

**B. Tech. Project: Phases I & II**  
Report

# **Statistical Downscaling of Rainfall Projections using Convolutional Neural Networks**

*Submitted in partial fulfillment of  
the requirements for the award of the degree of*

**Bachelor of Technology  
in  
Civil Engineering**

Submitted by

---

Videsh Suman  
150040095

---

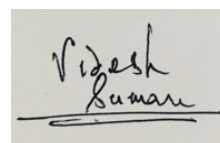
Under the guidance of  
**Prof. Subimal Ghosh (CE)**  
and co-guidance of  
**Prof. Amit Sethi (EE)**



Department of Civil Engineering  
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY  
Mumbai, Maharashtra, India – 400076  
Autumn 2018 - Spring 2019

## *Declaration*

I proclaim this composed submission speaks to my thoughts in my own words and where others' thoughts or words have been utilized, I have satisfactorily referred to, cited and referenced the first sources. I likewise announce that I have clung to all standards of scholarly trustworthiness and uprightness and have not distorted or created or misrepresented any thought, information, truth or source in my submission. I comprehend that any infringement of any of the above will be cause for disciplinary action by the Institute and bring out penal action from the sources which have hence not been appropriately referred to or from whom legitimate authorization has not been taken when required.

A handwritten signature in black ink on a light gray background. The signature reads "Videsh Suman" with a horizontal line underneath the name.

Videsh Suman  
(150040095)

Date: 23<sup>rd</sup> April, 2019

## Abstract

General Circulation Models (GCMs) and Earth System Models (ESMs) are tools, designed to simulate time series of climate variables globally for future. The spatial scales, on which these operate, are very coarse compared to that of any hydrologic process of interest. Direct use of the outputs of GCMs in hydrology is not desirable due to these limitations. Downscaling models [dynamical (physics based) and statistical (data driven)] are developed to address the limitations of GCMs by projecting high resolution climate variables, making use of coarse scale GCM simulations. These high resolution climate projections serve as key input not only in planning and management programs but also for obtaining future patterns of extreme events pertaining to different climate variables such as temperature, rainfall etc. These projections can play crucial role for country such as India (characterized by rainfed agriculture and high population density regions) in formulating different strategies regarding water food energy nexus, disaster mitigation planning etc. With the advent of parallel computing frameworks, there has been a meteoric boom in the use of Deep Neural Networks as some of most effective data driven methods. These have drawn a lot of attention due to their success in solving some of the most computationally difficult problems with highly non-linear relationships between input and output variables. The recent advances of deep learning has helped in solving complex computational problems in fields like Computer Vision, Image and Speech Processing, Natural Language Understanding, Language Translation, etc. Hence, the research summed up in this report contributes towards the main objective of obtaining high resolution rainfall projections for India using custom variants of Convolutional Neural Networks (CNNs). Across the two phases of this project, I have covered the associated literature, including the theory of Statistical Downscaling, some previous data driven models used for this task, and theory of CNNs in the context of visual recognition, as well as the details and results of the my implementations in the context of the goal of this project. The first phase was spent majorly in understanding the literature, which also included some preliminary implementations leveraging deep CNNs on the available spatio-temporal data. In the second phase, I explored the various techniques and heuristics with respect to the deep learning literature for obtaining close acceptable results.

# Contents

<b>I</b>	<b>BTP: Phase I</b>	<b>1</b>
<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Climate Forcing Mechanisms . . . . .	2
1.2	General Circulation Models . . . . .	3
1.3	Climate Projections using Downscaling Techniques . . . . .	3
1.4	Stationarity in Statistical Relationships . . . . .	5
1.5	Deep Learning Era . . . . .	5
1.6	Problem Definition . . . . .	5
<b>2</b>	<b>Literature Survey</b>	<b>7</b>
2.1	Downscaling of GCM Simulations . . . . .	7
2.2	Bias Correction Methodologies . . . . .	7
2.3	Dynamical Downscaling . . . . .	8
2.4	Statistical Downscaling . . . . .	9
2.4.1	Weather Generators . . . . .	9
2.4.2	Weather Typing . . . . .	9
2.4.3	Transfer Functions . . . . .	10
2.5	Model by Salvi et al.[2013]: Case Study . . . . .	10
2.5.1	Data . . . . .	10
2.5.2	The Kernel Regression Pipeline . . . . .	11
2.5.3	Results & Inference . . . . .	12
2.6	CNNs: An Overview . . . . .	14
2.6.1	What is Convolution? . . . . .	14
2.6.2	Why Convolutions? . . . . .	15
2.6.3	Non-linear Activation . . . . .	15
2.6.4	Loss Function . . . . .	15
2.6.5	Gradient Descent . . . . .	16
2.6.6	Training a CNN . . . . .	16
2.7	DeepSD by Vandal et al.[2017]: A Case Study . . . . .	17
2.7.1	Motivation . . . . .	17
2.7.2	Interpretation of the Climate Data as Images . . . . .	17
2.7.3	Data . . . . .	17
2.7.4	Super-Resolution CNN . . . . .	18
2.7.5	The DeepSD Framework . . . . .	19
2.7.6	Results and Inference . . . . .	20
2.8	Summary . . . . .	21

<b>3</b>	<b>Implementation and Experiments</b>	<b>22</b>
3.1	Data . . . . .	22
3.2	Model Definition . . . . .	22
3.3	Network Heuristics . . . . .	23
3.3.1	Transposed Convolutions for Up-sampling . . . . .	23
3.3.2	Dilated Convolution . . . . .	24
3.3.3	Training Strategies . . . . .	24
3.4	Results . . . . .	25
<b>4</b>	<b>Inference and Conclusion</b>	<b>27</b>
<b>5</b>	<b>Future Work</b>	<b>28</b>
<b>II</b>	<b>BTP: Phase II</b>	<b>29</b>
<b>6</b>	<b>Methodology and Implementation</b>	<b>30</b>
6.1	Problem Reformulation . . . . .	30
6.2	Data . . . . .	30
6.3	Data Preprocessing . . . . .	31
6.3.1	Disintegration into Zones . . . . .	31
6.3.2	Central Region . . . . .	31
6.3.3	Data Normalization . . . . .	32
6.4	Network Heuristics . . . . .	33
6.4.1	Dense Connections . . . . .	33
6.4.2	Complete Network Architecture . . . . .	34
<b>7</b>	<b>Results and Inference</b>	<b>35</b>
<b>8</b>	<b>Future Prospects</b>	<b>37</b>
	<b>Acknowledgements</b>	<b>38</b>
	<b>References</b>	<b>39</b>

# List of Figures

1.1	(Source - Salvi et al.[1]) Flowchart of Linear Regression based statistical downscaling methodology . . . . .	4
2.1	(Source - Salvi et al.[6]) A flowchart for the Multisite Statistical Downscaling Model which enlists various mathematical operations that are performed on predictors (the GCM simulated climate variables) and the predictand (rainfall) which take part in statistical downscaling as inputs. The current statistical downscaling model is a combination of the daily weather state generator and the transfer function method. Rainfall . . . . .	12
2.2	(Source - Salvi et al.[6]) The region of predictors (as shown with black rectangles) for each meteorologically homogeneous zone (as shown with gray shade) are illustrated (a) Central, (b) Jammu and Kashmir, (c) North, (d) Northeast hills, (e) Western, (f) South, and (g) Northeast. The extent of the region of predictors in terms of latitude and longitudes are detailed in (h). . . . .	13
2.3	(Source - Yamashita et al.[7]) An example of convolution operation with a kernel size of $3 \times 3$ , no padding, and a stride of 1. A kernel is applied across the input tensor, and an element-wise product between each element of the kernel and the input tensor is calculated at each location and summed to obtain the output value in the corresponding position of the output tensor, called a feature map. . . . .	14
2.4	(Source - Yamashita et al.[7]) Activation functions commonly applied to neural networks: a rectified linear unit (ReLU), b sigmoid, and c hyperbolic tangent (tanh). . . . .	15
2.5	(Source - Yamashita et al.[7]) Gradient descent is an optimization algorithm that iteratively updates the learnable parameters so as to minimize the loss, which measures the distance between an output prediction and a ground truth label. The gradient of the loss function provides the direction in which the function has the steepest rate of increase, and all parameters are updated in the negative direction of the gradient with a step size determined based on a learning rate . . . . .	16
2.6	(Source - Vandal et al.[3]) Augmented SRCNN Architecture. From the left to right: Precipitation and Elevation sub-image pair, filters learned in layer 1, layer 1 activations, layer 2 filters, layer 2 activations, layer 3 filters, and HR precipitation label. . . . .	19
2.7	(Source - Vandal et al.[3]) Layer by layer resolution enhancement from DeepSD using stacked SRCNNs. Top Row: Elevation, Bottom Row: Precipitation. Columns: $1.0^\circ$ , $1/2^\circ$ , $1/4^\circ$ and $1/8^\circ$ spatial resolutions. . . . .	20

2.8	(Source - Vandal et al.[3]) Comparison of predictive ability between all six methods for 1000 randomly selected locations in CONUS. Runtime is computed as the amount of time to downscale 1 year of CONUS. . . . .	20
2.9	(Source - Vandal et al.[3]) Comparison of Predictive Ability between DeepSD and BCSD for each season, Winter, Summer, Spring, and Fall. Values are computed at each location in CONUS and averaged. . . . .	20
3.1	(Source - Dumoulin et al.[9]) The transpose of convolving a $3 \times 3$ kernel over a $4 \times 4$ input using unit strides (i.e., $i = 4$ , $k = 3$ , $s = 1$ and $p = 0$ ). It is equivalent to convolving a $3 \times 3$ kernel over a $2 \times 2$ input padded with a $2 \times 2$ border of zeros using unit strides. . . . .	24
3.2	(Source - Fisher et al.[11]) Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) $F_1$ is produced from $F_0$ by a 1-dilated convolution; each element in $F_1$ has a receptive field of $3 \times 3$ . (b) $F_2$ is produced from $F_1$ by a 2-dilated convolution; each element $F_2$ has a receptive field of $7 \times 7$ . (c) $F_3$ is produced from $F_2$ by a 4-dilated convolution; each element in $F_3$ has a receptive field of $15 \times 15$ . The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly. . . . .	24
3.3	(Source - fast.ai) Increasing the learning rates every few iterations to restart the gradient descent. . . . .	25
3.4	6 pairs of predicted output and ground truth projections randomly picked from the testing samples. . . . .	26
6.1	(Source - Salvi et al.[6]) The region of predictors (as shown with black rectangles) for each meteorologically homogeneous zone (as shown with gray shade) are illustrated (a) central, (b) Jammu and Kashmir, (c) North, (d) Northeast hills, (e) Western, (f) South, and (g) Northeast. The extent of the region of predictors in terms of latitude and longitudes are detailed in (h). . . . .	31
6.2	The central region (1273 pixels) enclosed within the bounding box of size $48 \times 47$ , owing to the $0.25^\circ$ resolution of the ground truth. . . . .	32
6.3	Percentile plot of the observed rainfall of central region over 55 monsoon periods. . . . .	32
6.4	A 5-layer dense block with a growth rate of $k = 4$ . Each layer takes all preceding feature-maps as input. . . . .	34
7.1	3 sets of target, predicted output and error projections randomly picked from the testing samples. MAE (in mm/day) for cases from top to bottom: 2.68, 13.73 and 1.32 . . . . .	35
7.2	Some more sets of target, predicted output and error projections randomly picked from the testing samples. MAE (in mm/day) for cases from top to bottom: 4.24, 5.17 and 5.62 . . . . .	36

**Part I**

**BTP: Phase I**



# Chapter 1

## Introduction

Climate change is causing detrimental effects to society's well being as temperatures increase, extreme events become more intense, and sea levels rise. Natural resources that society depends on, such as agriculture, freshwater, and coastal systems, are vulnerable to increasing temperatures and more extreme weather events. Similarly transportation systems, energy systems, and urban infrastructure allowing society to function efficiently continue to degrade due to the changing climate. Furthermore, the health and security of human beings, particularly those living in poverty, are vulnerable to extreme weather events with increasing intensity, duration, and frequency. Scientists and stakeholders across areas such as ecology, water, and infrastructures, require access to credible and relevant climate data for risk assessment and adaptation planning. Hence, there is a strong socio-economic value in predicting the potential effects of climate change affecting the local hydrological processes.

The previous studies have used several other methods such as linear regression, quantile regression, kernel regression, beta regression, and artificial neural networks for solving statistical downscaling problems. Deep neural networks, particularly convolutional neural networks have been shown to be highly promising in modeling complex and highly non-linear relationships between input and output variables in different domains. This background serves as the motivation behind undertaking the research problem of predicting fine-resolution rainfall projections from coarse simulation models using the framework of a deep convolutional network.

The present chapter provides brief information on the aspects of climate, climate change, general circulation models, future projections, deep neural networks and problem definition.

### 1.1 Climate Forcing Mechanisms

Factors that shape the climate are called as 'Climate forcing mechanisms'. Such mechanisms are classified into (1) Internal forcing and (2) External forcing. Natural changes in the components of earth's climate system and their interactions are the causes of internal climate variability, or "Internal forcing". Ocean variability is a key component of internal forcing. Short-term fluctuations (years to a few decades) such as the El Nino-Southern Oscillation, the Pacific decadal oscillation, the North Atlantic oscillation, and the Arctic oscillation, represent climate variability rather than climate change. "External forcing" involves changes in solar irradiance or anthropogenic GHG emissions [IPCC, 2007]. The phenomenon of global warming, which consists of unequivocal and continuing rise in the

average temperature of Earth's climate system is one of the examples of climate change because of GHG emissions [IPCC, 2007]. Most of the observed increase in global average temperatures since the mid- 20th century is very likely due to the observed increase in anthropogenic GHG concentrations [IPCC, 2007]. The consequences of global warming are reflected in global as well as regional climate. These consequences are in terms of changes in frequency, intensity, and duration of key climatic variables such as precipitation, atmospheric moisture, snow cover, extent of land/sea ice, sea level, patterns in atmospheric and ocean circulation etc. Therefore, the study of climate change is necessary to understand its impacts on hydrological processes. While, downscaling of global scale climatic variables into local scale hydrologic variables is a very important aspect to it.

## 1.2 General Circulation Models

A General Circulation Model (GCM) is a mathematical model of the general circulation of a planetary atmosphere or ocean and based on the NavierStokes equations on a rotating sphere with thermodynamic terms for various energy sources (radiation, latent heat). These equations are the basis for complex computer programs, commonly used for simulating the atmosphere or ocean of the Earth. GCMs are widely applied for weather forecasting, understanding the climate, and projecting climate change. These computationally intensive numerical models are based on the integration of a variety of fluid dynamical, chemical, and sometimes biological equations. GCMs have been developed to simulate the present climate and have been used to project the change in future climate. While GCMs demonstrate significant skill at the continental and hemispheric spatial scales and incorporate a large proportion of the complexity of the global system, they are inherently unable to represent local sub grid-scale features and dynamics especially fail to reproduce non-smooth fields such as precipitation. Hence, while the impact of greenhouse gases on large-scale atmospheric circulation is well understood, regional changes in the hydrological cycle are far more uncertain in GCM simulations. To circumvent these problems, tools for generating high-resolution meteorological inputs are required for modelling hydrological processes. "Downscaling" approaches have subsequently emerged as a means to bridge the gap between the large-scale atmospheric predictor variables and the local or station-scale meteorological series of interest.

## 1.3 Climate Projections using Downscaling Techniques

The GCMs show different skill scores in simulations and projections of different climatic variables. The variables which show comparatively higher degree of spatial uniformity e.g. temperature or wind variables, are well simulated by GCMs. However, the variables like rainfall, which are highly affected by local parameters, are poorly simulated by GCMs. This is mainly because of the spatial resolution at which GCMs work. Hence, downscaling techniques are deployed to obtain climate projections at high resolution. These techniques involved obtaining fine resolution projections either by developing physics based Regional Climate Model (RCM) which takes inputs from GCMs or by establishing statistical relationship between coarse scale climate variables (predictors) which are relatively well simulated by GCMs and local scale rainfall (predictand). First approach is known as 'Dynamical Downscaling' and the later approach is known as 'Statistical Downscaling'.

Both the approaches have their pros and cons. Dynamical downscaling can be applied to obtain very high resolution projections; however, these are time consuming and require heavy computational facility. On the other hand, statistical downscaling techniques are very fast in obtaining projections, although, these techniques demand observed data availability over longer time scale. Also, the future projections are obtained using the same statistical relationship, which is established over past observed data. The credibility of statistical relationship, which is established over past data, is questioned under changing climatic conditions.

The statistical relationship between predictors and predictand can be as simple as linear regression to more complicated models. Predictand is the climate variables of interest at fine resolution and predictors are climate variables at coarse scale that influence predictand. Mathematically, predictors are independent variables  $X = (X1, X2..)^T$ , predictand is dependent variable  $Y$ , and relationship is the function which links  $X$  and  $Y$ . Figure 1.1 shows the flowchart of the most basic, single site (at a single location), linear regression based downscaling methodology. The set of predictors  $X = (X1, X2..)^T$  first undergoes mathematical operations which involve reduction of dimension and removal correlation among predictors. These are required for (1) reducing computational power and (2) fulfilling the assumption of regression that the predictors are not correlated with each other, respectively. Let  $X' = (X1', X2'..)$  be the modified predictors (after mathematical operations). These form statistical relationship with the predictand  $Y$ . This statistical relationship is assumed to be time invariant. Using future predictors that are simulated by GCMs and the established relationship (based on past observed data), future projections of predictand are obtained. The assumed stationarity in the relationship is major limitation of all data driven models.

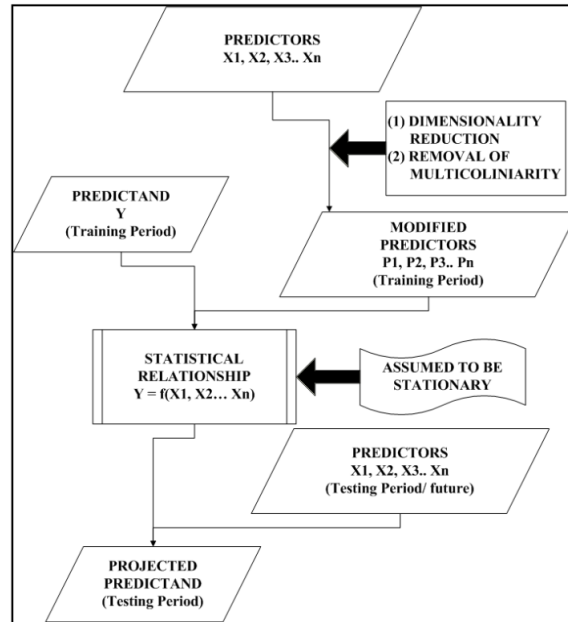


Figure 1.1: (Source - Salvi et al.[1]) Flowchart of Linear Regression based statistical downscaling methodology

## 1.4 Stationarity in Statistical Relationships

Statistical downscaling involves deriving empirical relationships that transform large-scale features of the GCM (Predictors) to regional-scale variables (Predictand) such as precipitation and streamflow. Statistical downscaling methodologies can be broadly classified into three categories viz. (1) weather generators, (2) weather typing, and (3) transfer functions. The most popular approach of statistical downscaling is the use of transfer function which is a regression based downscaling method that relies on direct quantitative relationship between the local scale predictand and the large scale predictors through some form of regression. Generally the relationship is established for observed period and it is assumed that the relationship holds good in future and projections are obtained. This assumption is known as 'Assumption of Stationarity'. The validity of this assumption is always in jeopardy especially under the changing climatic conditions. Hence, future projections of any climate variable, which are obtained using statistical downscaling always, face reliability issues. As these data driven models rely upon past data for establishing the relationship, this assumption stays as integral part of statistical downscaling models. This assumption shows its presence in dynamical downscaling techniques as well, where parameterization is involved.

## 1.5 Deep Learning Era

Deep neural networks, have drawn a lot of interest due to their success in solving some of the most computationally difficult problems with highly non-linear relationships between input and output variables. The success of neural networks is mainly attributed to their ability to learn hierarchical representations, unlike traditional machine learning models that build up on hand-engineered features. The most established algorithm among various deep learning models is convolutional neural network (CNN), a class of artificial neural networks that has been a dominant method in computer vision tasks since the astonishing results were shared on the object recognition competition known as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012[2]. CNNs are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. Various frameworks of CNN architectures are being used for a variety of Computer Vision tasks like object detection, image segmentation, image captioning, image super-resolution, video frame prediction, etc. The CNNs have been found to work excellently with images or other forms of gridded data. Recently, CNNs have been discovered to work phenomenally in the domain of Medical Image Analysis[4] including sub-fields of Radiology and Pathology. Several other areas have started looking deep learning techniques as much better alternatives for traditional statistical approaches. Motivated by two such research works in Climate Studies[3][5], I embarked on this research exposition of exploring deep learning methodologies for Statistical Downscaling of rainfall projections.

## 1.6 Problem Definition

Severe and possibly permanent alterations, brought by climate change (on account of anthropogenic activities) have led to the emergence of large-scale environmental hazards

to human health. In this case, obtaining high resolution climate projections of impact relevant climate variables (e.g. temperature and rainfall) will help in understanding the climate scenario, which is likely to be encountered in future at a regional scale. The problem being attempted in the present study involves using CNNs for obtaining future projections of rainfall at high resolution ( $0.25^\circ$ ) over Indian landmass. High resolution projections can then be used for impacts assessment and planning purpose. Motivation for this problem was provided by this research work by Salvi et al.[6]. They had used a kernel regression based statistical downscaling pipeline to obtained 21st century projections in  $0.5^\circ$  resolution over India. The DeepSD[3] framework by Vandal et al. has provided the motivation to my research for applying a CNN based model for this task.

# Chapter 2

## Literature Survey

A detailed review of literature on (a) different statistical downscaling techniques with special emphasis on the work of Salvi et al.[6] that was developed and applied to obtain high resolution rainfall projections and (b) convolutional neural networks with emphasis on the DeepSD architecture by Vandal et al.[3] is discussed in the present chapter. The research problem being attempted in my study, is designed to address the limitations of previously reported literature.

Note: A major portion of this chapter has been presented from Kaustubh Salvi's doctoral dissertation[1] and the DeepSD framework by Vandal et al.[3]

### 2.1 Downscaling of GCM Simulations

GCMs have been developed to simulate the present climate and used to project future climatic change under the influence of greenhouse gases and aerosols. These global scale models are generally not designed for local climate change impact studies and do not provide a good estimation of hydrological responses to climate change at local or regional scale. Some of the drawbacks, which restrict direct use of GCM output in hydrology, are

- Accuracy of GCMs decreases at increasingly finer spatial and temporal scales, while the needs of impacts studies conversely increase with higher resolution.
- Accuracy of GCMs decreases from climate related variables, i.e., wind, temperature, humidity and air pressure to precipitation, evapo-transpiration, runoff and soil moisture, while the latter variables are of key importance in hydrologic regimes.

Therefore there is a need to convert the GCM outputs into hydrologic variables (e.g. precipitation, stream flow) at a watershed scale to which the hydrological impact is to be investigated. Methodologies to model the hydrologic variables at a smaller scale (finer resolution) based on large scale GCM outputs are known as downscaling techniques.

### 2.2 Bias Correction Methodologies

Normally because of incomplete knowledge of physics behind the atmospheric processes and application of numerical schemes to solve the governing differential equations, GCM projections show systematic errors, known as bias. The GCM simulated climate variables are used as predictors in statistical downscaling methods. These impact models should

not be forced with native form GCM simulations due to the high biases in the data. The presence of such biases in GCM data seriously limits its applicability in climate impact studies and can result in unwanted uncertainty regarding projected climate change impacts. In the light of these remarks, a number of statistical methodologies have been proposed to correct the GCM outputs relative to the corresponding local observed data in order to make the simulations appropriate. Bias correction involves application of mathematical models which brings the statistical properties of simulated data (mean and standard deviation) close to observed for the same period.

Transfer functions establish statistical relationships between cumulative density functions (CDFs) of a common period between observed and simulated data and apply it to the projected data [Deque, 2007; Block et al., 2009; Piani et al., 2010a]. The transfer function may derive from mapping an empirical or a theoretical distribution (such as the gamma distribution) on GCM and observed precipitation CDFs. This kind of correction is also known as quantile [Wood et al., 2004] or distribution [Kirono et al., 2011] mapping.

Wood et al. [2004] proposed "bias correction and spatial disaggregation (BCSD) which is based on quantile remapping technique. It was applied to both, Parallel Climate Model (PCM) and its dynamically downscaled product. For the retrospective climate simulation, results were compared to an observed gridded climatology of temperature and precipitation, and gridded hydrologic variables resulting from forcing the hydrologic model with observations. The most significant findings are that the BCSD method was successful in reproducing the main features of the observed hydrometeorology from the retrospective climate simulation, when applied to both PCM and downscaled outputs.

Gudmundsson et al. [2012] compared the skills of bias correction methods, belonging to quantile mapping (QM) family. Three types of QM methods viz.

- distribution derived transformations (which involve fitting probability distributions to the observed and simulated data and perform the transformations at equal quantile levels),
- parametric transformations, (which involve establishing quantile-quantile relationship using parametric equations), and
- nonparametric transformations (which involve developing empirical CDFs or smoothing splines);

each differing with respect to their underlying assumptions are applied to RCM simulations. The study showed that nonparametric transformations have the highest skill in systematically reducing biases in RCM precipitation, based on skill score and ranking methods.

## 2.3 Dynamical Downscaling

Regional Climate Models (RCMs) are physics based models that are developed for the study regions incorporating local factors such as topography, land cover, etc. and run at a fine resolution. This type of modeling is also termed as dynamical downscaling.

## 2.4 Statistical Downscaling

Statistical downscaling involves deriving empirical relationships that transform large-scale features of the GCM (Predictors) to regional-scale variables (Predictand) such as precipitation and streamflow. There are three implicit assumptions involved in statistical downscaling:

- predictors are variables of relevance, realistically modeled by the GCM,
- empirical relationship is valid also under altered climatic conditions,
- the predictors employed fully represent the climate change signal.

Statistical downscaling methodologies can be broadly classified into three categories: weather generators, weather typing and transfer functions.

### 2.4.1 Weather Generators

Weather generators are statistical models of sequences of weather variables. They can also be regarded as complex number generators, the output of which resembles daily weather data at a particular location. The weather generators can broadly be classified into two categories viz., algorithm based and statistical tools based.

The algorithm based weather generators can further be classified into categories viz., Markov chain models and spell length models. In the Markov chain approach, a random process is constructed which determines a day at a station as rainy or dry, conditional upon the state of the previous day, following given probabilities. In case of spell length approach, instead of simulating rainfall occurrences day by day, spell length models operate by fitting probability distribution to observed relative frequencies of wet and dry spell lengths.

Similarly, the statistical tool based weather generators can also be classified into two categories viz. parametric and non-parametric weather generators. The most common models are parametric empirical-statistical models. They generate daily weather sequences based on a relatively simple stochastic process to which the underlying atmospheric physical processes are related only implicitly. The non-parametric stochastic weather generators uses non homogeneous hidden Markov model for spatial downscaling of multi-station daily rainfall occurrences using atmospheric circulations variables.

### 2.4.2 Weather Typing

Weather typing approaches involve grouping local, meteorological variables in relation to different classes of atmospheric circulation. Future regional climate scenarios are constructed, either by re-sampling from the observed variable distribution (conditional on circulation patterns produced by a GCM), or by first generating synthetic sequences of weather patterns using Monte Carlo techniques and re-sampling from the generated data. The mean or frequency distribution of the local climate is then derived by weighting the local climate states with the relative frequencies of the weather classes. Climate change is then estimated by determining the change of the frequency of weather classes.



### 2.4.3 Transfer Functions

The most popular approach of statistical downscaling is the use of transfer function which is a regression based downscaling method that relies on direct quantitative relationship between the local scale climate variable (predictand) and the variables containing the large scale climate information (predictors) through some form of regression. Individual downscaling schemes differ according to the choice of mathematical transfer function, predictor variables or statistical fitting procedure.

The Statistical downscaling techniques for prediction of multi-site rainfall in a river basin fail to capture the correlation between multiple sites and thus are inadequate to model the variability of rainfall. Kannan and Ghosh [2011] addressed this problem through representation of the pattern of multi-site rainfall using rainfall state in the Mahanadi river basin. A model based on K-means clustering technique coupled with a supervised data classification technique, namely Classification And Regression Tree (CART), is used for generation of rainfall states from large- scale atmospheric variables in a river basin. The non parametric Kernel regression is used for rainfall projections. In 2013, Salvi et al.[6] extended this model to downscale rainfall projections for the entire Indian landmass through region-wise predictions.

## 2.5 Model by Salvi et al.[2013]: Case Study

This study was performed with respect to the Indian Summer Monsoon Rainfall (ISMR) system that roughly spans over 4 months i.e. June to September each year. Its is an appropriate example to illustrate the complexity involved in understanding monsoon system. ISMR gets affected by multiple geophysical processes, simulating which is a big challenge. In order to simulate complex phenomenon such as ISMR, the model should have the capability to capture the large scale circulation as well as local scale parameter like topography, land use, etc. The entire exercise of obtaining rainfall projections using statistical downscaling was carried out on Indian landmass at 0.5 degree resolution.

### 2.5.1 Data

The choice of predictor variables is of the utmost importance when it comes to the accuracy of projected data. The selection of predictors should be dependent on the following criteria:

- the data for the particular predictor should be available for the desired period;
- the selected GCM should be capable of simulating the variable well; and
- the predictor should show a good correlation with the predictand.

For this study, the climatic variables described by Kannan and Ghosh [2011] were used as predictors viz. temperature, pressure, specific humidity, u-wind, and v-wind at the surface. Due to the systematic biases, GCM simulations can't be directly used as the input for the regression model. Rather, reanalysis data is used a proxy to train the model, and later the GCM simulations are bias-corrected with respect to the reanalysis before inferring the results from the bias corrected GCM simulations as input.

**Source of Predictors - The Reanalysis :** An outgrowth of the Climate Data Assimilation System (CDAS) project undertaken by the National Center for Environmental Prediction/ National Center for Atmospheric Research (NCEP/NCAR). The components of the assimilated datasets are the following: (1) global rawinsonde data, (2) a Comprehensive Ocean-Atmosphere Data Set (COADS) that comprises a collection of surface marine data, (3) aircraft data, (4) surface land synoptic data, (5) satellite sounder data, (6) Special Sensing Microwave/Imager (SSM/I) surface wind speeds, and (7) satellite cloud drift winds. For their work, the NCEP/NCAR reanalysis-I daily data for surface air temperature, mean sea level pressure, specific humidity, zonal wind velocity, and meridional wind velocity for a region delimited by the latitudes  $5^{\circ} - 40^{\circ}\text{N}$  and longitudes  $60^{\circ} - 120^{\circ}\text{E}$ , surrounding the entire study area for a period of 30 years from 1971-2000, was utilized for the bias correction, the training, and the validation of the downscaling model.

**Host GCM:** Coarse resolution climate variables, which were used as predictors in this study, are simulated using a third generation coupled GCM (CGCM3.1) developed by the Canadian Centre for Climate Modeling and Analysis.

**Source of the Predictand:** Gridded daily rainfall data for India (Latitude:  $6.5^{\circ}$  to  $38.5^{\circ}$ , Longitude:  $66.5^{\circ}$  to  $100.5^{\circ}$ ) at a  $0.5^{\circ}$  resolution is provided by IMD. Gridded data is derived from rainfall data collected from more than 6,000 rain-gauge stations over India.

## 2.5.2 The Kernel Regression Pipeline

- The predictors undergo a bias correction operation where the systematic error is removed using a quantile based remapping technique [Li et al., 2010]. However, using the bilinear interpolation technique, first the GCM simulated data is scaled to the NCEP/NCAR resolution, then the bias correction is performed.
- The bias corrected predictors go through a principal component analysis (PCA) that involves the application of orthogonal transformation on a set of correlated predictor variables, producing principal components. The PCA helps to reduce both dimensionality and multicollinearity. A reduction in the dimensions also results in a reduction in the computational effort.
- The meteorological homogeneous zones identified by the IMD are shown in Figure 6.1. Each zone is treated as an individual entity, and for each zone a corresponding region is fixed that is regular in shape (rectangle or square) and large enough to completely encompass the zone. Rainfall in a particular zone is assumed to be influenced more by the predictors in the selected region surrounding the zone than the predictors that are outside the region.
- To capture cross-correlation among the rainfall data (gridded data), the K-means clustering technique is adopted for generating the daily rainfall states for each zone. The technique reads the observed rainfall values for all nodes in a zone on any day, clusters them, and provides one representative value that is referred to as the state for that day. The step is important, provides the representative rainfall category for a particular day, and is linked to the predictors for establishing the statistical

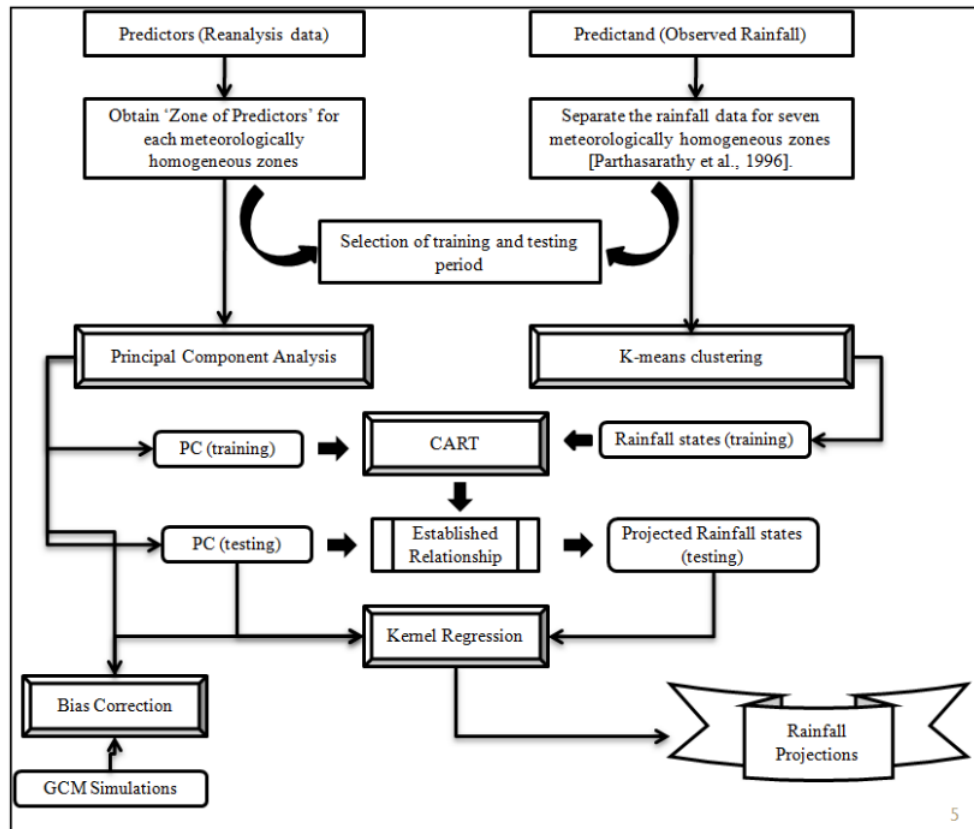


Figure 2.1: (Source - Salvi et al.[6]) A flowchart for the Multisite Statistical Downscaling Model which enlists various mathematical operations that are performed on predictors (the GCM simulated climate variables) and the predictand (rainfall) which take part in statistical downscaling as inputs. The current statistical downscaling model is a combination of the daily weather state generator and the transfer function method. Rainfall

relationship. Clustering is used to classify multi-site rainfall for each zone into different states (in the form of clusters).

- The Classification and Regression Trees (CART) is a supervised classification based model that builds classification trees for categorical dependent variables and regression trees for predicting continuous dependent variables. This technique uses historical data to construct decision trees. The decision trees are then used to classify new data. In the present study K-means clustering is applied to cluster multi-site rainfall data into rainfall classes/ states containing categorical values. These categorical values are used in the training sample for building the decision trees.
- Nonparametric kernel regression is utilized to obtain the projected daily rainfall at each node.

### 2.5.3 Results & Inference

- Rainfall projection results obtained for the 20th century revealed a good match to observed data in terms of statistical properties (i.e. mean and standard deviation).
- Although the maximum absolute difference between observed mean rainfall and

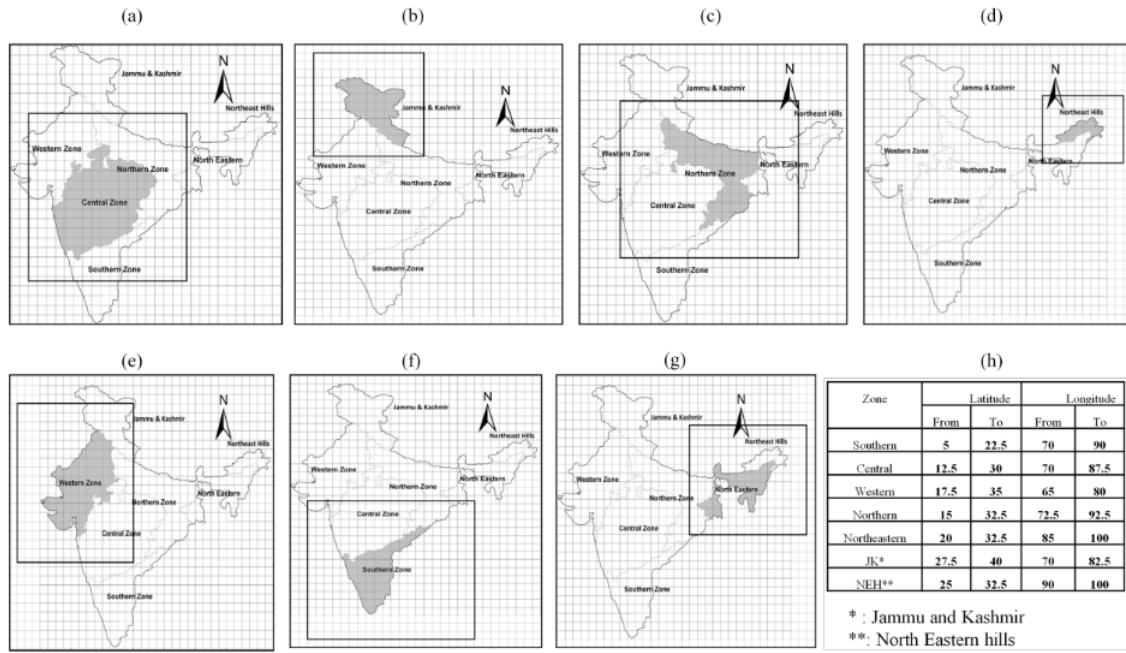


Figure 2.2: (Source - Salvi et al.[6]) The region of predictors (as shown with black rectangles) for each meteorologically homogeneous zone (as shown with gray shade) are illustrated (a) Central, (b) Jammu and Kashmir, (c) North, (d) Northeast hills, (e) Western, (f) South, and (g) Northeast. The extent of the region of predictors in terms of latitude and longitudes are detailed in (h).

projected mean rainfall is approximately 10mm, for most of the parts of India, the difference is around 3mm.

- Although the maximum difference is delimited between 20mm for the majority of the nodes (more than 75% of the grids points), the difference between the standard deviations for the observed and projected rainfall differed by 7.5mm/day.
- The cross-correlation plot displays the strength of the model in capturing the influence of rainfall at nearby nodes on the node at which the rainfall is projected.
- Future projections are performed over 21st century, the results indicated spatial non-uniformity for changes in mean rainfall. The magnitude of this change indicates an intensification with time.
- Consistency in the projection patterns may lead to a hypothesis of 'wet areas getting wetter and dry areas getting drier'
- The influence of orography is nicely captured by the model and is clearly evident from the projected rainfall pattern that shows high rainfall for the windward side of the Western Ghat, the Satpura, etc.; and lower rainfall for the leeward side.
- The projected rainfall time series for the future does not display any significant trend, indicating that, for the future, no major change in rainfall, as far as the magnitude is concerned, will occur, but that the spatial distribution will change.

## 2.6 CNNs: An Overview

CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low to high-level patterns. CNN is a mathematical construct that is traditionally composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. A convolution layer plays a key role in CNN, which is composed of a stack of mathematical operations, such as convolution, a specialized type of linear operation. In digital images, pixel values are stored in a two-dimensional (2D) grid, i.e., an array of numbers (Fig. 2), and a small grid of parameters called kernel, an optimizable feature extractor, is applied at each image position, which makes CNNs highly efficient for image processing, since a feature may occur anywhere in the image. As one layer feeds its output into the next layer, extracted features can hierarchically and progressively become more complex. The process of optimizing parameters such as kernels is called training, which is performed so as to minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

### 2.6.1 What is Convolution?

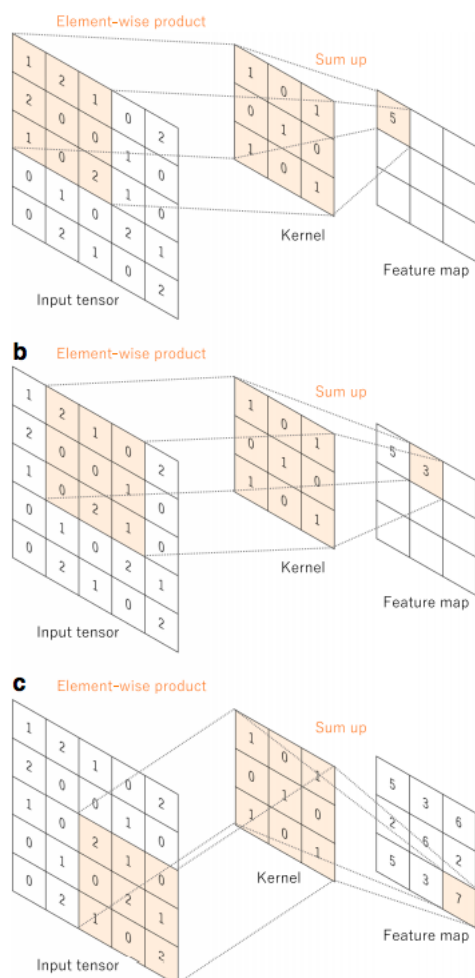


Figure 2.3: (Source - Yamashita et al.[7]) An example of convolution operation with a kernel size of  $3 \times 3$ , no padding, and a stride of 1. A kernel is applied across the input tensor, and an element-wise product between each element of the kernel and the input tensor is calculated at each location and summed to obtain the output value in the corresponding position of the output tensor, called a feature map.

## 2.6.2 Why Convolutions?

**Local Connectivity:** When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume. Instead, each neuron should be connected to only a local region of the input volume. The spatial extent of this connectivity is a hyperparameter called the receptive field of the neuron (equivalently this is the filter size). For an RGB image, depth corresponds the 3 channels of red, green and blue values, while for a greyscale image the channel size is just 1. The extent of the connectivity along the depth axis is always equal to the depth of the input volume. It is important to emphasize again this asymmetry in how we treat the spatial dimensions (width and height) and the depth dimension: The connections are local in space (along width and height), but always full along the channel space of the input volume.

**Parameter Sharing:** This is an important consequence of a convolutional layer. Kernels are shared across all the image positions. Parameter sharing creates the following characteristics of convolution operations: (1) letting the local feature patterns extracted by kernels translation invariant as kernels travel across all the image positions and detect learned local patterns, and (2) increasing model efficiency by reducing the number of parameters to learn in comparison with fully connected neural networks.

## 2.6.3 Non-linear Activation

The outputs of a linear operation such as convolution are then passed through a nonlinear activation function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) function, were used previously because they are mathematical representations of a biological neuron behavior, the most common nonlinear activation function used presently is the rectified linear unit (ReLU), which simply computes the function:  $f(x) = \max(0, x)$  (Figure 2.4).

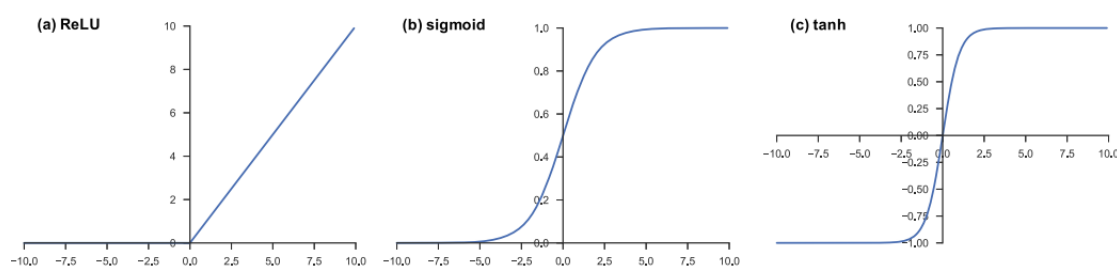


Figure 2.4: (Source - Yamashita et al.[7]) Activation functions commonly applied to neural networks: a rectified linear unit (ReLU), b sigmoid, and c hyperbolic tangent (tanh).

## 2.6.4 Loss Function

A loss function, also referred to as a cost function, measures the compatibility between output predictions of the network through forward propagation and given ground truth labels. Commonly used loss function for multiclass classification is cross entropy, whereas mean squared error is typically applied to regression to continuous values. A type of loss function is one of the hyperparameters and needs to be determined according to the given tasks.

## 2.6.5 Gradient Descent

Gradient descent is commonly used as an optimization algorithm that iteratively updates the learnable parameters, i.e., kernels and weights, of the network so as to minimize the loss. The gradient of the loss function provides us the direction in which the function has the steepest rate of increase, and each learnable parameter is updated in the negative direction of the gradient with an arbitrary step size determined based on a hyperparameter called learning rate (Figure 2.5). The gradient is, mathematically, a partial derivative of the loss with respect to each learnable parameter, and a single update of a parameter is formulated as follows:

$$w = w * \frac{L}{w}$$

where  $w$  stands for each learnable parameter,  $\alpha$  stands for a learning rate, and  $L$  stands for a loss function. It is of note that, in practice, a learning rate is one of the most important hyperparameters to be set before the training starts. In practice, for reasons such as memory limitations, the gradients of the loss function with regard to the parameters are computed by using a subset of the training dataset called mini-batch, and applied to the parameter updates. This method is called mini-batch gradient descent, also frequently referred to as stochastic gradient descent (SGD), and a mini-batch size is also a hyperparameter. In addition, many improvements on the gradient descent algorithm have been proposed and widely used, such as SGD with momentum, RMSprop, and Adam, though the details of these algorithms are beyond the scope of this article.

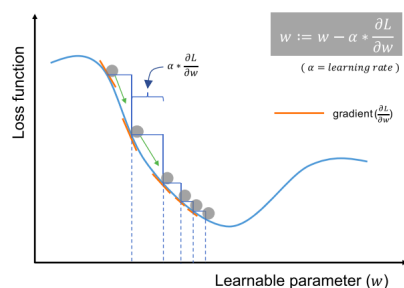


Figure 2.5: (Source - Yamashita et al.[7]) Gradient descent is an optimization algorithm that iteratively updates the learnable parameters so as to minimize the loss, which measures the distance between an output prediction and a ground truth label. The gradient of the loss function provides the direction in which the function has the steepest rate of increase, and all parameters are updated in the negative direction of the gradient with a step size determined based on a learning rate

## 2.6.6 Training a CNN

Training a network is a process of finding kernels in convolution layers and weights in fully connected layers which minimize differences between output predictions and given ground truth labels on a training dataset. Backpropagation algorithm is the method commonly used for training neural networks where loss function and gradient descent optimization algorithm play essential roles. A model performance under particular kernels and weights is calculated by a loss function through forward propagation on a training dataset, and learnable parameters, namely kernels and weights, are updated according to the loss value through an optimization algorithm called backpropagation and gradient descent, among others.

## 2.7 DeepSD by Vandal et al.[2017]: A Case Study

Local scale projections can be obtained using statistical downscaling, a technique which uses historical climate observations to learn a low-resolution to high-resolution mapping. Depending on statistical modeling choices, downscaled projections have been shown to vary significantly in terms of accuracy and reliability. The spatio-temporal nature of the climate system motivates the adaptation of super-resolution image processing techniques to statistical downscaling. In this work, they presented DeepSD, a generalized stacked super resolution convolutional neural network (SRCNN) framework for statistical downscaling of climate variables. DeepSD augments SRCNN with multi-scale input channels to maximize predictability in statistical downscaling. They provide a comparison with Bias Correction Spatial Disaggregation (BCSD) as well as three Automated-Statistical Downscaling approaches in downscaling daily precipitation from  $1^\circ$  (100km) to  $1/8^\circ$  (12.5km) over the Continental United States.

### 2.7.1 Motivation

Advances in single image super-resolution (SR) correspond well to statistical downscaling, which learns a mapping between low and high-resolution images. Moreover, as SR methods attempt to generalize across images, they aim to provide downscaled climate projections to areas without high-resolution observations through what may be thought of as transfer learning. They found that super-resolution convolutional neural networks were able to capture spatial information in climate data to improve beyond existing methods. The sparse coding generalization, non-linearity, network flexibility, and scalability to large datasets presents an opportunity to apply SRCNN to statistical downscaling.

### 2.7.2 Interpretation of the Climate Data as Images

Each of the earth science data products inherently possess rich spatial dependencies, much like images. However, traditionally statistical downscaling methods, particularly regression based models, vectorize spatial data, removing this spatial structure. While colored images contain channels consisting of, for example, red, green, and blue, climate data may be represented analogously such that the channels correspond to climate variables and topographical data. Similar approaches have been applied to satellite datasets for image classification and resolution enhancement. Though climate data is more complex than images due to its dynamics and chaotic nature, they propose that this representation allows scientists to approach the data in an unconventional manner and apply augmented models developed for image processing. Using the analogy between climate datasets and images, they relate statistical downscaling to image super-resolution, where one aims to learn a mapping from low to high-resolution image pairs. Specifically, single image super-resolution (SR), as the name suggests, increases the resolution of a single image, rather than multiple images, from a scene.

### 2.7.3 Data

Often, SD models are built to downscale GCM simulations directly to a observational station while others aim to downscale to a grid based dataset. Gridded observational datasets are often built by aggregating station observations to a defined grid.



They obtain precipitation through the PRISM dataset at a 4km daily spatial resolution which aggregates station observations to a grid with physical and topographical information. They then upscale the precipitation data to  $1/8^\circ$  ( 12.5 km) as the high-resolution observations. Following, they upscale further to  $1^\circ$  corresponding to a low-resolution precipitation. The goal is then to learn a mapping between these low-resolution and high-resolution datasets.

They use daily precipitation from the PRISM dataset and elevation from Global 30 Arc-Second Elevation Data Set (GTOPO30) provided by the USGS. These datasets are used to train and test the DeepSD framework, which is compared to BCSD, a widely used statistical downscaling technique, as well as three off-the-shelf machine learning regression approaches. The years 1980 to 2005 were used for training (9496 days) while the years 2006 and 2014 (3287 days) were used for testing.

### 2.7.4 Super-Resolution CNN

Super-resolution methods, given a low-resolution (LR) image, aim to accurately estimate a high-resolution image (HR). As presented by Dong et al.[8], a CNN architecture can be designed to learn a functional mapping between LR and HR using three operations, patch extraction, non-linear mappings, and reconstruction. The LR input is denoted as  $X$  while the HR label is denoted as  $Y$ .

A three layer CNN is then constructed as follows to produce a high resolution estimate and presented in Figure 2.6. Layer 1 is formulated as

$$F_1(X) = \max(0, W_1 * X + B_1),$$

where  $*$  is the convolution operation and the  $\max$  operation applies a Rectified Linear Unit while  $W_1$  and  $B_1$  are the filters and biases, respectively.  $W_1$  consists of  $n_1$  filters of size  $c \times f_1 \times f_1$ . The filter size,  $f_1 \times f_1$ , operates as an overlapping patch extraction layer where each patch is represented as a high-dimensional vector.

Correspondingly, layer 2 is a non-linear operation such that

$$F_2(X) = \max(0, W_2 * F_1(X) + B_2),$$

where  $W_2$  consists of  $n_2$  filters of size  $n_1 \times f_2 \times f_2$  and  $B_2$  is a bias vector. This non-linear operation maps high-dimensional patch-wise vectors to another high-dimensional vector. A third convolution layer is used to reconstruct an HR estimate such that

$$F(X) = W_3 * F_2(X) + B_3.$$

Here,  $W_3$  contains 1 filter of size  $n_2 \times f_3 \times f_3$ . The reconstructed image  $F(X)$  is expected to be similar to the HR image,  $Y$ . This end-to-end mapping is then required to learn the parameters  $\Theta = W_1, W_2, W_3, B_1, B_2, B_3$ . A Euclidean loss function with inputs  $\{X_i\}$  and labels  $\{Y_i\}$  is used where the optimization objective is defined as:

$$\operatorname{argmin}_{\Theta} \sum_{i=1}^n \|F(X_i; \Theta) - Y_i\|_2^2$$

such that  $n$  is the number of training samples (batch size).

The convolutions in layers 1, 2, and 3 decrease the image size depending on the chosen

filter sizes,  $f_1$ ,  $f_2$ , and  $f_3$ . At test time, padding using the replication method is applied before the convolution operation to ensure the size of the prediction and ground truth correspond. During training, labels are cropped such that  $Y$  and  $F(X_i; \Theta)$ , without padding, are of equal size.

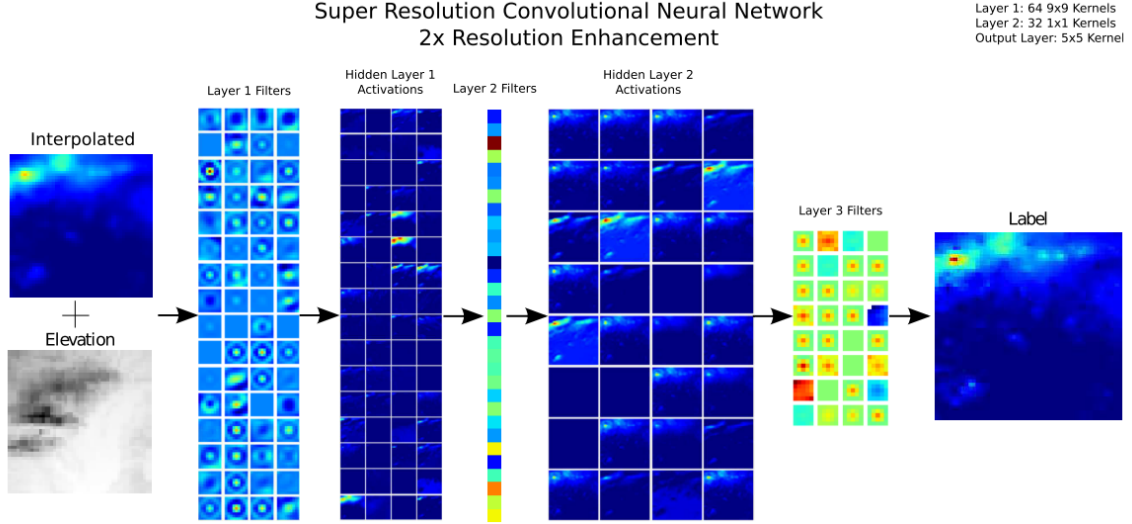


Figure 2.6: (Source - Vandal et al.[3]) Augmented SRCNN Architecture. From the left to right: Precipitation and Elevation sub-image pair, filters learned in layer 1, layer 1 activations, layer 2 filters, layer 2 activations, layer 3 filters, and HR precipitation label.

### 2.7.5 The DeepSD Framework

When applying SR to images we generally only have a LR image to estimate a HR image. However, during SD, we may have underlying high-resolution data coinciding with this LR image to estimate the HR images. For instance, when downscaling precipitation we have two types on inputs including LR precipitation and static topographical features such as HR elevation and land/water masks to estimate HR precipitation. As topographical features are known beforehand at very high resolutions and generally do not change over the period of interest they can be leveraged at each scaling factor. So, they train 3 SRCNNs stacked together, where each SRCNN is trained independently with its associated input/output pairs. Traditional SR methods are built for resolution enhancements of factors from 2 to 4 while statistical downscaling conservatively requires resolution increases of factors from 8 to 12. Hence, stacked SRCNNs are used in DeepSD.

Inference is executed by starting with the lowest resolution image with its associated HR elevation to predict the first resolution enhancement. The next resolution enhancement is estimated from the previous layers estimate and its associated HR elevation. This process is repeated for each trained SRCNN. Figure 2.7 illustrates this process with a precipitation event and its various resolution improvements. We see that this stacked process allows the model to capture both regional and local patterns.

The experiments downscale daily precipitation from  $1.0^\circ$  to  $1/8^\circ$ , an  $8\times$  resolution enhancement, using three SRCNN networks each providing a  $2\times$  resolution increase ( $1.0^\circ \rightarrow 1/2^\circ \rightarrow 1/4^\circ \rightarrow 1/8^\circ$ ).

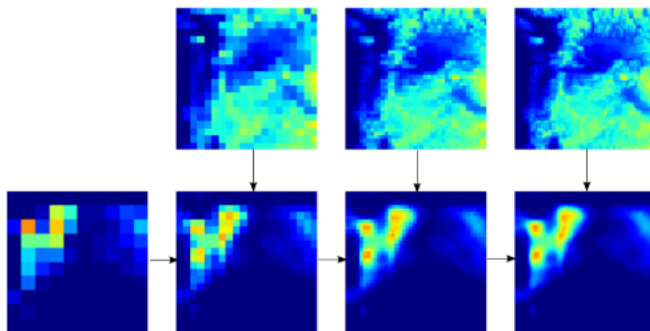


Figure 2.7: (Source - Vandal et al.[3]) Layer by layer resolution enhancement from DeepSD using stacked SRCNNs. Top Row: Elevation, Bottom Row: Precipitation. Columns:  $1.0^\circ$ ,  $1/2^\circ$ ,  $1/4^\circ$  and  $1/8^\circ$  spatial resolutions.

## 2.7.6 Results and Inference

- First experiment compares six approaches, DeepSD, SRCNN (w/o stacking), BCSD, Lasso, SVM, and ANN, on their ability to capture daily predictability, presented in Figure 2.8 four metrics computed and averaged over the 1000 randomly selected locations in CONUS where ASD methods were trained.

Model	Bias (mm/day)	Corr	RMSE (mm/day)	Skill	Runtime (secs)
Lasso	0.053	0.892	2.653	0.925	1297
ANN	0.049	0.862	3.002	0.907	2015
SVM	-1.489	0.886	3.205	0.342	27800
BCSD	-0.037	0.849	4.414	<b>0.955</b>	<b>13</b>
SRCNN	-0.699	0.894	2.949	0.833	24
DeepSD	<b>0.022</b>	<b>0.914</b>	<b>2.529</b>	0.947	71

Figure 2.8: (Source - Vandal et al.[3]) Comparison of predictive ability between all six methods for 1000 randomly selected locations in CONUS. Runtime is computed as the amount of time to downscale 1 year of CONUS.

- In the next experiment DeepSD and BCSD, the two scalable and top performing methods from the previous experiment, are chosen. Each metric is computed per location and season using the daily observations and downscaled estimates then averaged over CONUS, as presented in Figure 2.9. DeepSD has high predictive capabilities for all seasons, higher correlation and lower RMSE, when compared to BCSD.

Season	Model	Bias (mm/day)	Corr	RMSE (mm/day)	Skill <sup>1</sup>
DJF	BCSD	<b>0.02</b>	0.89	2.36	0.95
	DeepSD	-0.03	<b>0.95</b>	<b>1.53</b>	<b>0.94</b>
JJA	BCSD	<b>0.01</b>	0.78	4.15	<b>0.92</b>
	DeepSD	-0.05	<b>0.86</b>	<b>3.29</b>	0.91
MAM	BCSD	<b>0.01</b>	0.87	3.02	<b>0.94</b>
	DeepSD	-0.03	<b>0.93</b>	<b>2.29</b>	0.93
SON	BCSD	<b>0.01</b>	0.87	3.27	<b>0.94</b>
	DeepSD	-0.04	<b>0.93</b>	<b>2.31</b>	<b>0.94</b>

Figure 2.9: (Source - Vandal et al.[3]) Comparison of Predictive Ability between DeepSD and BCSD for each season, Winter, Summer, Spring, and Fall. Values are computed at each location in CONUS and averaged.

- In a separate experiment they find that BCSD over-estimates extremes at upper quantiles while DeepSD is relatively stable. Though RMSE, Corr, and Skill becomes worse at these extremes, DeepSD consistently outperforms BCSD, most often with thinner confidence bounds. These results show DeepSDs ability to perform well for increasingly extreme precipitation events. With this, they hypothesize that capturing nearby spatial information allows DeepSD to isolate areas where extreme precipitation events are more likely than others.

## 2.8 Summary

With the two case-studies mentioned, its easy to analyze the limitations of the two models. Though extensive handcrafted approach by Salvi et al. makes some great conclusions, we feel that the results can be improved upon using a Deep Learning pipeline for the same. On the other hand, even though Vandal et al. use CNNs for downscaling the rainfall projections they use observed historical data to train a model which can't predict future projections as they never cited experiments with any form of simulated data as the input. Hence, the aim of this research is to obtain high-resolution daily precipitation projections of the Indian landmass with simulations of coarse resolution as the input to the model.

# Chapter 3

## Implementation and Experiments

This chapter deals with the details of my implementation of multiple variants of convolutional neural networks for obtaining high-resolution rainfall projections from predictor variables. Essentially, my implementation is based on the analogy of treating climate data as images for training CNNs as cited by Vandal et al.[3]. However, the objective of my problem adheres to that of the work by Salvi et al.[6].

### 3.1 Data

- **Reanalysis:** Source of predictors is the reanalysis from NCEP/NCAR, the one used in Salvi et al.'s study[6]. The climate variables used for the input are include surface level predictors viz. sea level pressure, u-wind, v-wind, temperature; lower pressure level variables viz. specific humidity, temperature, u-wind, v-wind at 850 mbar and geopotential height, temperature at 500 mbar. In all these 10 predictors were selected as an input for the model. These predictors are of the resolution of  $2.5^\circ$  available from 1951 to 2005 for everyday.
- **Host GCM:** CESM simulations which can bias corrected with respect to the reanalysis and will serve as the input for the trained model. The available data is at  $1.25^\circ$  resolution from 1850 to 2005 (past phase) and from 2006 to 2100 (future phase).
- **Observed Rainfall:** Daily data from APHRODITE is available at  $0.25^\circ$  resolution serve as the high-resolution ground truth for the model. This data spans covers each day's rainfall values in the form of gridded data from 1951 to 2007.

### 3.2 Model Definition

- One deep convolutional network that takes the input of a stack of 10 predictors (low-resolution) of 2-D grids with latitudes  $5^\circ - 40^\circ$  and longitudes  $65^\circ - 100^\circ$ , returning the grid of observed rainfall in high-resolution ( $0.25^\circ$ ). This is essentially the single image super-resolution problem which was discussed earlier.
- The RGB pixel values of an image are analogous to the values of 10 predictors for each node in the  $2.5^\circ$  resolution stack (can be referred to as the input tensor). The output is a single channelled 2-D grid with rainfall values (can be compared to a greyscale image) of  $0.25^\circ$  resolution (output tensor).

- The network is being considered a black box for now, until I discuss the overall objective. The goal is to produce the corresponding output tensor for an input tensor of any date. This output tensor is the target prediction we would want the model to generate accurately with respect to the ground truth (observed rainfall in high-resolution).
- This way if the model performs well on unseen data from the past, we can assume it to work reasonably for the future (for which we don't have the observed rainfall).
- Data Distribution:
  - Training ( $\sim 70\%$ ): 38 years (1951-1988) i.e.  $38 \times 365 = 13870$  training samples.
  - Validation ( $\sim 20\%$ ): 11 years (1989-1999) i.e.  $11 \times 365 = 4015$  validation samples.
  - Testing ( $\sim 10\%$ ): 6 years (2000-2005) i.e.  $6 \times 365 = 2190$  testing samples

### 3.3 Network Heuristics

Stacked SRCNNs[3] will not be effective in this case. Why? Well, this would mean bicubic interpolation of the input tensor into the the target resolution, and since the the input tensor and output tensor represent different forms of data, learning a non-linear mapping for the model would be even more difficult. This problem demands the network to learn a scaling factor of  $10\times$  which is huge. Also, to build a network that can construct high-resolution tensors from low-resolution input (without interpolation to the target resolution) from learning the feature spaces requires the network to be significantly deep. The idea of implementing zone-based training[6] for the entire Indian landmass has been dropped in this model since the focus was on trying out CNNs on the entire gridded data first.

Transposed convolution and dilation were two important types of layers that were used to form the 26-layered network. Training strategies like dropout regularizer, Adam optimizer and cyclic learning rate scheduler with warm restarts were used during the training of the network. Except for the final layer, all other layers were followed by ReLU activation for effective non-linearity. Mean squared error was used as the loss function.

#### 3.3.1 Transposed Convolutions for Up-sampling

The reverse of convolution is a one-to-many problem and can't be fully true. However, there is a procedure of transposed convolution by which researchers have tried to visualize convolutional filters, and seems to act as a good enough proxy for convolution. This work by Dumoulin et al.[9] discusses it in great detail.

The forward and backward passes for this layer is swapped, compared to the usual convolution layer. The need for transposed convolutions generally arises from the desire to use a transformation going in the opposite direction of a normal convolution, i.e., from something that has the shape of the output of some convolution to something that has the shape of its input while maintaining a connectivity pattern that is compatible with said convolution. Figure 3.1 provides a good visualization of its arithmetic.

In this network, we used 3 transposed convolutional layers at different positions to achieve  $10\times$  upsampled output.

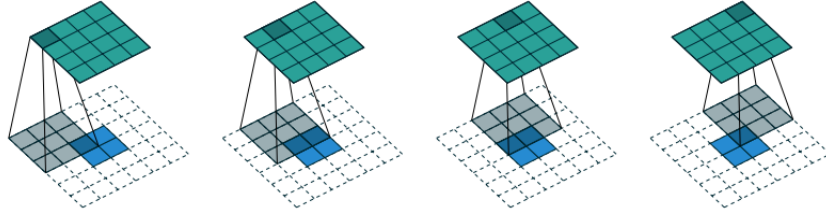


Figure 3.1: (Source - Dumoulin et al.[9]) The transpose of convolving a  $3 \times 3$  kernel over a  $4 \times 4$  input using unit strides (i.e.,  $i = 4$ ,  $k = 3$ ,  $s = 1$  and  $p = 0$ ). It is equivalent to convolving a  $3 \times 3$  kernel over a  $2 \times 2$  input padded with a  $2 \times 2$  border of zeros using unit strides.

### 3.3.2 Dilated Convolution

Dilation[11] is largely the same as run-of-the-mill convolution, except that it introduces gaps into its kernels, i.e. whereas a standard kernel would typically slide over contiguous sections of the input, its dilated counterpart may, for instance, "encircle" a larger section of the image –while still only have as many weights/inputs as the standard form.

In the present study, four 2-dilated convolutional layers appear in initial part of the network to capture multi-scale context from the input which might be useful in reconstruction.

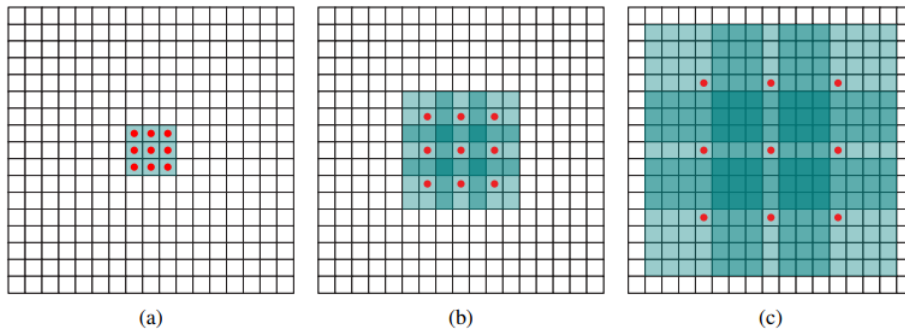


Figure 3.2: (Source - Fisher et al.[11]) Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a)  $F_1$  is produced from  $F_0$  by a 1-dilated convolution; each element in  $F_1$  has a receptive field of  $3 \times 3$ . (b)  $F_2$  is produced from  $F_1$  by a 2-dilated convolution; each element  $F_2$  has a receptive field of  $7 \times 7$ . (c)  $F_3$  is produced from  $F_2$  by a 4-dilated convolution; each element in  $F_3$  has a receptive field of  $15 \times 15$ . The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

### 3.3.3 Training Strategies

- **Dropout:** A regularization technique by Srivastava et al.[?] where randomly selected neurons are ignored during training. They are dropped-out randomly with some valid probability. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn

results in a network that is capable of better generalization and is less likely to overfit the training data.

- **Adam:** Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. Kingma describes Adam as combining the advantages of two other extensions of stochastic gradient descent i.e. AdaGrad and RMSProp.
- **SGDR:** "Stochastic gradient descent with warm restarts" is a variant of learning rate annealing[13], which gradually decreases the learning rate through training. In order to find a stable local minimum, we can increase the learning rate from time to time, encouraging the model to jump from one local minimum to another if it is in a steep trough. This is the restarts in SGDR.

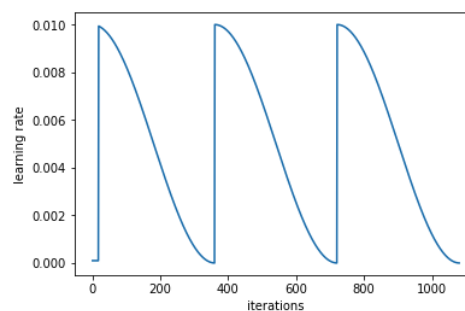


Figure 3.3: (Source - fast.ai) Increasing the learning rates every few iterations to restart the gradient descent.

## 3.4 Results

Unfortunately, the results are poor. The model is unable to capture the spatial complexity of the precipitation values. There seem to be too many layers in the network compared to the amount of data available due to which the model overfits. Following sets of images are 3 instances of how the prediction on the test data looks like when compared to the ground truth. Figure 7.2 clearly shows the shortcoming of the model of not being able to predict the rainfall projection values in the appropriate range. Though, the model seems approximately correct in capturing the rainfed areas for a given date.



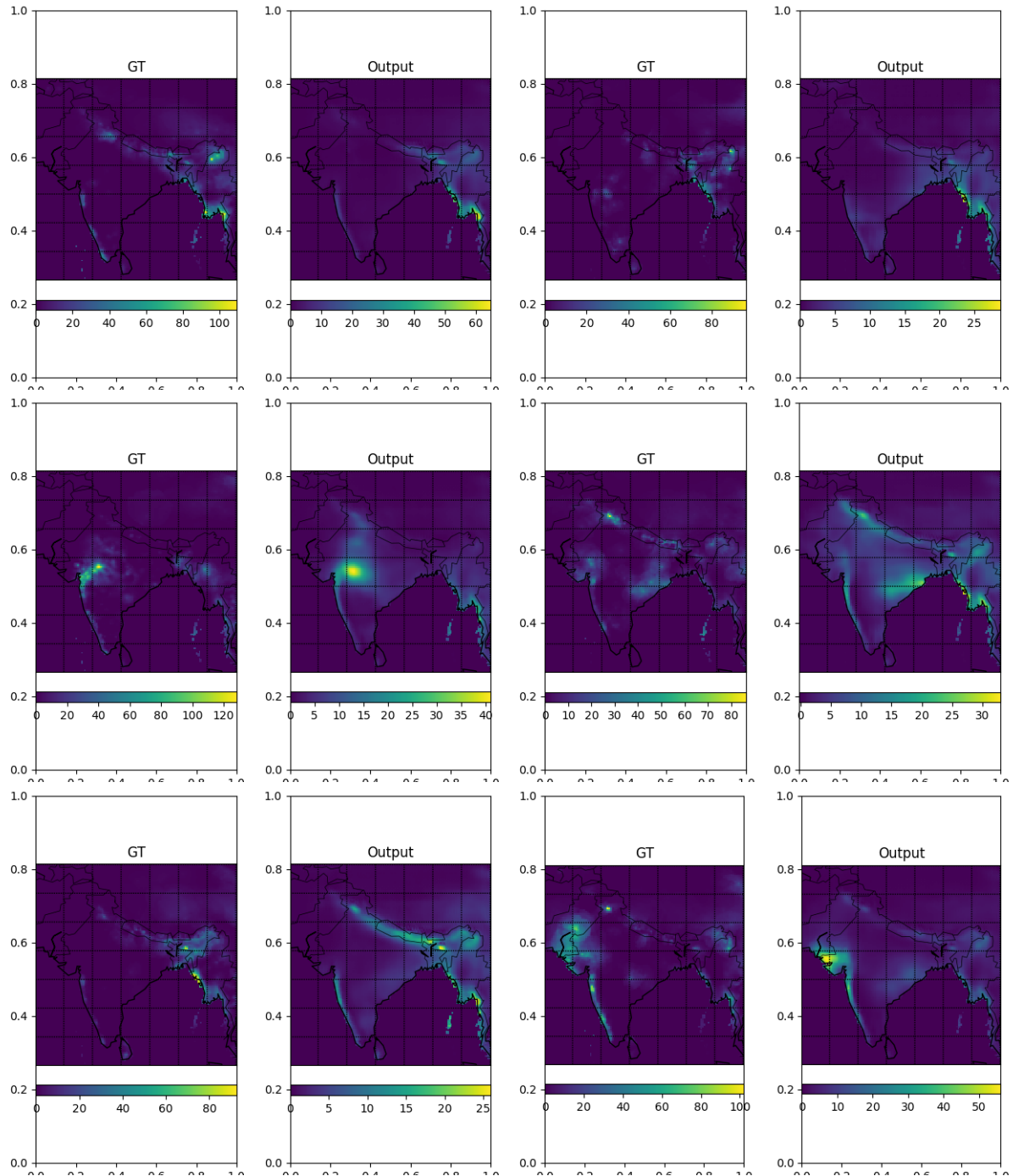


Figure 3.4: 6 pairs of predicted output and ground truth projections randomly picked from the testing samples.

# Chapter 4

## Inference and Conclusion

- With the available data and no extra information, its very difficult to train a single model, however deep, that could be generalized on all locations of the observed area. This suggests that we should use variables like elevation, land use/land cover as auxiliary parameters to the network since these are relatively time invariant and also available in very high resolution.
- The plots in Figure 7.2 show that the model is unable to capture the variability in the range of rainfall values across time and space. Hence, it is required that we take a shallower network and train the meteorologically homogenous zones individually and later combine the trained models into one ensemble.
- The  $10\times$  super-resolution is a difficult problem, especially when the input and ground truth samples do not have a very good correlation. If at all possible, we should try getting the reanalysis from some other prejects like CERA-20C that provide the reanalysis data at  $1.0^\circ$  resolution.
- A very deep network like this one doesn't have enough data to train on, which makes learning the super-resolution problem really difficult in such a case. The network devised, consists of 1.2M trainable parameters which is quite large for an input of size 13870 training samples.

# Chapter 5

## Future Work

- Write a shallower network with less parameters so that the number of training samples are just enough for the network to train.
- This network should receive the input in the form of zone-wise projections based on the meteorologically homogenous zones presented in the study by Salvi et al.[6]. This will ensure that only local factors affect the rainfall projections of a particular region and a more general model does not perturb the effects of the already existing phenomena responsible for rainfall in a region during that time. The different versions of the model could then be ensembled together.
- Use topographical features like elevation and land use/land cover from the NRSC project. These variables are available at very high-resolutions and aren't subject to change in time, hence they can be used very effectively as auxiliary parameters for the network.
- Once the model performs well with the reanalysis, we can obtain bias corrected GCM simulations for the required period and perform relevant experiments, comparing the overall downscaling quality with the previously mentioned kernel regression pipeline by Salvi et al.[6], assessing the quality of seasonal prediction, and later gauging the model's suitability to make extreme rainfall predictions.

# **Part II**

## **BTP: Phase II**

# Chapter 6

## Methodology and Implementation

This phase mainly deals with the experimentation with network architecture, model heuristics, and training strategies using some state-of-the-art methods to learn the most optimal transfer function for statistical downscaling of historical data.

### 6.1 Problem Reformulation

Considering the enormity of this full-fledged problem, I started off with setting a modest goal to this project. For this phase, my main focus has been to train a transfer function in the form of a deep convolutional network, with input being the NCEP/NCAR reanalysis simulations, and the ground truth being the observed rainfall.

### 6.2 Data

- **Reanalysis:** Source of predictors is the reanalysis from NCEP/NCAR, the one used in Salvi et al.'s study[6]. The climate variables used for the input are include surface level predictors viz. sea level pressure, u-wind, v-wind, temperature; lower pressure level variables viz. specific humidity, temperature, u-wind, v-wind at 850 mbar and geopotential height, temperature at 500 mbar. In all these 10 predictors were selected as an input for the model. These predictors are of the resolution of  $2.5^\circ$  available from 1951 to 2005 for everyday.
- **Observed Rainfall:** Daily data from APHRODITE is available at  $0.25^\circ$  resolution serve as the high-resolution ground truth for the model. This data spans covers each day's rainfall values in the form of gridded data from 1951 to 2007.

Although the data is available for each day of the year, the model was trained and validated only for the monsoon period (June-September i.e. 122 days) of each year. Modelling rainfall for just the monsoon period is supposedly less complicated. Also, Salvi et al.[6] stated that the reanalysis predictors showed correlation specifically for the monsoon period only. Following is the data split used in the experiments:

- Training ( $\sim 70\%$ ): 38 years (1951-1988) i.e.  $38 \times 122 = 13870$  training samples.
- Validation ( $\sim 20\%$ ): 11 years (1989-1999) i.e.  $11 \times 122 = 4015$  validation samples.
- Testing ( $\sim 10\%$ ): 6 years (2000-2005) i.e.  $6 \times 122 = 2190$  testing samples

## 6.3 Data Preprocessing

Without any extra information, learning this transfer function as a single generalized model for the whole of Indian landmass has been shown to be very difficult, owing to the high degree of super-resolution and varying local rainfall patterns for different regions of India. The earlier experiments (see Sections 3.4 and 4) proved that it is hard to capture the variability across time and space.

### 6.3.1 Disintegration into Zones

As identified by IMD, the Indian landmass can be disintegrated into meteorologically homogeneous zones[6] as per the Figure 6.1. For each zone entity in the observed rainfall data, a corresponding regularly shaped region from the reanalysis is considered as the input. Rainfall in a particular zone is assumed to be influenced by the large-scale predictors within the selected region in the reanalysis data.

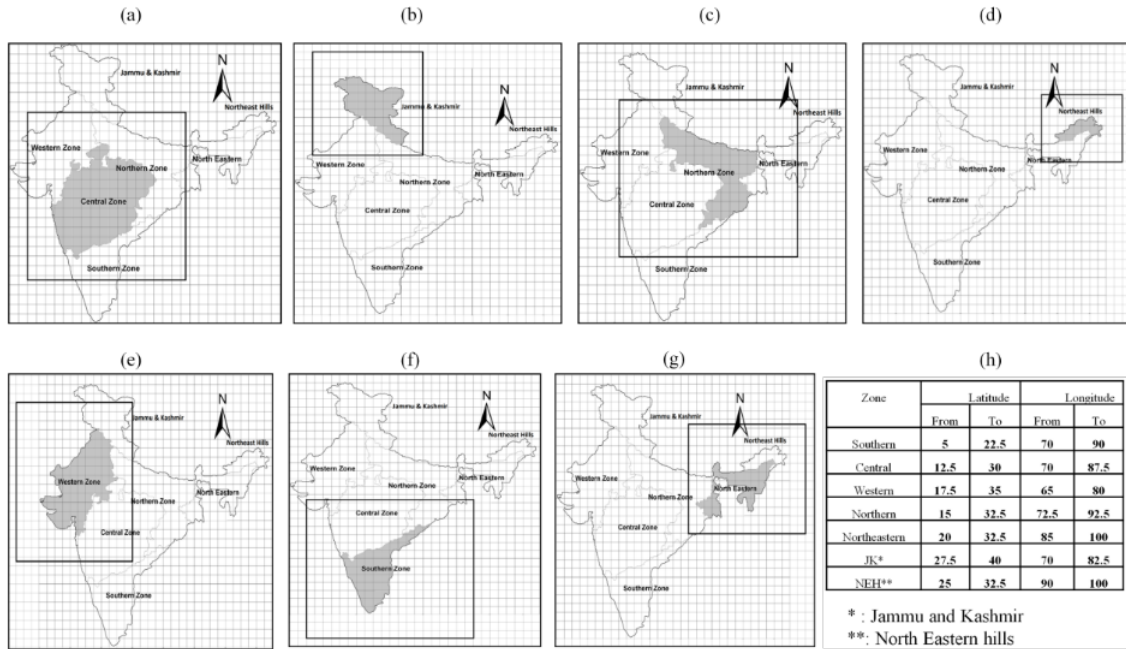


Figure 6.1: (Source - Salvi et al.[6]) The region of predictors (as shown with black rectangles) for each meteorologically homogeneous zone (as shown with gray shade) are illustrated (a) central, (b) Jammu and Kashmir, (c) North, (d) Northeast hills, (e) Western, (f) South, and (g) Northeast. The extent of the region of predictors in terms of latitude and longitudes are detailed in (h).

Therefore, it was decided to train a separate network for each zone based on its input and output size. As far as this project is concerned, I could only train the downscaling of the central region. Hence, all the experiments reported in the following sections, have been conducted on the central region only.

### 6.3.2 Central Region

As shown Figure 6.1(h), the central input region spans from latitudes  $12.5^\circ$  to  $30^\circ$ , and from longitudes  $70^\circ$  to  $87.5^\circ$  encompassing  $10 \times 8 \times 8$  volume as the input feature space.

The observed rainfall should be predicted for a masked region in  $0.25^\circ$  resolution as shown in Figure 6.2.

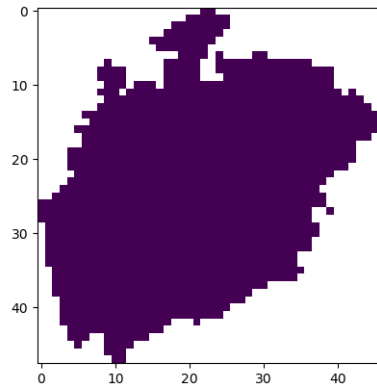


Figure 6.2: The central region (1273 pixels) enclosed within the bounding box of size  $48 \times 47$ , owing to the  $0.25^\circ$  resolution of the ground truth.

To study the distribution of rainfall in the central region over 55 monsoon periods (1951-2005), a percentile plot was created as shown in the Figure 6.3. For the central region, about 24.6% of the total rainfall values are zero. From the figure below, the skewness of the distribution is clearly evident. The rainfall values range from zero to 289.12 mm/day. However, the 98<sup>th</sup> percentile corresponds to a value as small as 38.44 mm/day.

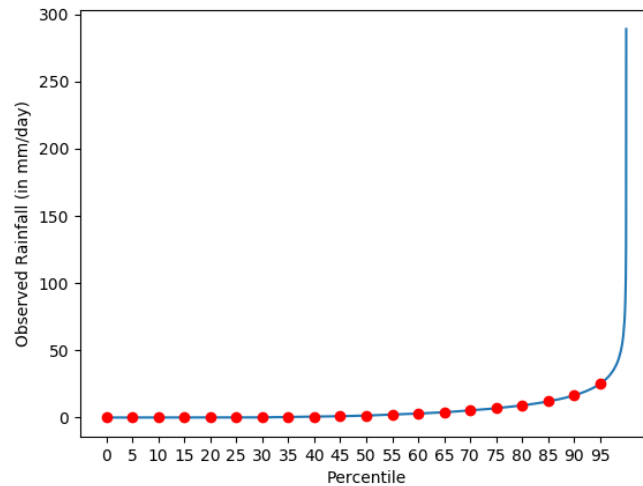


Figure 6.3: Percentile plot of the observed rainfall of central region over 55 monsoon periods.

### 6.3.3 Data Normalization

Since all of the 10 different predictors take different range of values, its necessary to standardize each predictor's values so that range of every predictor becomes  $[0, 1]$ . Such

a type feature scaling is important so that all the input features can be of the same range and hold the same level significance as far as predicting the regional rainfall is concerned. Besides, standardizing the data helps the model to converge faster while gradient descent.

It is also recommended to scale the target rainfall values in the range of  $[0, 1]$ . If the target variable has a large spread of values, it may result in large error gradient values causing weight values to change dramatically, making the learning process unstable. With reference to the percentile plot (Figure 6.3), very large values seem to act as outliers with respect to the rest of the distribution. For this reason, while standardizing the maximum value of the distribution was taken to be 100 mm/day which corresponds to 99.9% value according to that plot. Hence, the predictions obtained from the model will be within the range  $[0, 1]$ , where a prediction of 1 (100 mm/day) corresponds to the maximum rainfall. Although this approach lets the model suffer at extreme rainfall prediction, yet it provides more stable training for the model so that it predicts more accurately for the cases of low to moderate rainfall.

## 6.4 Network Heuristics

Apart from using layers of transposed and dilated convolutions, densely connected convolutional layers have been introduced at the very beginning of the network where most of the feature learning takes place. The training strategies used in the process are same as reported in the previous part (Section 3.3.3) i.e. Dropout, Adam and SGDR. Each convolutional layer is accompanied by batch normalization[15] and ReLU activation. Some of the last layers were trained with dropout regularization to overcome overfitting. RMSE (Root Mean Squared Error) is used as the evaluation metric over MAE (Mean Absolute Error) since it is much more sensitive to outliers in the training data.

### 6.4.1 Dense Connections

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. This observation is embraced in a dense convolutional network[14], which connects each layer to every other layer in a feed-forward fashion. For each layer, the feature-maps of all preceding layers are stacked and used as input, and its own feature-maps are used as input into all subsequent layers. A dense block has several compelling advantages: it alleviates the vanishing-gradient problem, strengthens feature propagation, encourages feature reuse, and substantially reduces the number of parameters.

Counter-intuitive effect of this dense connectivity pattern is that it requires fewer parameters than a traditional convolutional network, as there is no need to relearn redundant feature maps. Traditional feed-forward architectures can be viewed as algorithms with a state, which is passed on from layer to layer. Each layer reads the state from its preceding layer and writes to the subsequent layer. It changes the state but also passes on information that needs to be preserved.

Besides better parameter efficiency, one big advantage of densely connected blocks is their improved flow of information and gradients throughout the network, which makes them easy to train. Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision. Further, it is observed that dense connections have a regularizing effect, which avoids the overfitting



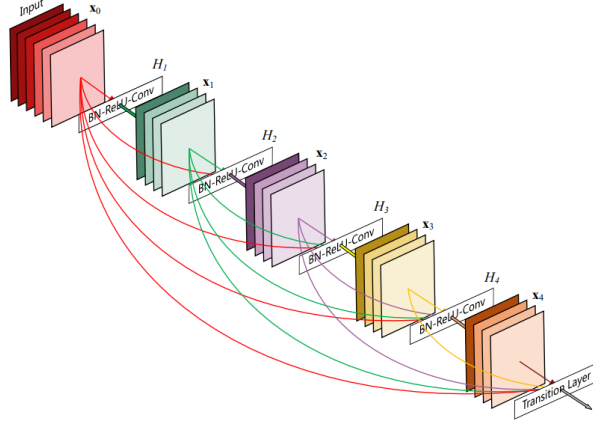


Figure 6.4: A 5-layer dense block with a growth rate of  $k = 4$ . Each layer takes all preceding feature-maps as input.

problem. Concatenating feature maps learned by different layers increases variation in the input of subsequent layers and improves efficiency.

As a direct consequence of the input concatenation, the feature maps learned by any of the densely connected layers can be accessed by all subsequent layers. This encourages feature reuse throughout the network, and leads to a more compact model.

In the context of the statistical downscaling problem, we need the encoding of the input predictor maps to be very efficient, so that the encoded feature maps can be decoded into high-resolution rainfall projections.

### 6.4.2 Complete Network Architecture

The network consists of 4 major parts: (a) an encoding block consisting of 7 densely connected convolutional layers, (b) a 5-layered feed forward dilated convolutional bloc, (c) 4 transpose blocks with each block increasing the height and width of the subsequent feature map by  $2\times$ , and (d) simple feed-forward convolutional block to make the size of the output same as the ground truth.

# Chapter 7

## Results and Inference

At optimal convergence, the training loss (RMSE) is achieved to be as 2 mm/day which is an average loss over all of the training data. Though the corresponding validation RMSE is 8 mm/day, the model didn't overfit according to my understanding. The trained model incurred an average of 6.3 mm/day RMSE loss for the test dataset i.e. monsoon periods for the last 6 years. As far as mean absolute error (MAE) is concerned, the validation MAE was 4.3 mm/day throughout the 11 monsoon periods. The MAE on the test dataset is 4.1 mm/day.

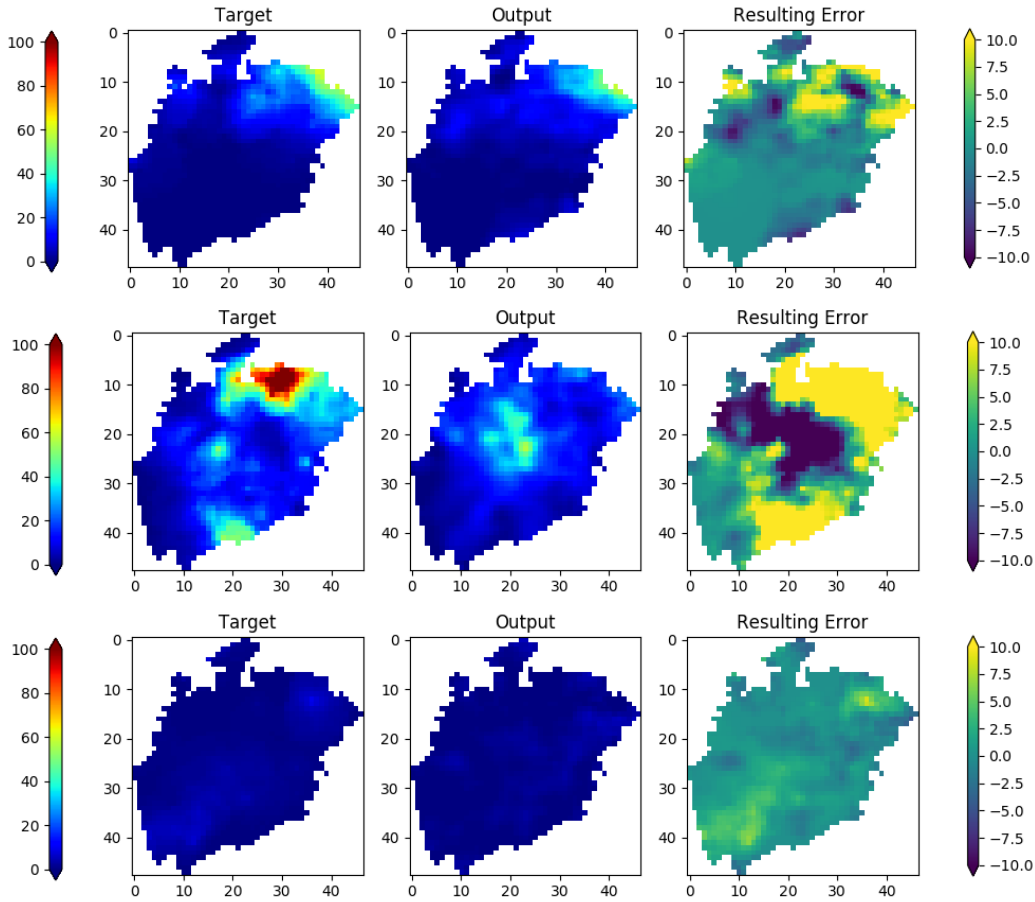


Figure 7.1: 3 sets of target, predicted output and error projections randomly picked from the testing samples. MAE (in mm/day) for cases from top to bottom: 2.68, 13.73 and 1.32

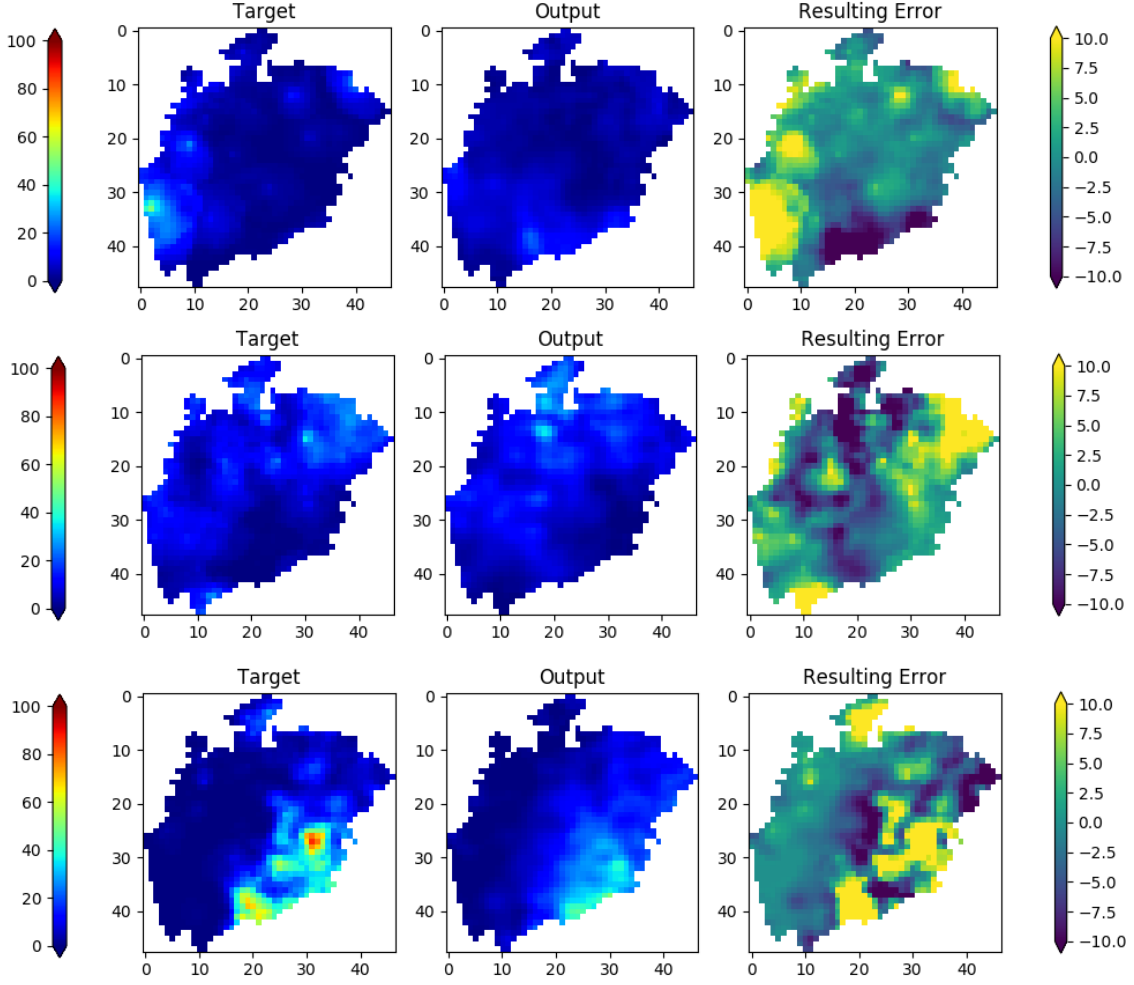


Figure 7.2: Some more sets of target, predicted output and error projections randomly picked from the testing samples. MAE (in mm/day) for cases from top to bottom: 4.24, 5.17 and 5.62

It's evident that regions having high rainfall in the target maps, often incur huge losses while prediction. Although regions with low to moderate rainfall are predicted with sufficiently low error. This is primarily due to the inability to find a robust method to model the skewness of the data distribution as far as the percentile plot (Figure 6.3) is concerned. Also, the predictions made are relatively smooth with respect to the neighbours' values. The zone-wise prediction seems to have delivered much better results. Although, this methodology is yet to be implemented with other regions. To improve on this model, more information in the form of elevation data can be used as the auxiliary information. Some non-linear invertible transformation of the target rainfall projection should be decided so that the very high values of rainfall can be predicted appropriately predicted with sufficiently small error. In fact, the last activation layer can also be designed in such a way that it is able to model the abruptness in the target rainfall projections.

# Chapter 8

## Future Prospects

- Experiment with non-linear transformations on the target rainfall projections so that the information in terms of the variability and extremity in the data can be captured more effectively.
- Devise a novel loss function that acts as a better critic for training the model, especially that is more sensitive to high rainfall values.
- Design and train similar network architectures for all the other 7 regions based on their input and output sizes.
- Use elevation map as a different channel input in the low-resolution (same as the predictors' resolution) along with the other predictors with the intent of providing the topographical feature as information for predicting local level rainfall.
- Once the model aces on the reanalysis, we can obtain bias corrected GCM simulations for the required period and perform relevant experiments similar to those done by Salvi et al.[6], assessing the quality of seasonal prediction, and later gauging the model's suitability to make extreme rainfall predictions for the future.

# Acknowledgments

I would like to express my gratitude to professors Subimal Ghosh and Amit Sethi for their valued guidance and continuous support during the entire course of work. I'm deeply indebted to Mr. Deepak Anand and Mr. Anamitra Saha, PhD scholars in the Departments of Electrical Engineering and Civil Engineering respectively, for their endless support in this research exposition. The compute requirements were fulfilled by Prof. Sethi's research lab MeDAL, Electrical Engineering Department.

Videsh Suman

23<sup>rd</sup> April, 2019

Indian Institute of Technology Bombay

# References

- [1] Kaustubh Anil Salvi, Subimal Ghosh. Fine Resolution Projections of Climate Variables and Meteorological Extremes with Special Emphasis on Assumption of Stationarity in Data Driven Methods
- [2] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks
- [3] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, Auroop Ganguly. DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution.
- [4] Justin Ker, Lipo Wang, Jai Rao, Tchoyoson Lim. Deep Learning Applications in Medical Image Analysis.
- [5] Sulagna Gope, Sudeshna Sarkar, Pabitra Mitra, Subimal Ghosh. Early Prediction of Extreme Rainfall Events: A Deep Learning Approach.
- [6] Kaustubh Salvi, Kannan S., Subimal Ghosh. Highresolution multisite daily rainfall projections in India with statistical downscaling for climate change impacts assessment.
- [7] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, Kaori Togashi. Convolutional neural networks: an overview and application in radiology.
- [8] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks.
- [9] Vincent Dumoulin, Francesco Visin. A guide to convolution arithmetic for deep learning.
- [10] Fisher Yu, Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions.
- [11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting.
- [12] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization.
- [13] Ilya Loshchilov, Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts.
- [14] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks.

- 
- [15] Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.