Simulations and Machine Learning for Parachute Navigation

Grace Malabanti, Joseph Scheufele, and Juliette Spitaels

Advisors: Randy Paffenroth and Oren Mangoubi

Sponsoring Co-Advisor: Greg Noetscher (CCDC-SC)

Sponsor

- U.S. Army Combat Capabilities Development Command
- Located in Natick, MA
- Experts in Soldier systems research and development

Problem

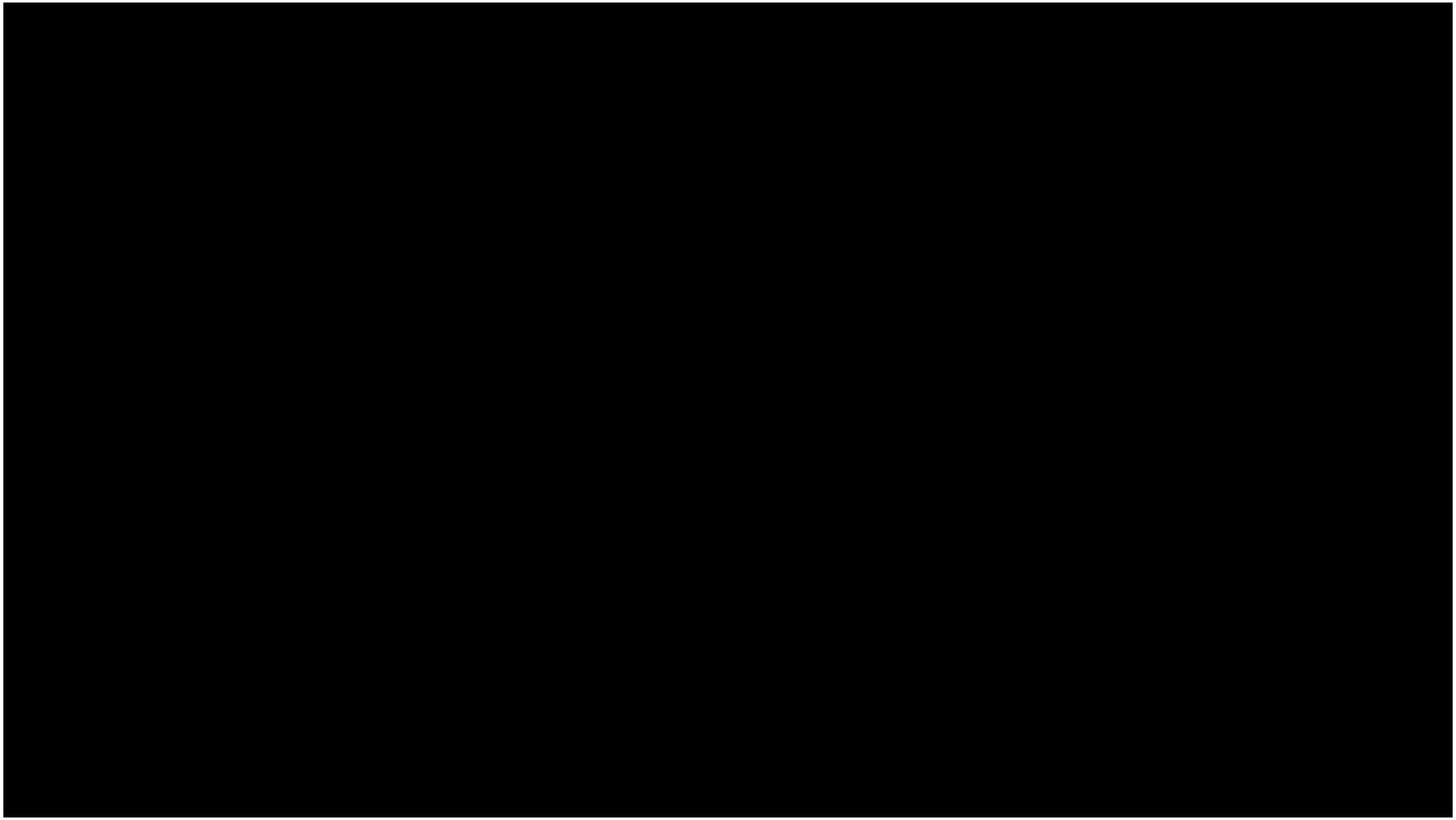
- Soldier required supplies in the field
- DEVCOM-SC specializes in the aerial delivery of these supplies
- Parachutes are navigated by GPS for these deliveries
- Issues with GPS

Solution

Simulator

Machine Learning

Comparison



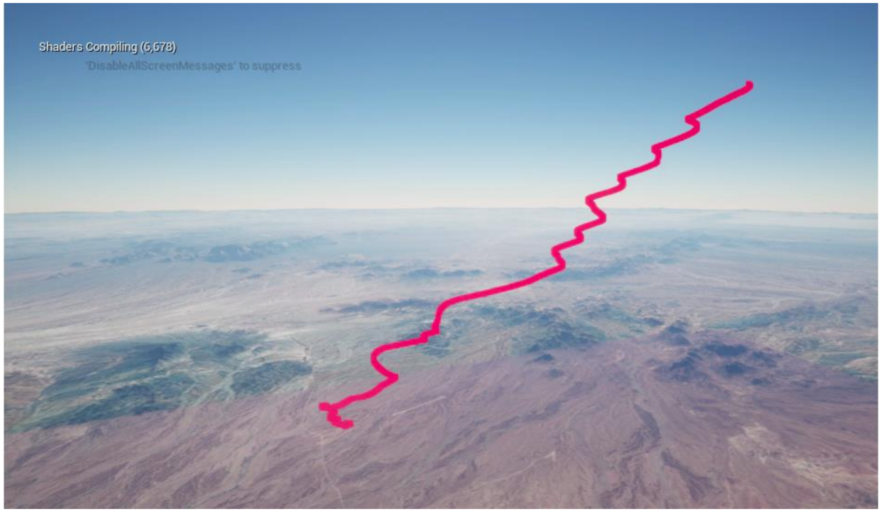
Flight Paths

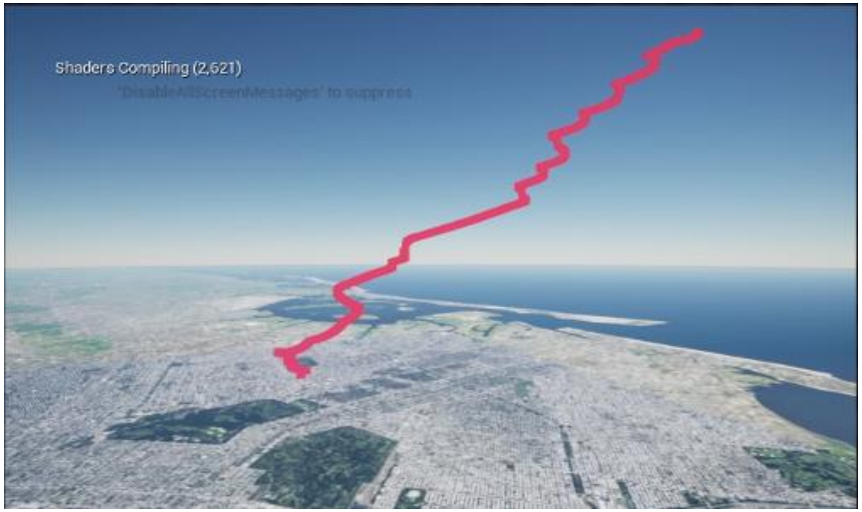
Extracted location information from in the field drops using DEVCOM-SC parser

Wrote code to clean and reformat coordinates to recreate drops in original location

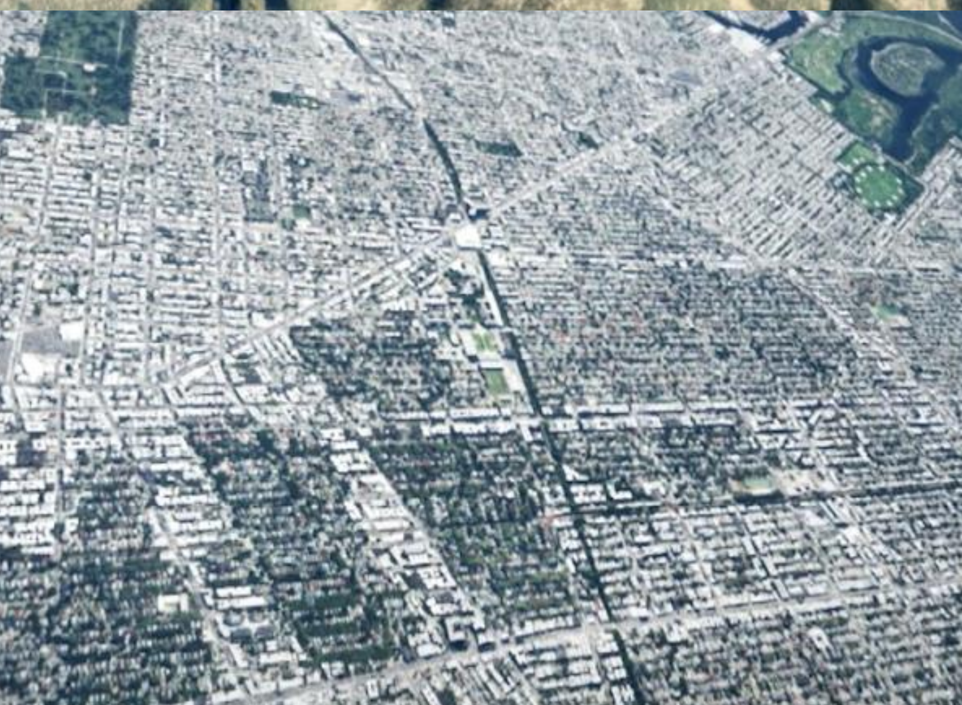
Wrote code to transform paths to new locations, based on user defined landing point

Built projects in new locations all over the world for variety of terrain

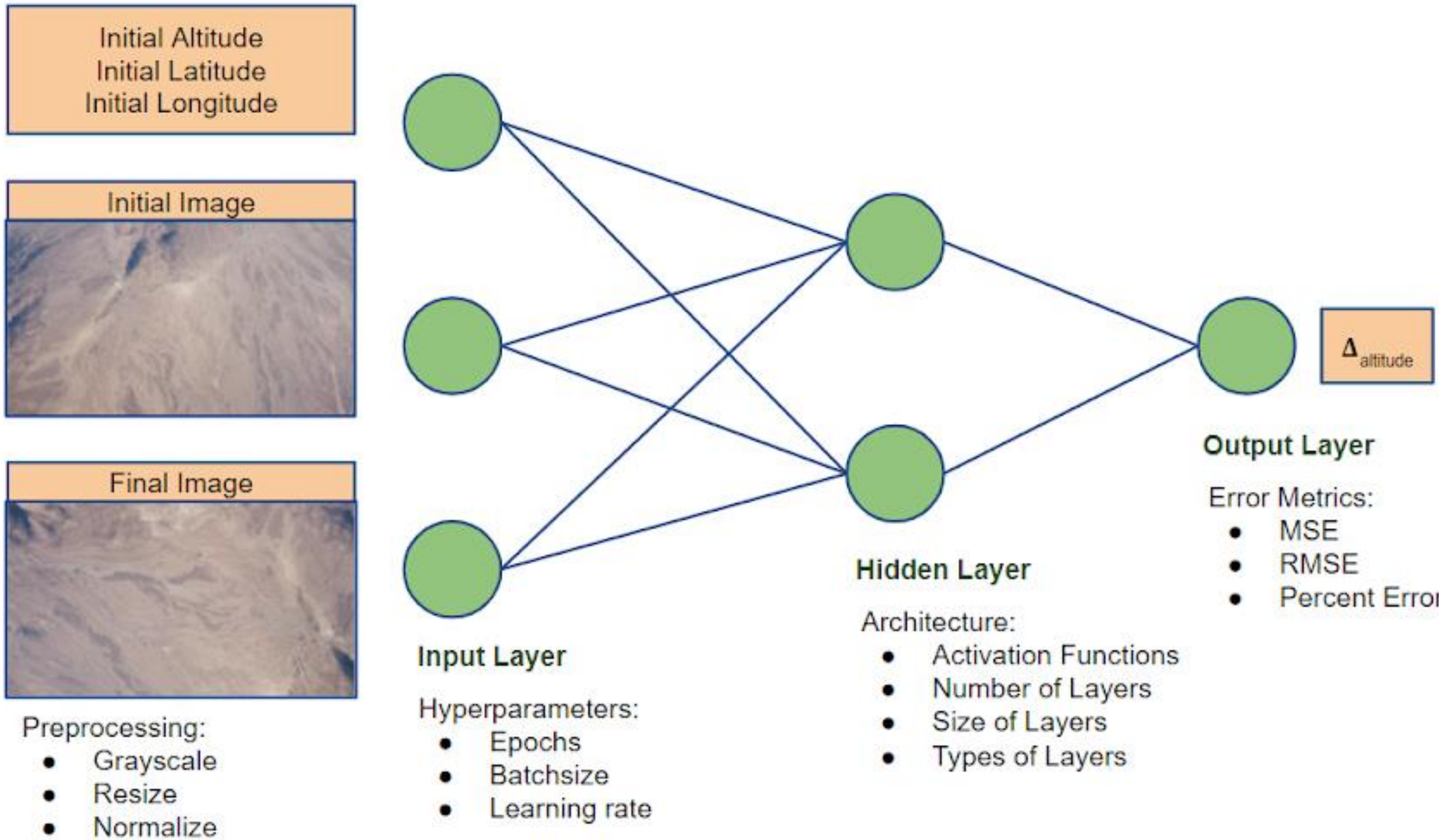




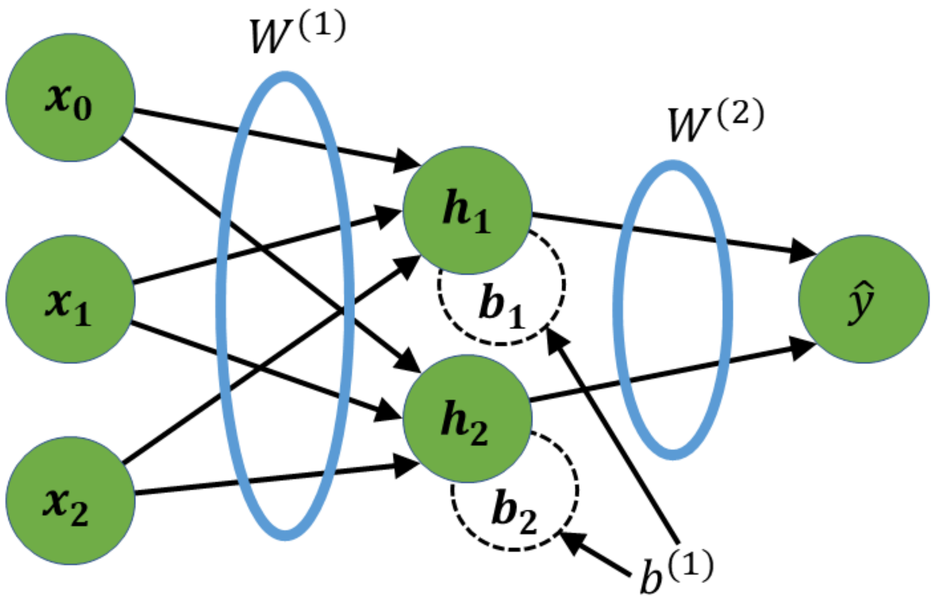





Our Neural Network



Neural Networks



$$f\left(W^{(2)}f\left(W^{(1)}X + b^{(1)}\right) + b^{(2)}\right) = \hat{y}$$

- Forward Pass
  - Densely connected network
  - Each neuron has a weight and bias
  - Activations control passing of information

Training

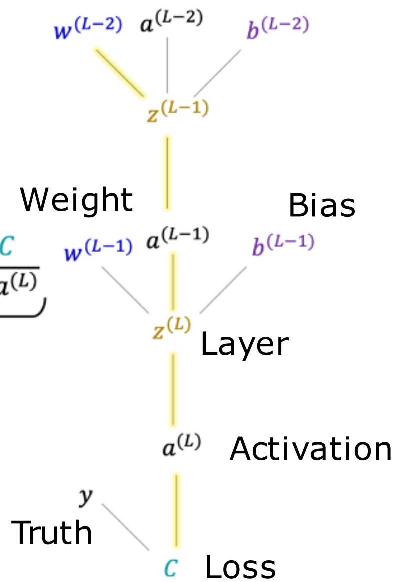
Backpropagation

What we want: $\dfrac{\partial C}{\partial w^{(L-2)}}$

Chain rule

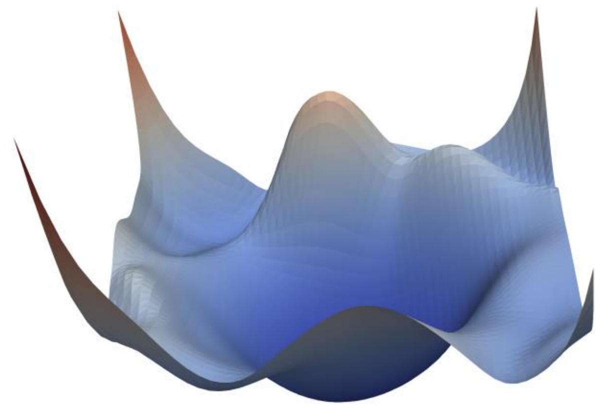
$$\frac{\partial C}{\partial w^{(L-2)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \underbrace{\frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}}}$$

$$\frac{\partial C}{\partial w^{(L-2)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \frac{\partial C}{\partial a^{(L-1)}}$$



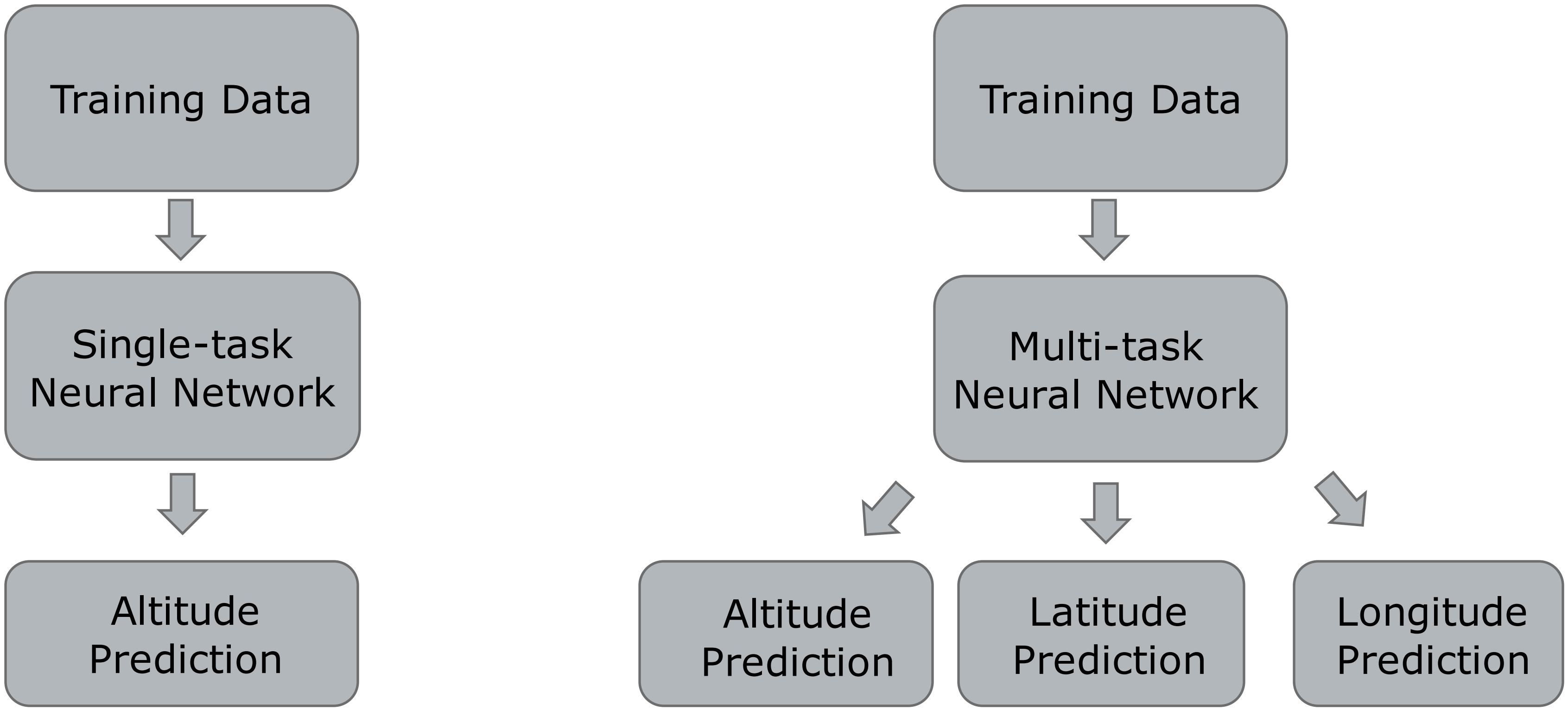
- Backward Pass
  – Finding the global min
  – Chain Rule for updating network
  – Surface depends on Loss Function

Loss Surface



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Single-task vs. Multi-task Neural Networks



$$MSE_{alt} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

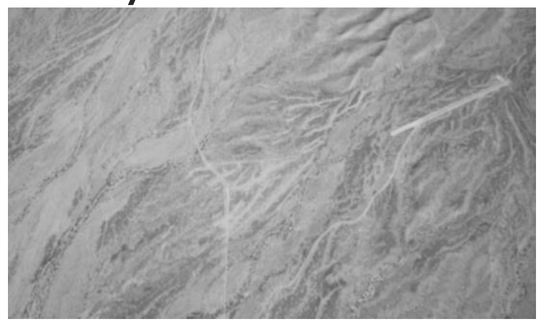
$$MSE_{multi} = \frac{1}{3}(MSE_{alt} + MSE_{lat} + MSE_{lon})$$

Caruana, R. Multitask Learning. *Machine Learning* 28, 41–75 (1997).

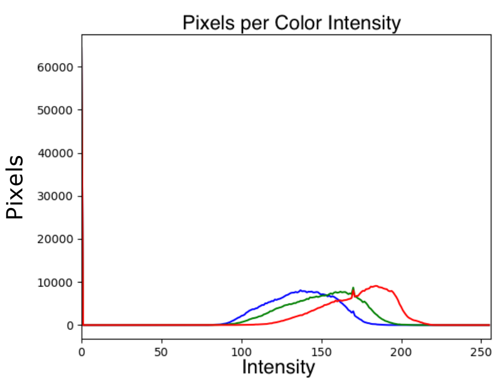
Image Processing

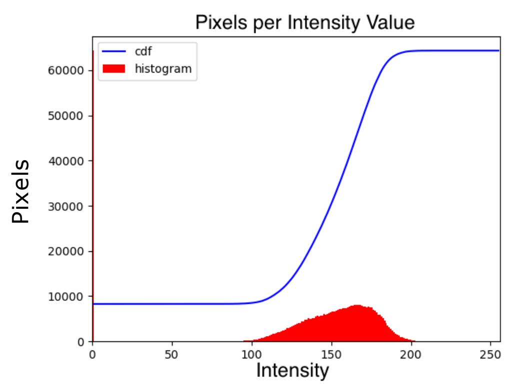
Original

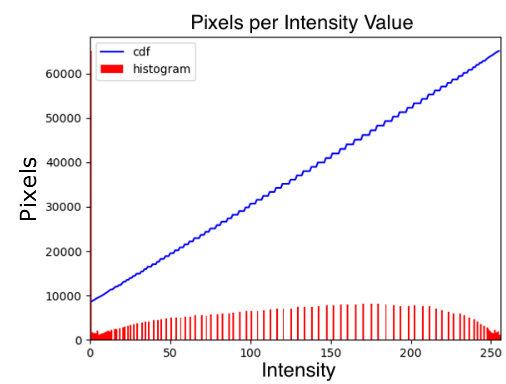
Grayscale



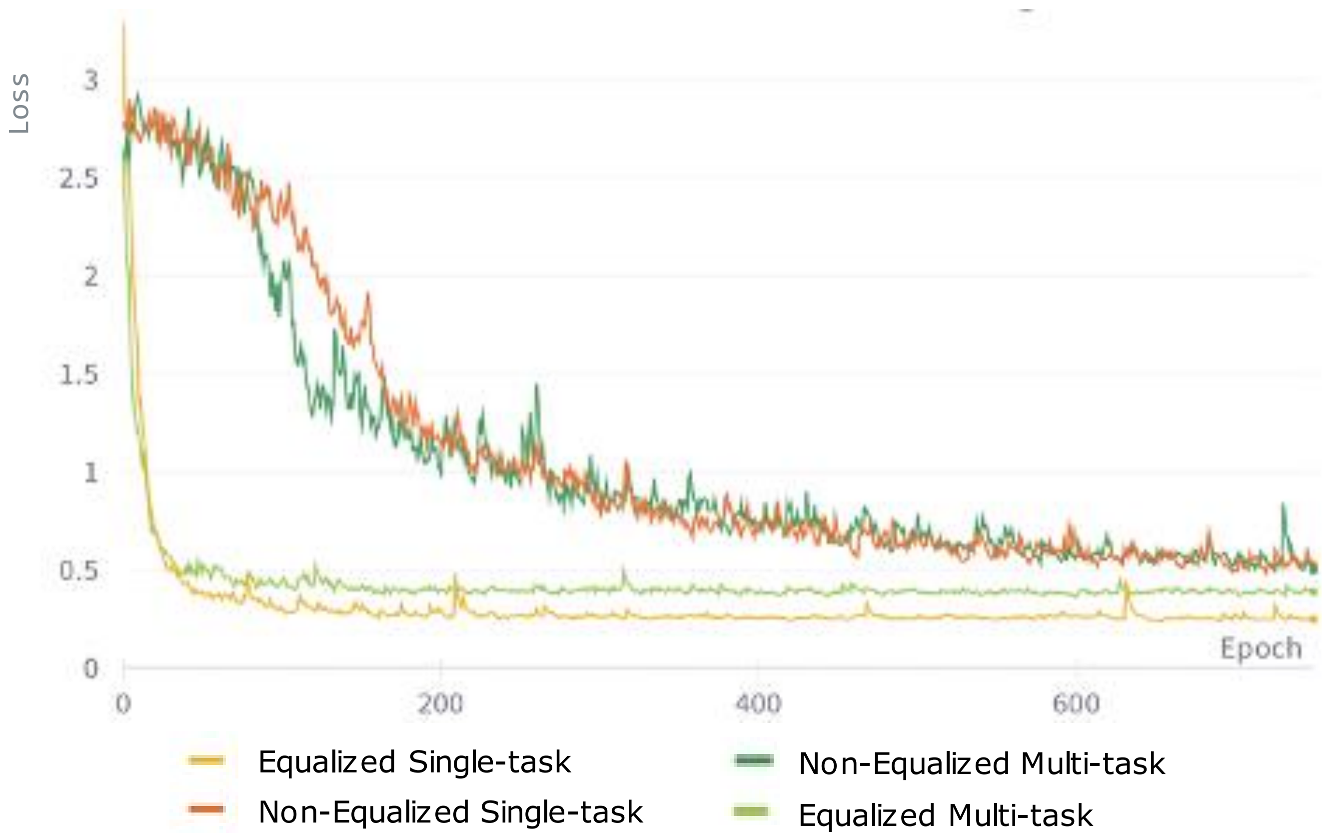
Equalize







Effects of Equalization



- Faster convergence
- Lower final error
- Smoother loss surface

Results

Variables			Testing Error		Validation Error	
Single or multi?	Equalized	Activation Function	RMSE	Percent Error	RMSE	Percent Error
Single	None	ReLU	25.607	0.489	78.474	1.379
Multi	None	ReLU	25.664	0.484	77.809	1.005
Single	Pairwise	ReLU	18.540	0.262	79.264	3.249
Multi	Pairwise	ReLU	18.673	0.291	18.907	0.294
Single	Pairwise	GELU	18.370	0.261	78.867	3.226
Multi	Pairwise	GELU	18.510	0.283	18.697	0.290
Single	Pairwise	Leaky ReLU	17.952	0.257	79.012	3.222
Multi	***Pairwise***	***Leaky ReLU***	***17.528***	***0.264***	**17.493**	**0.277**

Conclusions

- It could be that multi-task neural networks smooth the loss landscape, explaining our discrepancy in error between the testing and validation sets.

- The availability of unlimited simulated data allowed our team meaningfully apply ML models, and in turn showed additional evidence for the possibility of effective GAVN technology that could outperform traditional GPS signaling.

Thank you!

Questions?

Image References

DEVCOM-SC:

https://ccdcsoldiercenter.army.mil/#/whatwedo

Loss Surface:

https://www.cs.umd.edu/~tomg/projects/landscapes/

Back Propagation:

https://towardsdatascience.com/the-maths-behind-back-propagation-cf6714736abf

Error Metrics

n = number of data points

y_i = observed values

$\hat{y}_i$ = predicted values

Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Percent Error

$$\text{Percent Error} = \left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right|$$

Future Work

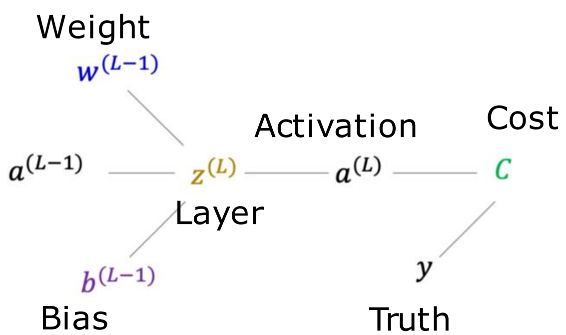
Future teams that work on this project teams could:

- Develop new features in the simulator
- Refine the preprocessing techniques
- Run the network with more diverse environments
- Improve the neural network

```python
from collections import OrderedDict
from torch import as_tensor, nn, no_grad, cat, flatten, float32, uint8, div, tensor
from torchvision import transforms
import numpy as np
import wandb
from random import sample

class BasicNetwork(nn.Module):
  def __init__(self, in_features, out_features):
    super(BasicNetwork,self).__init__()
    self.longer_stack = nn.Sequential(OrderedDict([ #Creating an ordered dictionary c
        ('Input', nn.Linear(in_features, 1024)),
        ('Relu 1', nn.ReLU()),
        ('Hidden Linear 1', nn.Linear(1024, 512)),
        ('Relu 2', nn.ReLU()),
        ('Hidden Linear 2', nn.Linear(512, 256)),
        ('Relu 3', nn.ReLU()),
        ('Hidden Linear 3', nn.Linear(256, 128)),
        ('Relu 4', nn.ReLU()),
        ('Hidden Linear 4', nn.Linear(128, 64)),
        ('Relu 5', nn.ReLU()),
        ('Hidden Linear 5', nn.Linear(64, 32)),
        ('Relu 6', nn.ReLU()),
        ('Hidden Linear 6', nn.Linear(32, 16)),
        ('Relu 7', nn.ReLU()),
        ('Hidden Linear 7', nn.Linear(16, 8)),
        ('Relu 8', nn.ReLU()),
        ('Output', nn.Linear(8, out_features))
    ]))
```

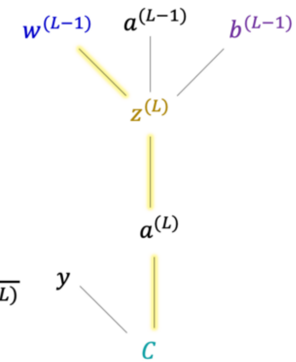
Back Propagation II



What we want: $\frac{\partial C}{\partial w^{(L-1)}}$

Chain rule

$$\frac{\partial C}{\partial w^{(L-1)}} = \frac{\partial z^{(L)}}{\partial w^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}}$$



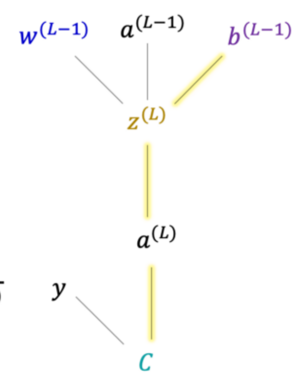
$$w^{(L-1)}(t+1) = w^{(L-1)}(t) - \eta \frac{\partial C}{\partial w^{(L-1)}}$$

$$b^{(L-1)}(t+1) = b^{(L-1)}(t) - \eta \frac{\partial C}{\partial b^{(L-1)}}$$

What we want: $\frac{\partial C}{\partial b^{(L-1)}}$

Chain rule

$$\frac{\partial C}{\partial b^{(L-1)}} = \frac{\partial z^{(L)}}{\partial b^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}}$$



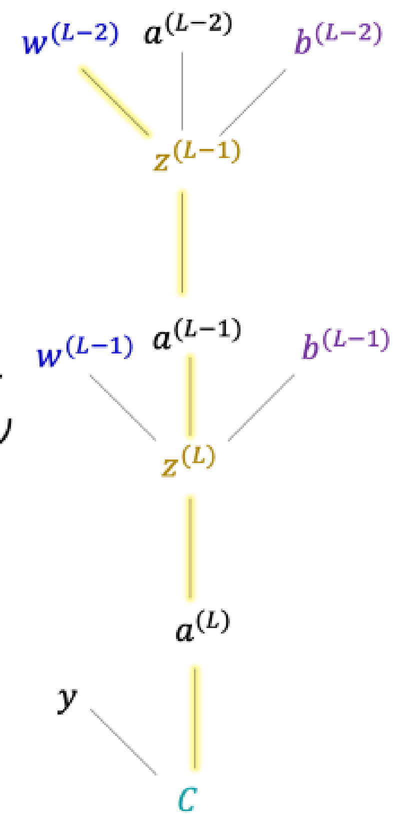
Back Propagation III

$$\boxed{\text{What we want: } \frac{\partial C}{\partial w^{(L-2)}}}$$

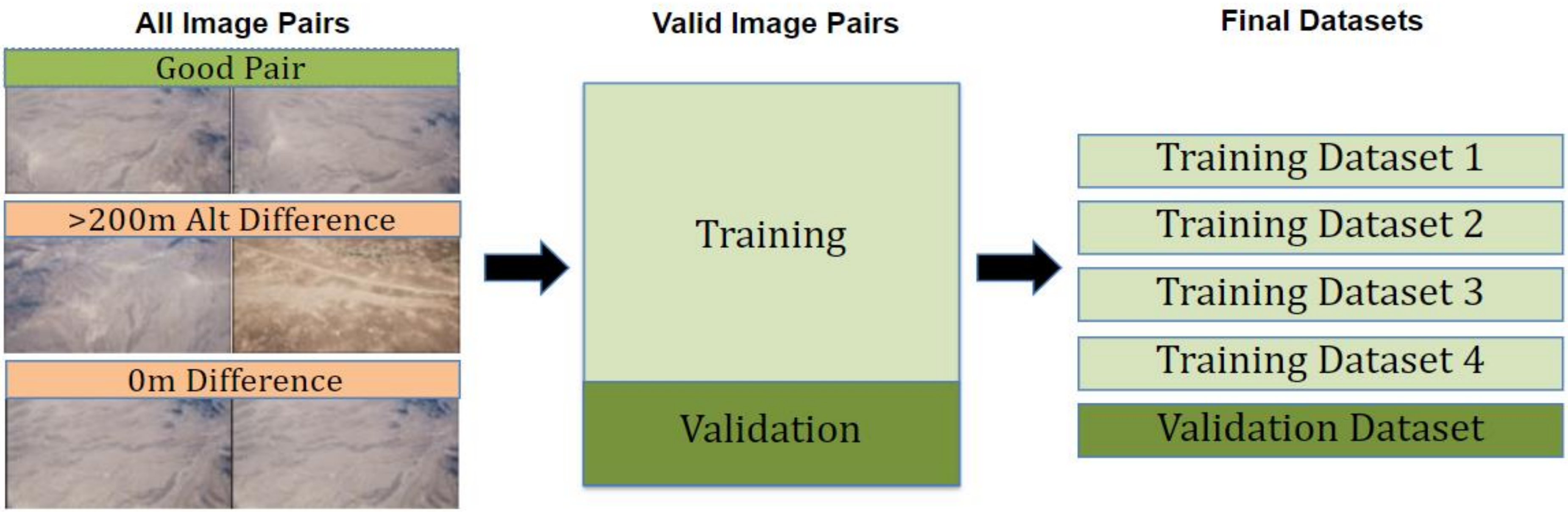
Chain rule

$$\frac{\partial C}{\partial w^{(L-2)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \underbrace{\frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C}{\partial a^{(L)}}}$$

$$\frac{\partial C}{\partial w^{(L-2)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-2)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \frac{\partial C}{\partial a^{(L-1)}}$$



Datasets


All Image Pairs
Good Pair
>200m Alt Difference
0m Difference
Valid Image Pairs
Training
Validation
Final Datasets
Training Dataset 1
Training Dataset 2
Training Dataset 3
Training Dataset 4
Validation Dataset

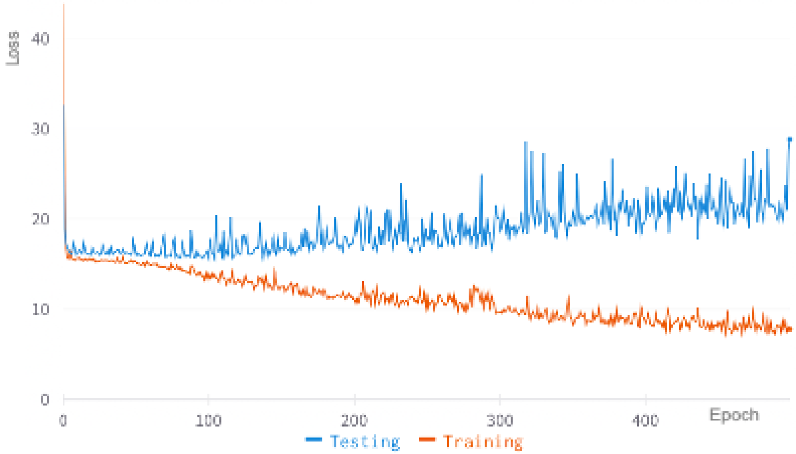
Random Forest

Trained on 5000 grayscale images

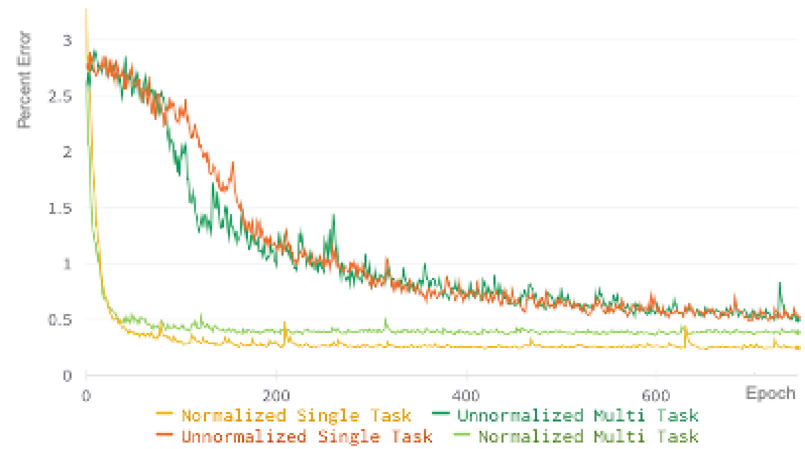
100 trees

RMSE of 55.74 m

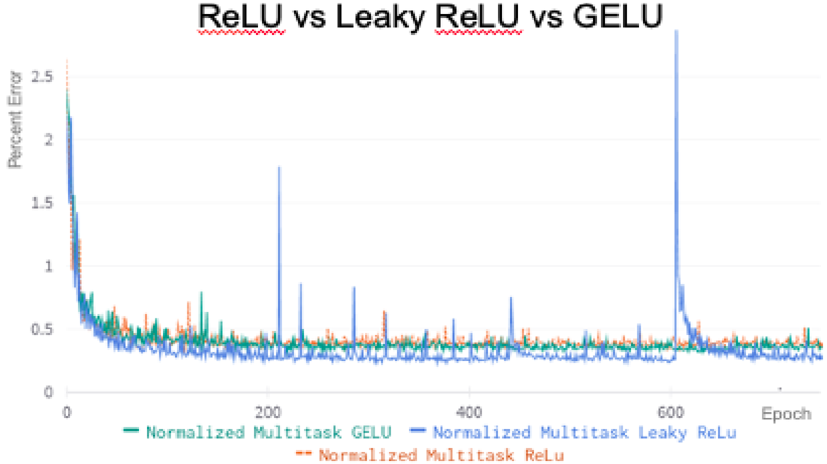
Poor Results



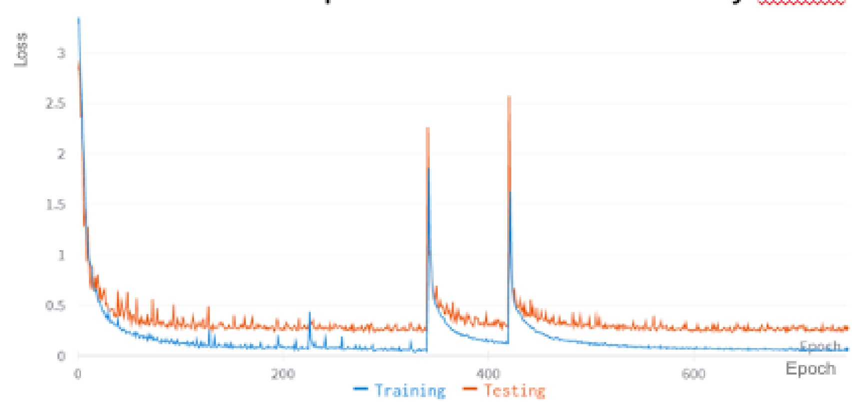
Effects of Equalization on Multitask vs Single Task

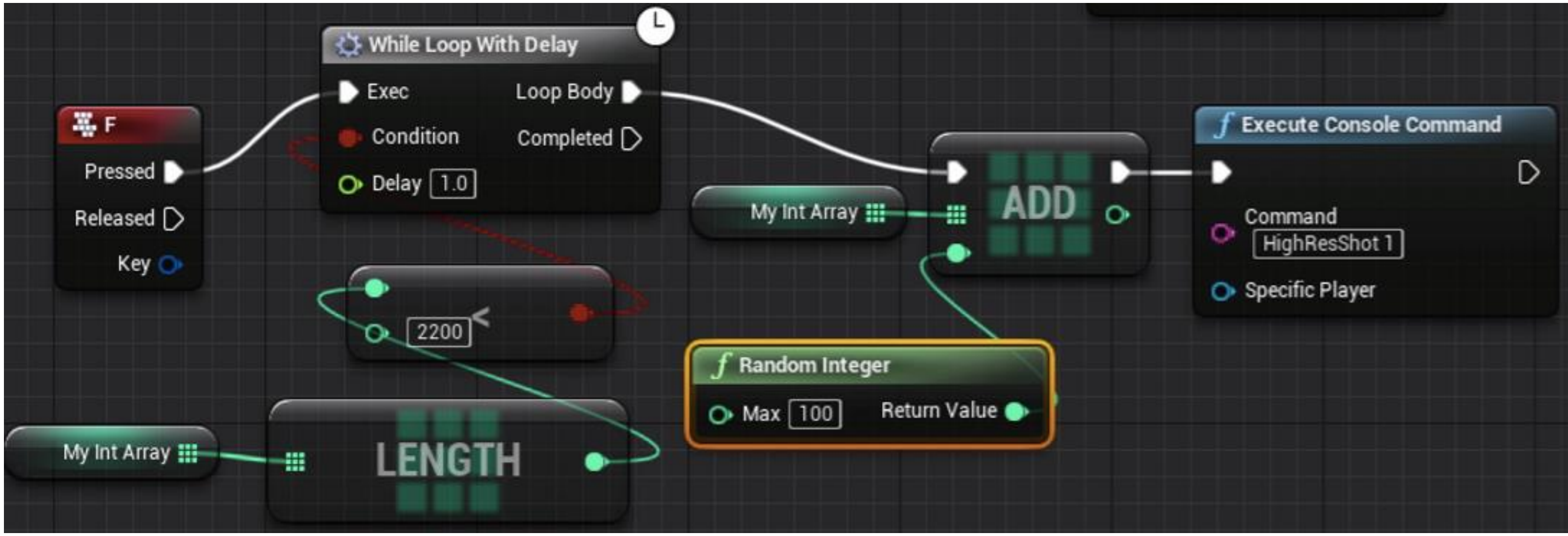


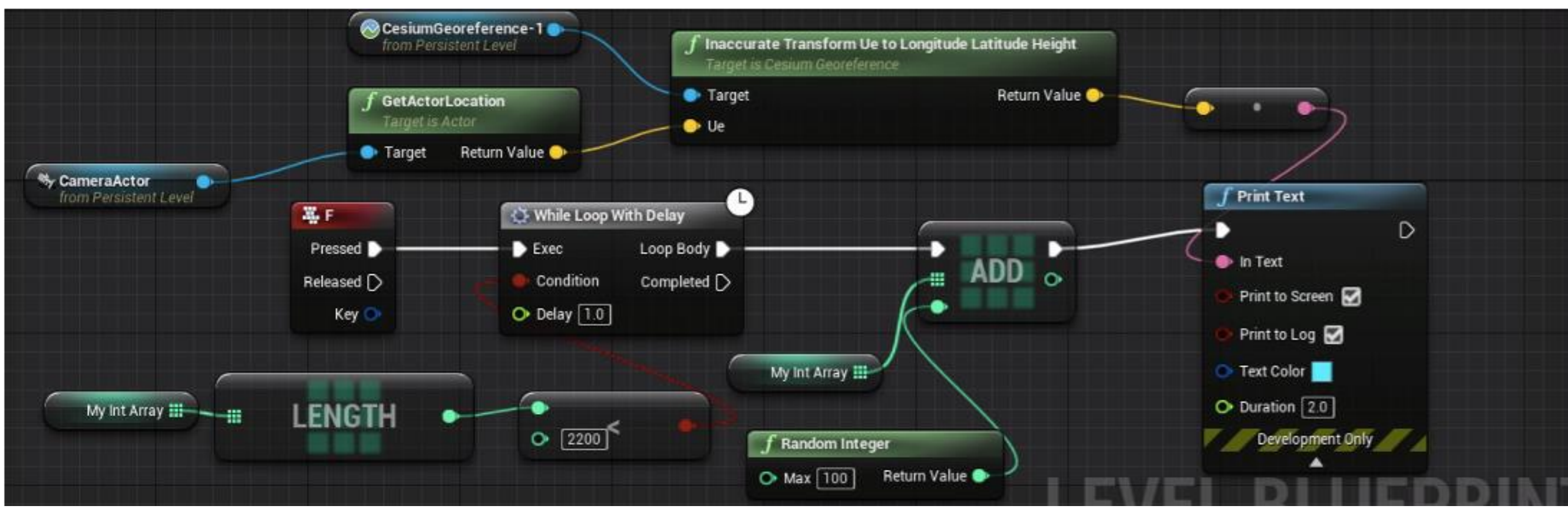
ReLU vs Leaky ReLU vs GELU

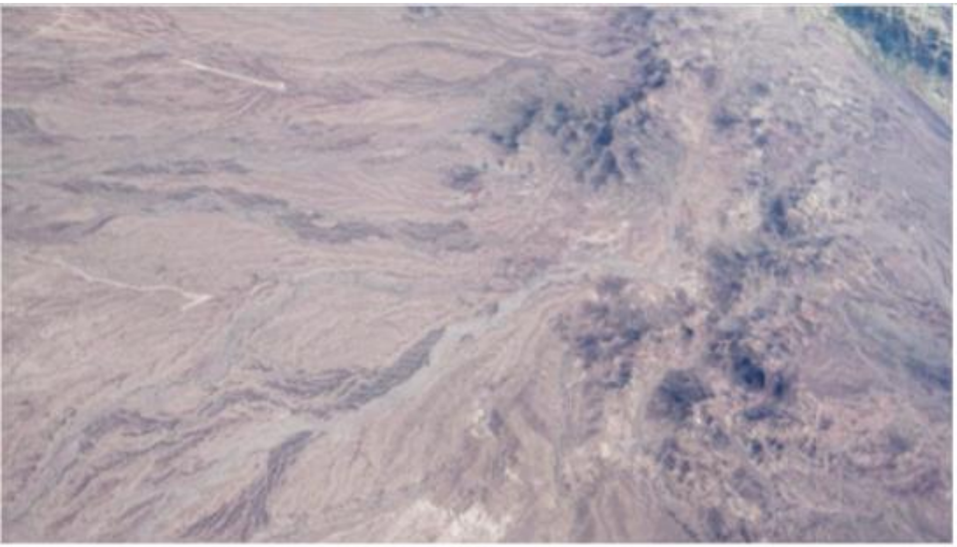


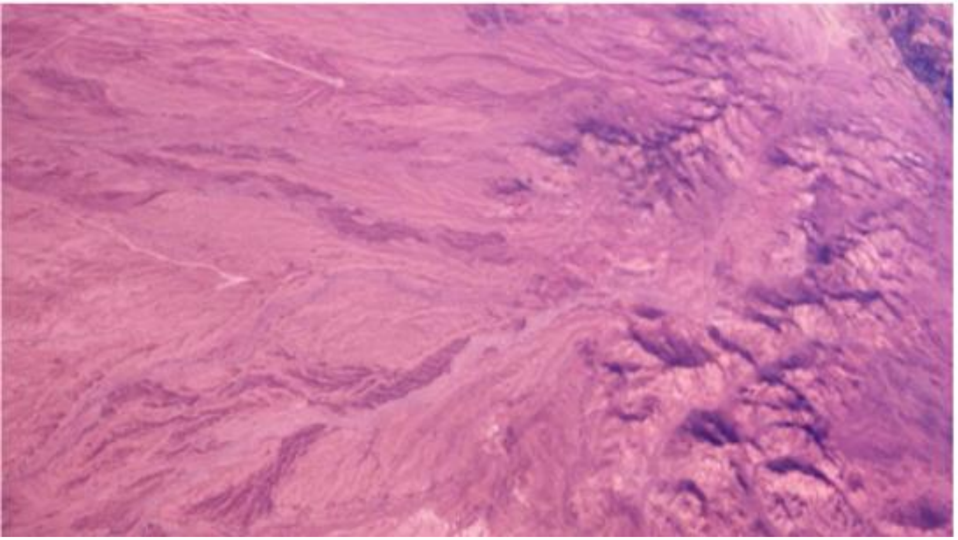
Best Network: Equalized Multitask with Leaky ReLU




While Loop With Delay
Exec
Loop Body
Condition
Completed
Delay 1.0
F
Pressed
Released
Key
Execute Console Command
Command
HighResShot 1
Specific Player
My Int Array
ADD
2200
Random Integer
Max 100
Return Value
My Int Array
LENGTH


CesiumGeoreference-1
from Persistent Level
Inaccurate Transform Ue to Longitude Latitude Height
Target is Cesium Georeference
Target
Return Value
Ue
GetActorLocation
Target is Actor
Target
Return Value
CameraActor
from Persistent Level
F
Pressed
Released
Key
While Loop With Delay
Exec
Loop Body
Condition
Completed
Delay 1.0
ADD
Print Text
In Text
Print to Screen
Print to Log
Text Color
Duration 2.0
Development Only
My Int Array
My Int Array
LENGTH
2200
Random Integer
Max 100
Return Value







CESIUM ion
Upgrade for commercial use. Data attribution

Neural Network Hyperparameters	
Epochs	750
Batch Size	128
Step Size	0.001

Normalized, Multi-task with Leaky ReLU			
	MSE	RMSE	Percent Error
Dataset 0	302.196	17.384	0.275
Dataset 1	287.701	16.962	0.274
Dataset 2	311.733	17.656	0.259
Dataset 3	*327.277*	*18.091*	*0.248*
Average	307.227	17.523	0.264

Normalized, Multi-task with GELU			
	MSE	RMSE	Percent Error
Dataset 0	313.538	17.707	0.251
Dataset 1	360.078	18.976	0.362
Dataset 2	357.839	18.917	0.283
Dataset 3	339.036	18.413	0.238
Average	342.623	18.503	0.283

Normalized, Multi-task with ReLU			
	MSE	RMSE	Percent Error
Dataset 0	327.976	18.110	0.254
Dataset 1	346.760	18.621	0.376
Dataset 2	365.185	19.110	0.262
Dataset 3	354.875	18.838	0.273
Average	348.699	18.670	0.291

Baseline

Layer	Input Size	Output Size
Input: 1	204,480	1024
Hidden: 2	1024	512
Hidden: 3	512	256
Hidden: 4	256	128
Hidden: 5	128	64
Hidden: 6	64	32
Hidden: 7	32	16
Hidden: 8	16	8
Output: 9	8	3

Shallower

Layer	Input Size	Output Size
Input: 1	204,480	2048
Hidden: 2	2048	1024
Hidden: 3	1024	512
Hidden: 4	512	128
Hidden: 5	128	32
Hidden: 6	32	8
Output: 7	8	single: 1 multi:3

Straight Layers

Layer	Input Size	Output Size
Input: 1	204,480	2048
Hidden: 2	2048	1024
Hidden: 3	1024	256
Hidden: 4	256	256
Hidden: 5	256	64
Hidden: 6	64	32
Hidden: 6	32	8
Output: 7	8	single: 1 multi:3

Bigger Input

Layer	Input Size	Output Size
Input: 1	204,480	4096
Hidden: 2	4096	2048
Hidden: 3	2048	512
Hidden: 4	512	256
Hidden: 5	256	128
Hidden: 6	128	64
Hidden: 6	64	32
Hidden: 6	32	16
Hidden: 6	16	8
Output: 7	8	single: 1 multi:3

Variables		New Architectures		
Single or Multi?	Layer Structure	MSE	RMSE	Percent Error
Single	Straight	318.320	17.842	0.342
Multi	Straight	322.673	17.963	0.001
Single	Larger	329.667	18.157	0.284
Multi	Larger	630.166	25.103	0.588
Single	Shorter	334.532	18.290	0.300
Multi	Shorter	300.133	17.324	0.250

NYC	RMSE	% error
	79.325	3.577

Contacts

- **Advisors:**
- Randy Paffenroth- rcpaffenroth@wpi.edu
- Oren Mangoubi- omangoubi@wpi.edu
- **Sponsoring Co-Advisor:**
- Greg Noetscher- gregn@wpi.edu
- **Students:**
- Grace Malabanti- gmalabanti@wpi.edu
- Joe Scheufele- jcscheufele@wpi.edu
- Juliette Spitaels- jspitaels@wpi.edu