Deep Convolutional Elman Jordan Neural Networks

A Project Report

submitted by

DHRUV CHOPRA

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2021

THESIS CERTIFICATE

This is to certify that the thesis titled **Deep Convolutional Elman Jordan Neural Networks**, submitted by **Dhruv Chopra**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor and Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. V. Srinivasa Chakravarthy
Research Guide
Professor
Dept. of Biotechnology
IIT-Madras, 600 036

Prof. Harishankar Ramachandran
Research Co-Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: June 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my guide Dr. V. Srinivasa Chakravarthy whose teachings motivated me to pursue my current area of research. I was able to learn a wide range of topics thanks to his guidance. His ideas always intrigued me and led me in new directions helping me expand my horizons. I would like to thank him for giving me the opportunity to work on this project. I'm extremely grateful to him for his constant support and guidance throughout and for enabling me to learn so much and execute the project so well.

I would like to thank my co-guide Dr. Harishankar Ramachandran for his ideas, constant guidance, motivation and support.

I would like to thank Sweta Kumari, PhD student at CNS lab, for all her ideas and for all the fruitful discussions that I had with her.

Lastly I would like to thank my parents who have been guiding and helping me at every step of my life. None of this would have been possible without their help.

ABSTRACT

KEYWORDS: Reconstruction, Attention, Deep Learning, Elman Jordan

Image reconstruction using attention windows is an important task with widespread potential applications ranging from autonomous driving, to memory modelling, to saliency detection and saliency map generation. Our brain scans the entire image piecewise by attending to only a small region of the entire big picture and part by part aggregates the entire information represented in the image, with fading memory of the information represented by the parts focused at very early on and best recollection of the most recently focused at regions. Exactly on these lines, in this paper, we propose a dual channel convolutional recurrent neural network architecture with both Elman and Jordan connections to solve the image reconstruction problem. The inputs across timesteps are the heatmaps signifying the location in the image where the current attention is focused at and a zoomed in version of the attention window and the output at each timestep is the aggregated image till the current moment with diminishing brightness of the regions encountered in the past, precisely how our brain memory works. We test the performance of our proposed architecture on various datasets such as the MNIST dataset and the Fashion MNIST dataset.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
MSE	Mean Squared Error
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
conv	convolutional
deconv	deconvolutional
concat	concatenate
convEJ	convolutional Elman Jordan
convFF	convolutional Flip Flop
convLSTM	convolutional Long Short Term Memory
GT	Ground Truth

CHAPTER 1

Introduction

Achieving Image Reconstruction from attention windows and their corresponding heatmaps to encode their spatial locations using neural networks is of significant importance. Image reconstruction finds widespread potential applications in various domains such as saliency detection and map generation, autonomous driving and importantly memory modelling. In this thesis, we propose a dual channel recurrent convolutional Elman Jordan neural network to address the Reconstruction Problem.

1.1 Problem Statement

The Reconstruction Problem is the following:

Design a neural network with the following characteristics:

Inputs: Cropped out portions of the image scanned piecewise with spatial information
Final Output: The entire Aggregated image over all the timesteps with:
Case 1: Diminishing brightness across the regions focused at in the past
Case 2: Constant brightness across all the regions

Let's look at this in more detail.

The inputs that will be provided would be as follows. We would follow a random trajectory and scan over the entire image part by part, piecewise, and crop out small sections from the image, which are the Attention Windows. Corresponding to the location of these Attention Windows, we would create heatmaps to encode the spatial location of these cropped out pieces. The heatmap would be a black image of the same size as the image from which the patches are cropped out, with a small white patch of the same size as Attention Window, in the exact location from which it was cropped out from the original image. Thus, the network would be given an input of a series of a pair of images, the Attention Window and its corresponding heatmap, and it would be required to aggregate the information contained in all those images and reconstruct the entire image, of which they are a part.

For this reconstruction, we consider further two cases. In the first case, the network should, in the final reconstruction, reconstruct the patches, that were shown to it in the past, that is which were input to it in earlier timesteps, with continually diminishing brightness, according to as the further back in time the Attention Window was encountered, the lesser should the brightness of its corresponding region in the final reconstructed image be. In the second case, we require complete reconstruction, that is reconstructing the entire image with constant brightness across all the regions of the image, irrespective of when their corresponding attention windows were input to the network.

Thus, in this thesis, we explore the performance of different network architectures on these reconstruction tasks, and propose a novel lightweight architecture to address the reconstruction problem.

1.2 Approach to Solve the Problem

Since the reconstruction problem requires us to generate the entire image from small attention windows and their heatmaps, we need to aggregate their entire information content, the image content and the spatial location content, over all the timesteps and use it to generate the complete image in the end. Hence, the problem statement inherently requires the neural network to have memory, and so we must use recurrent neural networks. Since we are dealing with images and convolution operations work best with images instead of dense fully connected operations, so we use the convolutional version of these recurrent neural networks. We start by looking at naive approaches using LSTMs and dense connections. Then we split the problem statement into sub-objectives. First, we design a static network to reconstruct partial images just for a single timestep, by providing only one pair of Attention Window and heatmap as inputs. Essentially, for each timestep we try to reconstruct the corresponding partial image separately, with just the Attention Window placed in the correct location as pointed to by its corresponding heatmap and the rest of the image blank. Next, to the static network we add memory in the form of convolutional Elman and Jordan connections and reconstruct the entire image using the series of Attention Window and heatmaps pairs as inputs, for both the cases, the case with the diminishing brightness and the case with the constant brightness across the image regions. Finally, we consider other recurrent convolutional neural networks such as convFFs and convLSTMs and we compare the performance of our convolutional Elman Jordan neural network to their performance.

CHAPTER 2

Background and Related Work

In this chapter, we will have a look at some theoretical background and some research papers related to the work presented in this thesis.

2.1 Theoretical Background

We use various concepts such as convolutional neural networks and recurrent neural networks in our work, the performance of which have been well established on various kinds of tasks. Let's have a look at these concepts in detail.

2.1.1 Convolutional Neural Networks

Convolutional Networks are widely used for image related tasks. The convolution operation helps to preserve the spatial information content in an image, while also extracting the important features present in it. In the convolution operation, the whole image is scanned by a moving filter with a specific stride, and the pixel values of the image, which are in the scope of the size of the filter are multiplied with the corresponding filter values, the products are summed and then the value is placed in the corresponding cell in the output. Thus, this process is repeated till all the pixels in the input are covered. The filter values form the parameters of the network which are to be learnt during training by backpropagation.Another operation which is of significance is deconvolution. Deconvolution operation is basically the inverse of the convolution operation and is used for objectives like increasing the height and width dimensions of the input tensor and so on.

Convolutional Layers have a variety of advantages over dense fully connected layers. They are less computationally expensive as they have fewer parameters than dense connections. Moreover, they inherently implement weight sharing, which is a regularization technique and hence they prevent overfitting. Thus, using convolutional operations, instead of regular dense operations is a more effective alternative for tasks which involve images.

2.1.2 Recurrent Neural Networks

Recurrent Neural Networks are widely used when dealing with time series data. Whenever we have series of inputs and we are supposed to extract features from them, accumulate those features over time and produce relevant outputs, we use recurrent neural networks. RNNs have a memory component, which enables them to store the features representations over time and combine it along with the additional input given to it at every new timestep and produce appropriate outputs, aggregating all the information that it has accumulated upto the current point.

To implement the memory aspect of RNNs, a large variety of techniques are used, varying from Elman and Jordan Connections, to Vanilla RNNs, GRUs and LSTMs, depending on the tasks and requirements at hand.

In this thesis, since we are working on time series data involving images, we need to combine the best aspects from both the types of neural networks, and hence, we work on convolutional recurrent neural networks.

2.2 Related Work

In this section, we will look at some previous related work in domains related to reconstruction, such as saliency map generation.

The first work of interest is by Zhao *et al.* (2015). In the paper, the authors propose a dual channel network architecture to detect image saliency. They follow the following pathway. First they segment the image into superpixels. This they achieve by clustering neighbouring pixels with similar color attributes. Once the image is divided into these large pixel chunks known as superpixels, they then set out upon the task of classifying each superpixel as being salient or not. That is, whether it is a part of the object in focus, which is the foreground, or the background. To do this, the use a dual channel deep neural network. The first channel is used to encode local contextual information. The input to this channel is a small cropped out portion of the image, such that the current superpixel cluster being classified is at the center of the image patch with the rest of the portion padded appropriately. The second channel is used to encode global contextual information, such as spatial information about the superpixel relative to the other image components and so on. For this channel, the input which is sent is the whole image, shifted, translated and padded appropriately such that the superpixel cluster under current consideration lies at the center. These two inputs pass through a number of layers and after that the information from the two channels is combined together. Post that, there are a few more convolutional layers and then finally in the end there is just a two neurons, which output the probability of the superpixel cluster being in the background or being salient. Thus, using the dual channel approach to encode local information and global information improves the results significantly.

The second work is by Li *et al.* (2017). In this paper, the authors try to understand and demarcate the most important or saliency wise relevant portions of the image, basically the regions of the image which contain the information most relevant to its classification into the right category. For this, first the authors start off with the prediction probability and then back propagate through the various layers until they reach the input image, where they identify those pixels of the image which contribute the most to the classification probability. Next, for each of these identified pixels, they consider the surrounding pixels within a certain radius, which they compute using a formula and further mark them as being salient or being a part of the background. This, process they repeat again at two different scales, at twice and four times the initial radius, to improve the saliency relevance map of the image. Thus, the key aspect of this paper is that various regions of the image have to be looked at at different scales to accurately determine whether they are of significance to the saliency relevance map of the image or not.

The third work of interest is by Jaderberg *et al.* (2015). In the paper, the authors introduce a spatial transformer network and demonstrate how by using a series of spatial transformations they can narrow down upon the most important or salient portion of the image and extract it out and use it for classification or other tasks. They perform the transformations by passing the image through a localization network channel, which outputs the parameters of a transformation matrix. This transformation matrix learns the appropriate parameters to perform a variety of transformations, ranging from affine transformations to projective transformations to thin plate spline transformations. Then, once the transformation matrix is obtained, both the matrix and the input image are sent to a sampler, which implements a differentiable function such as integer sampling or bilinear sampling and produces a transformed output which is then further used for tasks such as classification and so on. Thus, in this paper, the authors demonstrate the inherent use of saliency by extracting the most important parts of an image for performing various tasks without explicitly demarcating the most salient regions.

CHAPTER 3

Proposed Architectures for the Reconstruction Problem

In this chapter, we will look at a number of architectures to address the reconstruction problem. We first will look at a naive initial approach using basic LSTMs and then we will step by step build our model architecture by splitting the objective into sub-objectives, explore various architectures and their variants and solve the Reconstruction Problem.

3.1 Dataset Preparation

We conduct out our experiments on the following datasets.

- Synthetically Generated Dataset
- MNIST
- Fashion MNIST

Since, our reconstruction problem requires two inputs, the heat map to encode the spatial location being currently focused at and the attention window, to each input we assign a separate channel. For the first channel, which is the Attention Location Channel, for each timestep we create the heatmap. The heatmap is a binary image which is black all over with a white square patch corresponding to the location being focused at currently in the present timestep. Correspondingly, for the second channel, which is the Attention Window Channel, we crop that portion from the image, where the current attention focus is at, which is mapped by the white square patch of the heatmap being input in the first channel. An example of the inputs for both the channels is shown in Fig. 3.1. For a 28x28 image, the input sizes are 8x8 for the Attention Window and 28x28 for the heatmap.

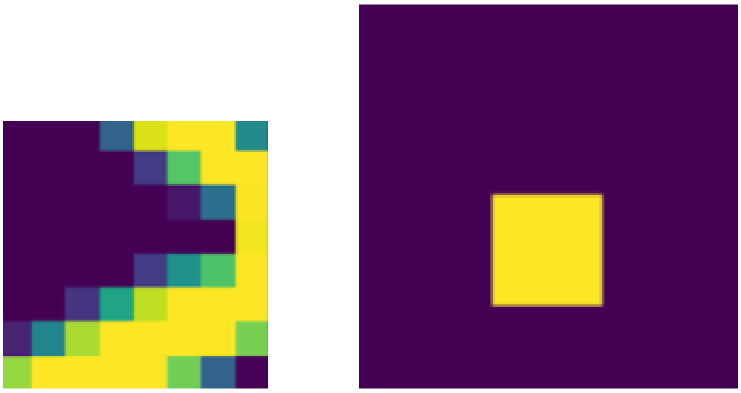

Figure 3.1: The inputs: The 8x8 Attention Window(left) and 28x28 heatmap(right)

We perform image reconstruction for the following two cases. In the first case, regions of the image which were encountered in the previous timesteps diminish in brightness over time by a constant factor. In the second case, we reconstruct the entire image, all the regions, with constant brightness, irrespective of the timestep at which the region was encountered. Accordingly, we construct the ground truths for each timestep, which are the aggregations of the regions of the image, which have been focused at upto the current timestep, diminishing in brightness for the first case and of constant brightness for the second. Examples of the expected outputs at the final timesteps have been shown for both the cases in Fig. 3.2.

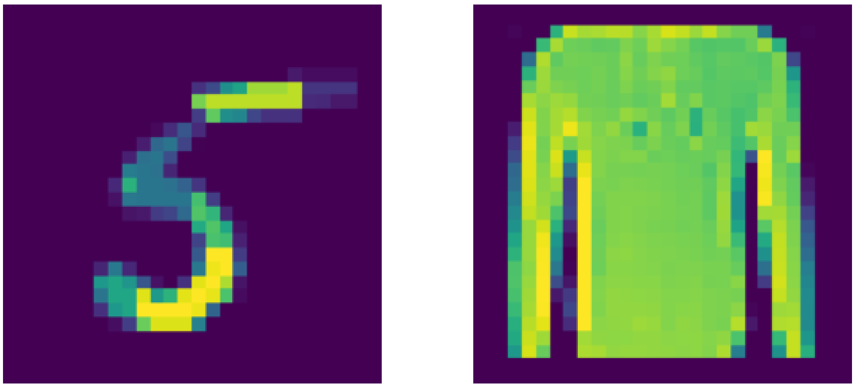

Figure 3.2: Expected outputs: Diminishing MNIST(left) and Fashion MNIST(right)

3.2 Initial Approach using LSTMs

First and foremost, whenever we have any data which is sequential in nature, LSTMs come to mind. LSTMs have memory, which enable the model to remember aspects and features of the previous timesteps and aggregate the entire information, along with the new inputs and information of the current timestep to generate the output of the present timestep. Given this, we carry out our first experiments on an LSTM based architecture to gauge their performance.

3.2.1 Architecture Details

The model architecture for this approach is shown in Fig. 3.3. First we apply convolutional operations to both the inputs to both the channels separately. Then we flatten them out and apply two fully connected layers to get two separate vectors for both the channels. Then, we concatenate both the channels and send the whole resultant vector to three contiguously stacked LSTM layers.

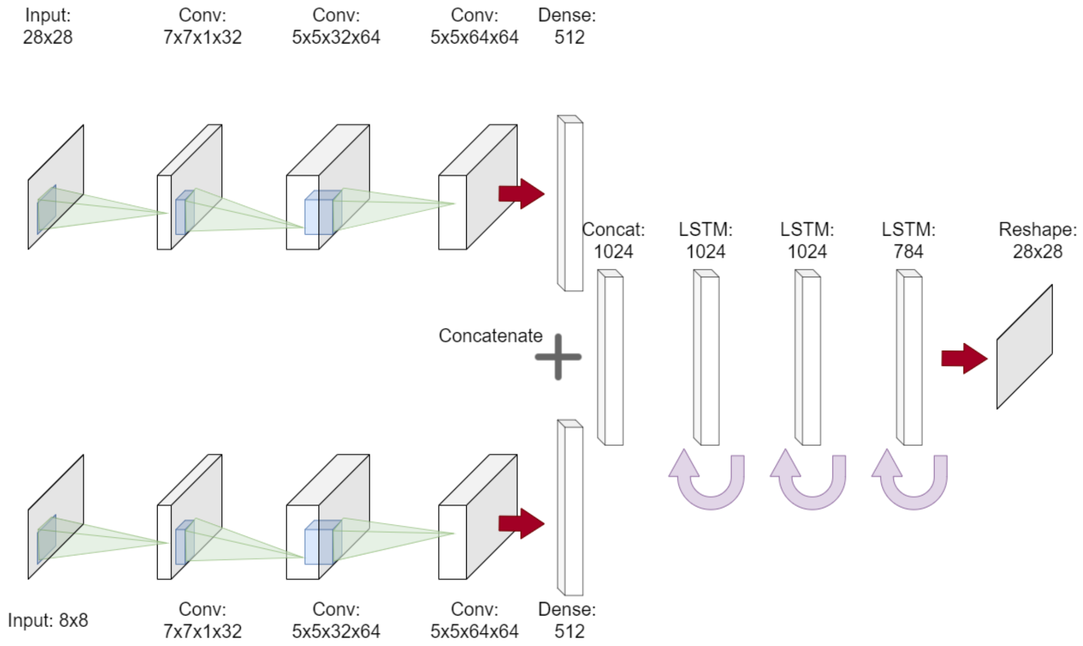

Figure 3.3: Architecture of the LSTM based network

3.2.2 Training Details and Results

We use the Mean Squared Error loss to train the network. The loss is computed by comparing the prediction and the expected image, which is the aggregation of the image only upto the current timestep, at all the timesteps and is back propagated through the network. We achieve a training MSE loss of 0.083 after training for 200 epochs.

The results are shown in Fig. 3.4. As we can see, the architecture has not performed well, as was reflected in the poor training loss also. The outputs are very blurred and no significant learning has occurred apart from the approximate location of the signs. This is primarily because the LSTM connections we used are dense connections composed of fully connected layers. However, to preserve the spatial information in images, fully connected layers are not the best approach. We will address these shortcomings and overcome the limitations of the architecture, step by step in the coming sections.

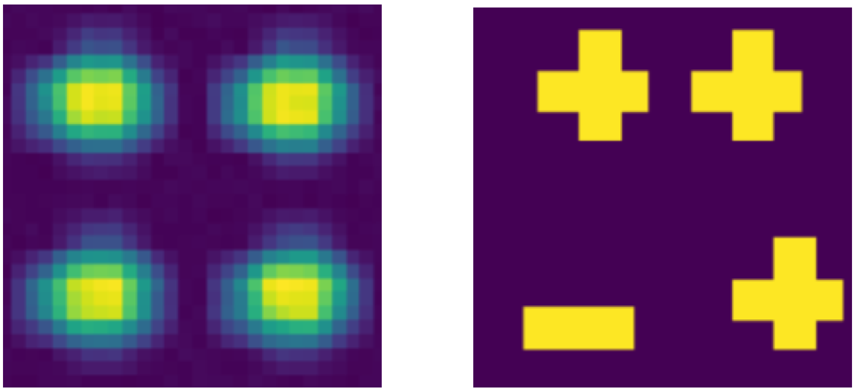

Figure 3.4: Generated Image(left) vs Ground Truth(right)

3.3 Sub Objective 1: Static Network

We solve the Reconstruction Problem in two steps. In the first step, we try to reconstruct a portion of the image at each timestep independently of each other. Essentially, we try to reconstruct every snapshot of the image as we scan through it, given just the attention window and the heatmap of the current timestep as inputs.

3.3.1 Architecture Details

The model architecture for this approach is shown in Fig. 3.5. Here, since the outputs of the network depend only upon the inputs at the current timestep there is no memory component needed and hence we devise a static network, to accomplish the task of piecewise partial reconstruction of the image. We reconstruct the image part by part separately for each timestep, by just providing the network the attention window and heatmap corresponding to the present region of being focused at, as inputs.

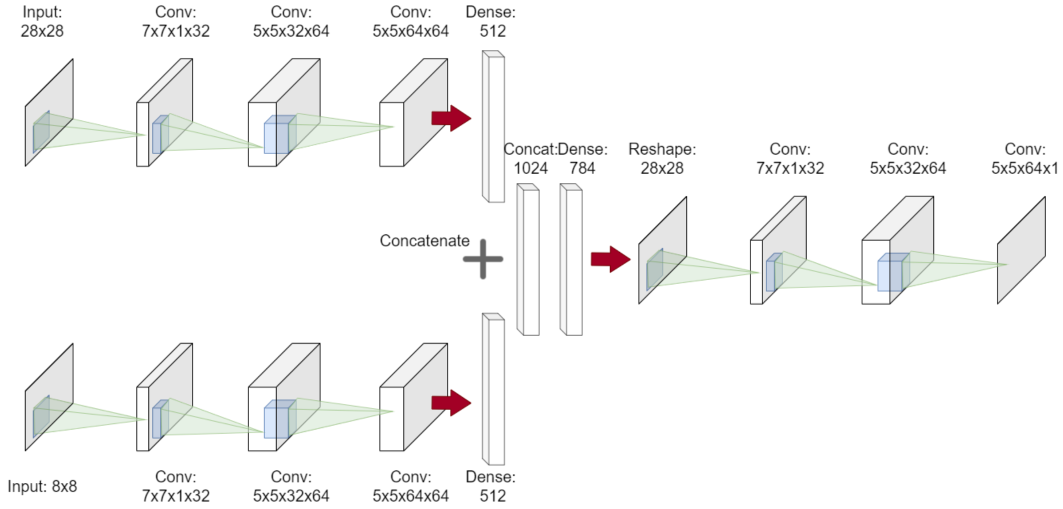

Figure 3.5: Architecture of the Static network

The Attention Location Channel and the Attention Window Channel have similar architectures.

The input to the Attention Location Channel is an image with same dimensions as the original image(28x28 for MNIST dataset images). The entire image is black except for a white patch. This patch denotes the exact area in the image where the attention is being focused at in the current timestep and is essentially being used as a heatmap to direct the network towards the current attention location.

Correspondingly, the input to the Attention Window Channel is the cropped out portion of the entire image where the attention is focused at in the current timestep, corresponding to the location of the white patch in the heatmap.

The inputs to both the channels are further fed in the the next layers of each channel respectively consisting of three convolutional layers. Additionally, after each convolutional layer, local response normalization is applied. Then both channels consist of fully connected layers. Those fully connected layers are then concatenated and now the combined information is sent through the combined channel, another series of three convolutional layers. The final output layer produces an output which is the reconstruction of the partial image, with the attention window placed in the correct location as was in the original image as pointed to by the heatmap.

3.3.2 Training Details and Results

For training this network, we use the Mean Squared Error loss. The prediction of the network is just the portion of the image reconstructed at the correct location, the spatial information of which is encoded by the heatmap, corresponding to the region in focus currently, using the provided inputs. We compare it with the constructed ground truth and obtain a training Mean Squared Error of 0.00012 and validation MSE loss of 0.00018 after training for 150 epochs.

The results for this sub objective are shown in Fig. 3.6. As we can observe, the results obtained are very sharp images and the network is able to place the attention window in the correct location as present in the original image, as directed by the heatmap.

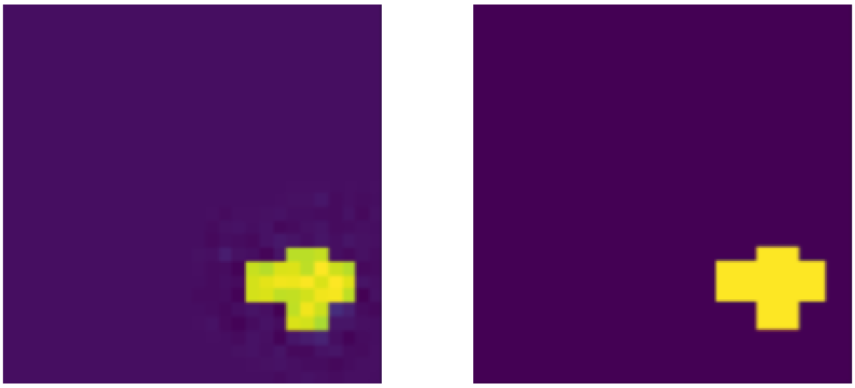

Figure 3.6: Generated Image(left) vs Ground Truth(right)

3.4 Sub Objective 2: Elman Jordan Aggregator

Now that we have been able to place the attention window in the correct corresponding location in the image separately for each timestep, we now need to design a network which can take a series of attention windows and heatmaps across various timesteps as inputs and generate the entire image for both cases, the first case in which brightness of the regions encountered in the past diminishes over time by a constant factor and the second case in which the brightness of the image is constant across all the regions. To accomplish this objective the network must possess memory of the features encountered in the past.

3.4.1 Architecture Details

The model architecture for this approach is shown in Fig. 3.7. We address the problem of adding memory to the previous static architecture by adding Convolutional Elman and Jordan Recurrent Connections to the network.

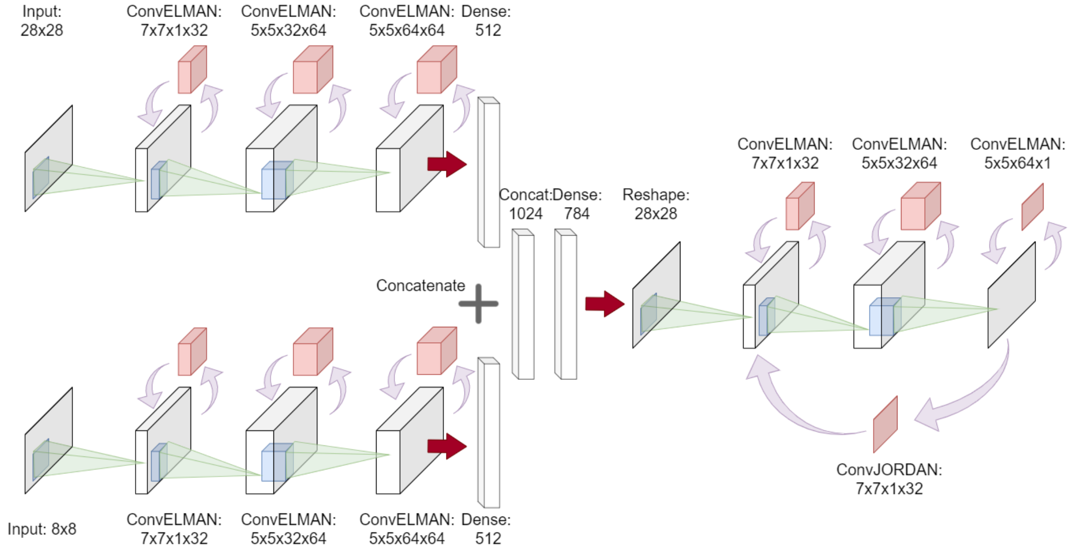

Figure 3.7: Architecture of the convElmanJordan Network

The Attention Location Channel and the Attention Window Channel have similar architectures as they had in the previous static network architecture.

We choose the attention location and hence the location of the white patch randomly in all of our experiments. Over the entire range of timesteps, we ensure that the entire image is being scanned so that it can be reconstructed appropriately.

The inputs to both the channels are further fed in the the next layers of each channel respectively consisting of three convolutional layers with convolutional Elman self connections at each layer. Additionally, after each convolutional layer, local response normalization is applied. Then both channels consist of fully connected layers. Those fully connected layers are then concatenated and now the combined information is sent through the combined channel, another series of three convolutional layers, each of which also have convolutional Elman self connections. Moreover, there is also a Jordan loop present from the final layer to the first layer of the combined channel. The final output layer produces an output which is the reconstruction of the image, with all the information seen upto the current timestep.

The equations for the Elman self connections are as follows:

$$h_t = \sigma_h(W_h * x_t + U_h * h_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y * h_t + b_y)$$

The equations for the Jordan connections are:

$$h_t = \sigma_h(W_h * x_t + U_h * y_{t-1} + b_h)$$

$$y_t = \sigma_y(W_y * g_t + b_y)$$

Here,

- x_t : input vector
- h_t :hidden convolutional layer vector
- g_t :hidden layer before the output layer
- y_t : output vector
- W_l, U_l : convolutional filters of the respective layers
- b_l : biases of the respective layers
- $*$: denotes the convolution operation
- t : denotes the current timestep

3.4.2 Training Details, Experiments and Results

For training this network, we use the Mean Squared Error loss. The prediction of the network is the entire aggregated image reconstructed by placing all the attention windows across all the timesteps at their correct respective locations, as directed by their respective heatmaps, with diminishing brightness across regions focused at in the past for the first case and constant brightness in the second. We compare it with the constructed ground truth and obtain the following validation MSE losses, for different datasets, after training for 100 epochs, which takes about 4.5 hours to complete. They are shown in Table. 3.1.

Table 3.1: MSE losses of convEJ network for various datasets

	MNIST	Fashion MNIST
Diminishing	0.0021	0.0032
Not Diminishing	0.0049	0.0084

The results for this sub objective are shown in Fig. 3.8, Fig. 3.9, Fig. 3.10 and Fig. 3.11 for different datasets for both the cases, diminishing and non diminishing brightness. As we can observe, the network is able to aggregate all the information fed to it in the form of heat maps and attention windows across all the timesteps and reconstruct the entire image almost perfectly, for both cases with diminishing brightness and with constant brightness for a large variety of datasets.

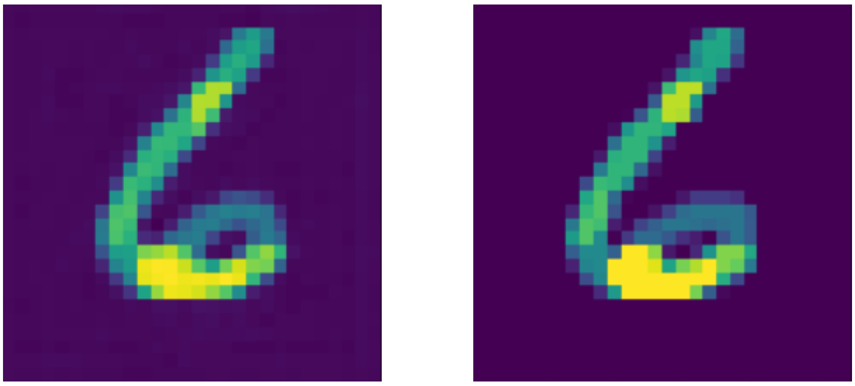

Figure 3.8: Diminishing MNIST: Reconstruction(left) vs Ground Truth(right)

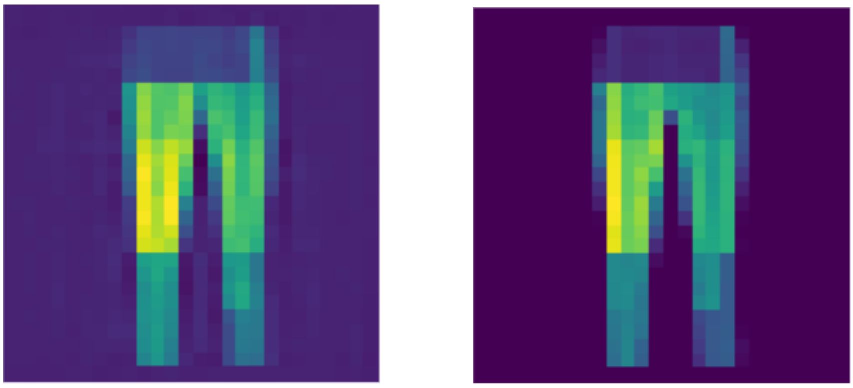

Figure 3.9: Diminishing Fashion MNIST: Reconstruction(left) vs Ground Truth(right)

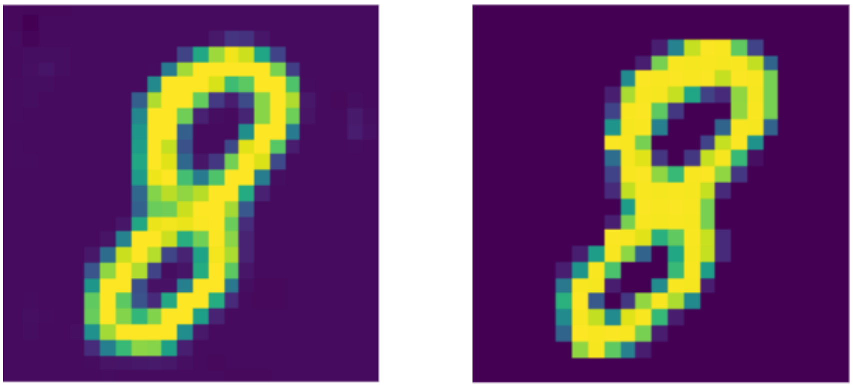

Figure 3.10: Non diminishing MNIST: Reconstruction(left) vs Ground Truth(right)

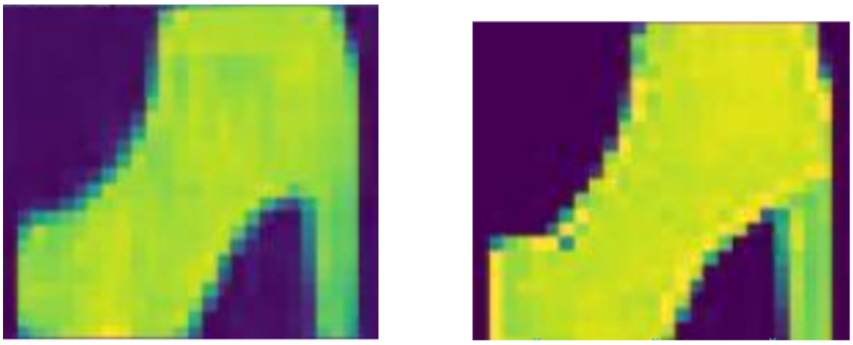

Figure 3.11: Fashion MNIST: Reconstruction(left) vs Ground Truth(right)

3.5 convFF and convLSTM

We now try adding memory to the static network using other kinds of recurrent architectures. Also, another aspect which we try to address is removing the fully connected dense layers and accordingly we modify the architecture of the two channels of the network to solve the reconstruction problem without them.

3.5.1 Architecture Details

The architecture details are shown in Fig. 3.12. We build upon the architecture of the static network. First off, we remove the fully connected layers and solely rely on convolution operations in both the channels, the Attention Location Channel and the Attention Window Channel. Post the convolution operations, we just add the tensors directly to combine the information from the features extracted from the inputs to both the channels. To enable this, we make the following change from the static architecture. Since the attention window is relatively smaller as compared to the heatmap, 8x8 compared to 28x28 in the case of MNIST and Fashion MNIST datasets, to the Attention Window Channel we need to add deconvolutional layers to bring it up to the same dimensions as the Attention Location Channel output tensor. As shown in Fig. 3.12 we add two deconvolutional layers to the Attention Window Channel. Post the addition of the tensors from both the channels, we pass them through convFF and convLSTM layers in the combined channel in two separate experiments, which serve as the memory component of the network.

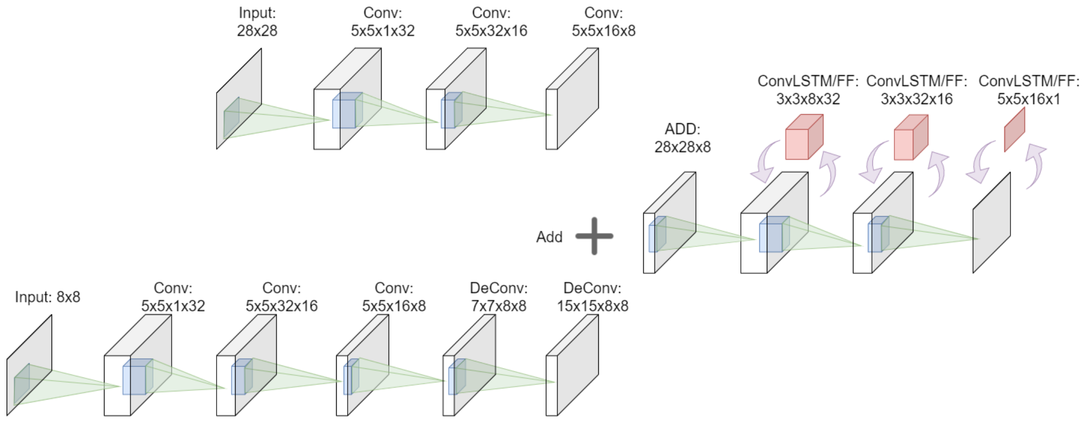

Figure 3.12: Architecture of the convFF/convLSTM Network

3.5.2 Training Details and Results

As in the previous Elman Jordan Network, we train both the networks, the convFF based network and the convLSTM based network. We train each network for 30 to 40 epochs, which takes nearly 16-22 hours to complete. For the convFF based network, as we can see in Fig. 3.13 and Fig. 3.14, the results are not good. The network is not able to reconstruct the image. For the convLSTM based network on the other hand, we are able to get good results, as we can see in Fig. 3.15, Fig. 3.16, Fig. 3.17 and Fig. 3.18. We obtain the following validation MSE losses, for different datasets, for both the cases diminishing and constant brightness. They are shown in Table. 3.2.

Table 3.2: MSE losses of convLSTM network for various datasets

	MNIST(30 epochs)	Fashion MNIST(40 epochs)
Diminishing	0.0043	0.0148
Not Diminishing	0.0065	0.0172

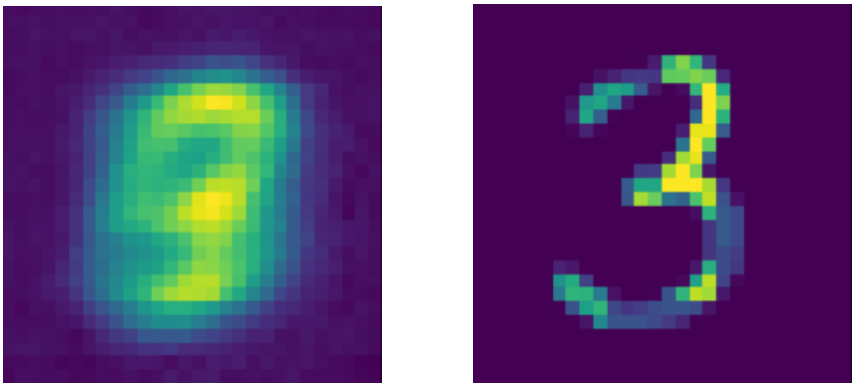

Figure 3.13: convFF Diminishing MNIST: Reconstruction(l) vs Ground Truth(r)

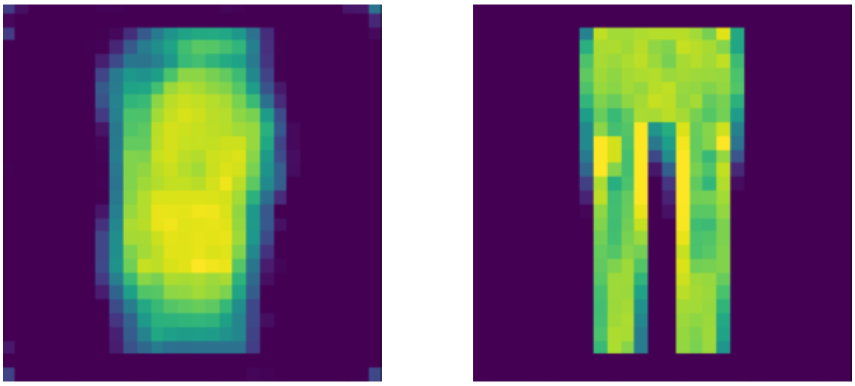

Figure 3.14: convFF Fashion MNIST: Reconstruction(l) vs Ground Truth(r)

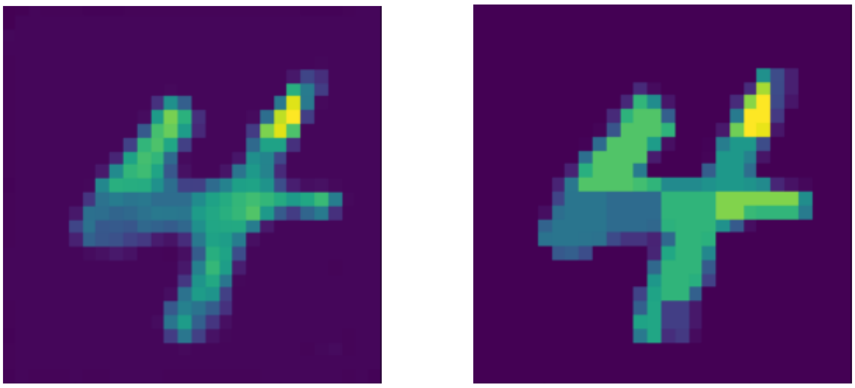

Figure 3.15: convLSTM Diminishing MNIST: Reconstruction(l) vs Ground Truth(r)

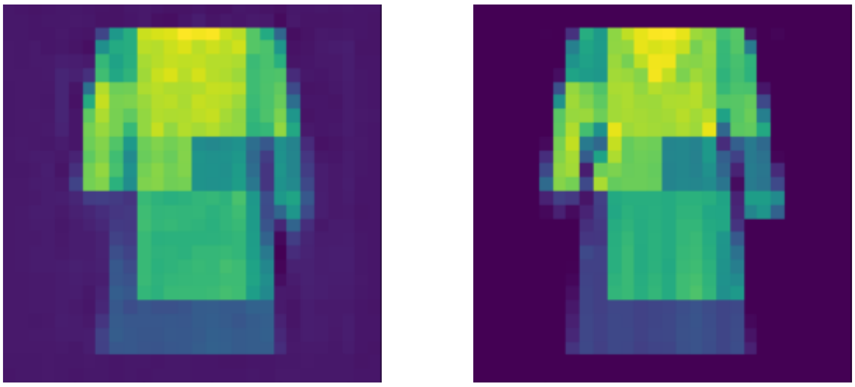

Figure 3.16: convLSTM Diminishing Fashion MNIST: Reconstruction(l) vs GT(r)

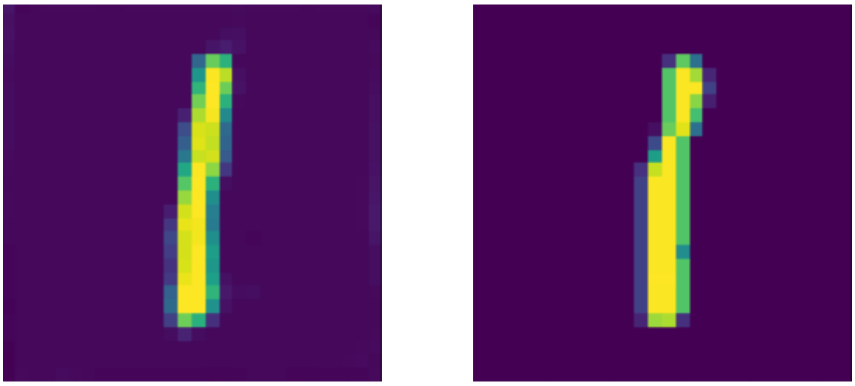

Figure 3.17: convLSTM MNIST: Reconstruction(l) vs Ground Truth(r)

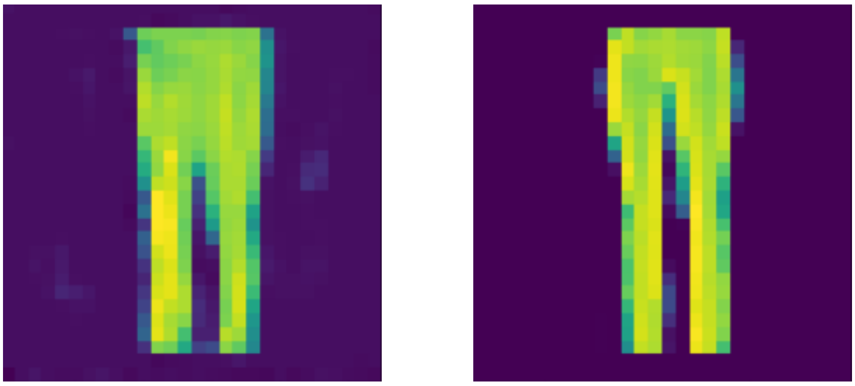

Figure 3.18: convLSTM Fashion MNIST: Reconstruction(l) vs Ground Truth(r)

3.5.3 Comparison with convElmanJordan Network

Hence, we see that, convLSTM although it is able to reconstruct the image, it is still not able to achieve as good MSE losses as the Recurrent Elman Jordan Network that we proposed in the previous section, even after taking so long to train as compared to it.The convLSTM network takes 16 hours for 30 epochs to train for the Diminishing MNIST dataset and 22 hours for 40 epochs to train for the Fashion MNIST dataset as compared to 4.5 hours for 100 epochs taken to train by the convELmanJordan network we proposed. There is a huge difference between the two. A comparison of the validation MSE losses for both the networks, convLSTM and convElmanJordan is plotted in Fig. 3.19 for the Diminishing MNIST case and in Fig. 3.20 for the Fashion MNIST case. Thus, we observe that, using our Recurrent Elman Jordan Network Architecture, we are able to achieve good image reconstruction results with a network architecture with lightweight recurrent connections by extending the Elman Jordan equations to a convolutional form and utilizing a dual channel architecture.

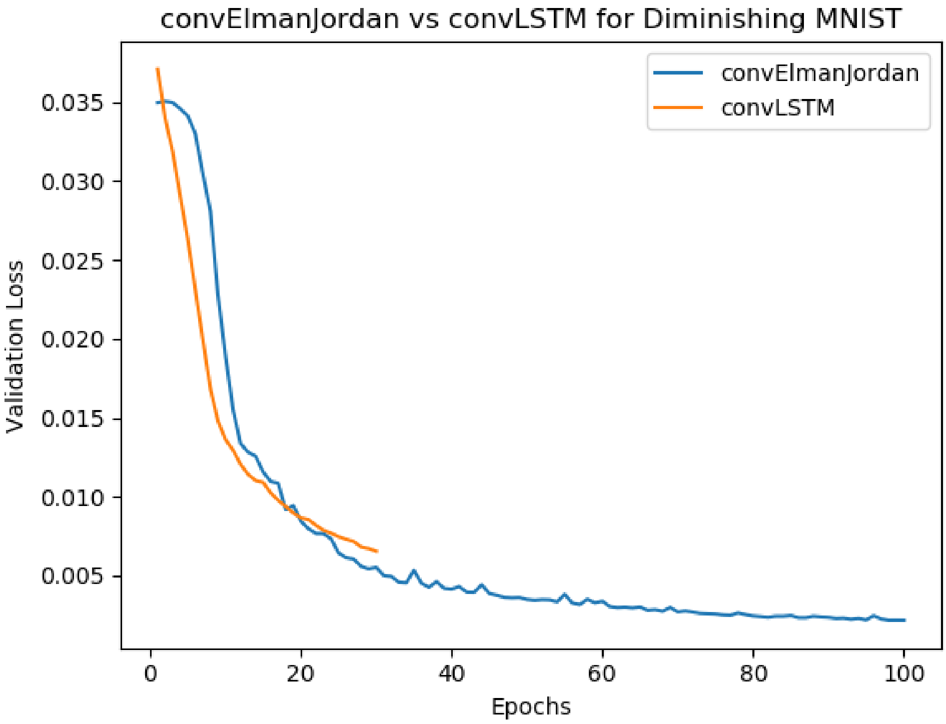

Figure 3.19: Validation Losses: convLSTM vs convEJ for Diminishing MNIST

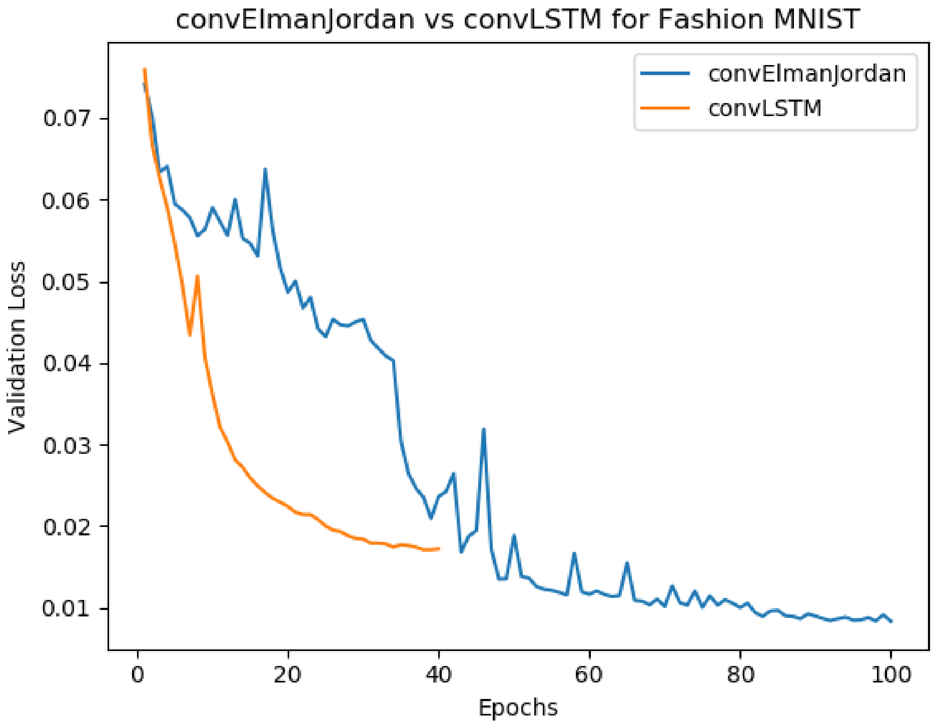

Figure 3.20: Validation Losses: convLSTM vs convEJ for Fashion MNIST

3.5.4 Conclusion

Thus, through various experiments, we see that, the recurrent convolutional ElmanJordan architecture that we propose performs really well as compared to other architectures that we experimented upon in the thesis. It is faster, lighter and reconstructs the images much more accurately.

REFERENCES

1. **Jaderberg, M.**, **K. Simonyan**, **A. Zisserman**, and **K. Kavukcuoglu** (2015). Spatial transformer networks. *CoRR*, **abs/1506.02025**. URL `http://arxiv.org/abs/1506.02025`.

2. **Li, H.**, **K. Mueller**, and **X. Chen** (2017). Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. *CoRR*, **abs/1712.08268**. URL `http://arxiv.org/abs/1712.08268`.

3. **Zhao, R.**, **W. Ouyang**, **H. Li**, and **X. Wang**, Saliency detection by multi-context deep learning. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

LIST OF PAPERS BASED ON THESIS

1. Dhruv Chopra, Sweta Kumari, V Srinivasa Chakravarthy
   Modelling Working Memory using Deep Convolutional Elman and Jordan Neural Networks
   Conference, CNS2021, Organization for Computational Neurosciences, (2021).