Fall 11-18-2020

Making Artificial CIPS Data With a Generative Adversarial Neural Network

Austin Hedges
Longwood University

Making Artificial CIPS Data With a Generative Adversarial Neural Network

Austin Hedges[1], Joe Mustachio[2], Dr. John McNabb[3]

[1]Longwood University, [2]Virginia Commonwealth University, [3]Hampton University Department of Atmospheric and Planetary Sciences

Abstract

Polar mesospheric clouds (PMCs) have been studied for thirteen years by NASA's Aeronomy of Ice in the Mesosphere (AIM) satellite. The Cloud Imaging and Particle Size (CIPS) instrument onboard AIM has taken many images of PMCs over this time. Such a large number of images makes CIPS data ideal for training neural networks which require large datasets. CIPS images were used to train a Generative Adversarial Network (GAN) to train towards being able to generate purely artificial CIPS-like images.

Introduction

Polar Mesospheric Clouds

- Highest clouds on Earth at 82km[1] above the poles, near the mesopause
- PMCs are thin ice clouds, theorized that ice nucleates on meteoric smoke[2]
- Methane oxidizes in stratosphere bringing water to the mesopause and so PMCs form, climate change leads to more methane and so more PMCs[1]

(a) (b)

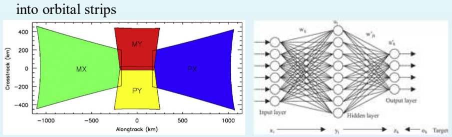
Fig. 1: (a) Image of PMCs taken from the ground[3], (b) artist's rendition of the AIM satellite[4]

Aeronomy of Ice in the Mesosphere

- NASA satellite that has studied PMCs since its launch in 2007
- Polar sun-synchronous orbit at 600km to study PMCs over both poles[5]
- Three instruments onboard SOFIE, CDE, and CIPS[6]

CIPS

- Four nadir facing cameras, angled outwards for a wider view of PMCs[7]
- Cameras image in ultraviolet with 15nm passband centered at 265nm[8]
- The cameras measure albedo differences to image PMC structures
- The four cameras projected onto Earth's surface form a "bowtie" shape
- As AIM takes images throughout orbit, the "bowtie" scenes are combined into orbital strips



(a) (b)

Fig. 2: (a) CIPS scene projected on Earth[7], (b) diagram of generic neural network with one hidden layer[9]

Neural Networks

- Computer program that mimics how brains learn information, great for recognizing patterns, classifying patterns, and making predictions[10]
- Neurons are connected to other neurons and are structured in layers
- Each neuron has a bias and the connections have weights, these values are real numbers and they affect how the network learns patterns[10]
- A network is fed data along with corresponding labels, the weights and biases are initially random and so the outputs are likely incorrect
- After outputting a result for an input, the network is given the correct label, the network then adjusts weights and biases to be more correct
- To do this, the error in the output is calculated using a loss function
- The network then backpropagates, going back and tweaking weights and biases in order to get more correct outputs
- Learns from success and failure, connections leading to correct outputs are strengthened and connections leading to incorrect outputs weakened

Methodology

Generative Adversarial Networks

- Generative Adversarial Networks (GANs) were developed in 2014 by Dr. Ian Goodfellow and colleagues at the University of Montreal[11]
- A type of neural network framework consisting of two neural networks
- The generator takes an input of random noise and manipulates it into a fake image
- The discriminator takes in the generators images as well as real images and determines if a given image is real or not[11]
- Once the discriminator can identify generated images at high confidence, its training is stopped
- The generator is then trained off of the discriminator's model
- At the beginning of this training, the generator will not be able to trick the discriminator much, but after many iterations it will be able to trick it again
- This process is repeated multiple times, the discriminator becomes very good at identifying generated images, and the generator becomes proficient in generating convincing images
- The end goal of a GAN is to have a convincing generator
- In the past, discriminators have been much more efficient than generators, the GAN's generator benefits from the discriminator's efficiency

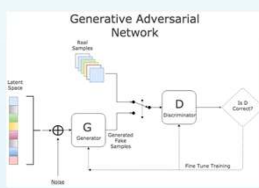
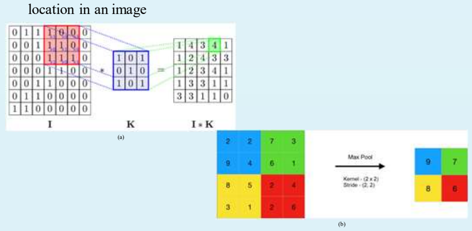


Fig. 3: Diagram of GAN architecture with Generator and Discriminator[12]

- Our GAN was written in Python using a machine learning library called PyTorch, PyTorch was developed in 2016 by Facebook
- PyTorch allows for the structures of networks to be easily written in Python, this includes creating the hidden layers of a neural network
- For the discriminator's input of real CIPS images, images from AIM's PY camera were used
- These images were squashed into 1D tensors to be inputted into the neural network's input layer
- The discriminator and generator were built with multiple layers of different types
- The discriminator used convolutional layers, where a square window of defined size scans over an image's pixels
- Pooling layers were also used in the discriminator, where an image is downsampled to a lower resolution by combining groups of pixels
- Pooling allows the neural network to identify structures regardless of their location in an image



(a) (b)

Fig. 4: (a) Visual representation of a convolutional layer[13], and (b) a visual representation of a Max Pooling layer[14]

- The discriminator also used linear layers, the simplest type of layer where a single layer of neurons is connected to the layers before after
- The hidden layer in Fig. 2b is an example of a linear layer

- Each layer of the neural network also has an activation function
- The activation of a neuron is calculated by passing the weighted sum of the previous layer's activations plus the bias into the activation function
- Activation functions mimic how the brain's neurons will activate differently with different stimuli
- With pooling and convolutional layers, the ReLU activation function is commonly used

$$ReLU(x) = \max(0, x)$$

- The ReLU activation function returns zero for all negative inputs and returns the value back if it is positive
- ReLU is a very simple calculation and so it reduces the time required to train a network
- ReLU causes certain neural connections to not have any activation making the network model more concise and reducing noise

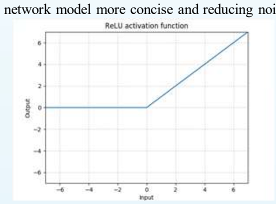


Fig. 5: Graph of the ReLU activation function[15]

- To identify fake images, the discriminator must modify its weights and biases, this is done by first calculating how wrong an output
- A loss function is used to calculate how wrong a given output is, the discriminator used the Cross-Entropy Loss function, its formula[16] is

$$loss_{CE} = -\sum_{i=1}^{C'=2} t_i \log(s_1) = -t_1 \log(s_1) - (1 - t_1)\log(1 - s_1)$$

- With C' being the number of classes, t_i being the correct label (0 for a generated image and 1 for a real image), and s_i being the value for the given output neuron (between 0 and 1)
- The network then propagates back through the layers
- Backpropagation calculates what connections need to be modified to lead to more correct outputs
- The network strengthens connections that lead to correct outputs and weakens connections that lead to incorrect outputs
- Through back propagation, the network minimizes the loss function via stochastic gradient descent

- In gradient descent, the gradient of the entire loss function across all training iterations is computed in order to minimize the loss function
- The larger the gradient is, the larger "step" is taken toward the minimum
- In stochastic gradient descent, random samples from training are chosen in order to reduce the computational expense

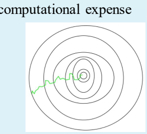


Fig. 6: Visual representation of stochastic gradient descent with the green line being its path toward the minimum[17]

Results

- Six iterations of swapping between the generator and the discriminator were performed
- An image from each of these iterations can be seen in Fig. 7 alongside a real CIPS image

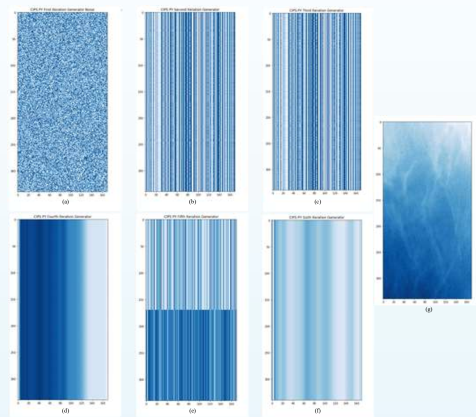


Fig. 7: (a) Image of the GAN's first iteration with the random noise that the generator begins with, (b) second iteration, (c) third iteration, (d) fourth iteration, (e) fifth iteration, (f) sixth iteration, and (g) a real CIPS PY image

Conclusion

- Between these six iterations of the GAN, it can be seen that the generator makes improvements in its image generation
- The GAN is functioning as intended, the discriminator trains to identify fake images to a high level of confidence and then the networks swap
- The changes between the iterations show the improvement that each network is having and the affect this has on the generator
- Clearly the images from these six iterations look nothing like the real image
- The generator is easily fooling the discriminator, especially by creating a structure that repeats vertically
- The generator is more powerful than the
- In order for convincing images to be generated a few things are needed
- First, many more iterations of the GAN are needed
- Second, in order to avoid stagnation in image generation, changes in the generator and discriminator layer structures would also be needed
- This includes using different types of layers and adding more layers

I'd like to thank Joe Mustachio, Dr. John McNabb, and Elliot Shiben for their contributions, as well as NASA, and the AIM mission team.

References

[1]Hervig, M., Thompson, R. E., McHugh, M., Gordley, L. L., Russell III, J. M., and Summers, M. E. (2001). First confirmation that water ice is the primary component of polar mesospheric clouds. *Geophysical Research Letters*, 28(6):971–974.

[2]Hervig, M. E., Deaver, L. E., Bardeen, C. G., Russell III, J. M., Bailey, S. M., and Gordley, L. L. (2012). The content and composition of meteoric smoke in mesospheric ice particles from sofie observations. *Journal of Atmospheric and Solar-Terrestrial Physics*, 84:1–6.

[3]Boardman J. (2006). [Noctilucent Clouds taken from 36,000 feet above the South of Nunivak Island, Alaska] [Photograph]. NASA. https://www.nasa.gov/mission_pages/aim/multimedia/noctiulant_clouds.html.

[4]NASA (2020). [Spacecraft icon of AIM satellite] [Photograph]. NASA. https://science.nasa.gov/toolkits/spacecraft-icons.

[5]Russell III, J. M., Bailey, S. M., Gordley, L. L., Rusch, D. W., Horányi, M., Hervig, M. E., Thomas, G. E., Randall, C. E., Siskind, D. E., Stevens, M. H., et al. (2009). The aeronomy of ice in the mesosphere (aim) mission: Overview and early science results. *Journal of Atmospheric and Solar-Terrestrial Physics*, 71(3-4):289–299.

[6]Hampton University (n.d.). Aim instruments sofie, cde, and cips. http://aim.hamptonu.edu/index.html.

[7]Rusch, D., Thomas, G., McClintock, W., Merkel, A., Bailey, S., Russell III, J., Randall, C., Jeppesen, C., and Callan, M. (2009). The cloud imaging and particle size experiment on the aeronomy of ice in the mesosphere mission: Cloud morphology for the northern 2007 season. *Journal of Atmospheric and Solar-Terrestrial Physics*, 71(3-4):356–364.

[8]Bailey, S. M., Thomas, G. E., Rusch, D. W., Merkel, A. W., Jeppesen, C. D., Carstens, J. N., Randall, C. E., McClintock, W. E., and Russell III, J. M. (2009). Phase functions of polar mesospheric cloud ice as observed by the cips instrument on the aim satellite. *Journal of Atmospheric and Solar-Terrestrial Physics*, 71(3-4):373–380.

[9]Templeton, G. (2015). Artificial neural networks are changing the world. what are they? https://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they.

[10]Agatonovic-Kustrin, S. and Beresford, R. (2000). Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5):717–727.

[11]Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[12]Gharakhanian, A. (2017). Generative adversarial net-works – hot topic in machine learning. https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html.

[13]Moutarde, F. (n.d.). Convolution Network. https://people.minesparis.psl.eu/fabien.moutarde/ES_MachineLearning/TP_convNets/convnet-notebook.html.

[14]Khosla, S. (2019). Introduction to Pooling Layer. https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/.

[15]PyTorch (2018). Pytorch master documentation. https://pytorch.org/docs/stable/index.html.

[16]Gomez, R. (2018). Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names. https://gombru.github.io/2018/05/23/cross_entropy_loss/.

[17]Roy, R. (2020). ML | Stochastic Gradient Descent (SGD). https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/.