

Progress Report for MOSES Data Inversion with CNNs

ROY SMART

Montana State University
roytsmart@gmail.com

I. INTRODUCTION

Simplified MOSES data inversion was attempted using a convolutional neural networks. Data was prepared randomly and then fed through the MOSES forward model. Results from the forward model were used as input to a convolutional neural network and then compared to the input of the MOSES forward model to train the network. This test was very successful, achieving very high accuracy. While this is a very simple example of MOSES data inversion, it is an important step to learning how to use convolutional neural networks.

II. DATA PREPARATION

The MOSES instrument observes the sun in spatial directions x and y . This measurement is the projection of a spectral cube in x , y , and λ . In the MOSES coordinate system x is the dispersion direction. Therefore, (if we neglect the point-spread function,) we may ignore the y component of the images without loss of generality.

For this test we randomly construct a slice of the spectral cube in x and λ , with dimensions 4 by 4. We will call this image the "truth" image, and the output of the neural network should strive to create this image.



Figure 1: *An example of a 4 pixel by 4 pixel truth image.*

From the truth image, we simulate the measurements captured by the instrument using the so-called MOSES forward model (fomod). Fomod works by summing along each spectral direction for the $m = 0, 1$, and -1 spectral orders. Since the y dimension has been ignored, we are left with a graph, described in Figure 2

Unfortunately, convolutional neural networks do not work as well on graphs, they are designed to work with images. We can transform the graphs from the three orders back into an image by backprojecting along each spectral direction. The result is essentially an image that contains the maximum possible pixel value at every location in the spectral slice. For our purposes we will store the projections from each order on separate color channels (RGB). We will call this image the

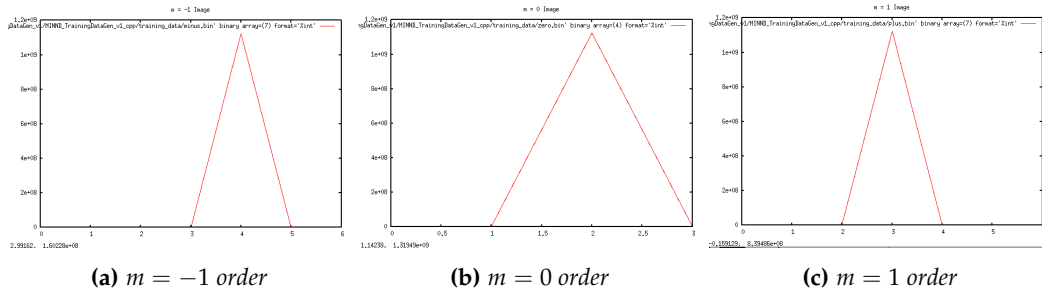


Figure 2: Simulated results of the MOSES instrument for the simple case described in Figure 1.

"input" image, and it will be the image that is fed into the neural network. An example of this image is given in Figure 3. This process gives the neural network important information about the operation of the MOSES forward model and may help the network train faster.

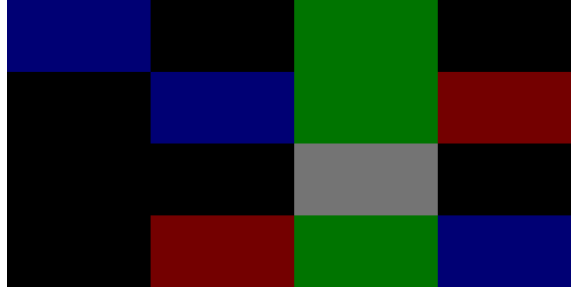


Figure 3: An example of a 4 pixel by 4 pixel input image.

We created 10,000 4 by 4 pixel, 8 bits per pixel truth-input image pairs and used them to train the network.

III. NEURAL NETWORK ARCHITECTURE

The neural network architecture was slightly arbitrary, in that we haven't had too much time to explore the hyperparameter space. The architecture we developed consisted of one convolutional layer, one max pooling layer, followed by three fully connected layers, with ReLU layers in between. This is outlined in Figure 4.

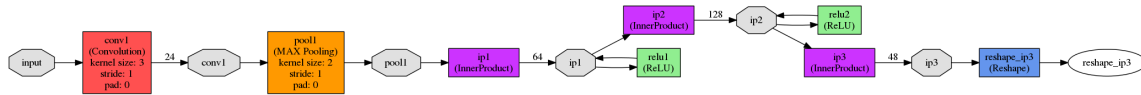


Figure 4: Flow chart of the layout of the neural network used for this test.

The performance of each layer is not yet understood very well, and will become more clear as the image size and complexity increases. However, we will explain the intended effect of each layer here. The convolutional layer should consist of a series of orthogonal filters described in Figure 5.

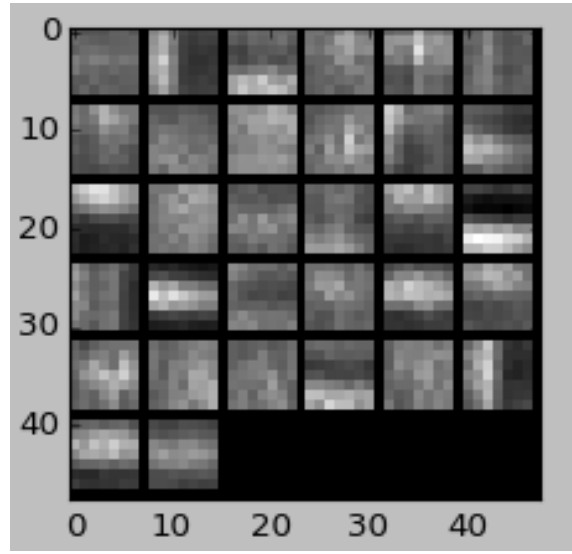


Figure 5: Example of orthogonal filters in convolutional layer. NOTE: This is not from our network.

The max pooling layer helps determine which convolutional filter is most prevalent. In larger/more complex images, there will be more convolutional/max pooling layers to handle the information increase.

After the max pooling layers, the information is fed into a more classical neural network, with each neuron fully connected to one another. (In Caffe terms, this fully-connected layer is called an inner product layer.) This layer takes the information from the convolutional filter output and reconstructs the truth image.

IV. RESULTS

I. 4 x 4 Pixels, 1 active pixel

The network was trained for 10,000 iterations. Each truth image was 4 by 4 pixels and had one active pixel between the values of 0 and 255 randomly placed on the image. The loss computed by Caffe after 10,000 iterations was $5.38543e-05$. Since $4 \times 4 \times 255 = 4080$, it is likely that the neural network simply memorized the training dataset.

Iteration	Input	Output	Truth	Difference
1				

Table 1: Characteristic output for the 4x4 case. MINND performs perfectly each time.

II. 8 x 8 Pixels, Up to 5 active pixels

For this size of image, we used the network configuration described in Figure 6.

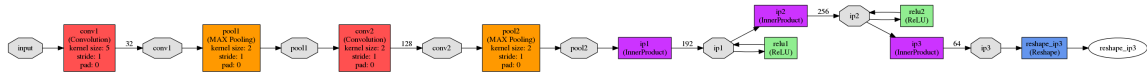


Figure 6: Neural Network configuration for 8x8 pixel input/output image

The network was trained for 1 million iterations. Each truth image was 8 by 8 pixels and had between 1 and 5 active pixels between the values of 0 and 255 randomly placed on the image. The loss computed by Caffe after 1,000,000 iterations was 0.127717.

This is much worse than the 4x4 case reported above. This is to be expected as there are many possible solutions to any non-trivial input image. However, the neural network doesn't perform perfectly, as the neural network output isn't necessarily a solution to an input cube.

Iteration	Input	Output	Truth	Difference
1				
2				
3				
4				

Table 2: Output of the MINND neural network for four randomly selected input images. Notice how not all output images are solutions to the inversion problem.

V. CONCLUSION

We have proved that a neural network can give a reasonable answer to the MOSES inversion problem for simple datasets. The network performed really well in the 1 active pixel getting the right answer every time. However the performance was disappointing in the 5 active pixel case, not only getting the answer wrong, but the solution presented by the network was not a potential solution to the inversion problem.

The performance of the network could potentially be improved in many ways. Selection of hyperparameters such as convolutional kernel size, learning rate, and number of hidden neurons may all help to increase the accuracy of the network. Also, the size of the training dataset was fairly small, consisting of only 1 million images in the 8x8 case. It is possible that larger training datasets will allow for better convergence.

Furthermore while the data presented to this neural network was completely random, the next dataset (the IRIS dataset) will have a structure with solar characteristics. This will provide details for the neural network to recognize, trimming potential solutions to the inversion problem.