

- AI & Computer Vision for Nat Hazards
- Enhancing Data with AI



AI & Computer Vision for Nat Hazards

What is AI?

The science/engineering that makes machines/programs smart.

SVM

Decision tree

KNN

PCA

Artificial neural nets

Random forest

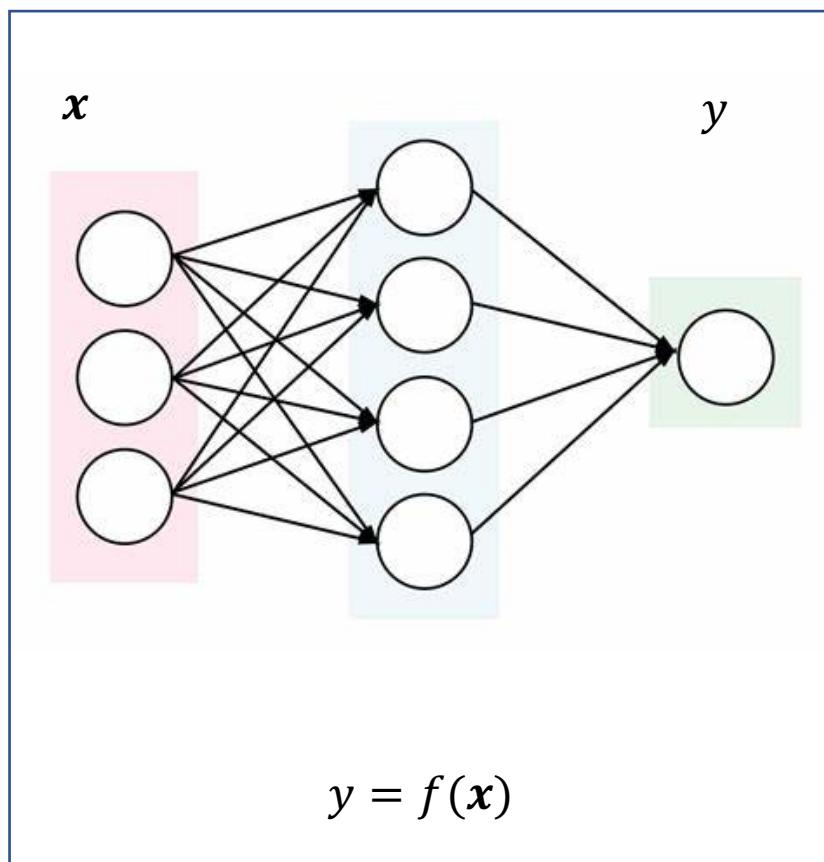
Markov chains

Adaboost

AI & Computer Vision for Nat Hazards

What is AI?

The science/engineering that makes machines/programs smart.



Artificial neural nets

Markov chains

Decision tree

Adaboost

KNN

SVM

PCA

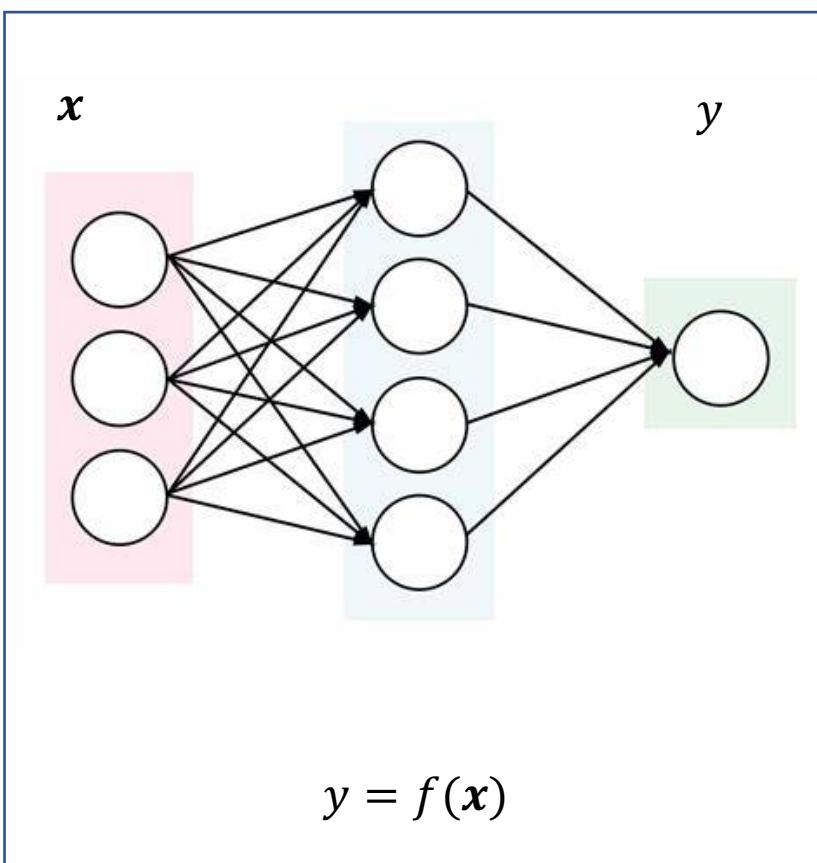
Random forest

AI & Computer Vision for Nat Hazards

What is AI?

Neural nets (NN)

NN architecture



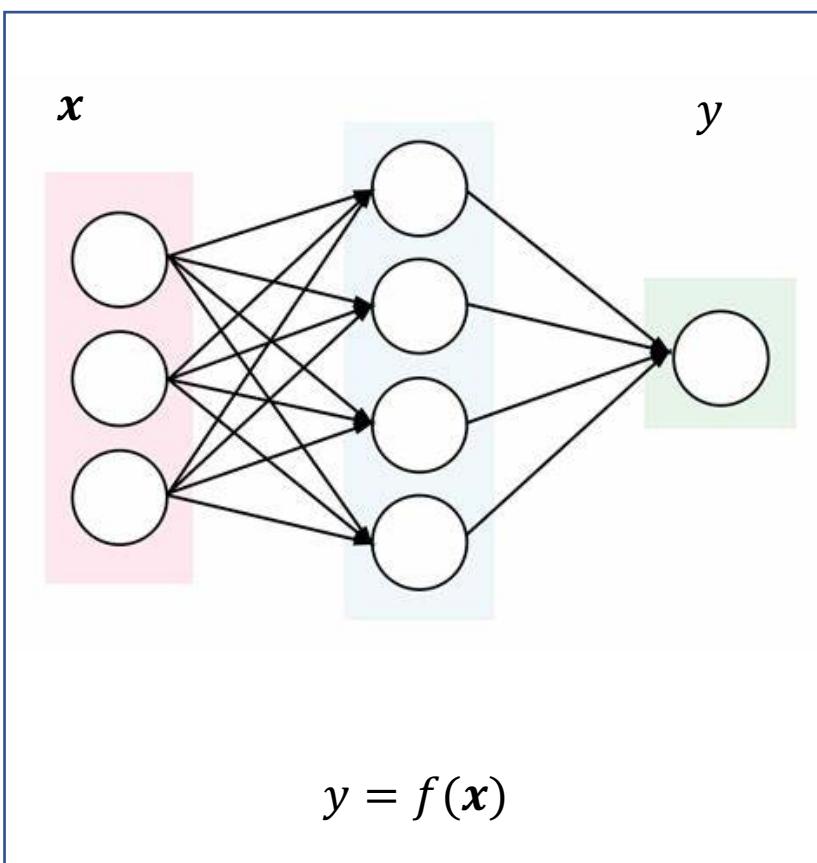
A neuron

AI & Computer Vision for Nat Hazards

What is AI?

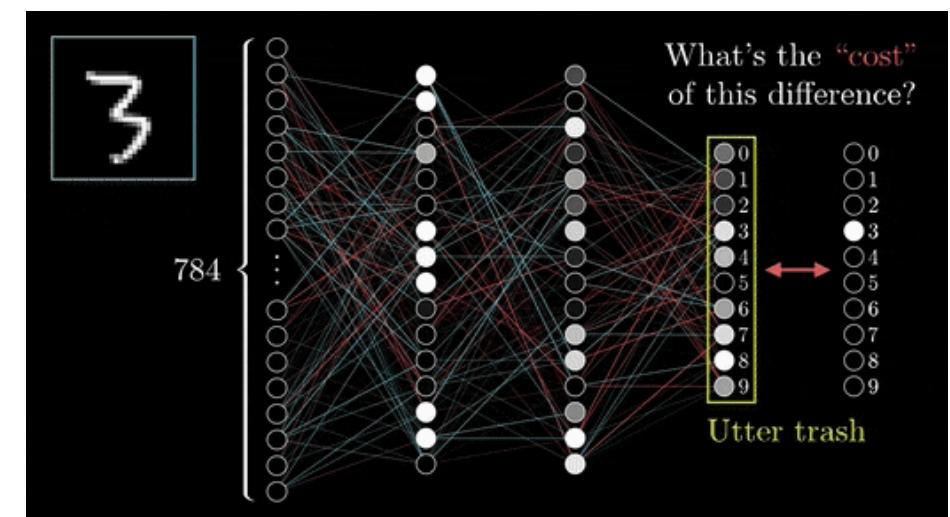
Neural nets (NN)

NN architecture



A neuron

How does it work?

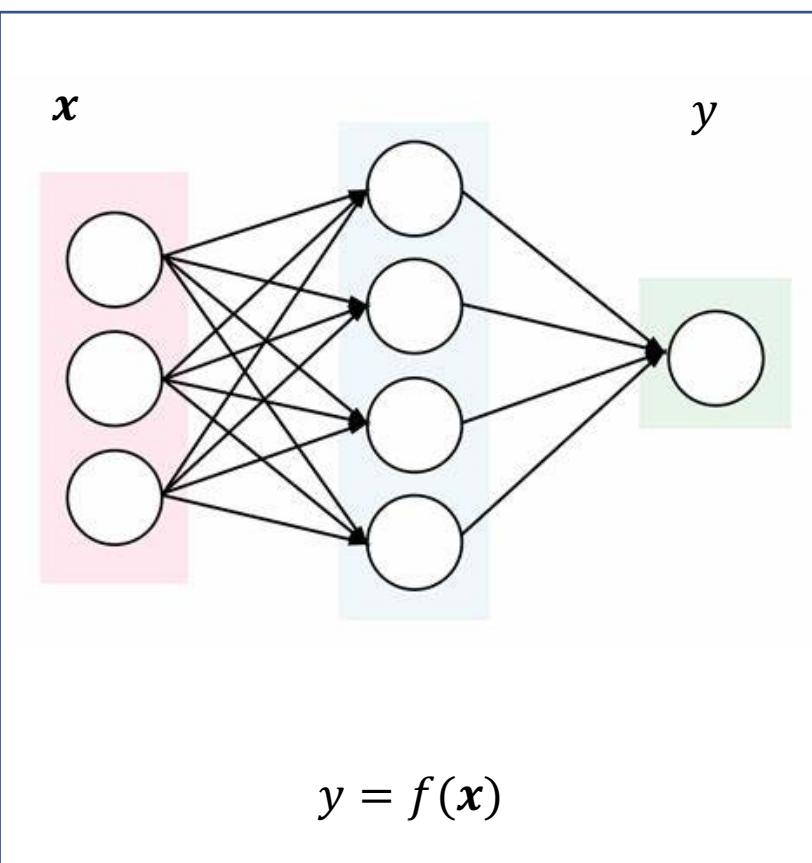


AI & Computer Vision for Nat Hazards

What is AI?

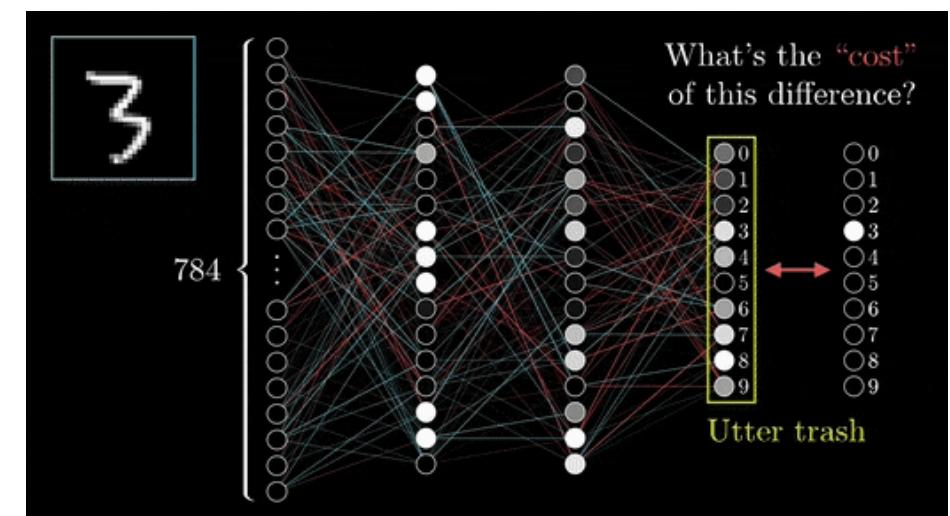
Neural nets (NN)

NN architecture



A neuron

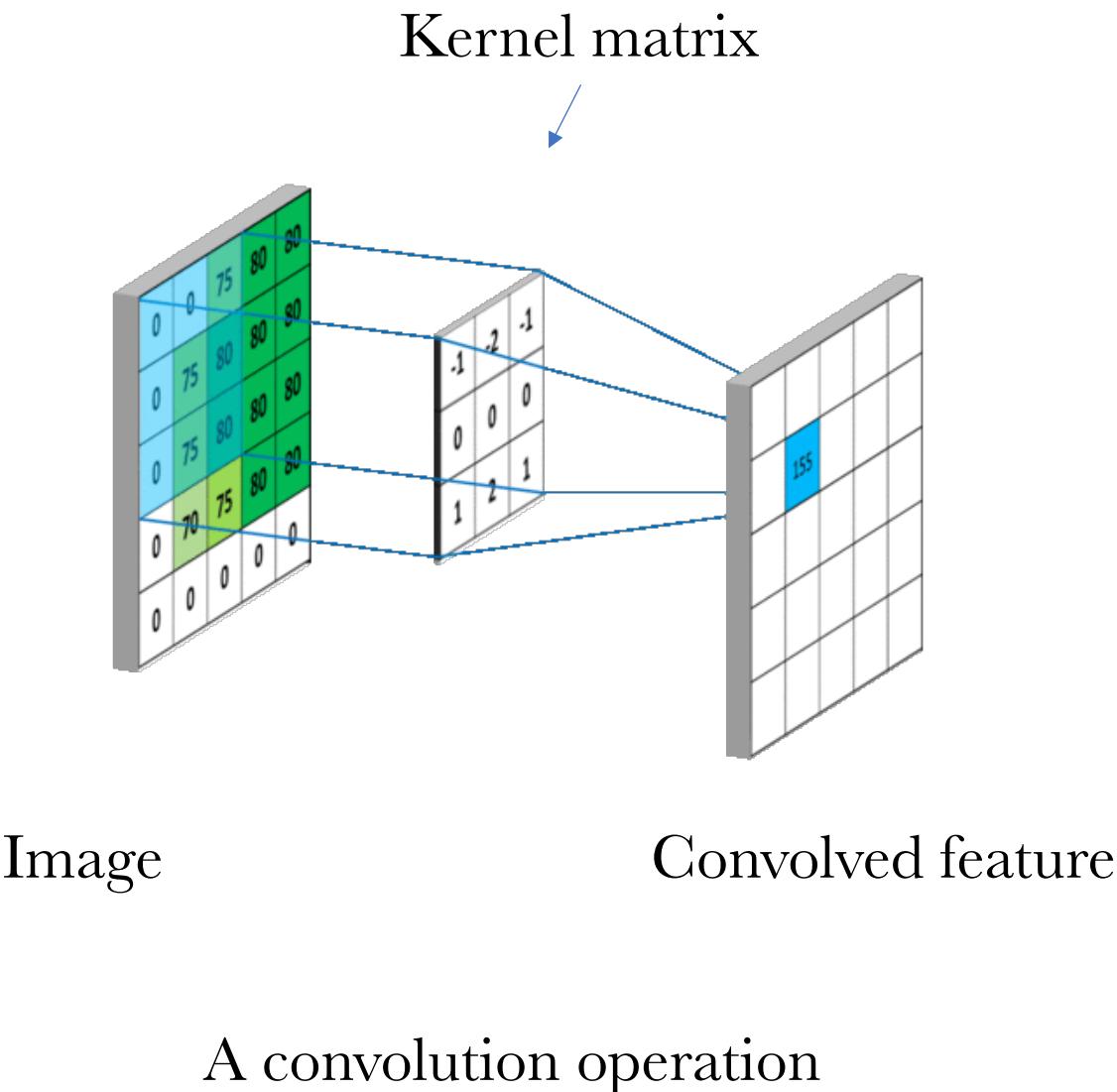
How does it work?



The “weights” are learned results.

AI & Computer Vision for Nat Hazards

What is a convolutional neural net (CNN)?



AI & Computer Vision for Nat Hazards

Learning resources

Machine Learning, Andrew Ng

<https://www.coursera.org/learn/machine-learning>

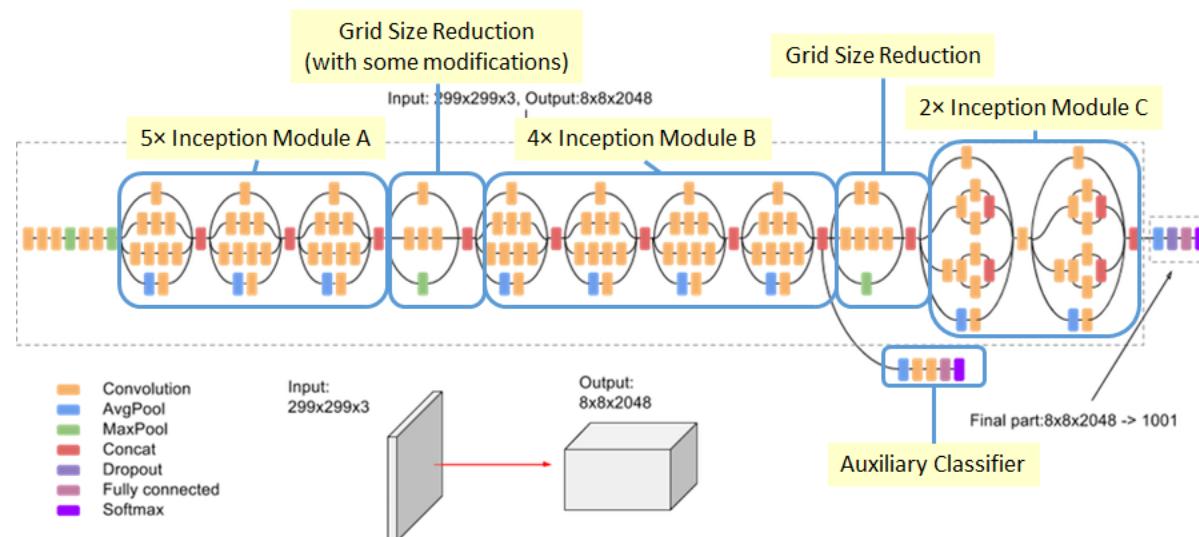
CS231n: Convolutional Neural Networks for Visual Recognition

<http://cs231n.stanford.edu/>

AI & Computer Vision for Nat Hazards

CNN architectures

- AlexNet
- VGGNet
- GoogLeNet
- Microsoft ResNet
- Google Inception
- ...



AI & Computer Vision for Nat Hazards

Tensorflow and TF-slim



TensorFlow is a good start point for ML
It makes it easy for you to build and deploy ML models.

TF-slim <https://github.com/tensorflow/models/tree/master/research/slim>

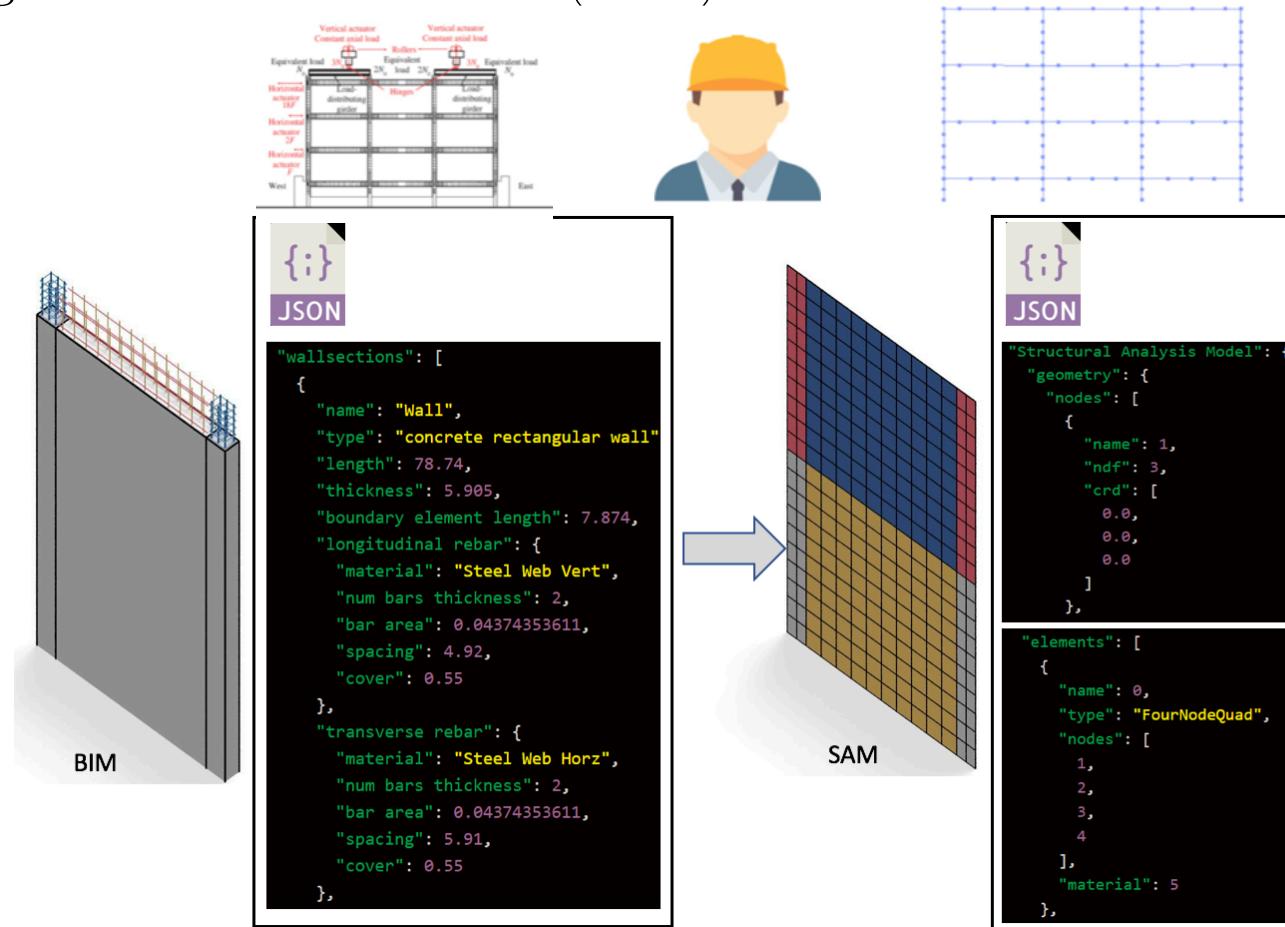
TF-slim is lightweight high-level API of TensorFlow for defining,
training and evaluating complex models.

AI & Computer Vision for Nat Hazards

SimCenter AI projects

BIM2SAM (<https://github.com/charlesxwang/BIM2SAM.AI>)

Converting Building Information Models (BIM) to Structural Analysis Models (SAM)



AI & Computer Vision for Nat Hazards

SimCenter AI projects

Soft story building detection

Detecting soft story building that are vulnerable to earthquakes using convolutional neural nets (CNN)



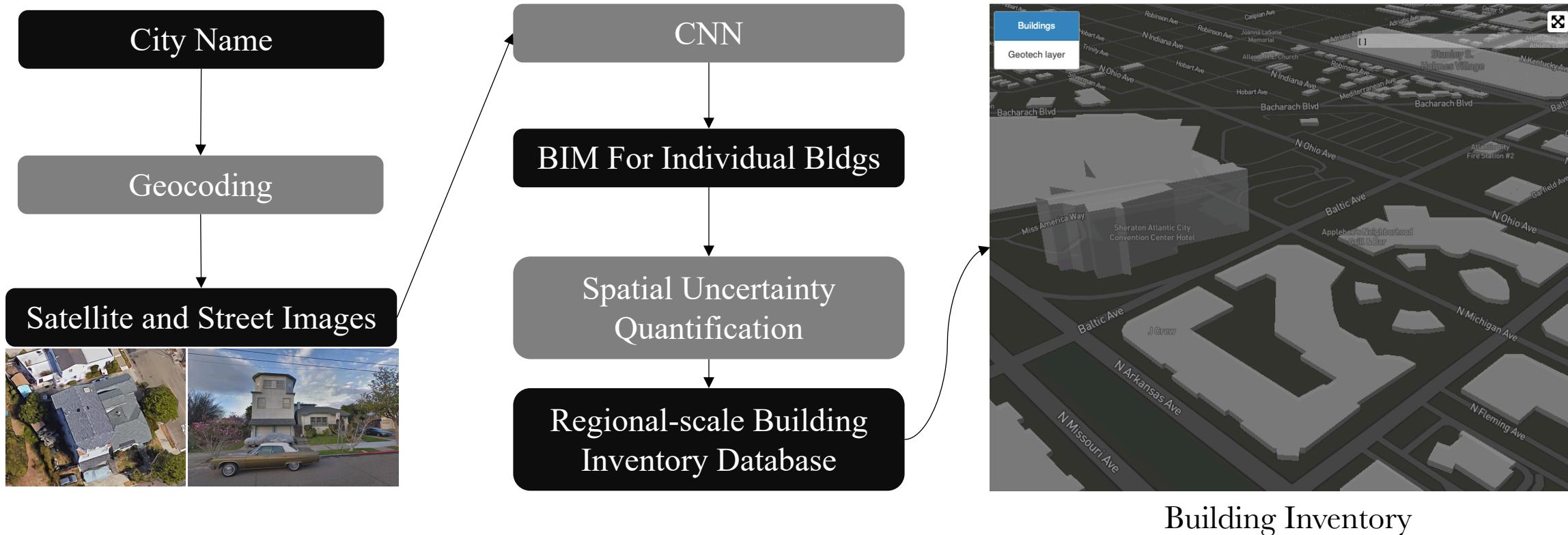
CNN
→

Soft story building
Not soft story building

AI & Computer Vision for Nat Hazards

SimCenter AI projects

Regional building information model procurement (<https://github.com/charlesxwang/BIM.AI>)

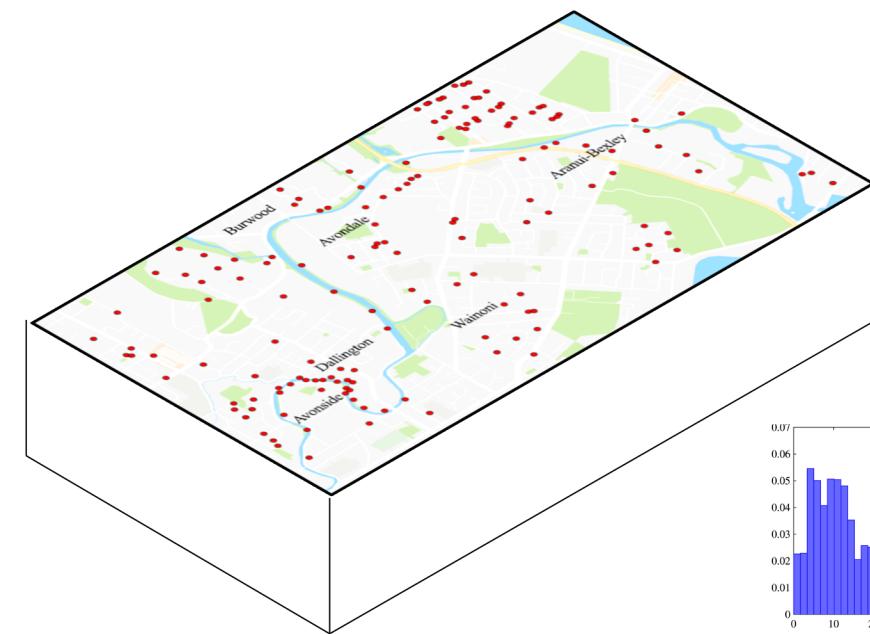


AI-Based Pipeline for Regional Building Inventory Procurement

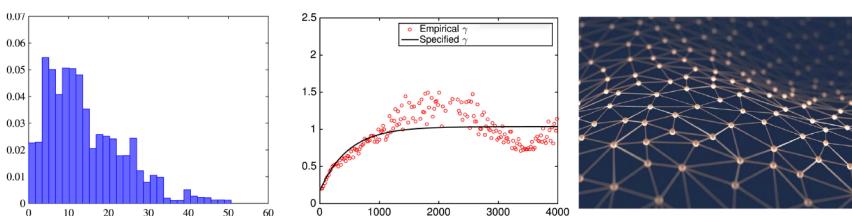
AI & Computer Vision for Nat Hazards

SimCenter AI projects

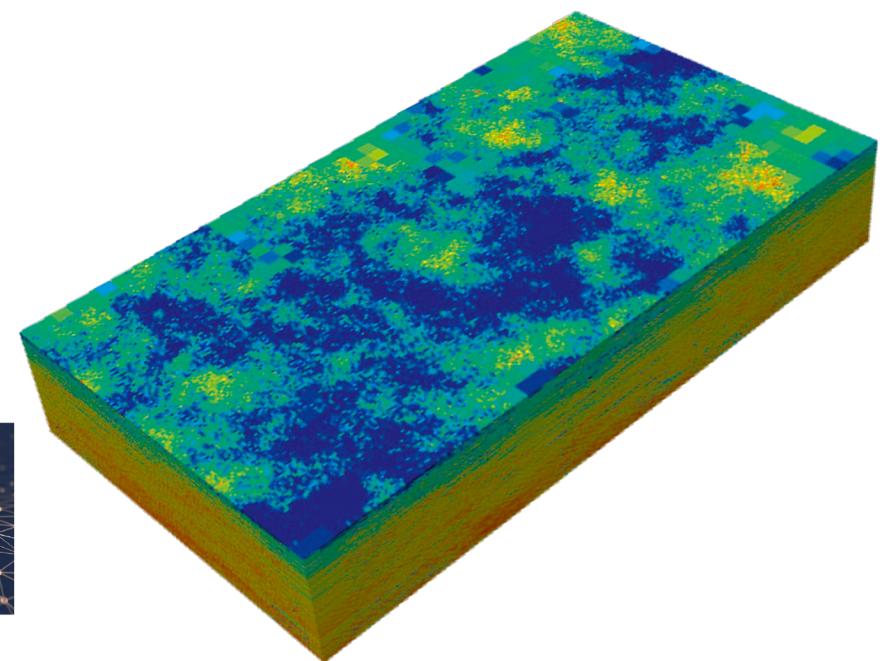
Spatial Uncertainty Quantification (<https://github.com/charlesxwang/SURF>)



Scattered Raw Data



Statistical & Spatial Structure Analysis

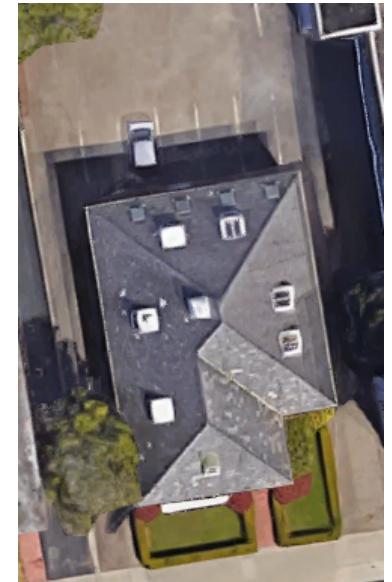
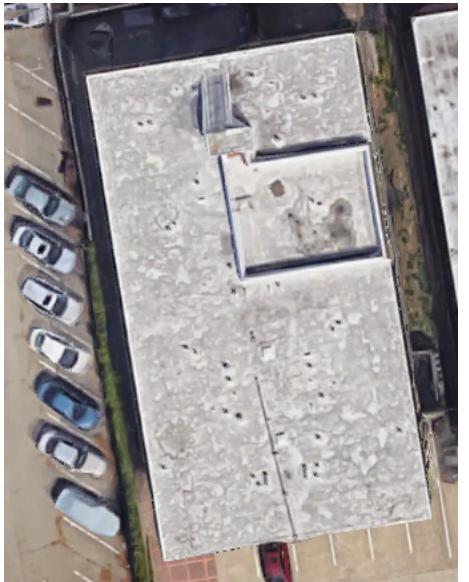


Random Field Realization

Hands on example 1. Image classification

AI & Computer Vision for Nat Hazards

Image classification: roof type



Flat



Gabled



Hipped

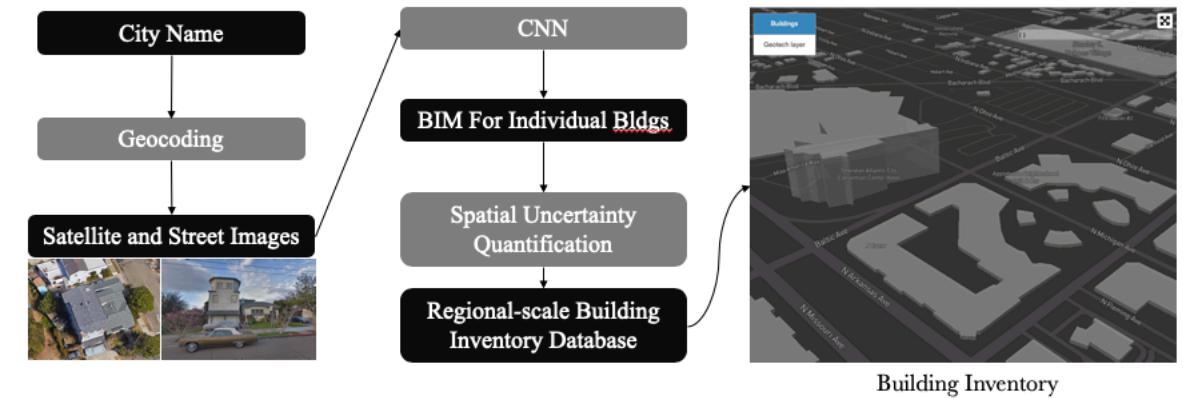
AI & Computer Vision for Nat Hazards

Image classification: roof type

Regional building information model procurement (<https://github.com/charlesxwang/BIM.AI>)

Get the code and prepare the environment:

```
cd to-the-dir-you-like  
git clone https://github.com/charlesxwang/BIM.AI  
cd BIM.AI/src  
pip install -r requirements.txt
```



AI & Computer Vision for Nat Hazards

Image classification: roof type

a. Download the images

```
mkdir training/roof/tmp/  
cd training/roof/tmp/  
wget -v -O roof.zip https://berkeley.box.com/shared/static/x2sslz2scsut6yo1ro2x5gvun2tumj6d.zip  
unzip roof.zip
```

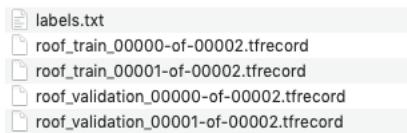
AI & Computer Vision for Nat Hazards

Image classification: roof type

b. Images => TFRecords

```
cd ../../training/roof/1_create_tfrecords  
python3 create_tfrecord.py --dataset_dir=dir/containing/roofImages --tfrecord_filename=roof
```

You get these:



labels.txt

```
0:flat  
1:gabled  
2:hipped
```

AI & Computer Vision for Nat Hazards

Image classification: roof type

c. Train a CNN (inception-v3) using Tensorflow-slim

```
cd ..../2_train  
sh finetune_inception_v3_on_roof_train.sh
```

The training takes a long time on laptops. If you don't want to run the training process, we have a CNN trained on TACC and can be

downloaded <https://berkeley.box.com/shared/static/awyycc22sjwknn9xg3p7wru4v5zwnlkjp.zip>. Put the downloaded file inside src/training/roof/tmp/roof-traindir/ and unzip it.
You can use Ctrl+c in the terminal to stop the training process.

AI & Computer Vision for Nat Hazards

Image classification: roof type

d. Test the trained CNN

```
sh finetune_inception_v3_on_roof_eval.sh
```

AI & Computer Vision for Nat Hazards

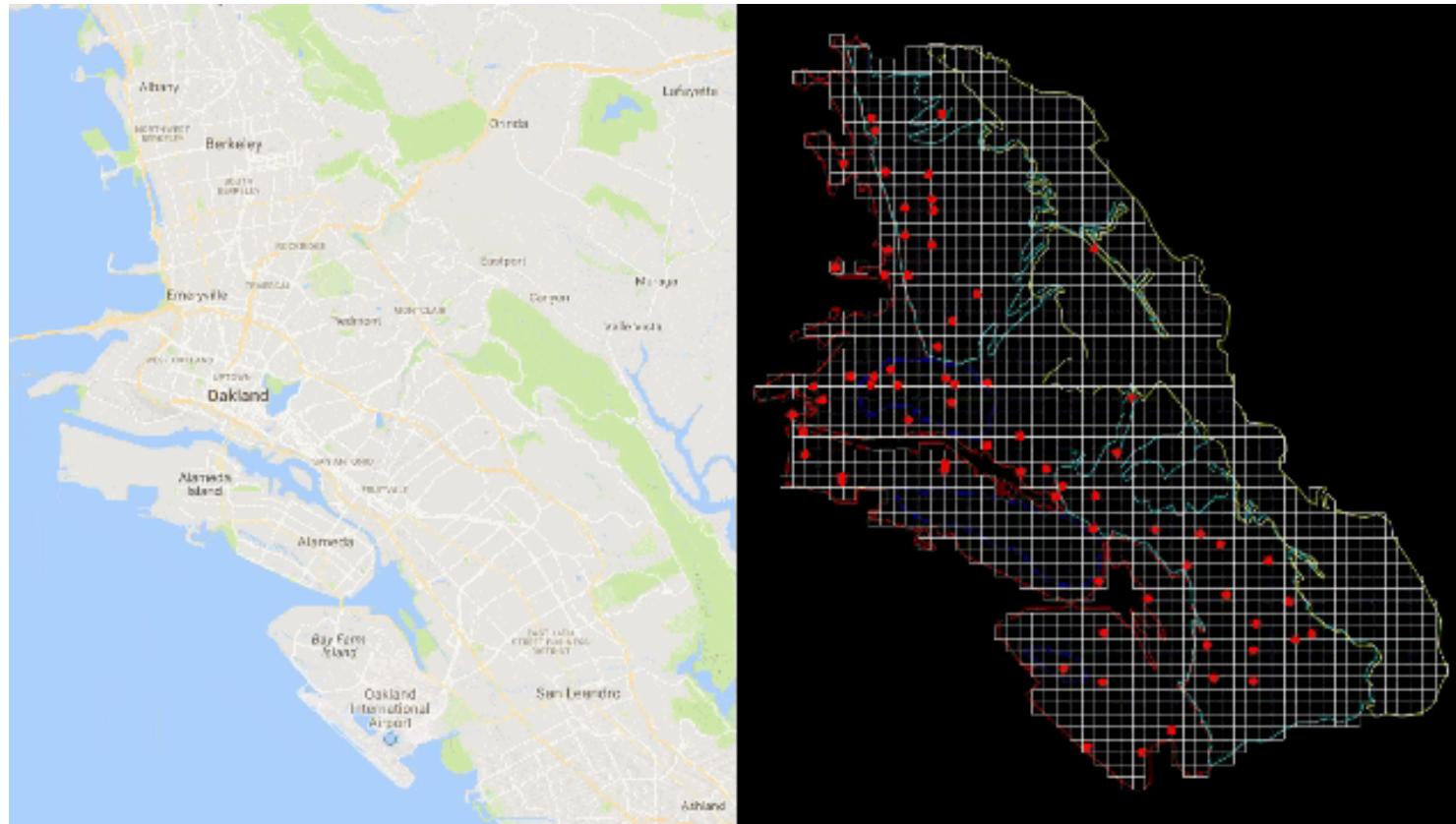
Image classification: roof type

e. Practice: try to set up a training process for soft-story building classification

Data is here: <https://berkeley.box.com/shared/static/0ejgwp56aux32opa4mwfawj0vglxnact.zip>

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

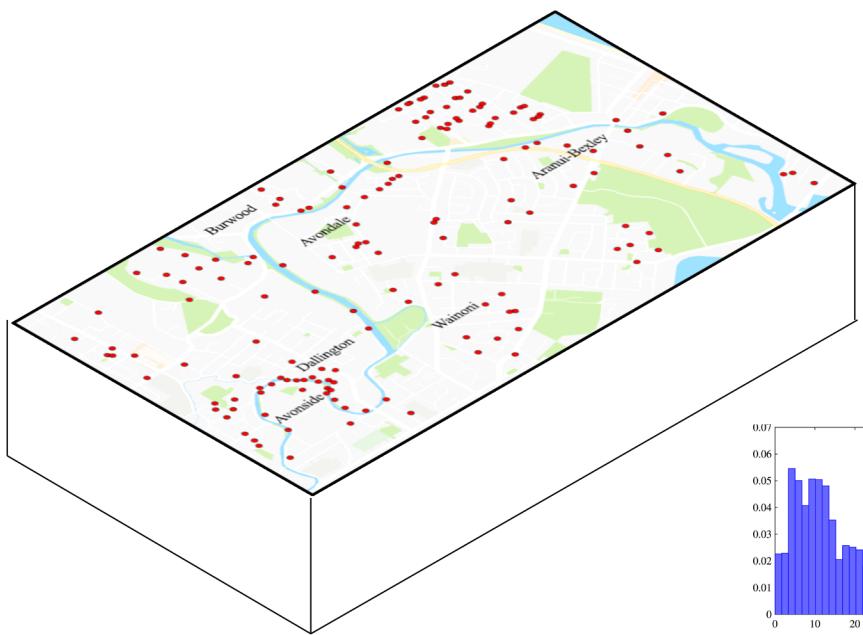


A framework for spatial uncertainty quantification

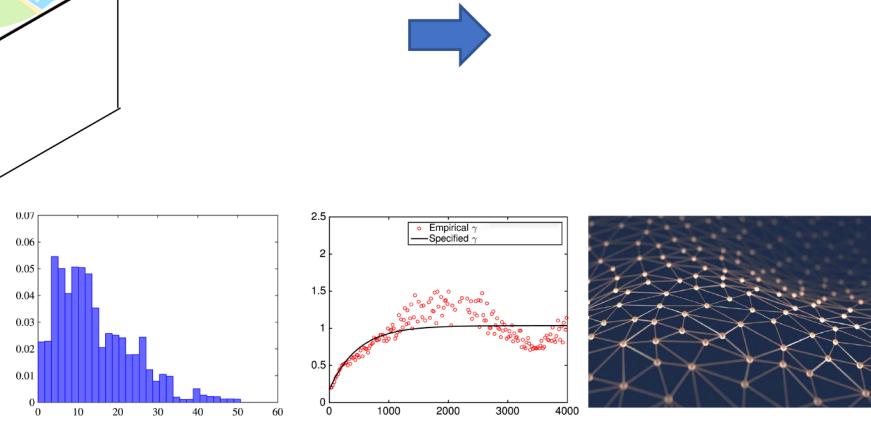
<https://github.com/charlesxwang/SURF>

Enhancing Data with AI

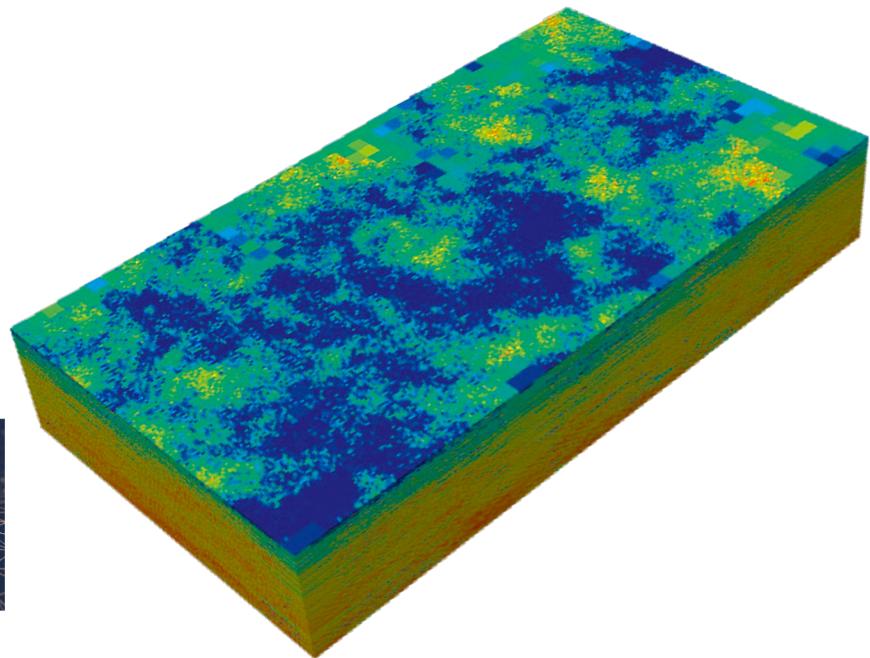
Spatial Uncertainty Research Framework (SURF)



Scattered Raw Data



Statistical & Spatial Structure Analysis



Random Field Realization

<https://github.com/charlesxwang/SURF>

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

Random field engine

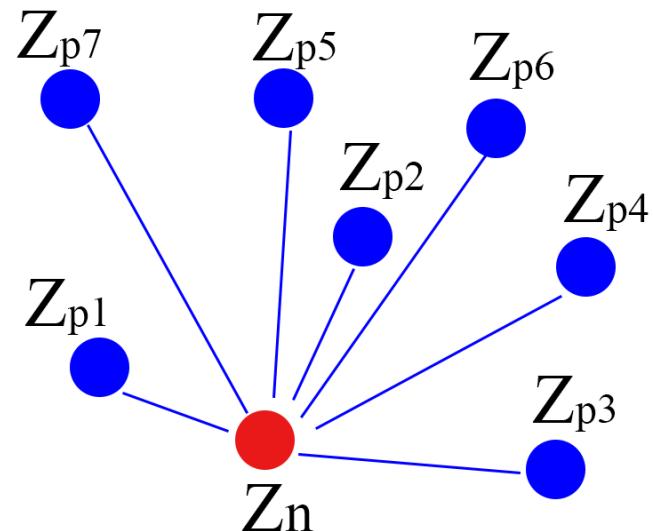
$$Z \sim N(\mu, \Sigma) \xrightarrow{\text{partition}} \begin{bmatrix} Z_n \\ Z_p \end{bmatrix} \sim N \left(\begin{bmatrix} \mu_n \\ \mu_p \end{bmatrix}, \begin{bmatrix} \Sigma_{nn} & \Sigma_{np} \\ \Sigma_{pn} & \Sigma_{pp} \end{bmatrix} \right)$$

multivariate Gaussian distribution

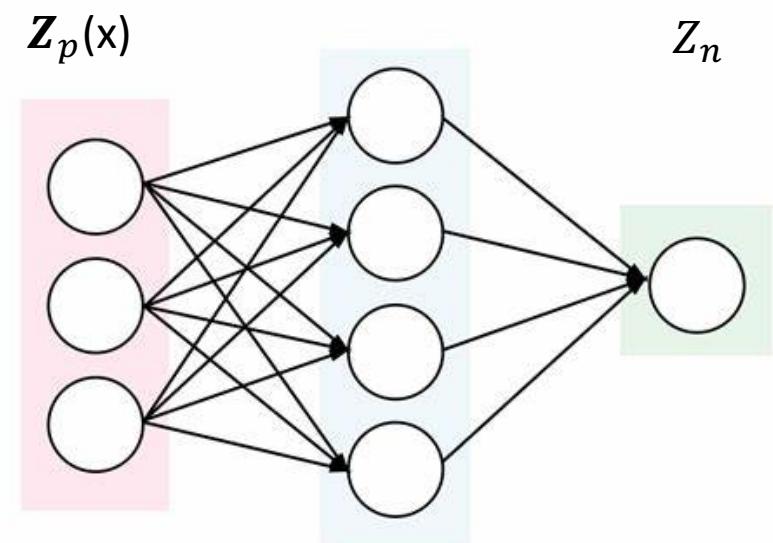
conditional distribution of RV Z_n :

$$Z_n | Z_p \sim N(\mu_{Z_n|Z_p}, \Sigma_{Z_n|Z_p})$$

C. Wang and Q. Chen, A hybrid geotechnical and geologic data-based framework for multiscale regional liquefaction hazard mapping, *Géotechnique*, 68(7):614-625, doi:10.1680/igeot.17.P.074, 2018.



Machine learning engine



$$Z_n = f(Z_p(x))$$

Hands on example 2. Data enhancement (build your own neural nets)

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

We have this BIM.geojson, <https://berkeley.box.com/shared/static/mmcjsjfkfl6q4it83cv16xzku1jdogz4.geojson> which contains BIM for each building.

But the information of some parts of some buildings is missing.

```
{  
    "type": "Feature",  
    "id": "8461",  
    "properties": {  
        "id": "8461",  
        "lat": 39.3716807,  
        "lon": -74.4513949,  
        "address": "1619 COLUMBIA AVE, ATLANTIC CITY, NJ",  
        "stories": 2,  
        "yearBuilt": 1979,  
        "occupancy": "Residential",  
        "structureType": "Frame",  
        "buildingDescription": "2SF",  
        "city": "ATLANTIC CITY CITY"  
    },  
    "geometry": {  
        "type": "Polygon",  
        "coordinates": [  
            [  
                [-74.451353, 39.371717],  
                [-74.451493, 39.371755],  
                [-74.451526, 39.37168],  
                [-74.451386, 39.371643],  
                [-74.451353, 39.371717]  
            ]  
        ]  
    }  
}
```

→ For example, yearBuilt is missing for some buildings

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

Construct the input vector $\mathbf{Z}_p(x)$

Zn_lon

Zn_lat

Distance_Zn_Zp1

Value_Zp1

Distance_Zn_Zp2

Value_Zp2

Distance_Zn_Zp3

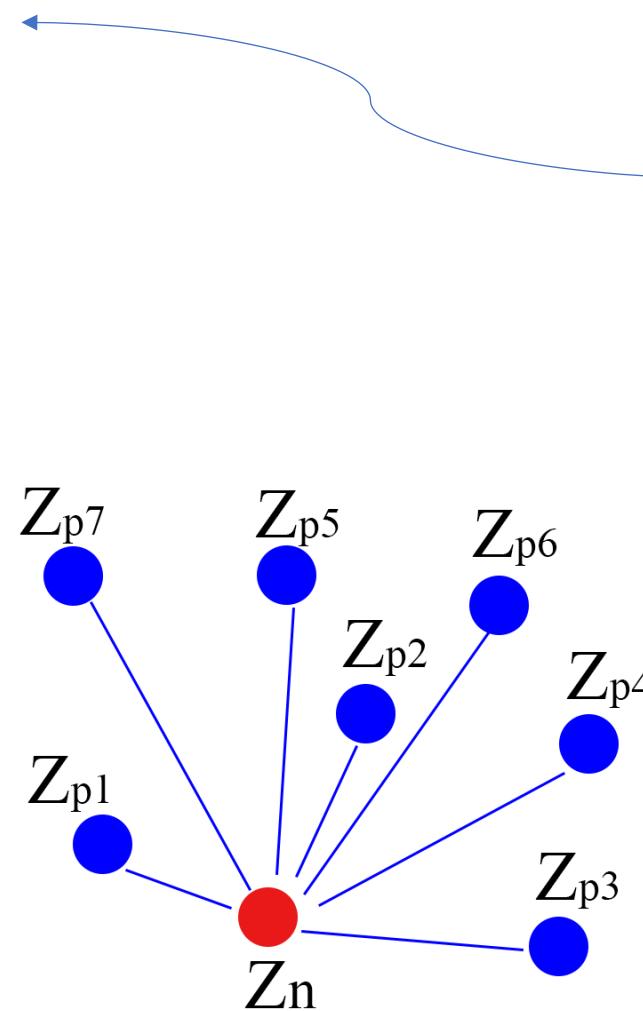
Value_Zp3

.

.

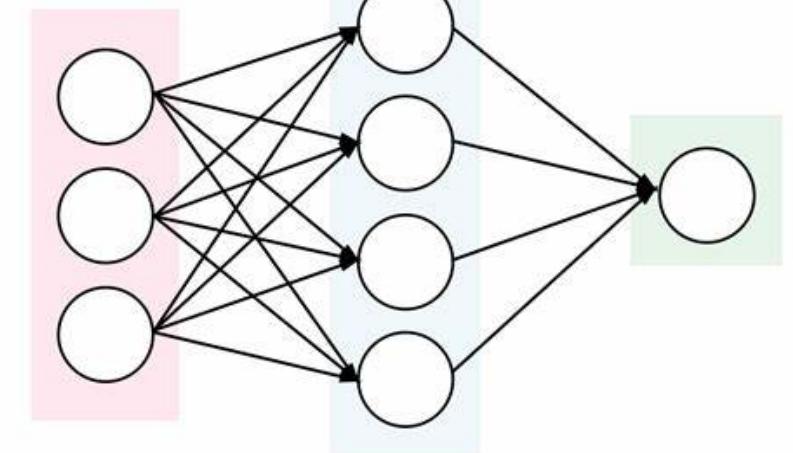
.

.



Machine learning engine

$\mathbf{Z}_p(x)$



$$Z_n = f(\mathbf{Z}_p(x))$$

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

Continuous

Put the following into SURF-bootcamp.py:

```
from surf.NN import SpatialNeuroNet
```

```
# construct dada (columns are: lon,lat,value) as numpy.ndarray # construct dada (columns are: lon,lat,value) as numpy.ndarray
```

```
nn = SpatialNeuroNet(rawData = data, numNei = 2)
nn.build_model()
nn.train()
nn.test()
```

Categorical

```
from surf.NN import SpatialNeuroNet
```

```
nn = SpatialNeuroNet(rawData = data, numNei = 2)
nn.build_classification_model(numofTypes)
nn.train_classification_model()
nn.test_classification_model()
```

Then issue this in your terminal under SURF/
`python SURF-bootcamp.py`

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

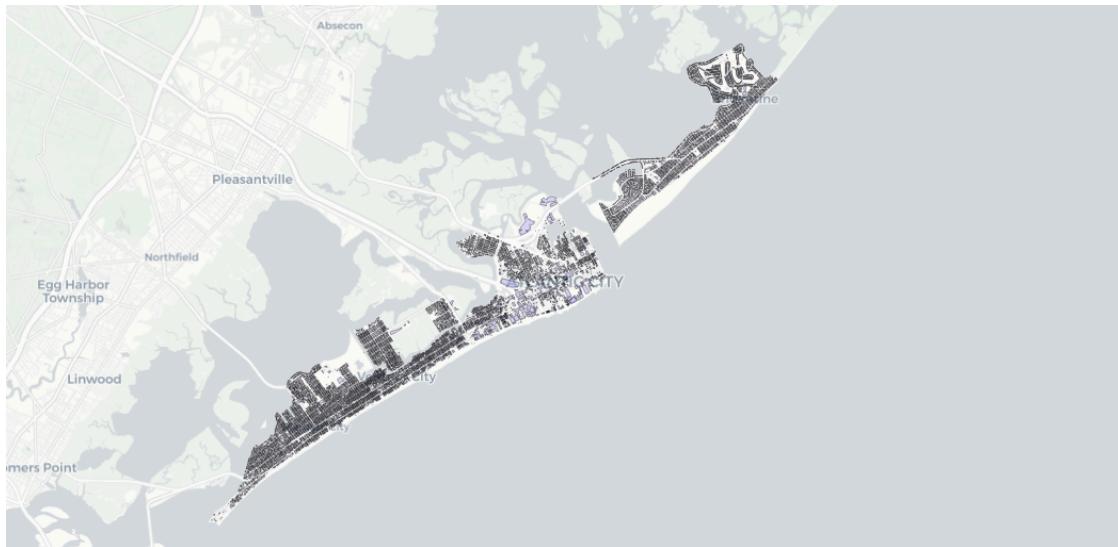
Architecture (SURF/surf/NN.py)

```
model = keras.Sequential([
    layers.Dense(256, activation=tf.nn.relu, input_shape=[len(self.train_dataset.T)]),
    layers.Dense(64, activation=tf.nn.relu),
    layers.Dense(64, activation=tf.nn.relu),
    layers.Dense(64, activation=tf.nn.relu),
    layers.Dense(1)
])
```

Enhancing Data with AI

Spatial Uncertainty Research Framework (SURF)

Results



Feature	Value
▼ Atlantic_Cities_BIM OGRGeoJSON Polygon	
▼ buildingDescription	F3S
► (Derived)	
► (Actions)	
city	ATLANTIC CITY CITY
structureType	Frame
lon	-74.4601876
roofType	hipped
occupancy	Residential
stories	3
buildingDescription	F3S
address	55 S RALEIGH AVE, ATLANTIC CITY,NJ
lat	39.3474871
yearBuilt	1920
id	964101