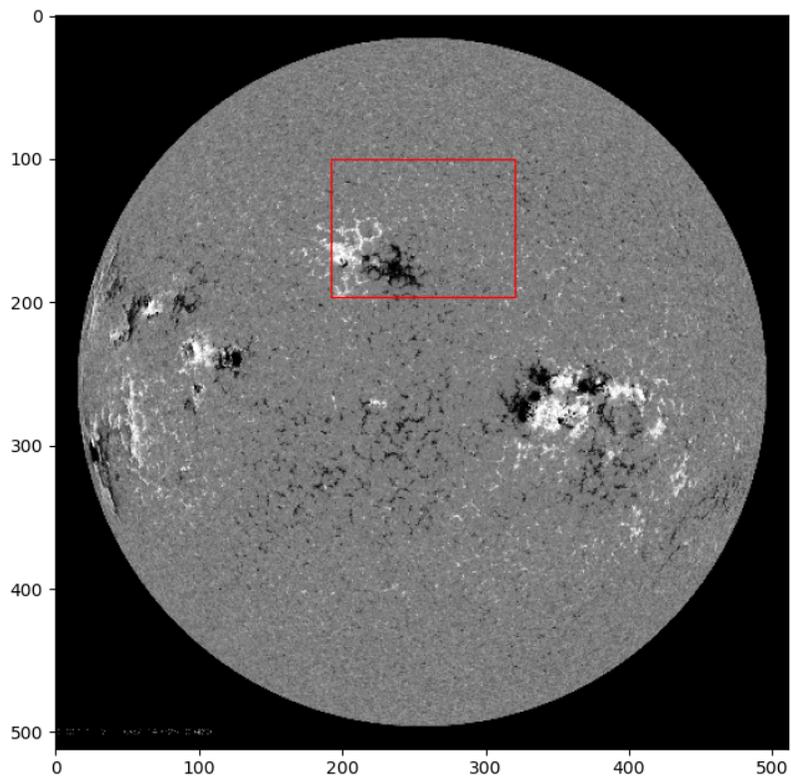


Figure 4.2: Region cutout shown with the red rectangle

- Minimum-Maximum Normalization (Min-Max Norm) - Using the minimum and maximum values of the images to normalize them to 0 – 1 range as shown in equation (4.1).
- Sigmoid Normalization (Sig-Norm)- Using the sigmoid function to normalize them to 0 – 1 range as shown in equation (4.2).

The operations done on an image, X , in equation (4.1) and equation (4.2) are carried out in an element-wise fashion. Since the images are of size 96×128 , the matrices returned in the two equations are of the same size as the image. The value of k is set to be 0.03 in equation (4.2). And the maximum pixel value of the images is known to be as high as 5000 [46] therefore b is set to be 5000 whilst a , representing the minimum value, is set to be -5000 .

$$\text{min_max} = \frac{X - a}{b - a}. \quad (4.1)$$

$$\text{sig_norm} = \frac{1}{1 + e^{-X*k}} \quad (4.2)$$

We applied different values of k in equation 4.2 to see how the distribution of the data were affected. Two magnetograms were randomly chosen, one from the solar minimum and one from the solar maximum. We then picked two values of k , 0.01 and 0.001 with a stepsize of 0.01 and 0.001 respectively. Now, to select which value to use for k , we consider the graph in figures 4.3 and 4.4 by focusing more on the graph from the solar minimum. Specifically, we chose 0.03 since there was a desirable bell-shape characteristic and small changes to the extreme values (0, 1) but with values from 0.001 to 0.007 there was not an appreciable change in the distribution of the pixel values as seen in figures 4.5 and 4.6.

Also in processing the data, we used the following time intervals:

- input = 1 hour; output = 1 hour

hmi.M_720s.20140225_004800_TAI.1.magnetogram.fits

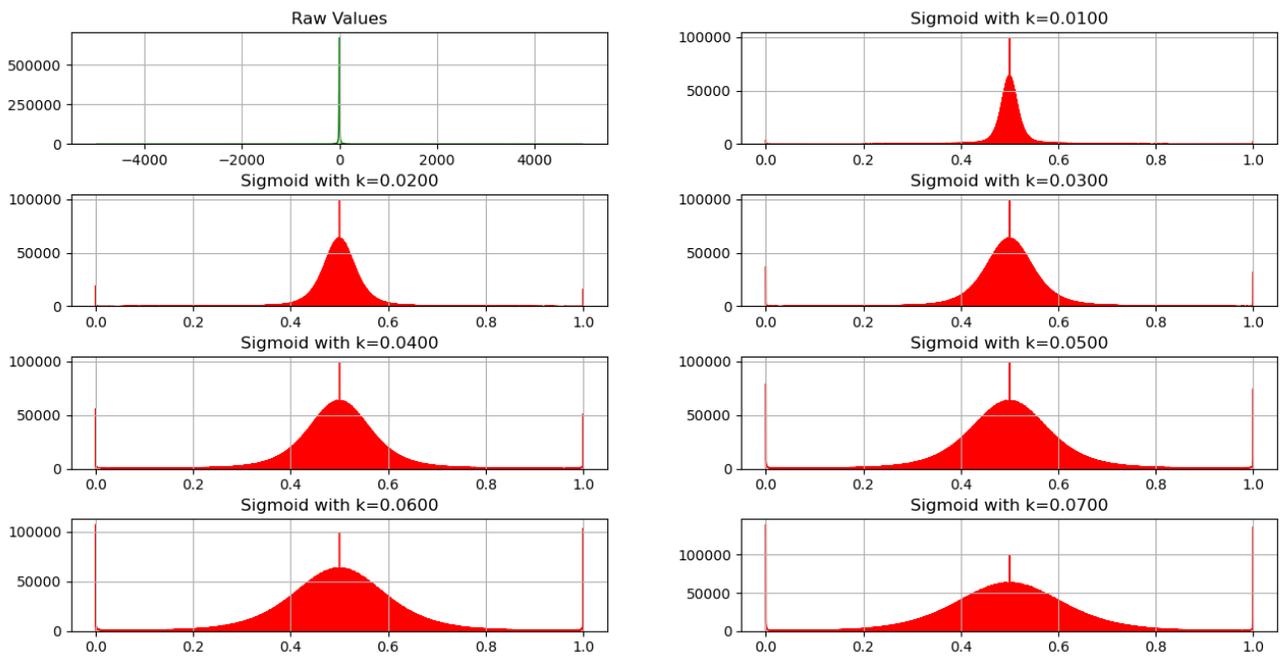


Figure 4.3: Plot of distribution of pixel values of image from solar maximum after transformation using k ranging from 0.01 – 0.07.

hmi.M_720s.20100501_001200_TAI.1.magnetogram.fits

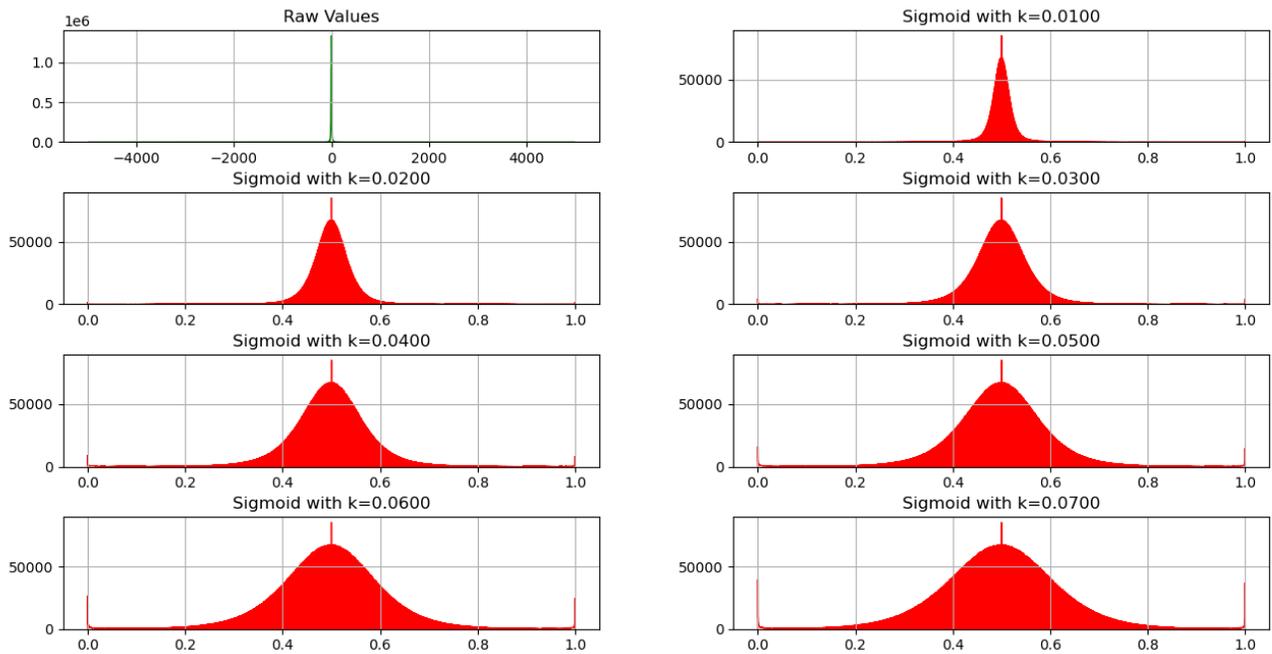


Figure 4.4: Plot of distribution of pixel values of image from solar minimum after transformation using k ranging from 0.01 – 0.07.

hmi.M_720s.20140225_004800_TAI.1.magnetogram.fits

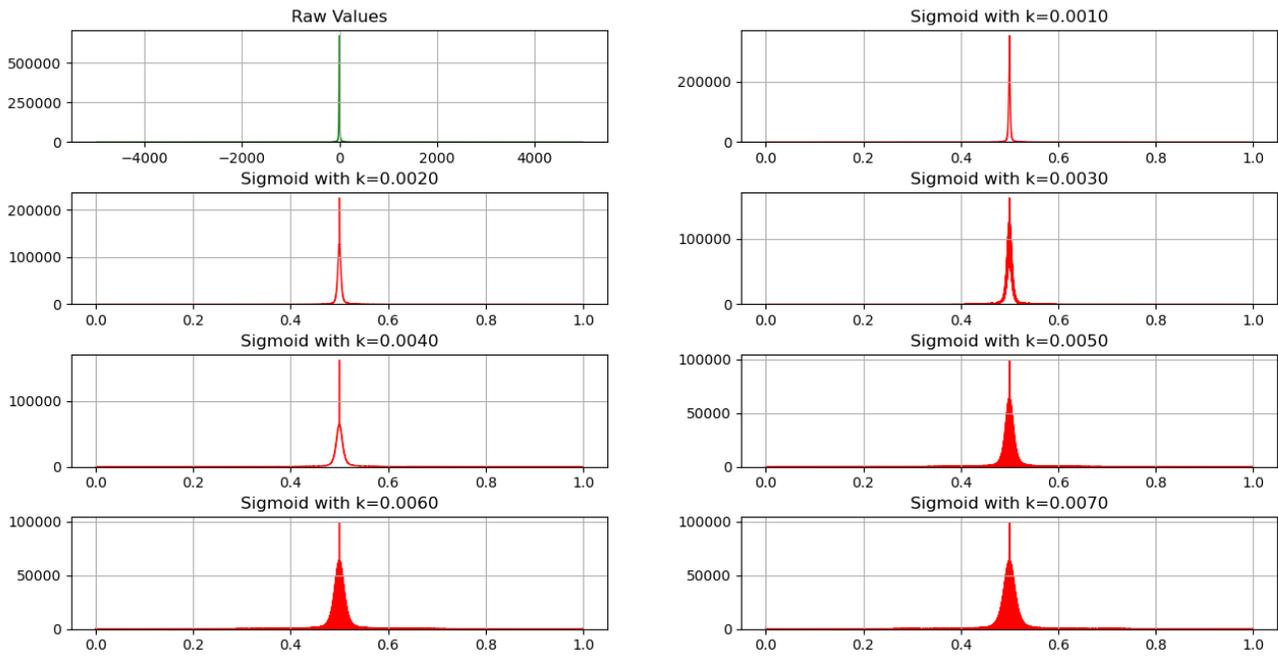


Figure 4.5: Plot of distribution of pixel values of image from solar maximum after transformation using k ranging from 0.001 – 0.007.

hmi.M_720s.20100501_001200_TAI.1.magnetogram.fits

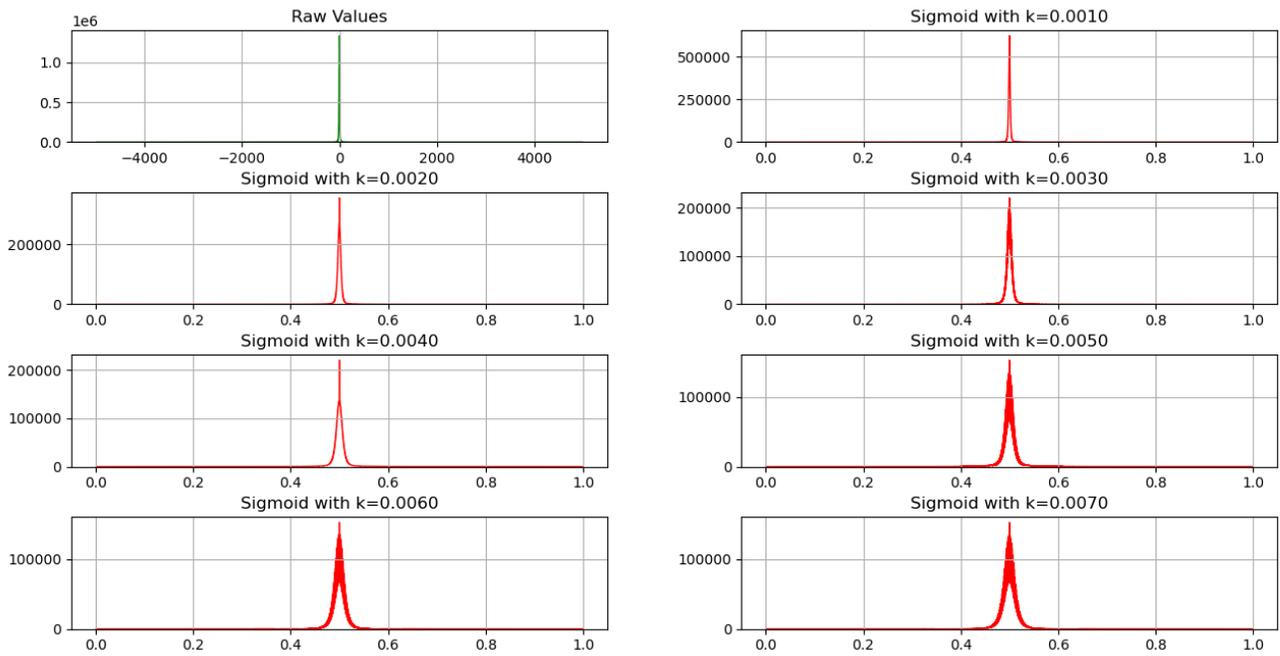


Figure 4.6: Plot of distribution of pixel values of image from solar minimum after transformation using k ranging from 0.001 – 0.007.

- input = 3 hour; output = 3 hour
- input = 6 hour; output = 6 hour

In order to maintain the length of the sequence to be 10, we increased the cadence of the dataset to 36 minutes and 72 minutes for the time interval *3 hour input; 3 hour output* and *6 hour input; 6 hour output* respectively. For the *1 hour input; 1 hour output* we kept the 12 minutes cadence used in the original dataset. The dataset curated based on the parameters above gave us six different datasets to perform our experiment. Table 4.1 shows the number of sequences generated for each time interval.

Table 4.1: Number of Sequences

Input/Output	Training	Testing	Validation
1 Hour / 1 Hour	161880	76457	17040
3 Hour / 3 Hour	39330	18442	4140
6 Hour / 6 Hour	9120	4050	922

Each dataset was subdivided into training, testing and validation using contiguous sampling figure as illustrated in figure 4.7. For example, for 1000 sequences generated, the first 20 sequences are placed in the training portion, the next 5 in the validation portion and the next 10 after that in the testing portion, and the process repeats until all the 1000 sequences are partitioned. This was done to capture the changes on the Sun as it moves from solar minimum to solar maximum in each partition and also prevent leakages of information from the testing portion into the training portion.



Figure 4.7: Contiguous Sampling from 2011 to 2013

4.3 Models

4.3.1 Memory In Memory Network (MIM)

Memory In Memory (MIM) Network is a variant of RNN architecture and also sequence-to-sequence architecture. It was designed as an update to Predrnn [55] to solve the problems such as saturation that occurs in forget gates.

To solve the problem of saturation, memory units are arranged in a cascaded fashion which also helps in modeling the non-stationary variations. The MIM network is composed of two main parts, that is, the non-stationary module (MIM-N) and the stationary module (MIM-S) illustrated in figures 4.8 and 4.9 respectively. These two parts together form the MIM-block. Two consecutive hidden states serving as inputs to the MIM-N module (H_t^{l-1} and H_{t-1}^{l-1}) are differenced to capture the non-stationary variations. This leads to the generation of D_t^l as seen in figure 4.8. In the other module, MIM-S, the output D_t^l of the MIM-N module is taken as input together with the outer temporal memory C_{t-1}^l to represent the approximate stationary variations in spatiotemporal sequences.

S and N denote the horizontally-transited memory, D represents the differential features and C represents the memory cells. The following shows the calculations taking place in MIM-N and MIM-S:

MIN-N

$$g_t = \tanh(W_{xg} * (H_t^{l-1} - H_{t-1}^{l-1}) + W_{ng} * N_{t-1}^l + b_g)$$

$$i_t = \sigma(W_{xi} * (H_t^{l-1} - H_{t-1}^{l-1}) + W_{ni} * N_{t-1}^l + b_i)$$

$$f_t = \sigma(W_{xf} * (H_t^{l-1} - H_{t-1}^{l-1}) + W_{nf} * N_{t-1}^l + b_f)$$

$$N_t^l = f_t \odot N_{t-1}^l + i_t \odot g_t$$

$$o_t = \sigma(W_{xo} * (H_t^{l-1} - H_{t-1}^{l-1}) + W_{no} * N_t^l + b_o)$$

$$D_t^l = \text{MIM-N}(H_t^{l-1}, H_{t-1}^{l-1}, N_{t-1}^l) = o_t \odot \tanh(N_t^l),$$

MIM-S

$$g_t = \tanh(W_{dg} * D_t^l + W_{cg} * C_{t-1}^l + b_g)$$

$$i_t = \sigma(W_{di} * D_t^l + W_{ci} * C_{t-1}^l + b_i)$$

$$f_t = \sigma(W_{df} * D_t^l + W_{cf} * C_{t-1}^l + b_f)$$

$$S_t^l = f_t \odot S_{t-1}^l + i_t \odot g_t$$

$$o_t = \sigma(W_{do} * D_t^l + W_{co} * C_{t-1}^l + W_{so} * S_t^l + b_o)$$

$$T_t^l = \text{MIM-S}(D_t^l, C_{t-1}^l, S_{t-1}^l) = o_t \odot \tanh(S_t^l),$$

The MIM network is created by stacking multiple MIM blocks together to help create a representation for the spatiotemporal process and increase the accuracy of future frame prediction. Figure 4.10 shows how the MIM blocks are setup to create the network. The hidden states H_{t-1}^{l-1} and H_t^{l-1} are sent to MIM block at timestamp $t \neq 1$ and layer $l \neq 1$ to generate the differenced features for further processing and are shown as green arrows in figure 4.10. The spatiotemporal LSTM (ST-LSTM) is used to generate the hidden representations in the first layer since there is no layer before that. Also by differencing temporally adjacent hidden representations, the non-stationarity of the sequence becomes more evident as the spatiotemporal dynamics in local areas are encoded into the hidden states through the bottom ST-LSTM layer [56].

The learned stationary and non-stationary features by the MIM-N and MIM-S are sent along the yellow arrows in figure 4.10. The MIM network produces one frame at one

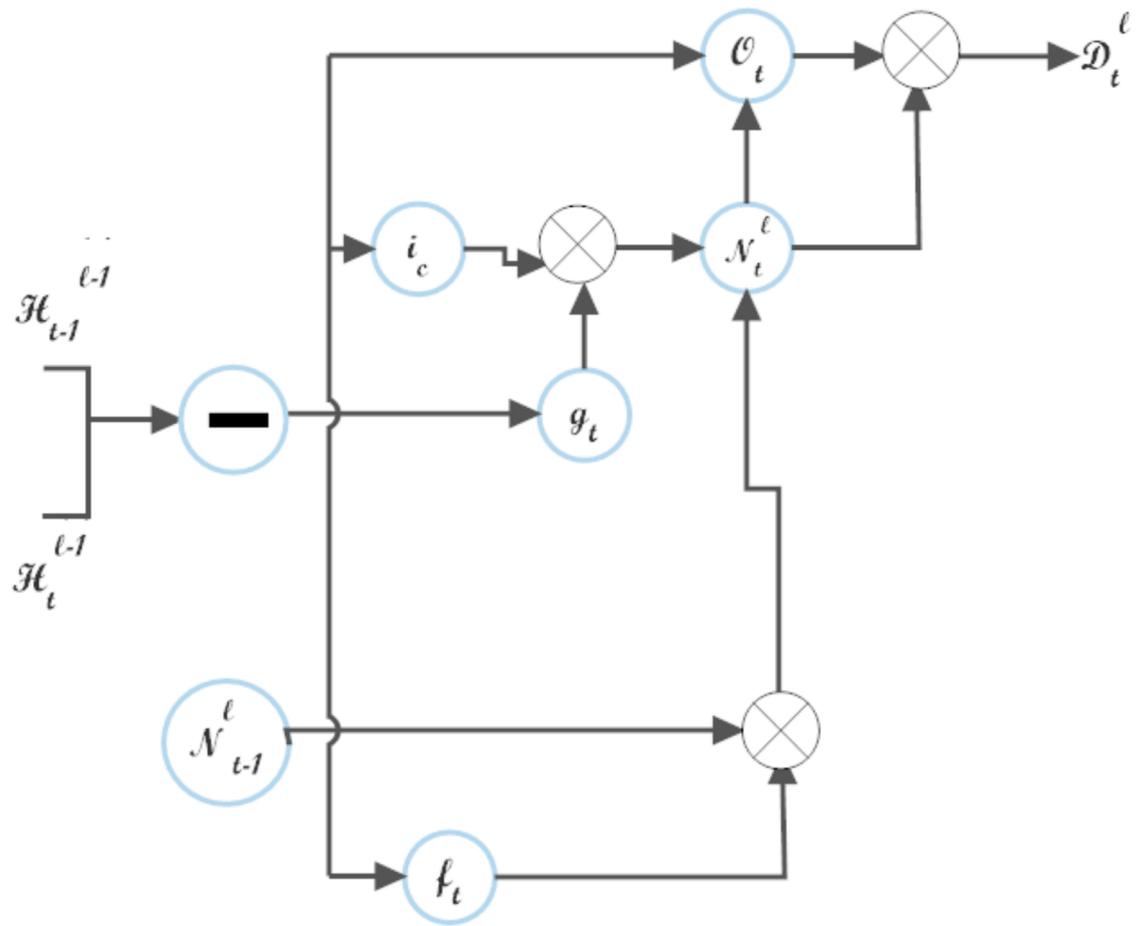


Figure 4.8: Schematic of MIM-N [56]

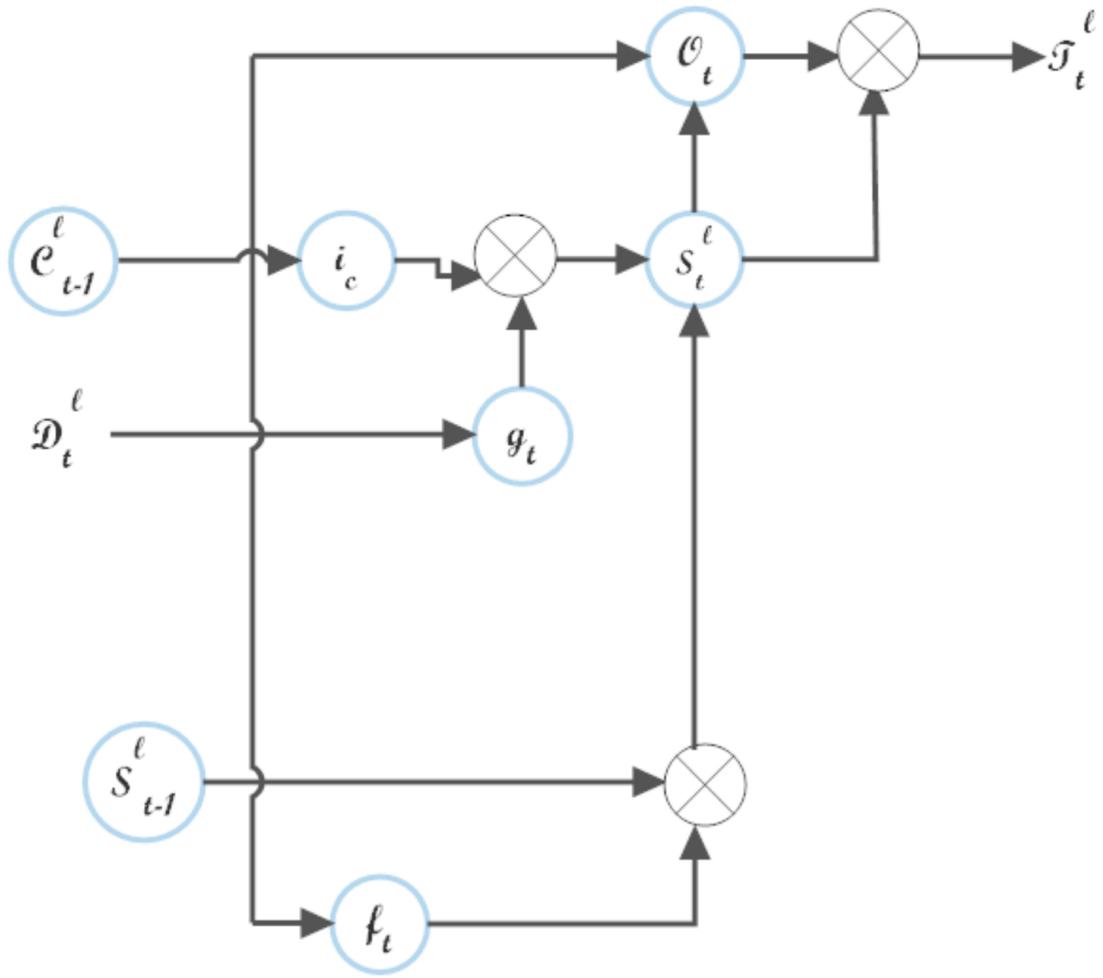


Figure 4.9: Schematic of MIM-S [56]

timestamp. The green arrows in figure 4.10 show the diagonal state transition paths of hidden representations for differential modeling. The black arrows in figure 4.10 represent the zigzag state transition paths of the memory module. The input to the network as shown in figure 4.10 can either be the ground truth frame for input sequence or the generated frame at previous timestamp [56].

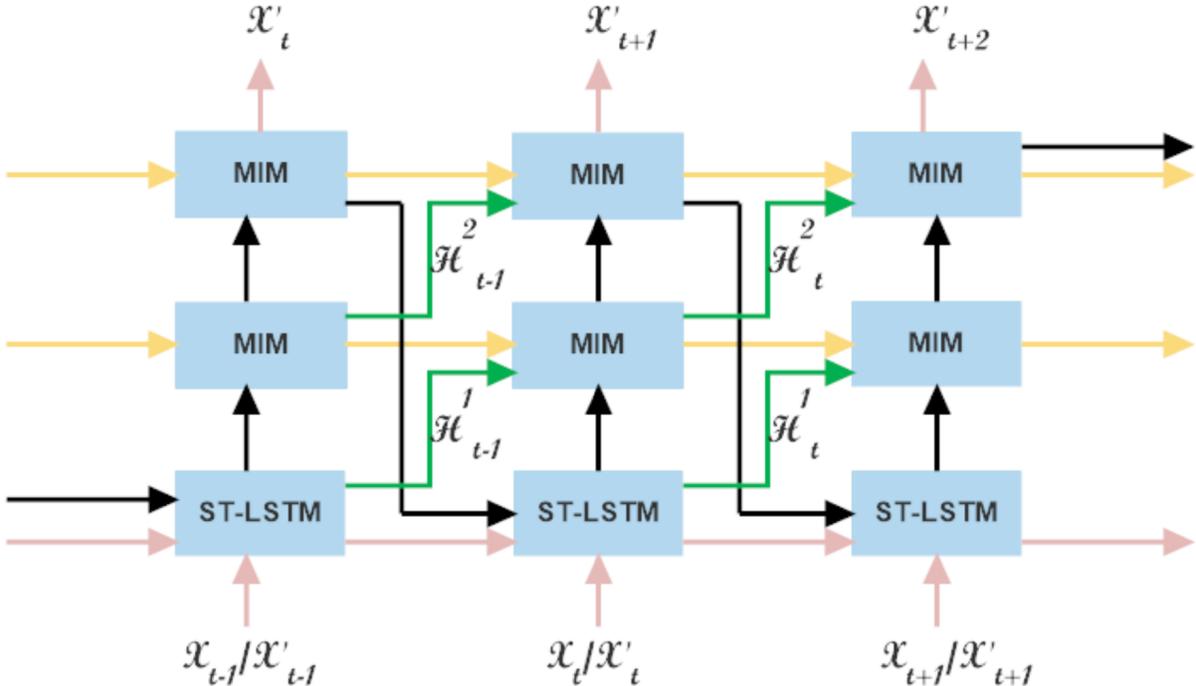


Figure 4.10: A MIM network with two MIMs and one ST-LSTM. [56]

4.3.2 Baseline

Since this work is the first of its kind, we could not compare our work against others. We therefore utilized a simple baseline by which we can compare our work. Based on the complexity of the spatiotemporal dynamics on the surface of the Sun, that is, the rotation and the changes of the magnetic field on the Sun, our baseline is to repeat the last input frame as the predicted image for the output sequence. Basically, we used a no-change

baseline where the predicted images are the same as the last input frame.

4.3.3 Regression

Additional to the no-change baseline, we use the regression model to model the spatiotemporal changes on Sun. To do this, we pick the first rows of the sequences in the training set, then we trained a regression model to map the inputs (the first five rows) to the output (the last five rows). After this training, we apply it to the first row of the input testing portion of the dataset. This done for each row till the last row for a total of 96 rows.

Chapter 5

Preliminary Results

We first applied the no-change baseline on the testing portion for each dataset and we trained the regression model and the Memory In Memory (MIM) Network using the training portion of each dataset and compare their performance against each other. We use the structural similarity index measure (SSIM) [57] and the mean square error (MSE) averaging over their respective test sets for evaluation. MSE gives an indication of how different two images are and SSIM provides a quantitative evaluation of the similarity of two images. Therefore a lower MSE and a higher SSIM indicates a better prediction. The time interval (cadence) between say $t = 1$ and $t = 2$ as presented in the results is 12 minutes. Subsequently, $t = 3$ and $t = 6$ presents a time interval of 36 minutes and so on.

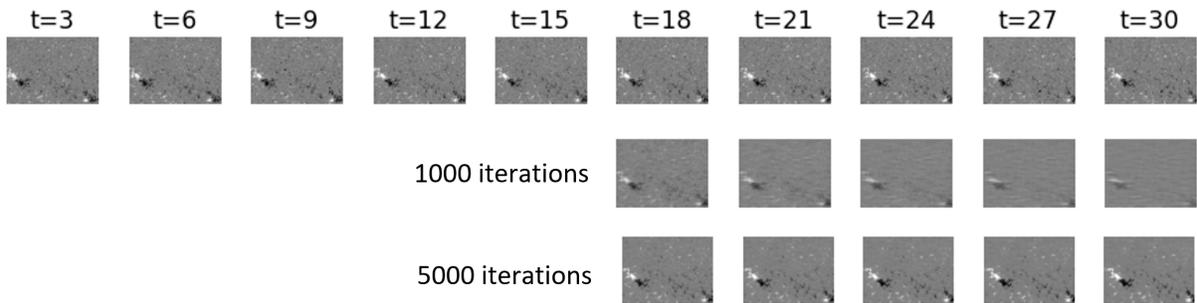


Figure 5.1: Sequence tested at different iteration step

Changing the batch size to 1, total length of the sequence to 10 and input length to 5 and keeping all other parameters same as used by [56] including the L1 loss function, we trained the MIM network for 170000, 160000 and 80000 iterations for 1 hour, 3 hour and 6 six hour time interval respectively on a single NVIDIA GeForce RTX 2080 Ti and it took about 3 to 8 days to finish training. The difference in the number of iterations is

due to the difference in the number of sequences available for each time interval and one iteration represent a pass through of one sequence through the network. Figure 5.1 shows a validation of the network after 1000 and 5000 iterations during training of the network using the sig-norm 3 hour time interval dataset.

Also, the figures 5.1, 5.2, 5.3 and 5.4 shown have been color-coded with 1 being white and 0 being black to help visualize the changes in images both in the ground truth and the predicted one. The ground truths are presented in the first row, whilst the predicted ones are shown in the second and/or third row.

Upon visual inspection in figure 5.3 for example, it can be seen that the sig-norm images show good detail on what is happening and produced good prediction visually though it performed poorly against the min-max Norm in all reported metrics.

Deep neural network tend to work well when images are normalized to the 0-1 range. Since the values in the images ranges from -5000 to 5000 , we apply the min-max norm to normalize them. But the distribution of the pixel values are not evenly distributed since most values lie between the $0.4 - 0.6$ range. This brings about the problem where values appear similar especially during training when values are round-off. Therefore the network predicted that nothing happened in the image predicting similar pixel values centered around the midpoint (0.5). In figure 5.2, that is the min-max norm for time interval of 6 hours, this problem becomes more evident as the network fails to make a good prediction. In this figure 5.2, gray images (almost no visual changes) were predicted which shows that the network predicted the images to have pixel values within a very small range. This led to the application of sig-norm (equation(4.2)) to help solve this issue. Based on the results as seen in Figure 5.3, that is, sig-norm for 6 hour time interval, sig-norm help solve this issue in normalization.

For the results obtained for MIM network in tables 5.1, 5.2, and 5.3, the SSIM for min-max norm is close to 1 and shows slight improvement over the baseline but in their corresponding MSEs there is over 43% improvement.

In terms of the regression model used, it perform significantly better than the baseline

both in MSE and SSIM but only in MSE when it is compared against MIM. This good performance in MSE can be attributed to the fact that regression model seek to minimize the mean square error whilst the MIM minimize L1 error. There is a slight deviation in performance in table 5.3 where the regression model did better than MIM in SSIM by 0.0002. Figure 5.4 shows a prediction made with the trained regression model. This figure is the same test sample used in figure 5.3. Comparatively, figure 5.3 shows greater detail especially around the upper areas and the sunspot area at the bottom right of the predicted images.

Since MIM network was created and trained for perceptual task, it shows this characteristic in the SSIM metric where it did better than the baseline and the regression model in almost all the results.

In figures 5.2 and 5.3 for the 6 hour interval, the test sample captured a sunspot. From the ground truth row in all the figures, it can be seen that the sunspot is moving towards the right(rotation of the Sun). This non-stationarity is captured by the MIM network and it is seen in the row showing the predicted images.

In the SSIM and MSE metrics under sig-norm in tables 5.1, 5.2 and 5.3, the MIM network and the regression model did well not to deviate by higher margins as the time intervals were increased.

Table 5.1: One Hour Input to Predict the Next One Hour

Model		Min-Max Norm		Sig Norm	
		<i>MSE</i>	<i>SSIM</i>	<i>MSE</i>	<i>SSIM</i>
Baseline		4.52e-05	0.9927	0.0408	0.2845
Reg Row-Row		2.23e-06	0.9979	3.18e-03	0.4930
MIM		9.62e-06	0.9982	0.0145	0.5778
% Improvement	Reg Row-Row	95.1	0.52	92.2	73.3
Against Baseline	MIM	78.7	0.55	64.5	103.1

Table 5.2: Three Hour Input to Predict the Next Three Hour

Model		Min-Max Norm		Sig Norm	
		<i>MSE</i>	<i>SSIM</i>	<i>MSE</i>	<i>SSIM</i>
Baseline		6.09e-05	0.9914	0.0502	0.1404
Reg Row-Row		3.44e-06	0.9969	3.97e-03	0.3898
MIM		1.51e-05	0.9973	0.0182	0.4478
% Improvement	Reg Row-Row	94.4	0.55	92.1	177.6
Against Baseline	MIM	75.2	0.60	63.7	218.9

Table 5.3: Six Hour Input to Predict the Next Six Hour

Model		Min-Max Norm		Sig Norm	
		<i>MSE</i>	<i>SSIM</i>	<i>MSE</i>	<i>SSIM</i>
Baseline		3.13e-05	0.9949	0.0427	0.1274
Reg Row-Row		2.61e-06	0.9975	4.01e-03	0.3135
MIM		1.76e-05	0.9973	0.0184	0.3668
% Improvement Against Baseline	Reg Row-Row	91.7	0.26	90.6	146.1
	MIM	43.8	0.24	56.9	187.9

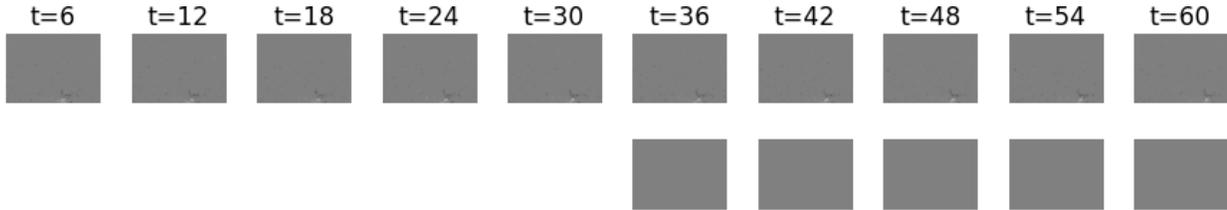


Figure 5.2: MIM: Min-Max Norm - Input: 6 Hour Output: 6 Hour

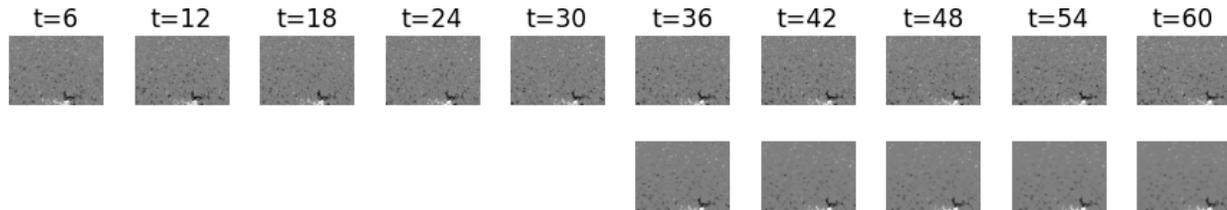


Figure 5.3: MIM: Sig Norm - Input: 6 Hour Output: 6 Hour

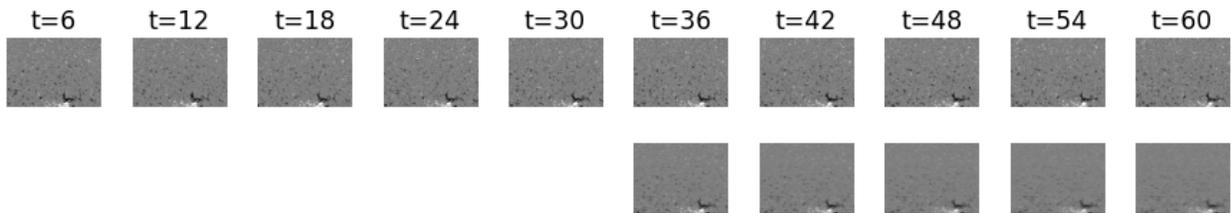


Figure 5.4: Regression: Sig Norm - Input: 6 Hour Output: 6 Hour

Chapter 6

Research Plan

6.1 Proposed Work

In order to achieve the research objectives, the following steps have been outlined to serve as guidelines:

- Generating high quality synthetic data: Generative Adversarial Networks(GANs) have made impressive gains since its inception in 2014 [15] and now are known to produce plausible realistic images. This has provided an alternative to previous methods used in data augmentation which involves crops, flips, rotation, cutouts and other image transforming techniques. These data augmentation steps are usually applied to datasets with a limited number of examples especially in the case of categorical datasets. Most solar-related datasets suffer from this limitation such as the solar flare dataset where there are more examples of A, B, and C class flares than M and X class flares. Therefore we seek to employ the GAN architecture to create more realistic examples for Solar Data. In the work done by Diaz Baso et al. [9], they use a deep fully convolutional neural network called Enhance to super-resolve HMI data. Enhance was able to produce good HMI images after up-scaling them by a factor of 2. Their network is basically an autoencoder with residual components and skip connections. Using their idea and we will modify their network to incorporate the GAN architecture to produce high quality synthetic data since GANs are known to produce good and domain-specific images. Specifically we will be modifying the SRGAN architecture proposed by [25] to fit the solar images domain.

- Incorporating the knowledge of the 3D viewing geometry and the Sun’s rotation: Data from Helioseismic and Magnetic Imager (HMI) are mapped using the Helioprojective-cartesian coordinates. World Coordinate System (WCS) has been implemented in the Astropy [3] library which contains functions for converting from one coordinate system to the other. In the results obtained so far, we found that the network was able to learn the Sun rotation which might prevent the network from learning other non-stationarity of active regions. We seek take out this information from the dataset by leveraging on the Astropy library to warp the images. In warping the images, we will do linear translation by interpolating the pixels from one frame to other where the temporal difference will be 12 minutes in one test case and morph the images in other test set incorporating the curvature of the Sun. We will train the network with the warped images to make predictions excluding information from the Sun rotation. To evaluate our results, we will warp back the predictions and compare with real images. We will warp the images linearly and also warp them incorporating the curvature of the Sun and compare the
- Improving upon the predictions of changes in active regions: Galvez et al. [14] in their work on developing a dataset fit for machine learning algorithms, they were able to show that there exist not only a correlation in the physical properties of data from Atmospheric Imaging Assembly (AIA) and HMI but a deep convolutional neural network can be trained to translate HMI observables to AIA observables. With this, we will modified our deep learning model used in predicting changes in active regions to incorporate data from other data sources like AIA and Geostationary Operational Environmental Satellites (GOES) to improve upon our results. In combining the data, we will apply the method used by Bobra et al. [6] in selecting their features to find out which data is useful to improve our results. Modifying the deep learning model will involve changing the input of the network to accept the mixed data and training it end-to-end either by combining the data channel-wise or using two different inputs

and concatenating their hidden layers. we will compare these two approaches.

- Predicting changes in images in other wavelengths: From Galvez et al. [14], we will apply our findings in predicting changes to active regions and improving upon our results to data from AIA which has already been shown that they are highly correlated. After this, we will incorporate this into our model used in generating synthetic data to produce missing data in other data sources and vice versa.

6.2 Timeline

Table 6.1: Timeline

Activity	Dates	Description
Literature review	August 2021- January 2022	Complete Literature review both in computer science and astrophysics under spatiotemporal changes on the Sun
Publication	August 2021- December 2022	Write papers based on our research findings
Generate Synthetic data	August- December 2021	Modify and apply deep learning models used in image-to-image to generate high-quality synthetic solar data
Incorporate viewing geometry and Sun's rotation	January-April 2022	Write a paper on our results achieved by incorporating the viewing geometry and the Sun rotation into predicting changes of active regions
Adapt different deep learning models used in next frame prediction	May-December 2022	Test, modify and evaluate deep learning models used in next frame prediction and incorporate solar data from other sources
Apply deep learning models used in video prediction to data products from AIA instrument	May-December 2022	We will test the models using the data products from AIA and evaluate their performance
Dissertation writing	July-December 2022	Finish up adding all the findings made to the dissertation write-up
Dissertation defense	October- December 2022	Speak with committee members and set a date for the defense

References

- [1] O. W. Ahmed, R. Qahwaji, T. Colak, P. A. Higgins, P. T. Gallagher, and D. S. Bloomfield. Solar flare prediction using advanced feature extraction, machine learning, and feature selection. *Solar Physics*, 283(1):157–175, 2013.
- [2] I. Arel, D. C. Rose, and T. P. Karnowski. Deep machine learning - a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine*, 5(4):13–18, 2010.
- [3] Astropy Collaboration, A. M. Price-Whelan, B. M. SipHocz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. VanderPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D’Avella, C. Deil, E. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Z. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska,

- T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terr'on, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *aj*, 156(3):123, Sept. 2018.
- [4] G. Barnes and K. D. Leka. Evaluating the performance of solar flare forecasting methods. *The Astrophysical Journal Letters*, 688(2), 2008.
- [5] D. S. Bloomfield, P. A. Higgins, R. T. J. McAteer, and P. T. Gallagher. Toward reliable benchmarking of solar flare forecasting methods. *The Astrophysical Journal Letters*, 747(2), 2012.
- [6] M. G. Bobra and S. Couvidat. Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm. *The Astrophysical Journal*, 798(2):135, 2015.
- [7] S. Couvidat, J. Schou, R. Shine, R. Bush, J. Miles, P. Scherrer, and R. Rairden. Wavelength dependence of the helioseismic and magnetic imager (hmi). instrument onboard the solar dynamics observatory (sdo). *Sol Phys*, 275:285–325, 2012.
- [8] Y. Cui, R. Li, L. Zhang, Y. He, and H. Wang. Correlation between solar flare productivity and photospheric magnetic field properties. *Sol Phys*, 237:45–59, 2006.
- [9] C. J. Díaz Baso and A. Asensio Ramos. Enhancing sdo/hmi images using deep learning. *Astronomy & Astrophysics*, 614:A5, Jun 2018.
- [10] J. Eastwood, E. Biffis, M. Hapgood, L. Green, M. Bisi, R. Bentley, R. Wicks, L.-A. McKinnell, M. Gibbs, and C. Burnett. The economic impact of space weather: Where do we stand? *Risk Analysis*, 37(2):206–218, 2017.

- [11] D. A. Falconer. A prospective method for predicting coronal mass ejections from vector magnetograms. *Journal of Geophysical Research: Space Physics*, 2001.
- [12] D. A. Falconer, R. L. Moore, A. F. Barghouty, and I. Khazanov. Prior flaring as a complement to free magnetic energy for forecasting solar eruptions. *The Astrophysical Journal*, 757(1), 2012.
- [13] G. H. Fisher, D. J. Bercik, B. T. Welsch, and H. S. Hudson. Global forces in eruptive solar flares: The lorentz force acting on the solar atmosphere and the solar interior. *SoPh*, 277:59, 2011.
- [14] R. Galvez, D. F. Fouhey, M. Jin, A. Szenicer, A. Muñoz-Jaramillo, M. C. M. Cheung, P. J. Wright, M. G. Bobra, Y. Liu, and J. Mason. A machine learning dataset prepared from the nasa solar dynamics observatory mission. *The Astrophysical Journal*, 242(7), 2019.
- [15] I. J. Goodfellow, J. Pougetabadie, M. Mirza, B. Xu, D. Wardefarley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *In NeurIPS Volume 3*, pages 2672–2680, 2014.
- [16] M. J. Hagyard, J. B. S. Jr., D. Teuber, and E. A. West. A quantitative study relating observed shear in photospheric magnetic fields to repeated flaring. *Sol Phys*, 91:115–126, 1984.
- [17] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [18] J. T. Hoeksema, Y. Liu, K. Hayashi, X. Sun, J. Schou, S. Couvidat, A. Norton, M. Bobra, R. Centeno, K. Leka, G. Barnes, and M. J. Turmon. The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: Overview and performance. *Sol Phys*, 289:3483–3530, 2014.

- [19] X. Huang and H.-N. Wang. Solar flare prediction using highly stressed longitudinal magnetic field parameters. *Research in Astronomy and Astrophysics*, 13(3), 2013.
- [20] X. Huang, D. Yu, Q. Hu, H. Wang, and Y. Cui. Short-term solar flare prediction using predictor teams. *Sol Phys*, 263:175–184, 2010.
- [21] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. In *In ICML*, 2017.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [23] B. J. LaBonte, M. K. Georgoulis, , and D. M. Rust. Survey of magnetic helicity injection in regions producing x-class flares. *The Astrophysical Journal*, 671:955, 2007.
- [24] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.
- [25] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.
- [26] K. Lee, Y.-J. Moon, J.-Y. Lee, K.-S. Lee, and H. Na. Solar flare occurrence rate and probability in terms of the sunspot classification supplemented with sunspot area and its changes. *Sol Phys*, 281:639–650, 2012.
- [27] K. D. Leka and G. Barnes. Photospheric magnetic field properties of flaring versus flare-quiet active regions. ii. discriminant analysis. *The Astrophysical Journal*, 595(2), 2003.
- [28] K. D. Leka and G. Barnes. Photospheric magnetic field properties of flaring versus flare-quiet active regions. iv. a statistically significant sample. *The Astrophysical Journal*, 656:1173, 2007.

- [29] R. Li, Y. Cui, H. He, and H. Wang. Application of support vector machine combined with K-nearest neighbors in solar flare and solar proton events forecasting. *Advances in Space Research*, 42(9):1469–1474, 2008.
- [30] R. Li, H.-N. Wang, H. He, Y.-M. Cui, and Z.-L. Du. Support vector machine combined with k-nearest neighbors for solar flare forecasting. *Chinese Journal of Astronomy and Astrophysics*, 7(3):441, 2007.
- [31] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning, 2017.
- [32] J. P. Mason and J. T. Hoeksema. Testing automated solar flare forecasting with 13 years of michelson doppler imager magnetograms. *The Astrophysical Journal*, 723(1), 2010.
- [33] P. S. McIntosh. The classification of sunspot groups. *Solar Physics*, 125(2):251–267, 1990.
- [34] R. L. Moore, D. A. Falconer, and A. C. Sterling. The limit of magnetic-shear energy in solar active regions. *The Astrophysical Journal*, 750(24), 2012.
- [35] NASA. https://www.nasa.gov/mission_pages/sunearth/spaceweather/index.html.
- [36] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, and M. Ishii. Deep flare net (defn) model for solar flare prediction. *The Astrophysical Journal*, 858(2):113, May 2018.
- [37] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, S. Watari, and M. Ishii. Solar flare prediction model with three machine-learning algorithms using ultraviolet brightening and vector magnetograms. *The Astrophysical Journal*, 835(2):156, 2017.
- [38] C. Olah. <http://colah.github.io/>.

- [39] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *In ICLR Workshop*, 2016.
- [40] K. G. Pillai, R. A. Angryk, J. M. Banda, M. A. Schuh, and T. Wylie. Spatio-temporal co-occurrence pattern mining in data sets with evolving regions. In *2012 IEEE 12th International Conference on Data Mining Workshops, Brussels*, pages 805–812, 2012.
- [41] E. Priest and T. Forbes. The magnetic nature of solar flares. *The Astron Astrophys*, 10:313–377, 2002.
- [42] R. Qahwaji and T. Colak. Automatic short-term solar flare prediction using machine learning and sunspot associations. *Solar Physics*, 241(1):195–211, 2007.
- [43] A. Raboonik, H. Safari, N. Alipour, and M. S. Wheatland. Prediction of solar flares using unique signatures of magnetic field images. *The Astrophysical Journal*, 834(1), 2016.
- [44] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.
- [45] M. Rempel and M. C. M. Cheung. Numerical simulations of active region scale flux emergence: from spot formation to decay. *The Astrophysical Journal*, 785(2):90, Mar 2014.
- [46] J. Schou, P. H. Scherrer, R. I. Bush, R. Wachter, S. Couvidat, M. C. Rabello-Soares, R. S. Bogart, J. T. Hoeksema, Y. Liu, T. L. D. Jr., D. J. Akin, B. A. Allard, J. W. Miles, R. Rairden, R. A. Shine, T. D. Tarbell, A. M. Title, C. J. Wolfson, D. F. Elmore, A. A. Norton, and S. Tomczyk. Design and ground calibration of the helioseismic and magnetic imager (hmi) instrument on the solar dynamics observatory (sdo). *Sol Phys*, 275:229–259, 2012.

- [47] C. J. Schrijver. A characteristic magnetic field pattern associated with all major solar flares and its use in flare forecasting. *The Astrophysical Journal Letters*, 655(L117), 2007.
- [48] U. SciEd. <https://scied.ucar.edu/sun-active-region>.
- [49] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. chun Woo. Convolutional lstm network: A machine learning approach for precipitation now-casting. In *In NeurIPS*, 2015.
- [50] H. Song, C. Tan, J. Jing, H. Wang, V. Yurchyshyn, and V. Abramenko. Statistical assessment of photospheric magnetic features in imminent solar flare predictions. *Solar Physics*, 254(1):101–125, 2009.
- [51] space.com. <https://www.space.com/14736-sunspots-sun-spots-explained.html>.
- [52] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *In ICML*, 2015.
- [53] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *In NeurIPS*, 2014.
- [54] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *In ICLR*, 2017.
- [55] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *In NeurIPS*, 2017.
- [56] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. *cs.LG*, 2019.

- [57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. In *IEEE Transactions on Image Processing* 13(4):600. IEEE, 2004.
- [58] M. S. Wheatland. A bayesian approach to solar flare prediction. *The Astrophysical Journal*, 609(2), 2004.
- [59] D. Yu, X. Huang, H. Wang, and Y. Cui. Short-term solar flare prediction using a sequential supervised learning method. *Sol Phys*, 255:91–105, 2009.
- [60] Y. Yuan, F. Y. Shih, J. Jing, and H.-M. Wang. Automated flare forecasting using a statistical learning technique. *Research in Astronomy and Astrophysics*, 10(8), 2010.

Curriculum Vitae

Godwill Amankwa is the second son born to Mr Michael Mensah and Mrs Georgina Asare. He completed St. Augustine's college in the year 2012 for his high school education and entered Kwame Nkrumah University of Science and Technology to pursue a bachelor degree in Computer Engineering. After successfully completing all coursework, project, and project defense, he graduated in June 2016. He further stayed in the school for one academic year to serve as a teaching assistant in the department of computer and biomedical engineering. In the year 2017, we worked as card system analyst for PaySwitch Company Limited, a fintech company. After four months, he got admission to pursue a master's program in Computational Science in the University of Texas at El Paso.

Email: gamankwa@miners.utep.edu