Superresolution Techniques for Climate Simulation Output

Daniel Korytina

CSCI 5922

Spring 2019

Project Description

- This project aims to deploy a GAN model to upscale low resolution daily precipitation simulations over North America to a high resolution, specifically 4x in both x and y dimensions.

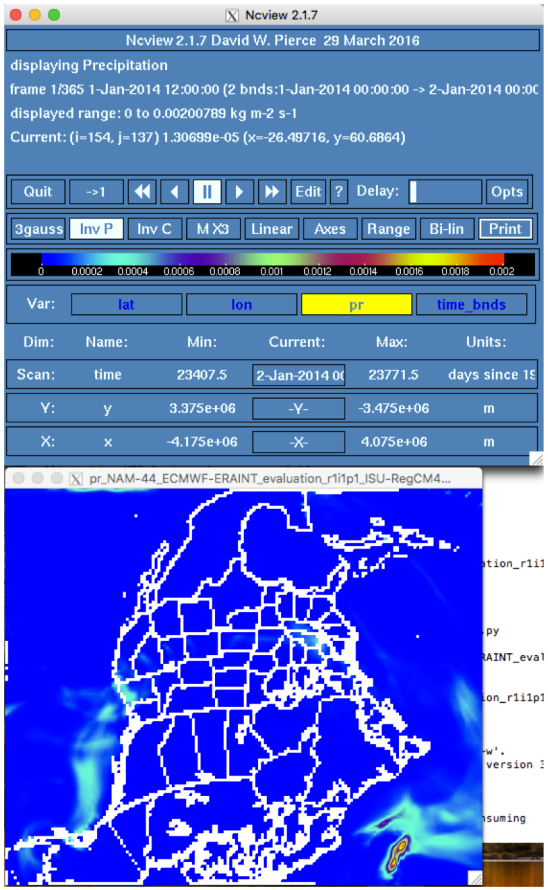
Differences from Proposal

- Interpolating 200km to 50km (instead of 1km)
  - Easier to spot low res artifacts, faster to train
- Break image into 20x20 patches
  - Native (166,138) is resized to (180,140) via cv2 then broken into 63 20x20 samples
  - Each nc4 file has 365 frames;
    - This then gives 63*365=22995 examples just for year 2014
- Data is extracted from NetCDF4 file on server and saved as a numpy .npy file to be Torched on my laptop.

Data set

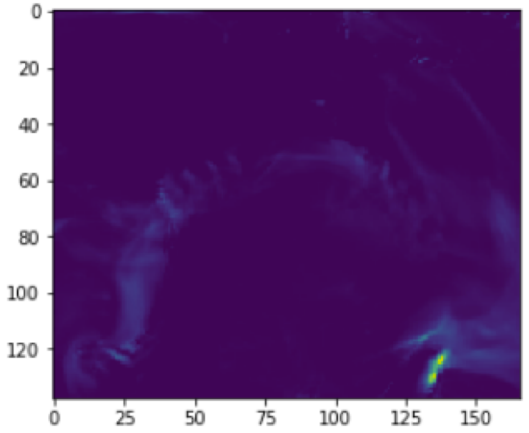
- CORDEX (standard) WRF and RegCM4 models of resolutions NAM-22 and NAM-44. These consist of 1 or 3-hourly snapshots of simulated weather from 1950-2100, over North American, with 25km and 50km resolutions, respectively. Most important variables include tas (surface temperature) and pr (rain). Other variables include: ps (surface pressure), huss (humidity), wind and many others.
  http://www.cordex.org/
  - In this project I'm specifically looking at RegCM4 Era-interim 50km pr for year 2014, daily averages.

Example 2D slice of 3D data, variable: pr




```
# load precip
d = np.load('pr.npy')
print(d.shape)
%matplotlib inline
import matplotlib.pyplot as plt
plt.imshow(d[0])
plt.show()
```

```
(365, 138, 166)
```



Tools

- NetCDF - NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
- cv2
- Sklearn
- Torch
- Matplotlib

Methods

Patch subsets

CV2 MM interpolation

Sklearn *linear_model.LinearRegression*

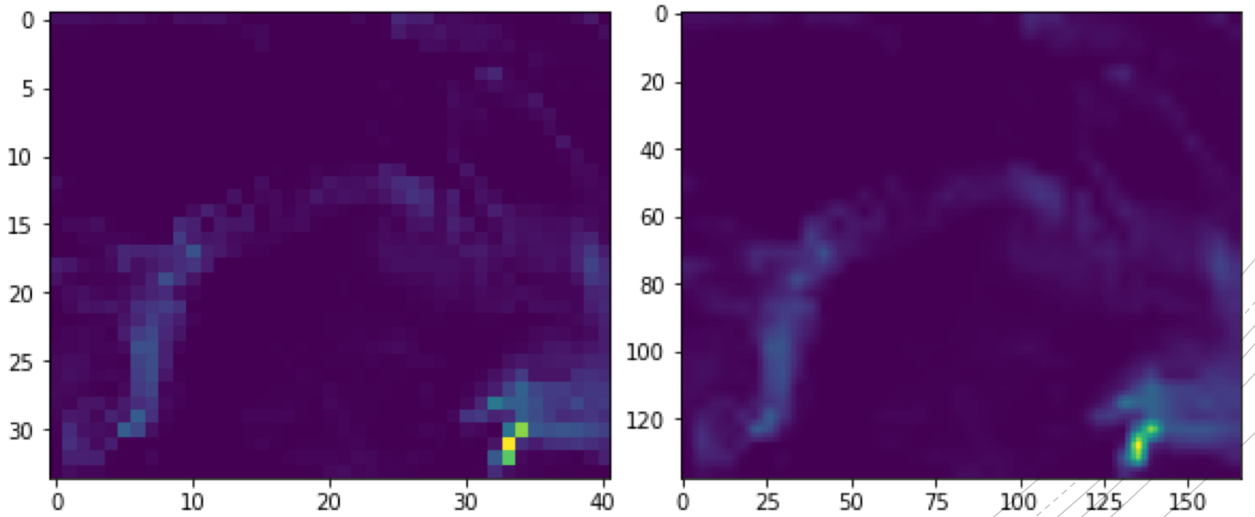
Simple 4-layer CNN GAN

Patch sub-sets

```
# break up into many samples
osamples = []    # 20x20 samples
isamples = []   # 5x5 input samples
for t in range(365):
   # rescale to dims divisible by 20, i.e. (166,138)->
(180,140)
   or_img = cv2.resize(d[t],(180,140))
   for y in range(0,140,20):
      for x in range(0,180,20):
         osample = or_img[y:y+20, x:x+20]
         isample = cv2.resize(osample,(5,5))
         osamples.append(osample)
         isamples.append(isample)
```

CV2 Interpolation

```
us_shape = (166,138)
ds_shape = (us_shape[0]//4, us_shape[1]//4)
print(ds_shape)
ds_img = cv2.resize(d[0],ds_shape)
plt.imshow(ds_img)
plt.show()
```



CNN GAN Interpolation

- Started with "Vanilla GAN" from medium.com
  - Generator:
    - 3 conv layers with Relu()
    - 4th covn activates with Tanh()
  - Discriminator:
    - 3 conv layers with LeakyRelu()
    - 4th conv layers activates with Sigmoid()
  - Originally tuned to generate MNIST data 64x64
    - Used cv2.resize to feed it my 20x20 data patches

Subtasks

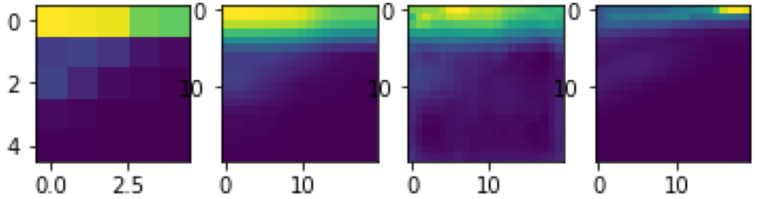
- Data collection – Decided to use 50km Era-Int RegCM4 daily interval precip simulation over North America
- Preprocessing – Open NetCDF file on server and saved as Numpy .npy file
- Management – Data is contained in single file. 20x20 data sets are created on the fly from this data. Initial analysis – baseline.ipynb and baseline.py
- Design – MSE for 4x cv2'ed images
- Evaluation – How well do different interpolation methods compare? How about different regional models (WRF vs RegCCM4)? Training on historical data (1950-2005), how well do these variable associations in future data (2006-2100)?
- Visualization – NetCDF tool *ncview* (see previous slide), matplotlib graphs.

Results

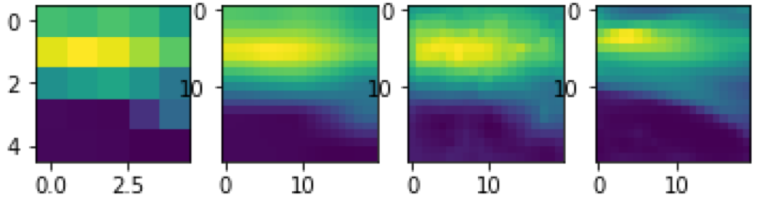
Interpolation from 200km to 50km

Baseline (see notebook for more)

PATCH# 1
mse 5x5: n/a
mse cv2: 0.0017179612
mse intr: 0.0017847365
mse orig: 0.0



PATCH# 63
mse 5x5: n/a
mse cv2: 0.007944433
mse intr: 0.007294818
mse orig: 0.0



Future work: temporal cues and other features

- Using 1 or 3 hourly samples (instead of daily avgs)
  - X,Y movement of precipitation vs time can inform interpolation
- Preserving statistical properties via 2-day average
  - Simultaneously interpolate day(n) + day(n+1)
    - Such that, Avg(I(day(n))+ Avg(I(day(n+1))==Avg(day(n), day(n+1))
- Wind vector
  - Wind direction and strength should affect precip patterns
- Tas
  - (near-surface temps) should have an effect on precipitation

See more on Github

https://github.com/korytina/csci5922