

Applying Machine Learning to Jet Noise Prediction

Jonny Dowdall

Acoustics Technical Working Group Meeting
NASA Glenn Research Center
October, 2018

About me

- Intern at NASA Glenn Research Center
 - Scientific Computing and Visualization Team
- Student at Michigan State University
 - B.S. Computer Science
 - Pursuing M.S. Computer Science
- Two years experience in data science and machine learning

Objectives

Predictive modeling

Can we use machine learning to create accurate predictive models for sound modeling?

Data Insight

Can we learn more about the domain?

Do we need to modify our experiments?

Requirements

Hardware requirements?

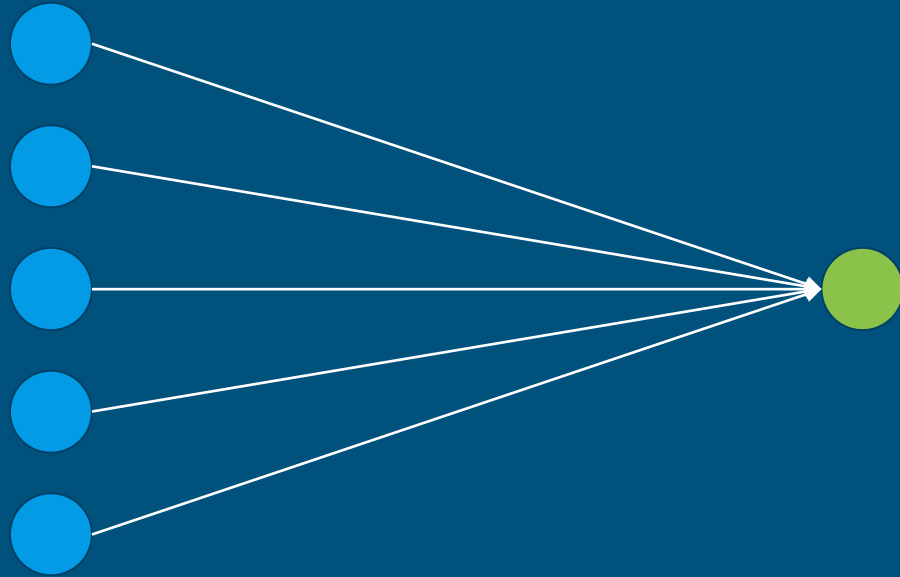
Time requirements?

Machine Learning:



A crash course.

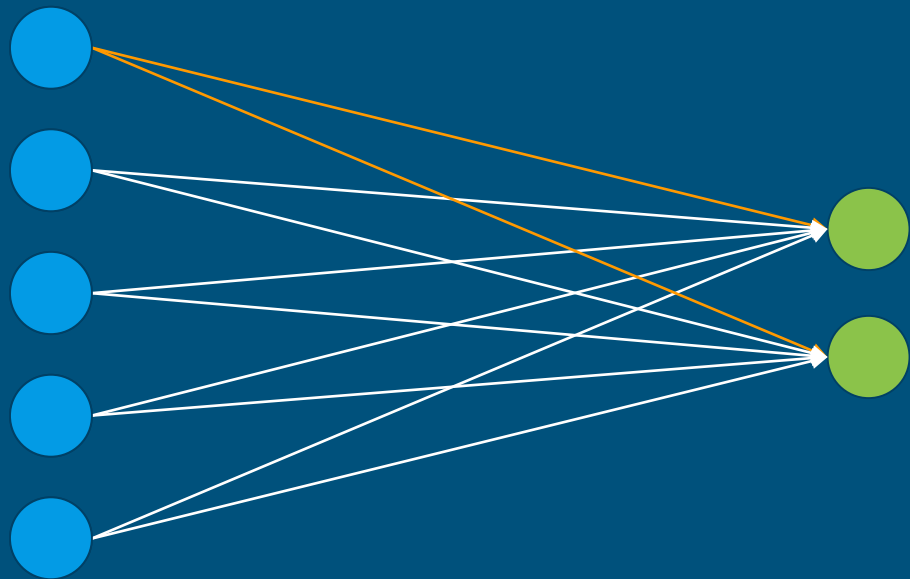
Linear Model



Output value is weighted sum of inputs (or features)



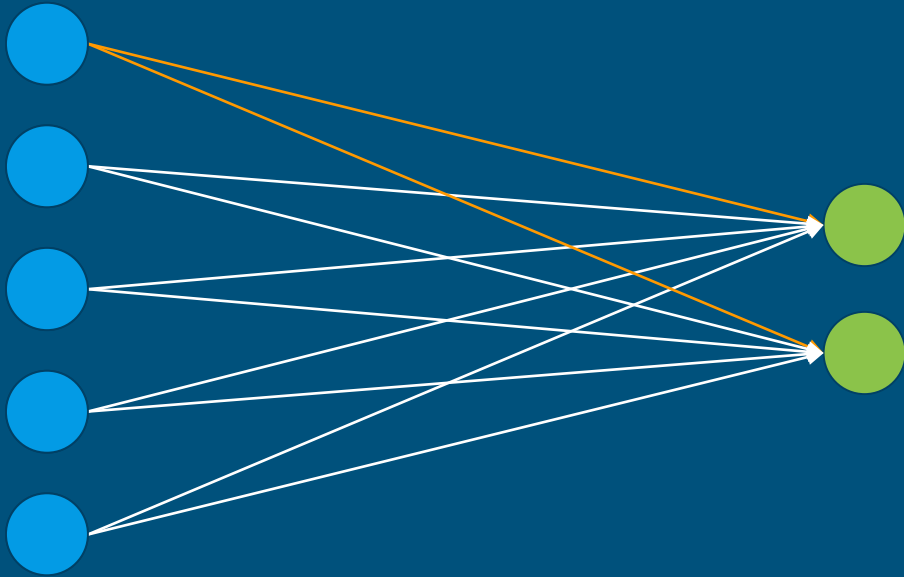
Linear Model

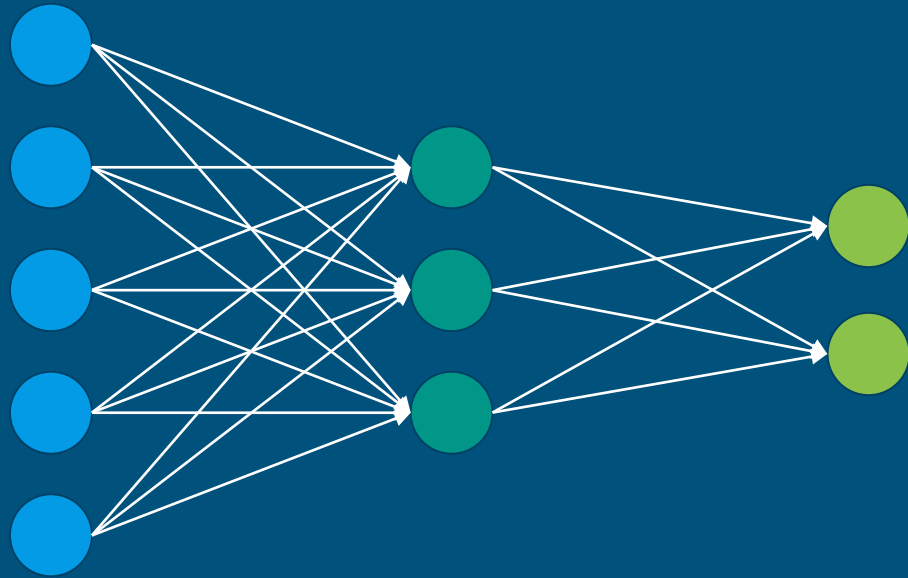


Output can be a vector (multiple values). Each **feature** will have a different weight, or **parameter**, for every predicted **output** value.

Linear Model

Often **output** cannot easily be modeled by a linear combination of **features**.



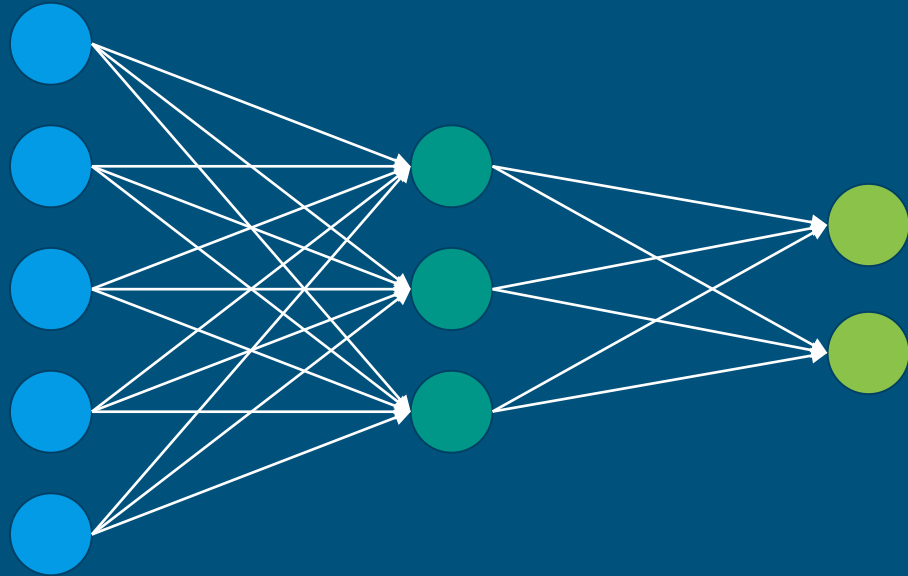


Neural Network

Add **hidden layer** which applies some non-linear function (**activation**) to the output of the first linear system.

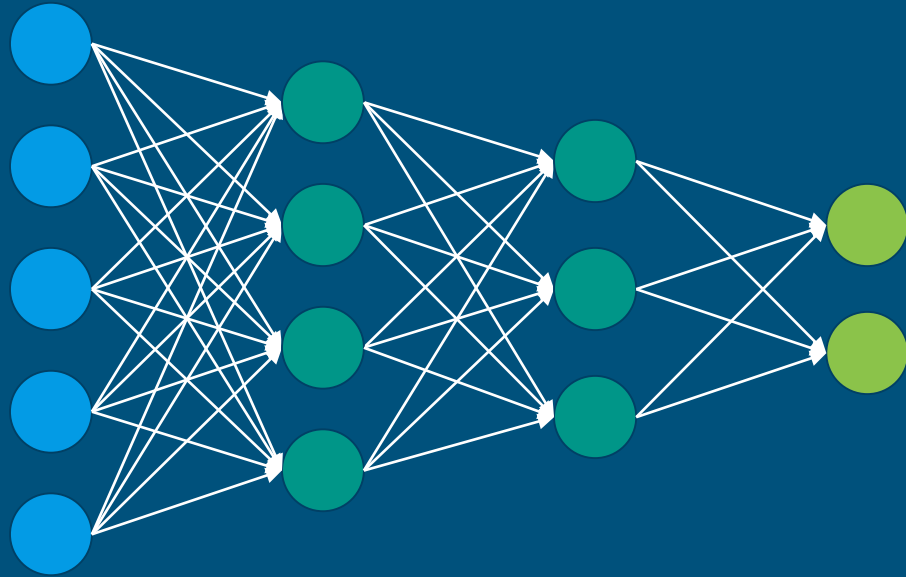
Many options for non-linear **activation**.

Neural Network



Output is now the weighted sum of the **hidden** layer activations.

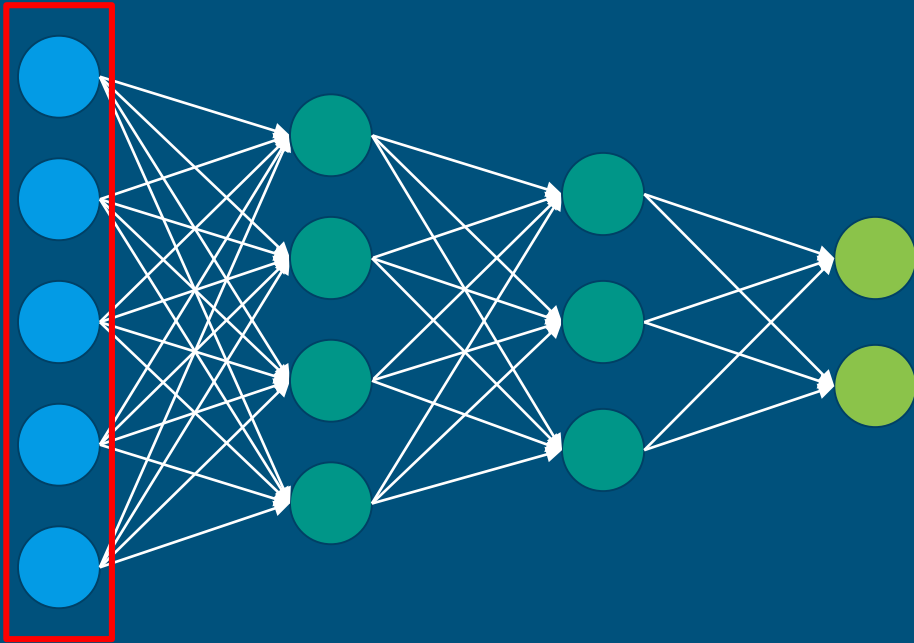
Neural Network



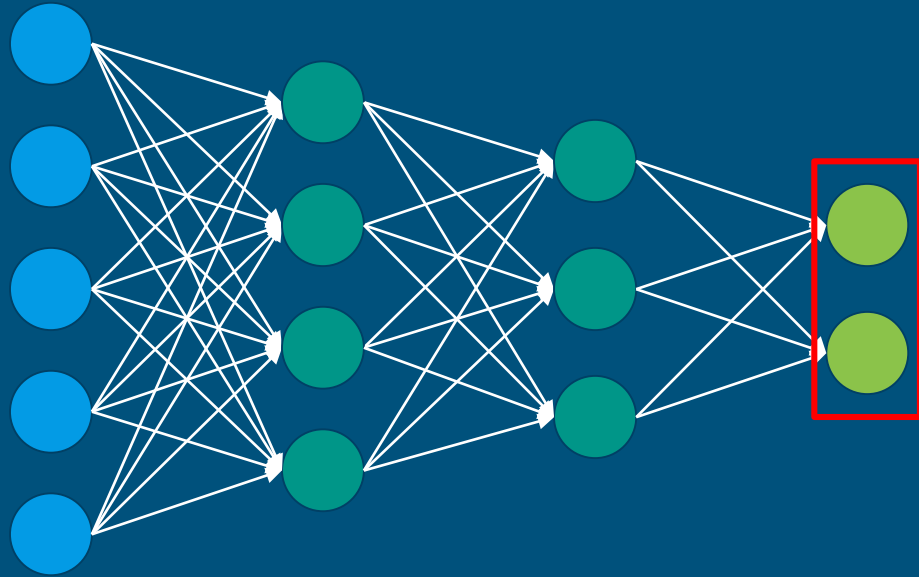
Each **hidden** layer abstracts things a little more.

Neural Network

Input shape is defined by your data.

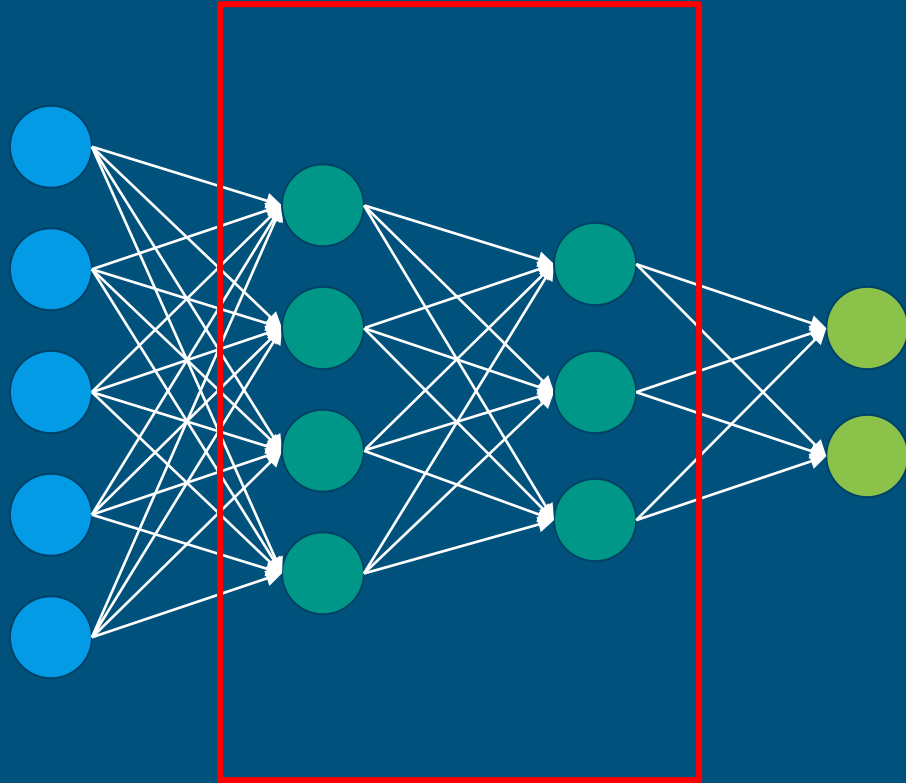


Neural Network



Output shape is defined by your data.



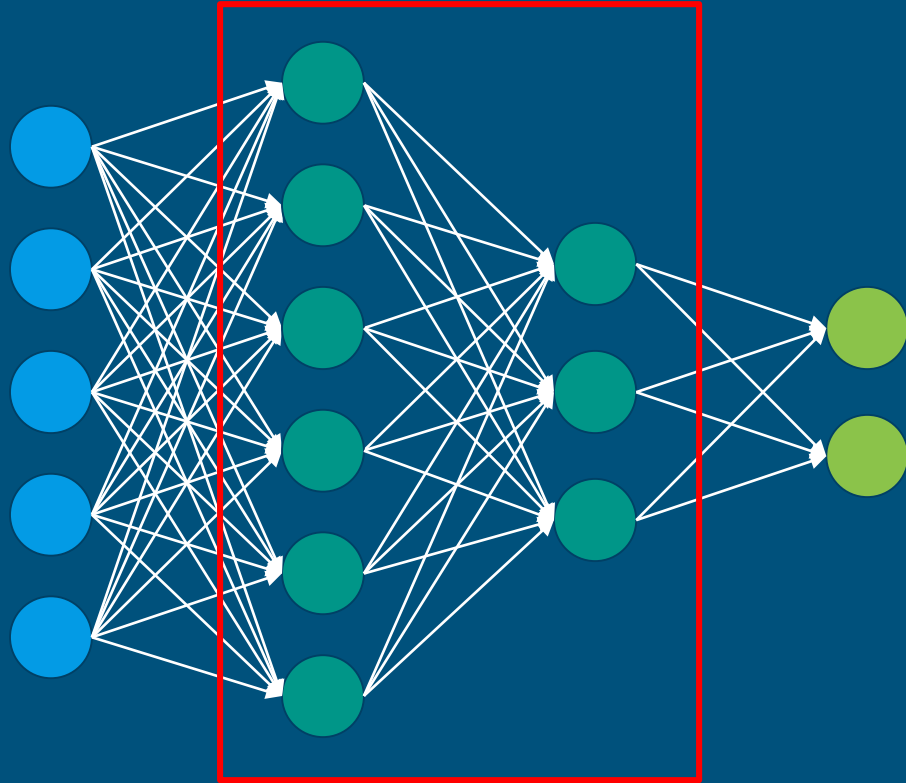


Neural Network

Hidden layers..?

Up to you. Typical to start with something in between your input and output sizes.



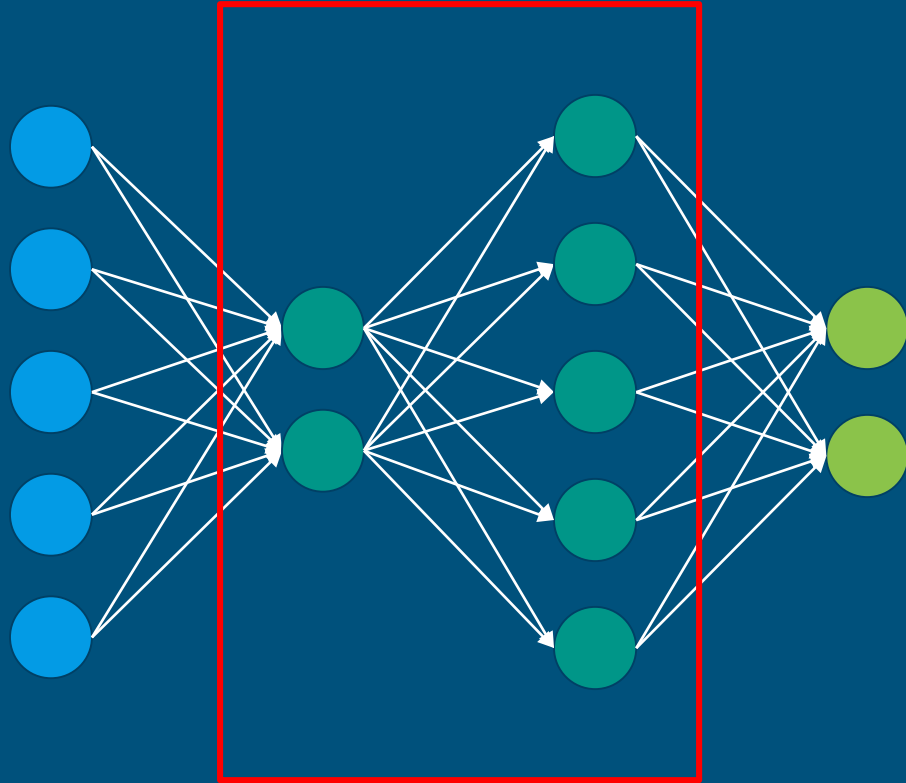


Neural Network

Hidden layers..?

Up to you. Typical to start with something in between your input and output sizes.



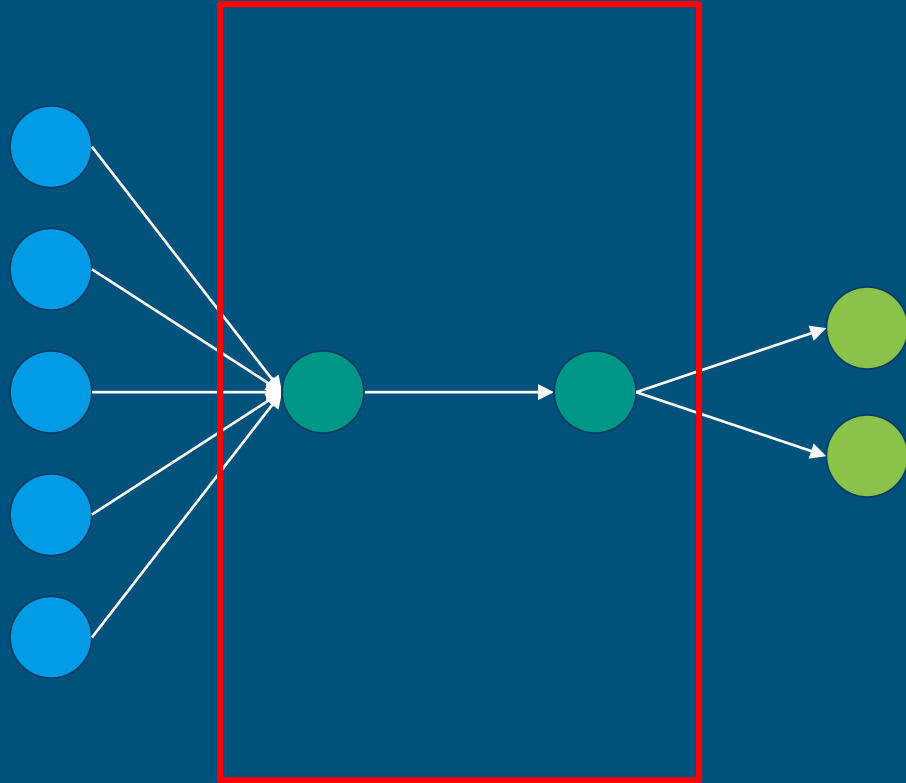


Neural Network

Hidden layers..?

Up to you. Typical to start with something in between your input and output sizes.



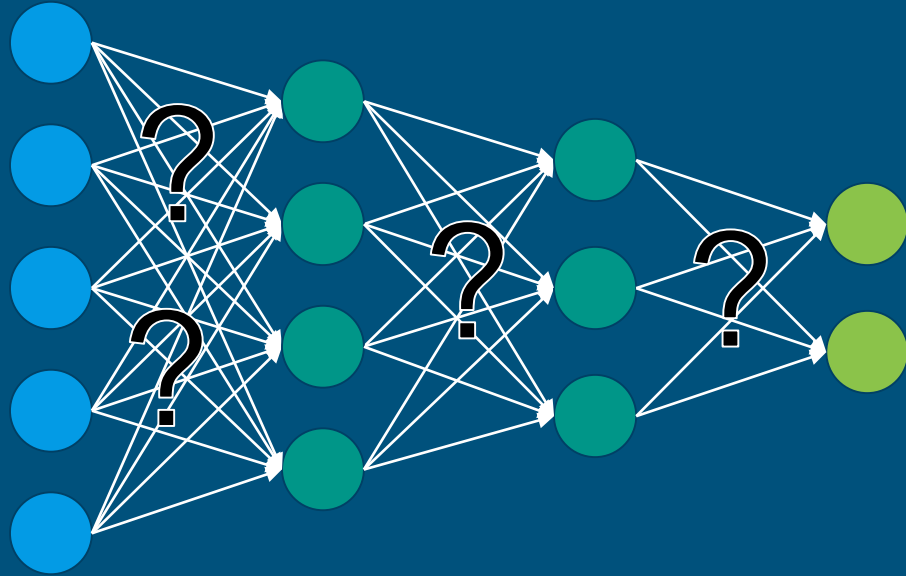


Neural Network

Hidden layers..?

Up to you. Typical to start with something in between your input and output sizes.

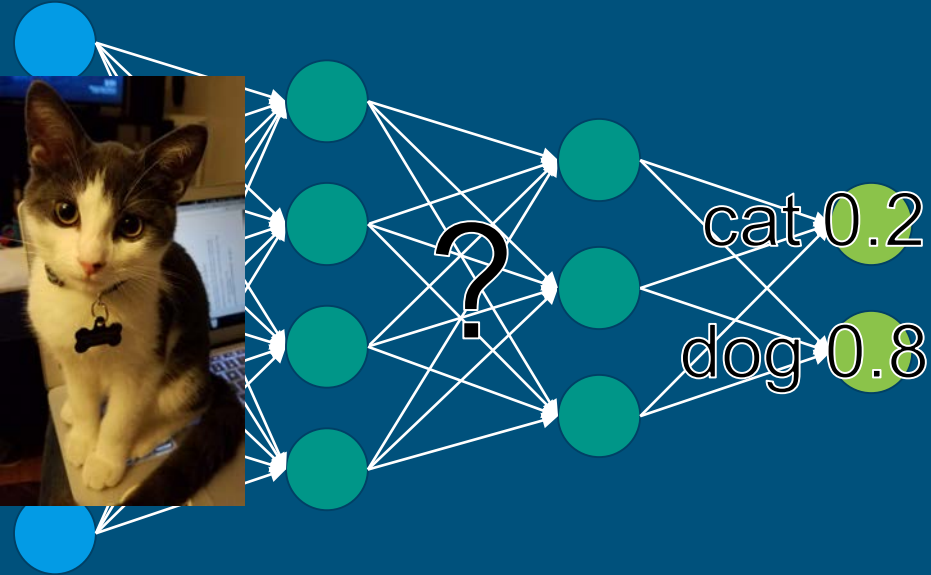
Training



Initialize random weights.

Training

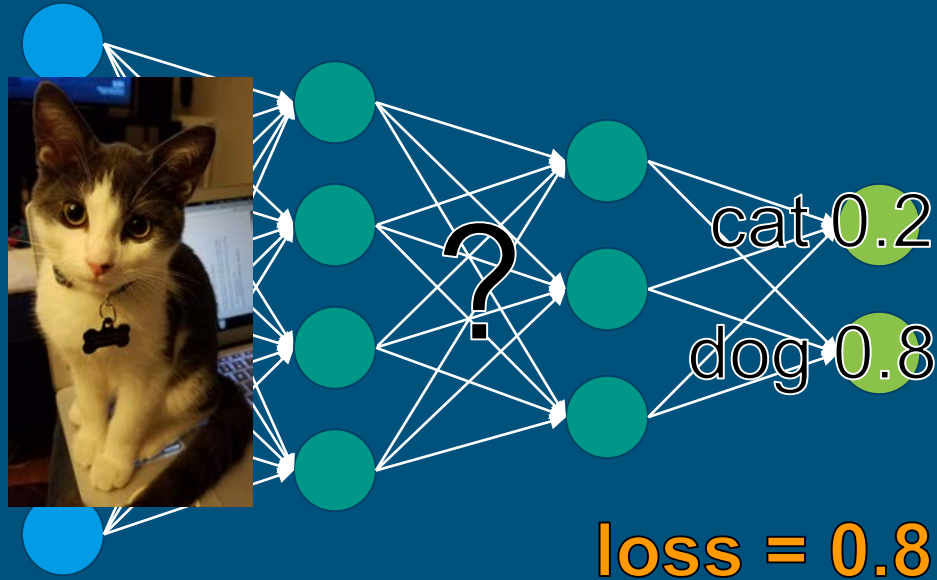
Feed network an observed sample input, compute a predicted output.



Training

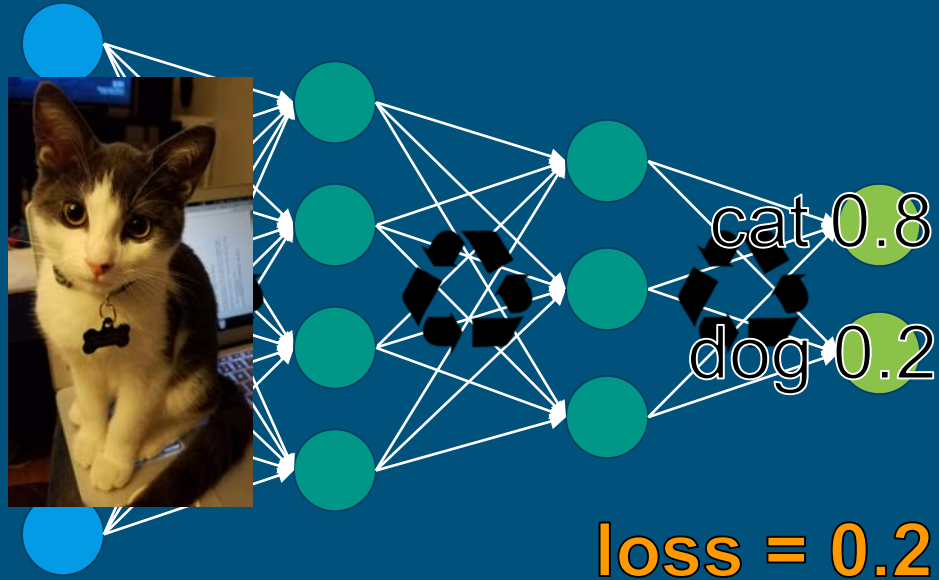
Compute some error (or **loss**) between the predicted output and the observed sample output.

Many options for loss functions.



Training

Update weights such that the error will be less when we predict this sample again.



Training vs. Validation

Training data

- Samples used to fit your parameters to your data.
- Learn the parameters which minimize loss for training set.

Validation data

- Samples used to measure your model's performance

Training

Validation

Good Poor

Poor

Good

| | |
|--|--|
| | |
| | |

Training

Validation

| | Poor | Good |
|------|--------------|------|
| Poor | Underfitting | |
| Good | | |

Training

Validation

| | Poor | Good |
|------|--------------|-------------|
| Poor | Underfitting | Overfitting |
| Good | | |

Training

Validation

| | Poor | Good |
|------|--------------|--------------|
| Poor | Underfitting | Overfitting |
| Good | | Generalizing |

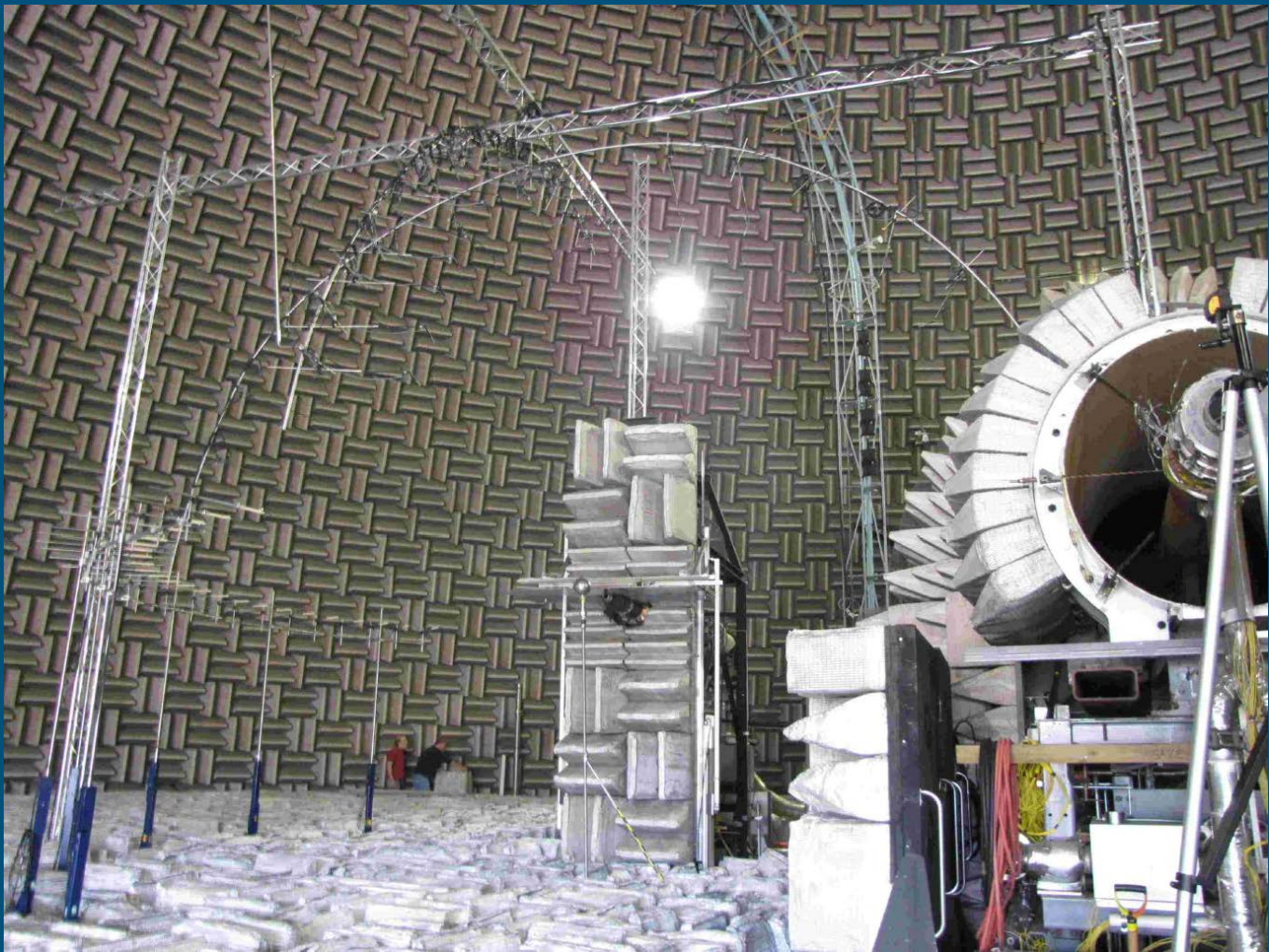
Training

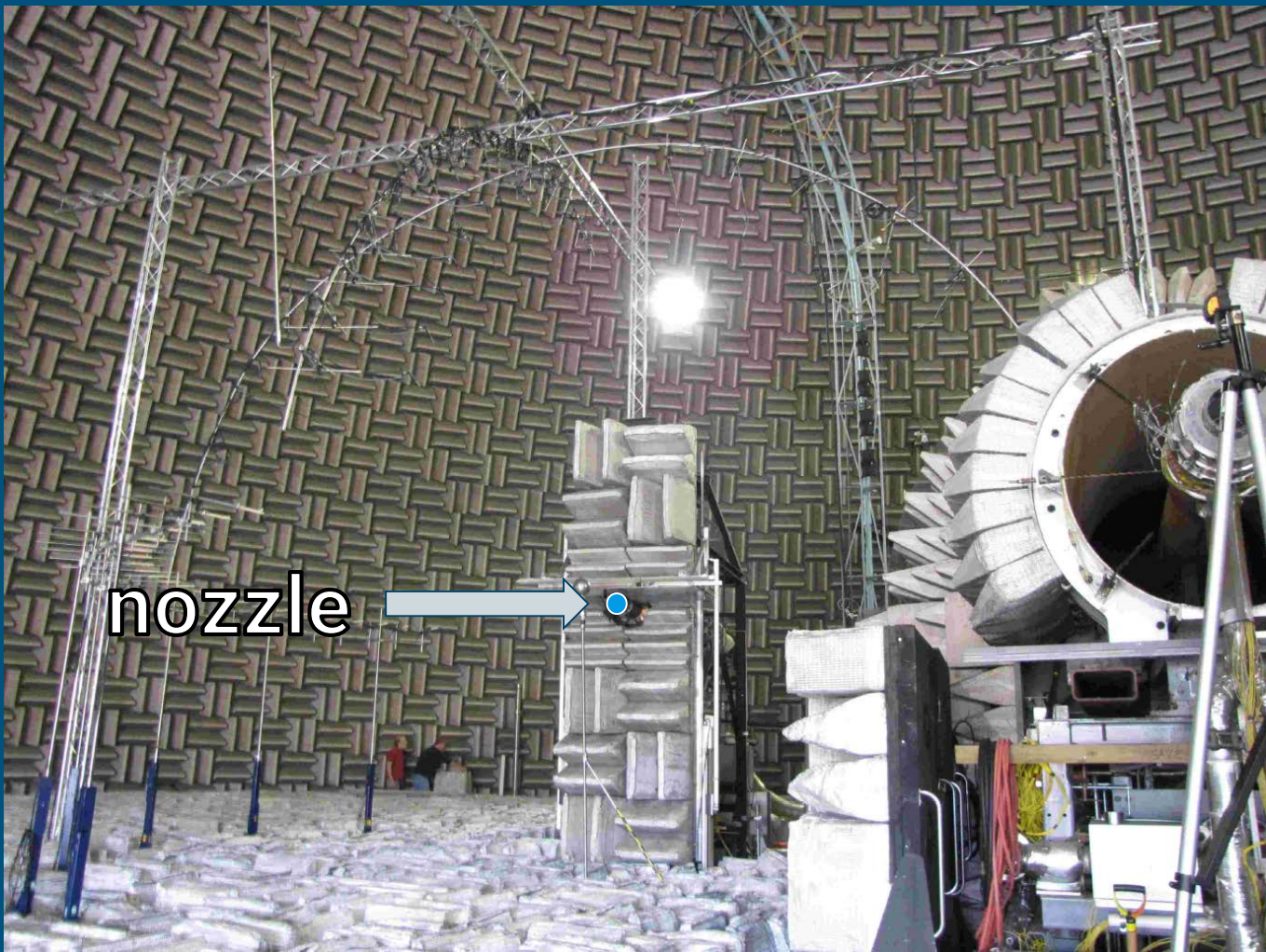
Validation

| | Poor | Good |
|------|--------------|--------------|
| Poor | Underfitting | Overfitting |
| Good | ??? | Generalizing |

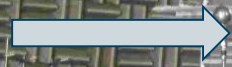
The Experiment

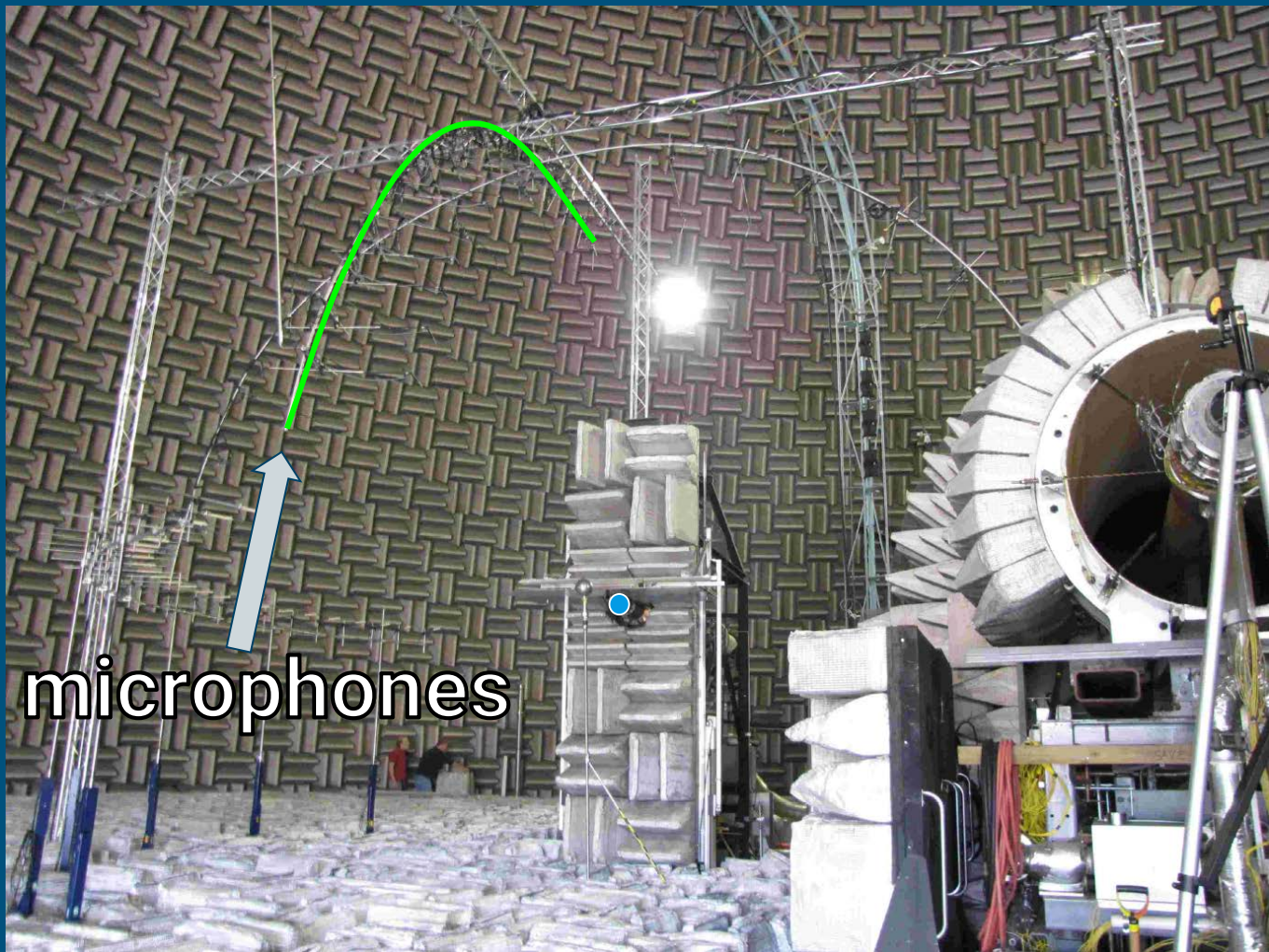




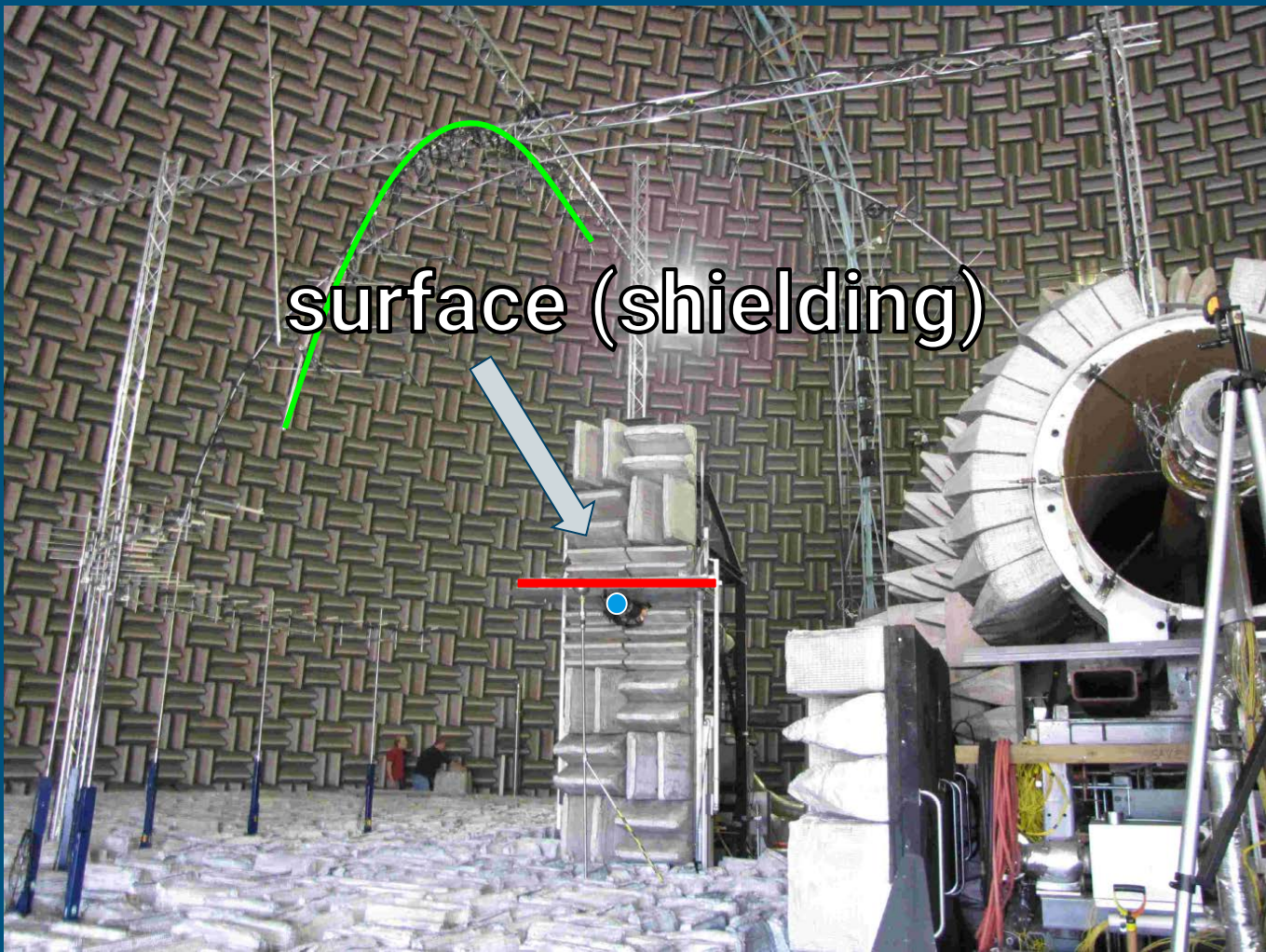


nozzle

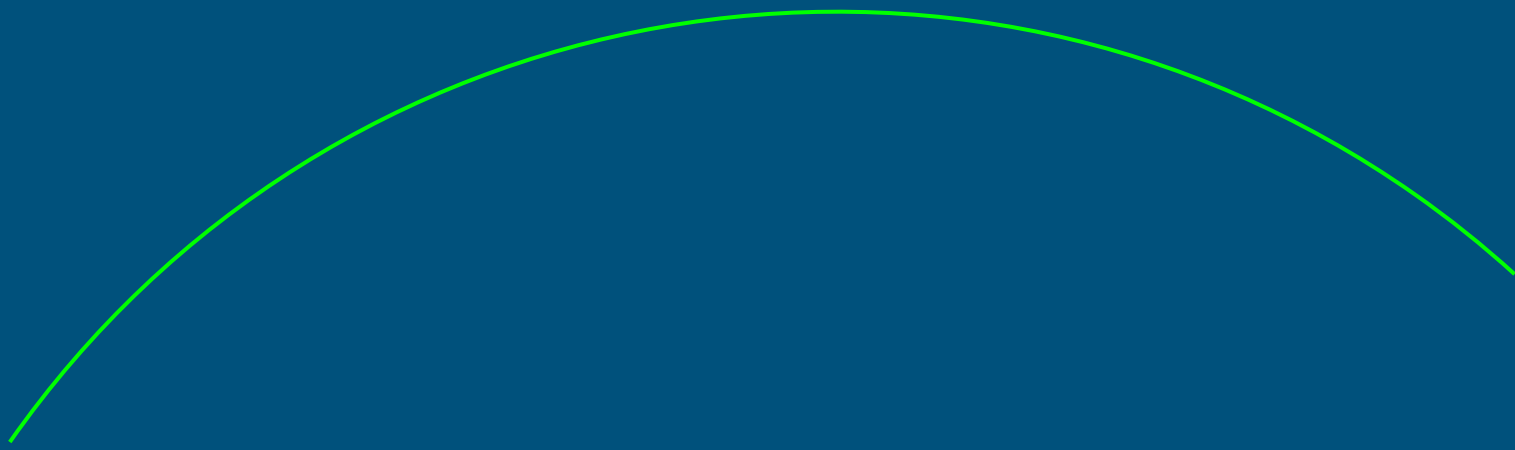


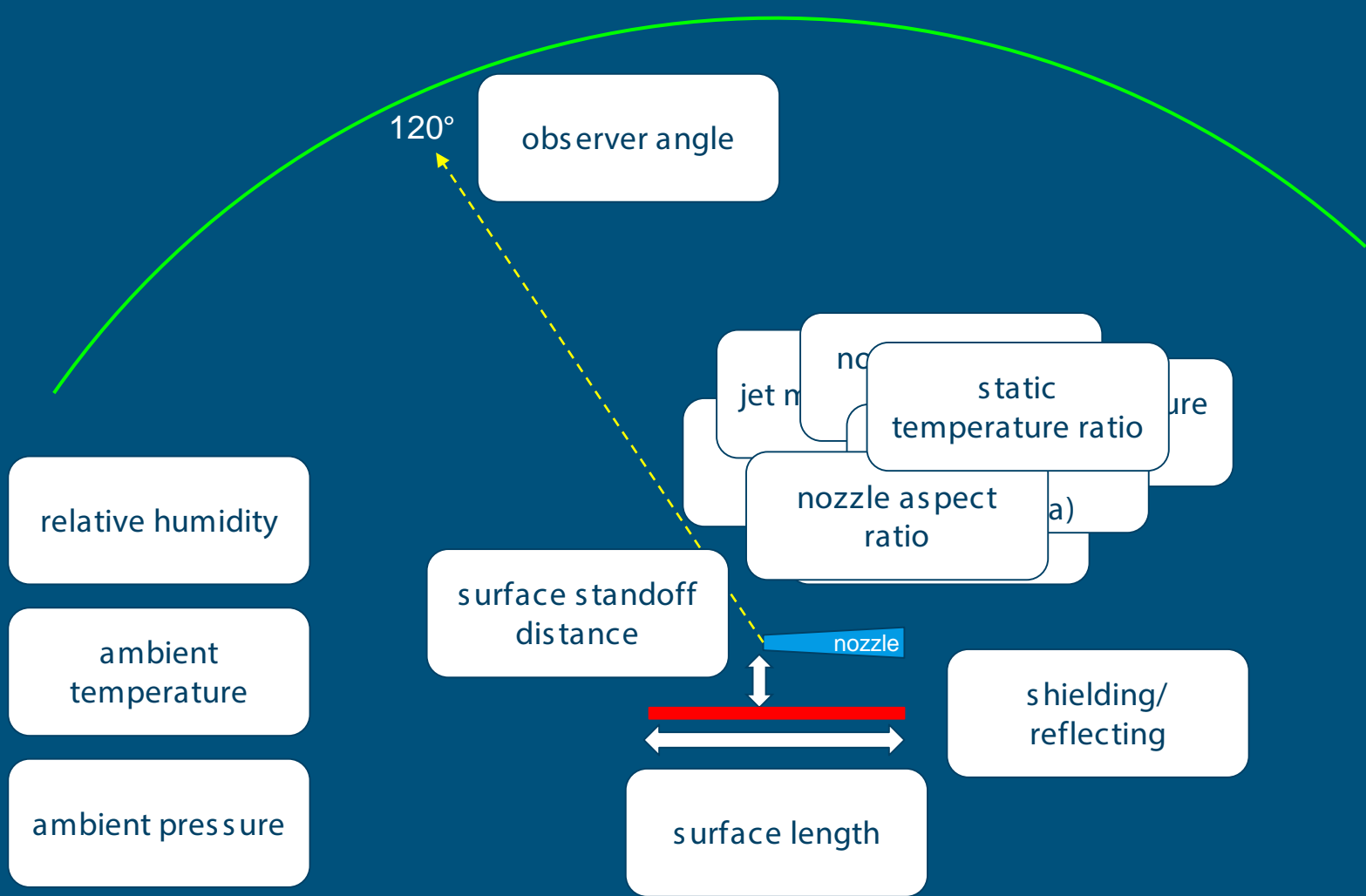


microphones

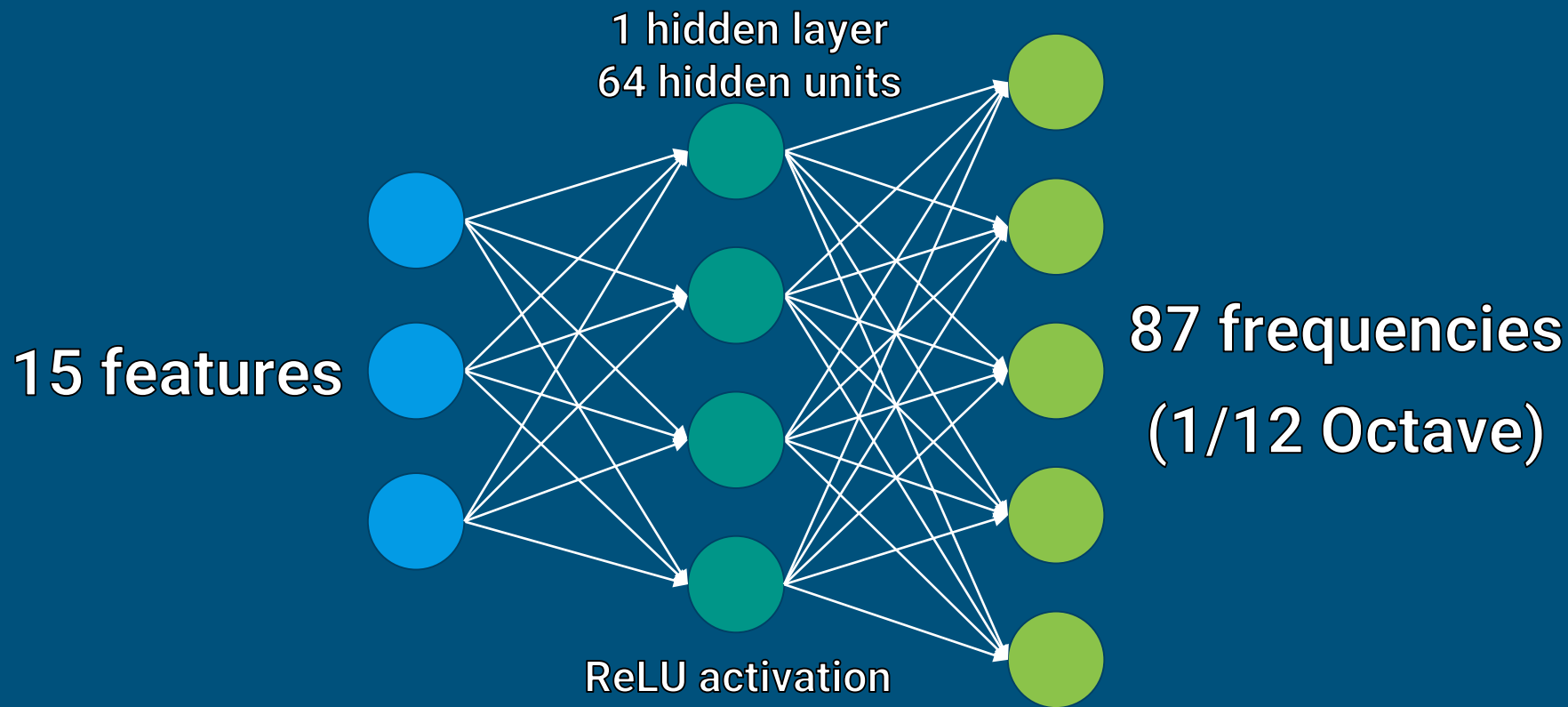


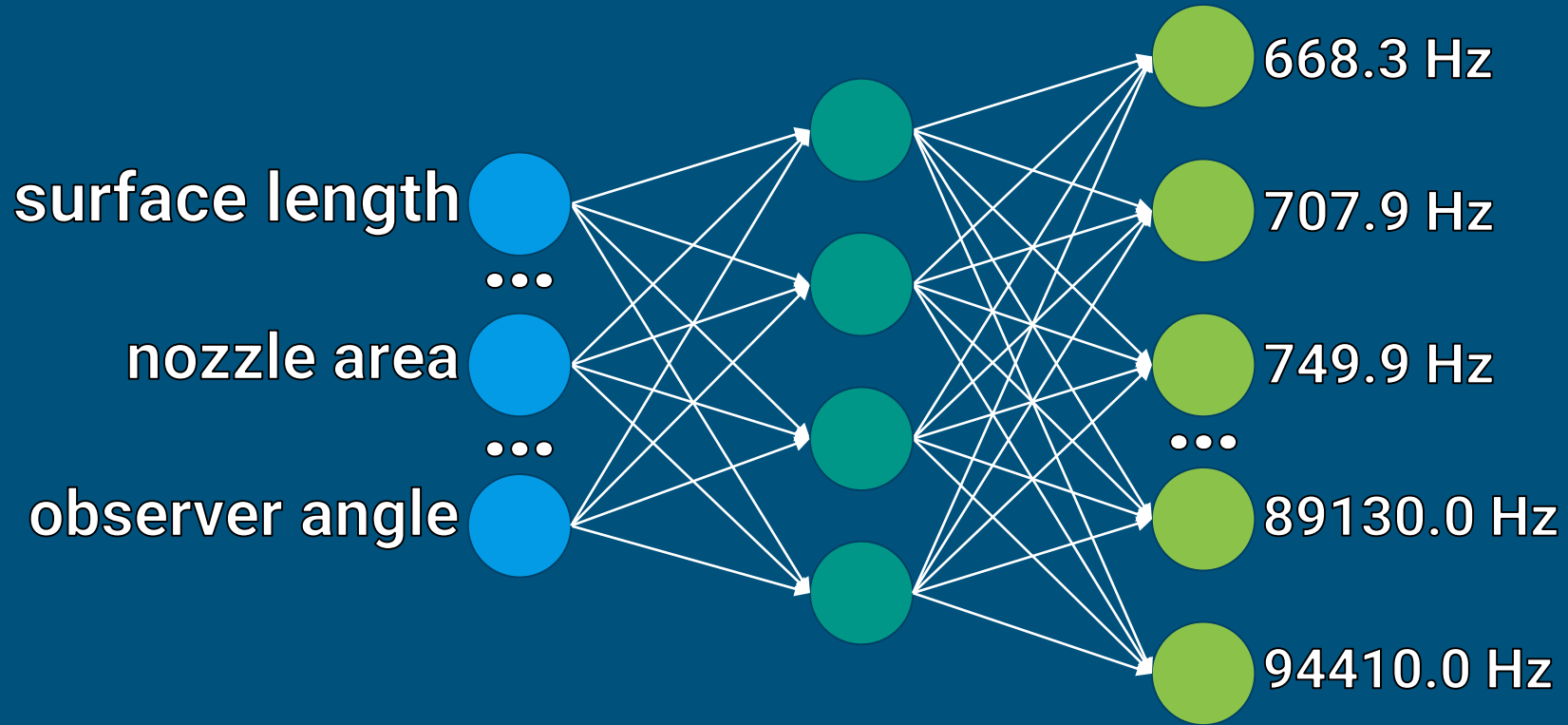
surface (shielding)



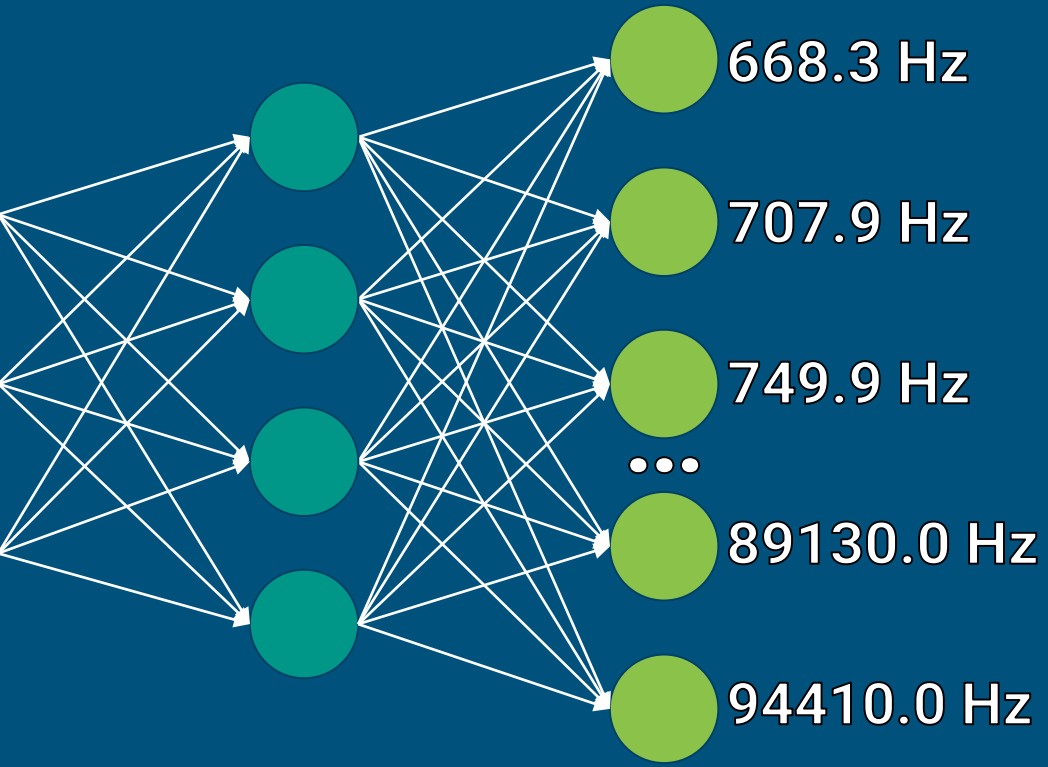


The Approach

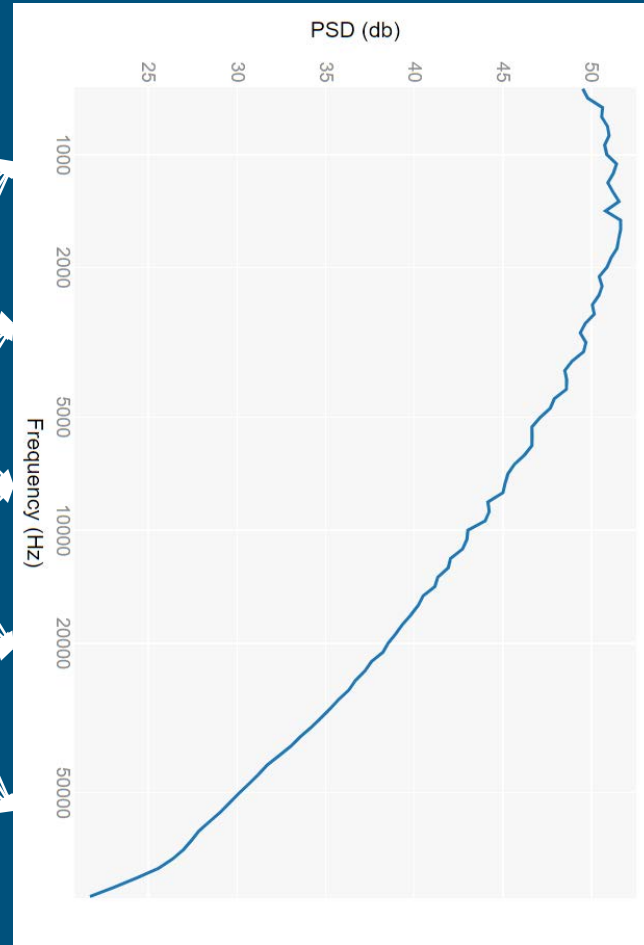
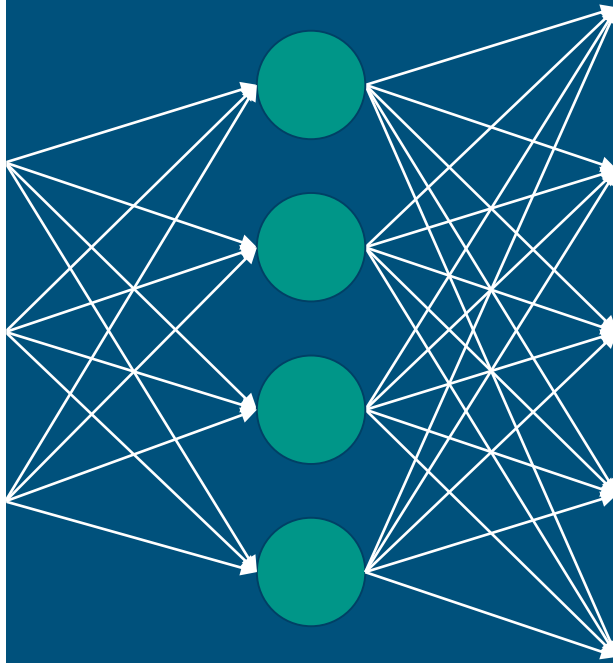




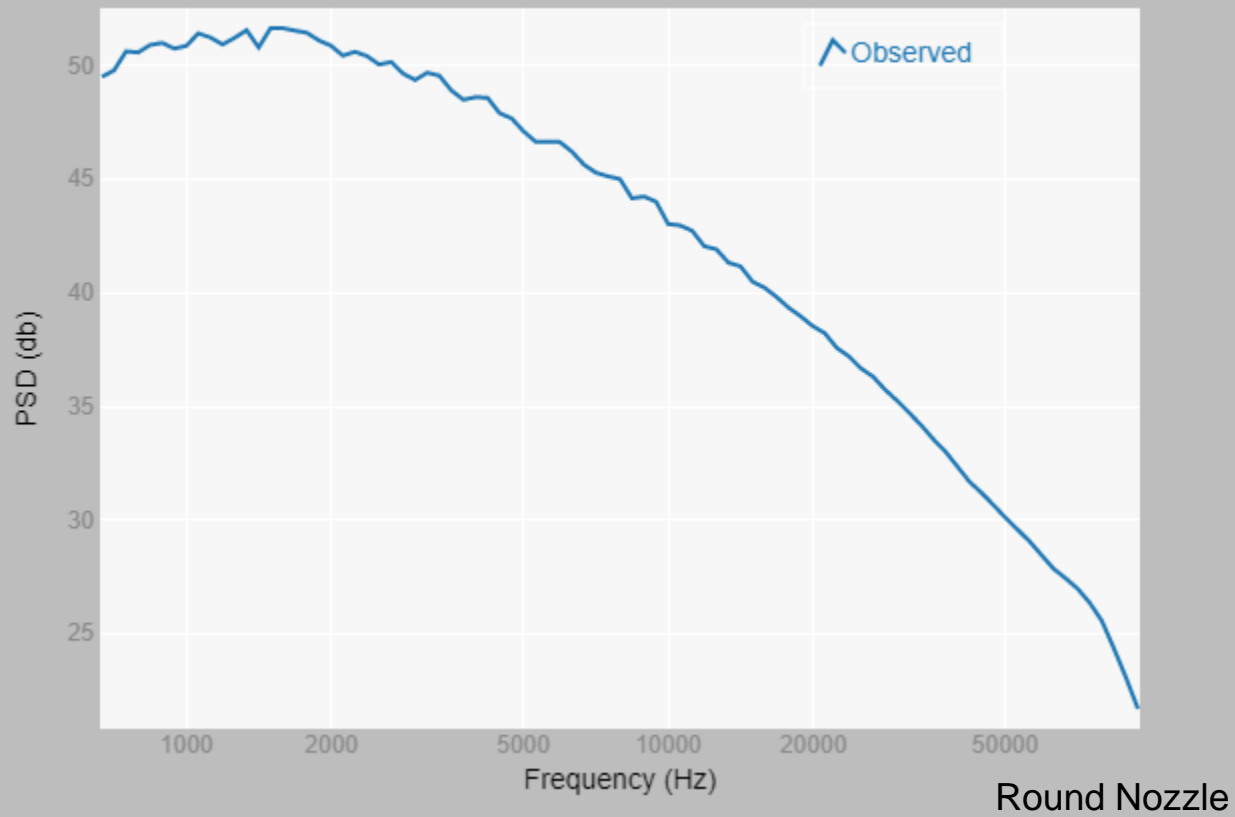
| | |
|----------------|------------|
| NozzleArea | 3.141590 |
| xE | 1.300000 |
| hE | 0.000000 |
| Ma | 0.699508 |
| Mj | 0.734632 |
| NPR | 1.431810 |
| TTR | 1.004820 |
| TSR | 0.906729 |
| Tamb | 490.749000 |
| Pamb | 14.257300 |
| RelHumidity | 57.931600 |
| SSS_Beta | 5.120000 |
| angle | 115.000000 |
| shield/reflect | 1.000000 |
| AspectRatio | 1.000000 |



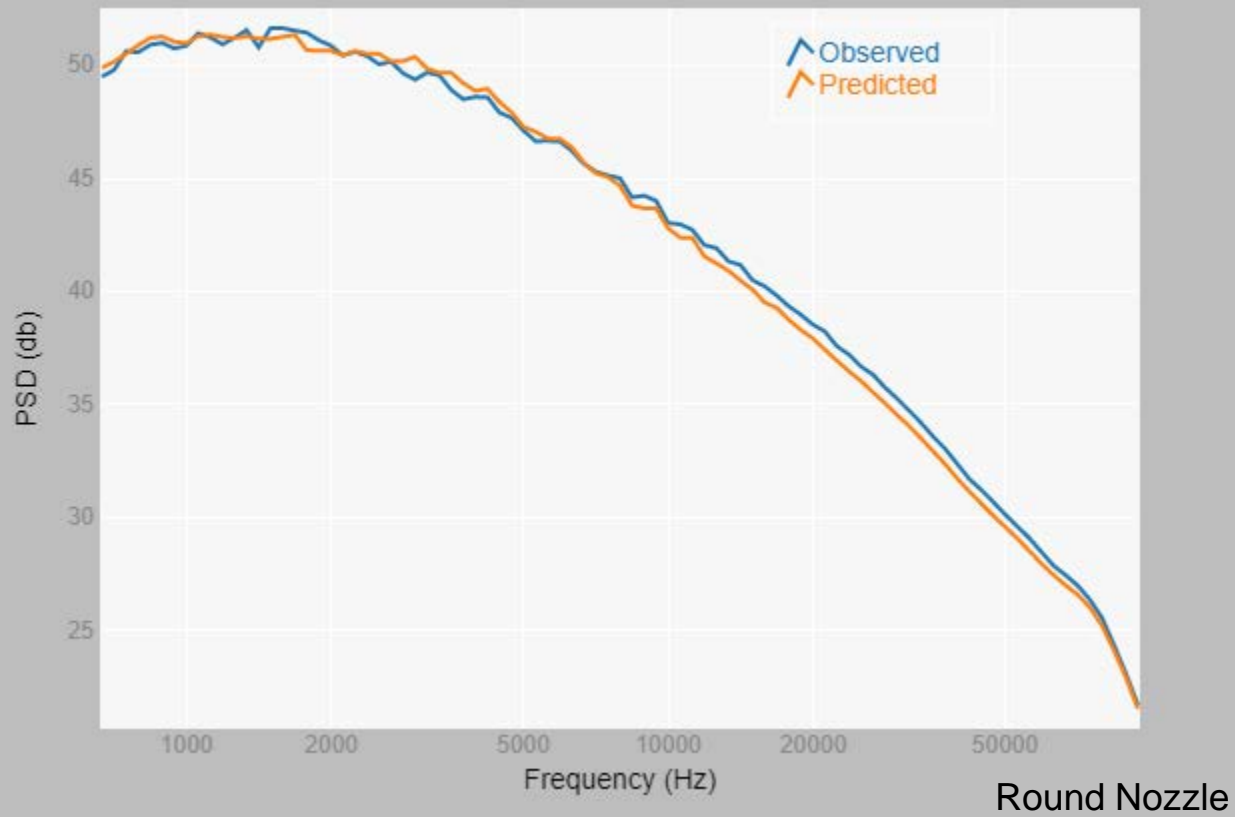
| | |
|----------------|------------|
| NozzleArea | 3.141590 |
| xE | 1.300000 |
| hE | 0.000000 |
| Ma | 0.699508 |
| Mj | 0.734632 |
| NPR | 1.431810 |
| TTR | 1.004820 |
| TSR | 0.906729 |
| Tamb | 490.749000 |
| Pamb | 14.257300 |
| RelHumidity | 57.931600 |
| SSS_Beta | 5.120000 |
| angle | 115.000000 |
| shield/reflect | 1.000000 |
| AspectRatio | 1.000000 |



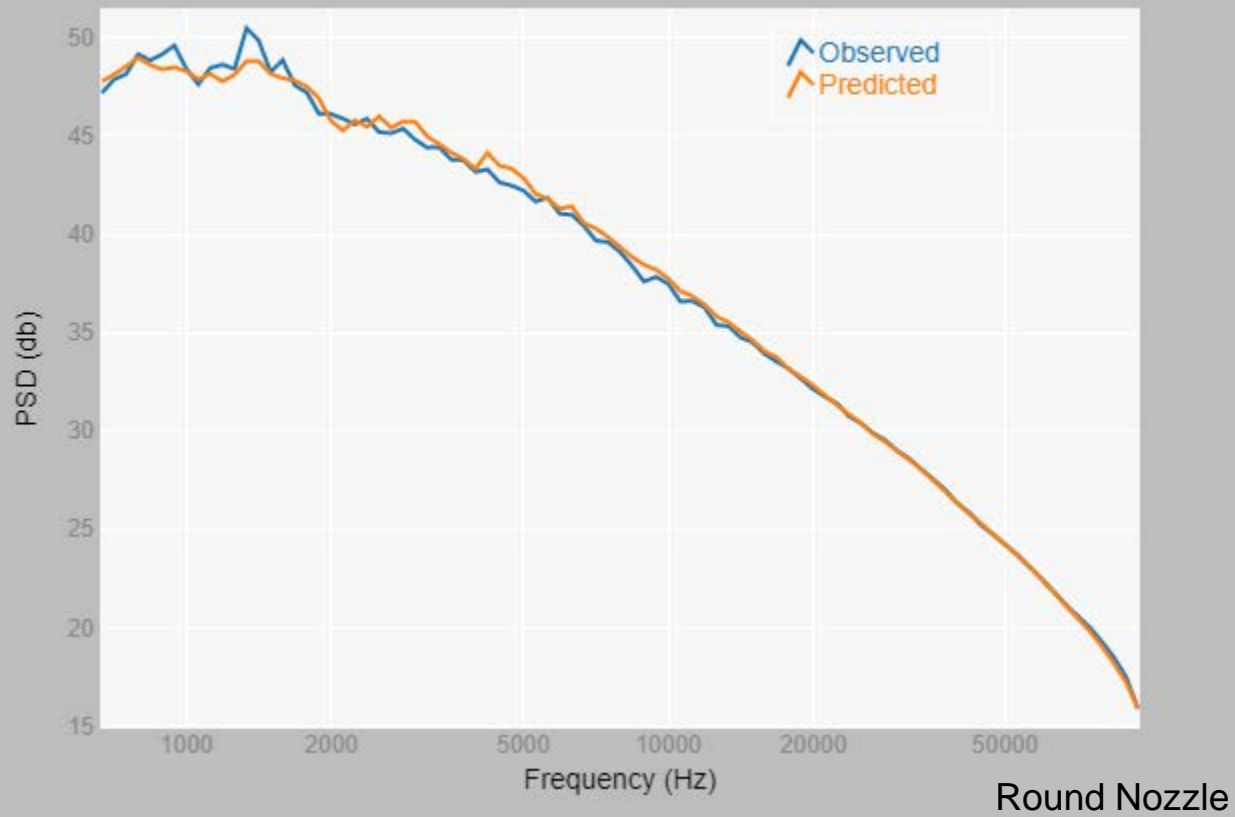
Results



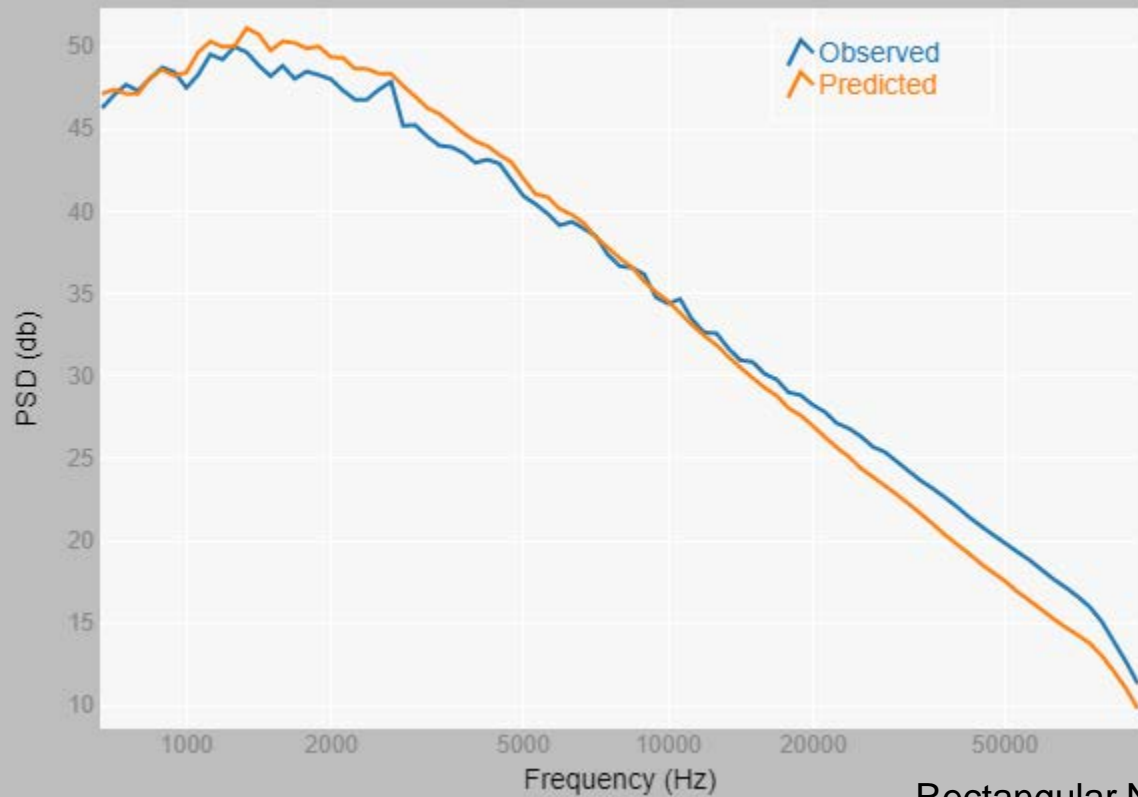
Mean Absolute Error: 0.412



Mean Absolute Error: 0.32



Mean Absolute Error: 1.292



Rectangular Nozzle

Error

Round Nozzle Model

Validation Samples: 4272

Average: +/- **0.575** dB

Worst sample: +/- 2.23 dB

Rectangular Nozzle Model

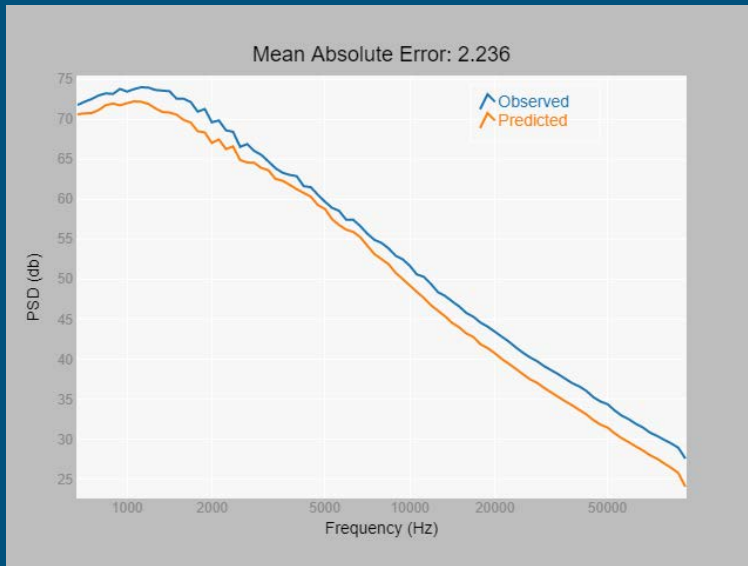
Validation Samples: 12594

Average: +/- **0.599** dB

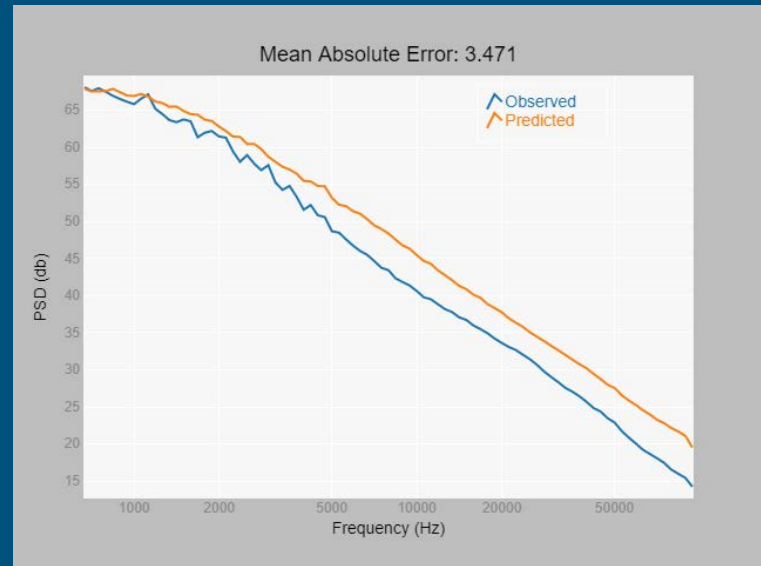
Worst sample: +/- 3.47 dB

Error

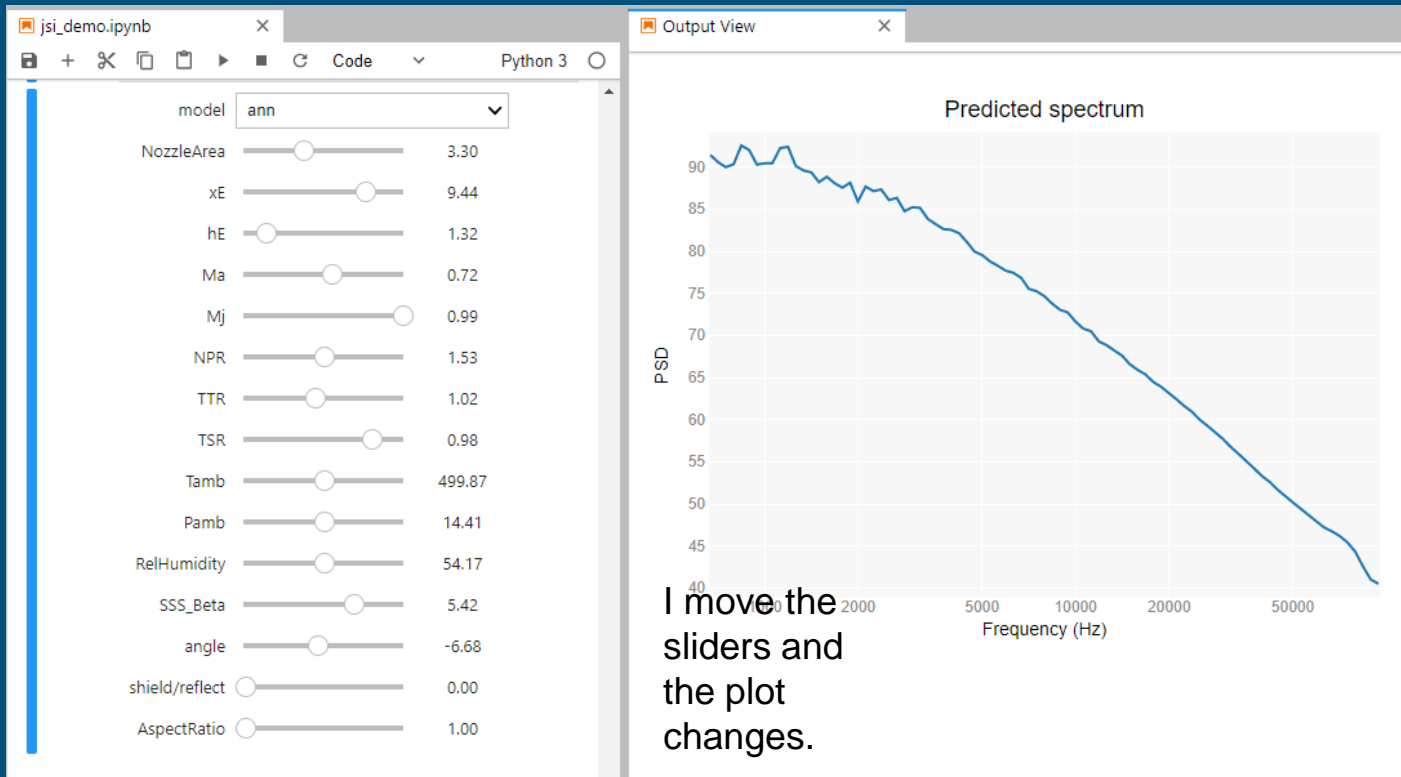
Round Nozzle Model



Rectangular Nozzle Model



Live Demo



Go deeper?

No, go shallower

Modeling the data looks promising.

Rapidly **prototype small models**,
turn knobs, flip switches, see what
happens.

Error: 2.60

relative humidity

shielding or
reflecting

nozzle aspect
ratio

jet potential core
length

ambient pressure

ambient
temperature

static
temperature ratio

total temperature
ratio

nozzle pressure
ratio

observer angle

jet mach number

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.60

shielding or
reflecting

nozzle aspect
ratio

jet potential core
length

ambient pressure

ambient
temperature

static
temperature ratio

total temperature
ratio

nozzle pressure
ratio

observer angle

jet mach number

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.60

shielding or
reflecting

nozzle aspect
ratio

jet potential core
length

ambient pressure

static
temperature ratio

total temperature
ratio

nozzle pressure
ratio

observer angle

jet mach number

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.60

shielding or
reflecting

nozzle aspect
ratio

jet potential core
length

ambient pressure

static
temperature ratio

total temperature
ratio

nozzle pressure
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.60

shielding or
reflecting

nozzle aspect
ratio

ambient pressure

static
temperature ratio

total temperature
ratio

nozzle pressure
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.60

shielding or
reflecting

nozzle aspect
ratio

ambient pressure

static
temperature ratio

total temperature
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.61

shielding or
reflecting

nozzle aspect
ratio

static
temperature ratio

total temperature
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

nozzle area

Error: 2.62

shielding or
reflecting

nozzle aspect
ratio

static
temperature ratio

total temperature
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

Error: 2.66

shielding or
reflecting

nozzle aspect
ratio

static
temperature ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

Error: 2.67

shielding or
reflecting

nozzle aspect
ratio

observer angle

acoustic mach
number

surface standoff
distance

surface length

Error: 2.72

shielding or
reflecting

observer angle

acoustic mach
number

surface standoff
distance

surface length

Reduce the feature space

Simplify experiments

Collect more data

How do we do it?



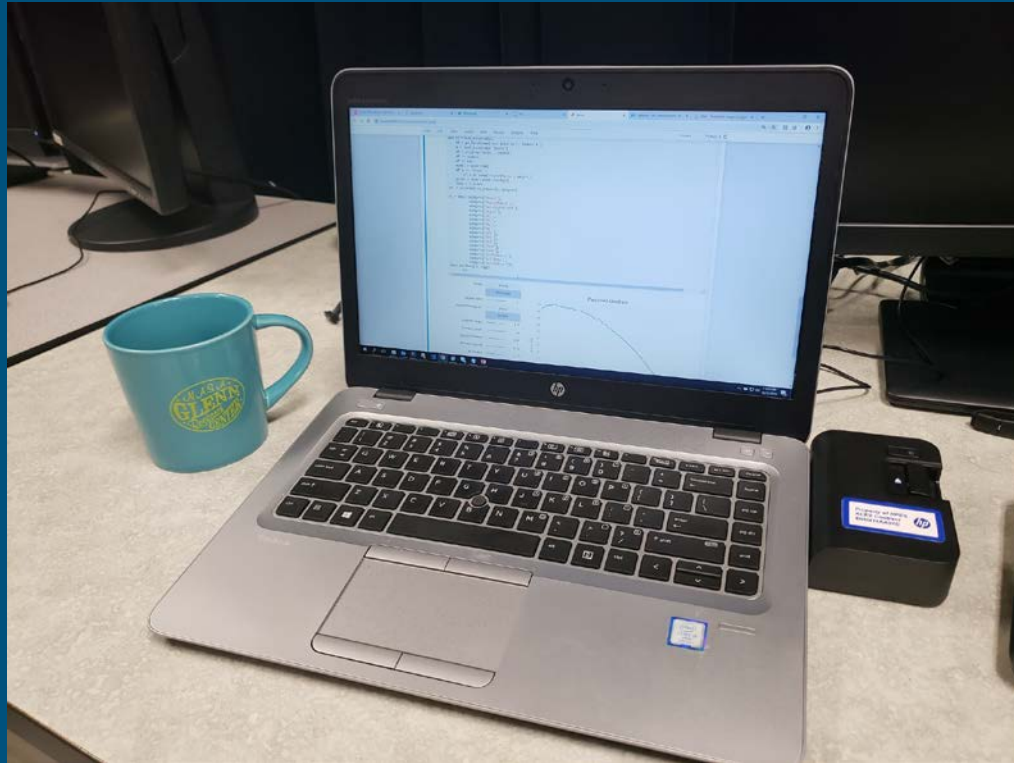
1. Python 3.6 (free)
2. Keras + TensorFlow (free)

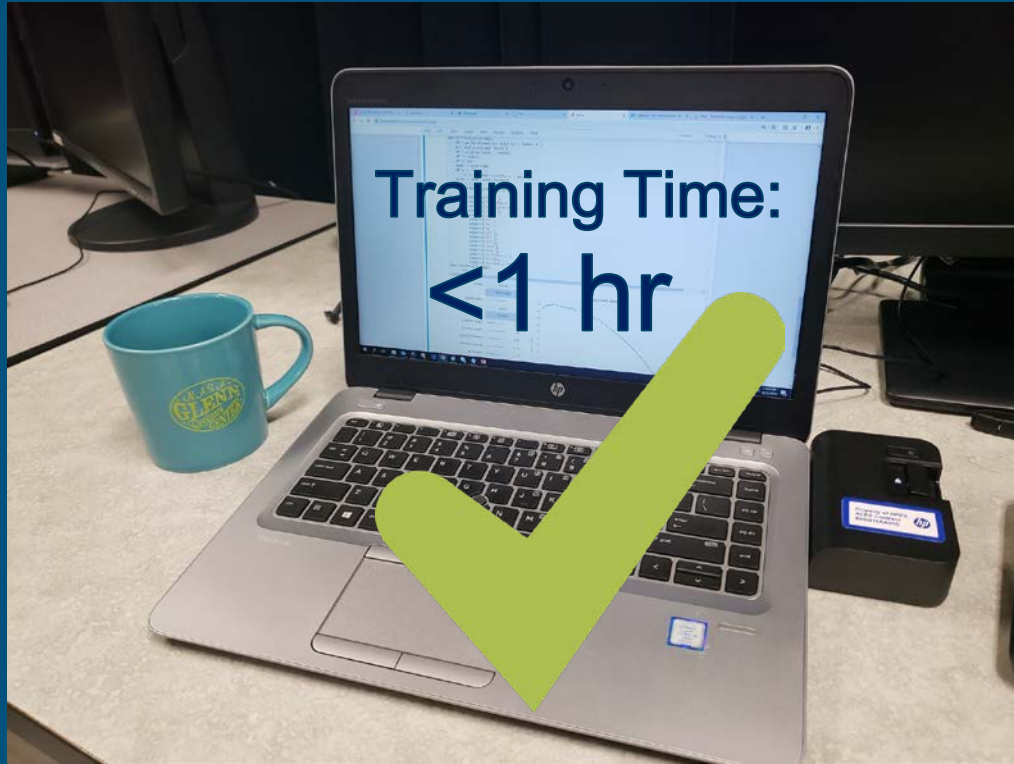
```
from keras.layers import Input, Dense
from keras.models import Model

input_layer = Input(shape=(INPUT_SIZE,))
hidden_layer = Dense(NUM_HIDDEN_UNITS, activation='relu')(input_layer)
output_layer = Dense(OUTPUT_SIZE)(hidden_layer)
model = Model(input_layer, output_layer)
model.compile(optimizer='adadelta', loss='mse')
```









DISCLAIMER:

Results may vary...

Neural network

short, thin, and simple

- Only 15 input features
- 7,000 trainable parameters



Consider...

image classifier

- 224x224 color image over 150,000 features
- Good classifiers contain millions to hundreds of millions of trainable parameters



Data

~20mb

- 1700 experiments X 24 microphones
 - 42,000 total data points
-

This is a good application
of machine learning.

What's next?

Try holding out certain ranges of features, test extrapolation ability.

What's next?

Add another dimension to observer position.

What's next?

Modify loss function to incentivize close fit on frequencies of annoyance.

What's next?

Investigate poor predictions,
identify weak points in data
acquisition or feature range.

Acknowledgements

Cliff Brown (NASA) – Data Acquisition & Research Advisor

Chris Miller (NASA) - Research Advisor

Lauren McIntyre (NASA) – Technical Mentor

Questions?