Generalizing Gradient Descent

Generalizing gradient descent

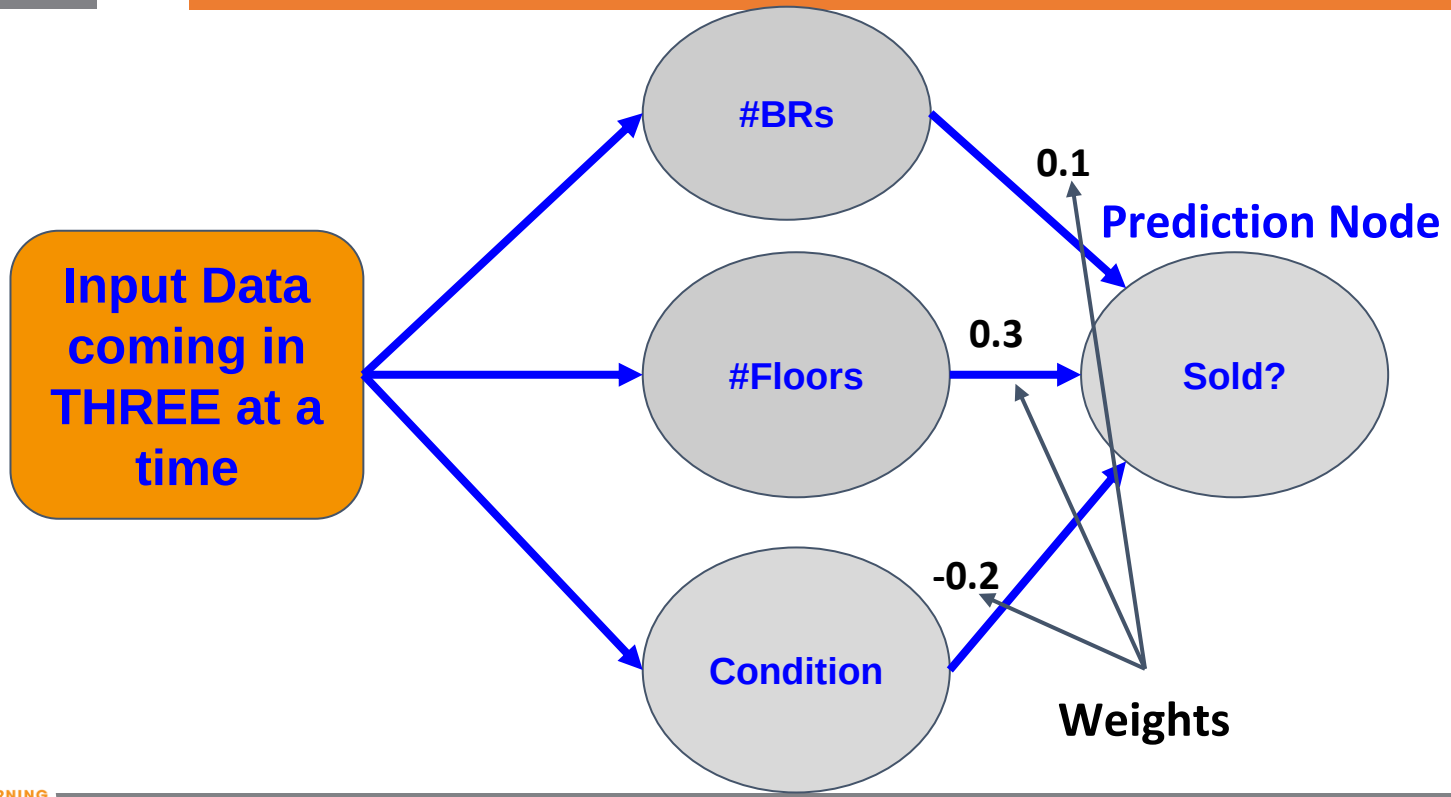
In this lesson

- How to build NNs with multiple inputs
- Freezing weights
- Gradient Descent with multiple outputs
- Gradient Descent with multiple inputs and multiple outputs
- Building a NN with real data… finally

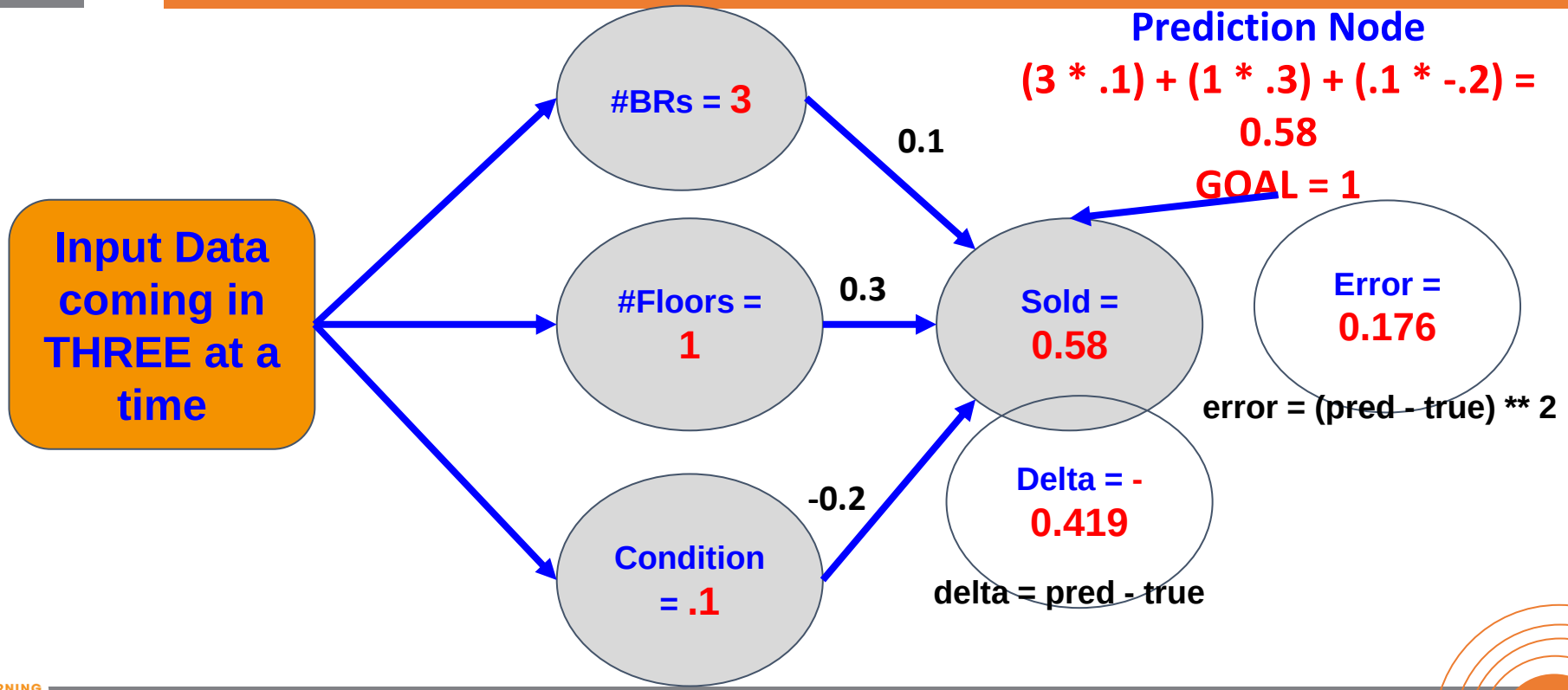
Gradient Descent with multiple inputs

Gradient descent also works with multiple inputs.

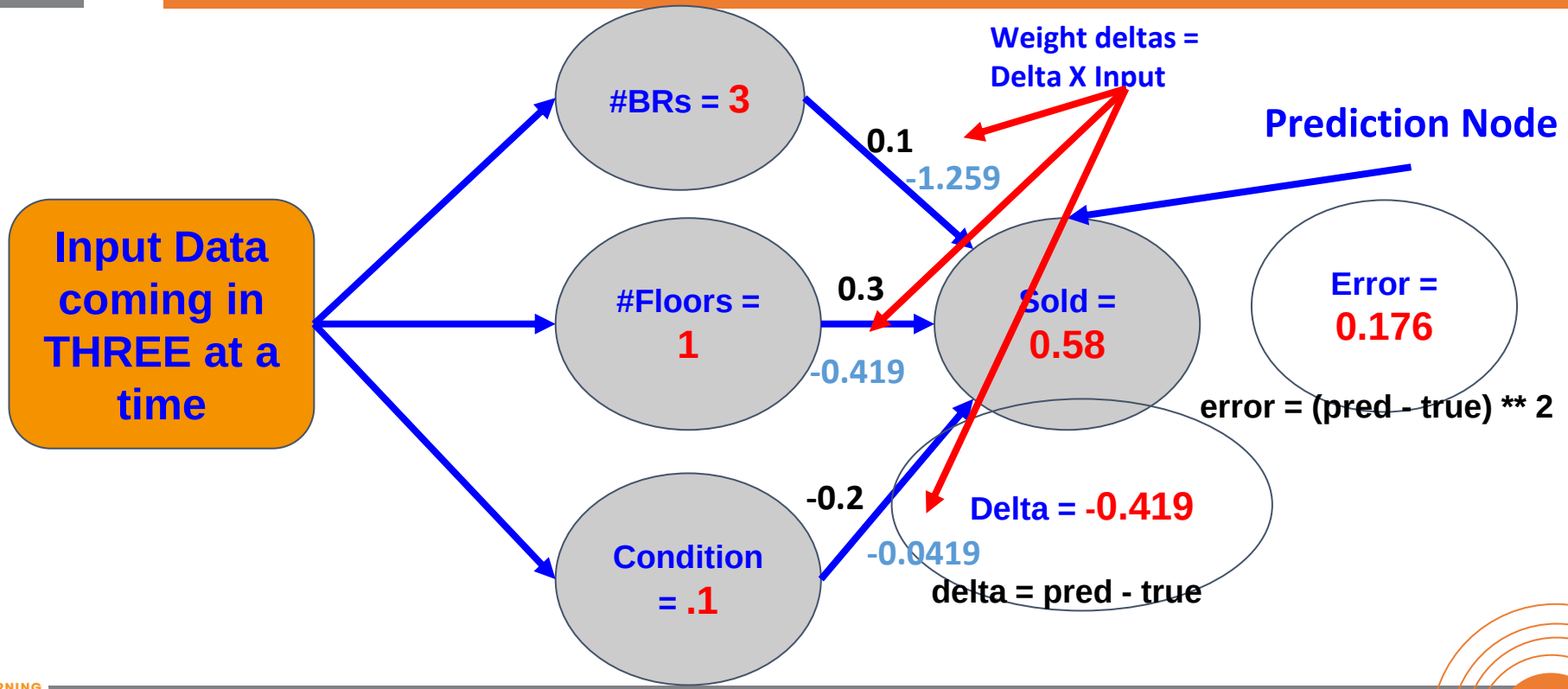
Gradient Descent with multiple inputs



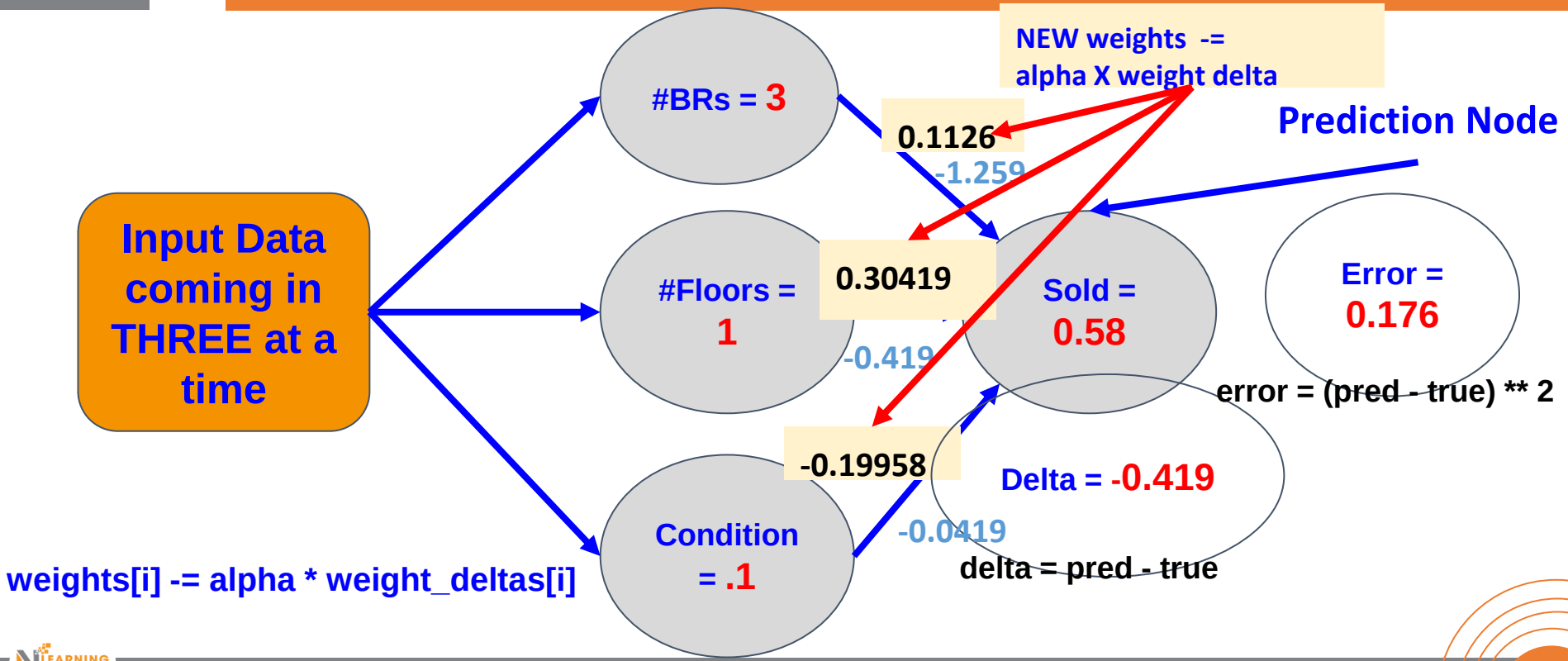
Gradient Descent with multiple inputs



Gradient Descent with multiple inputs



Gradient Descent with multiple inputs



Deeper Dive in Gradient Descent

INPUT

Bedrooms

= 3

PREDICTION

Input X

weight =

237K

Goal Prediction

= 250K

Error =

pred - goal=

-13K

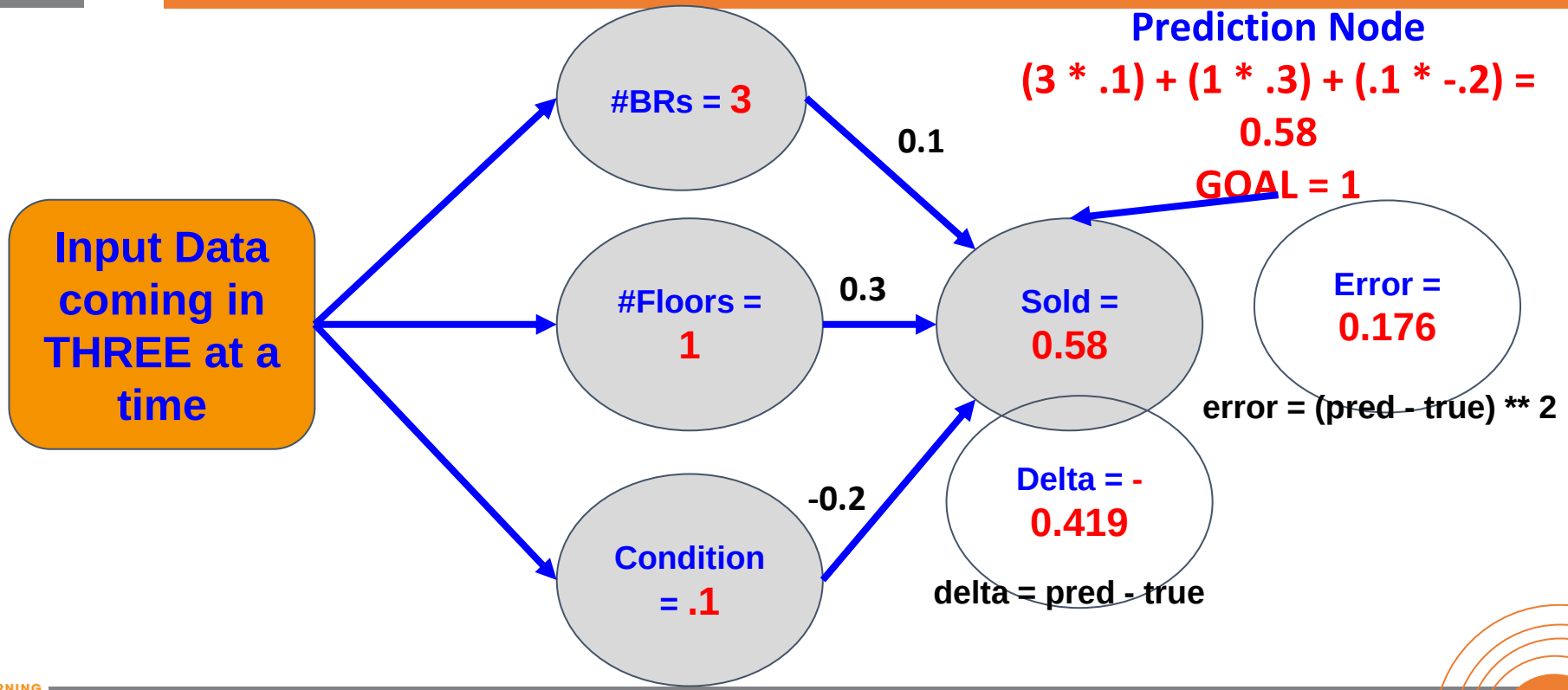
Weight Delta=

raw error * input

Weight = 79K

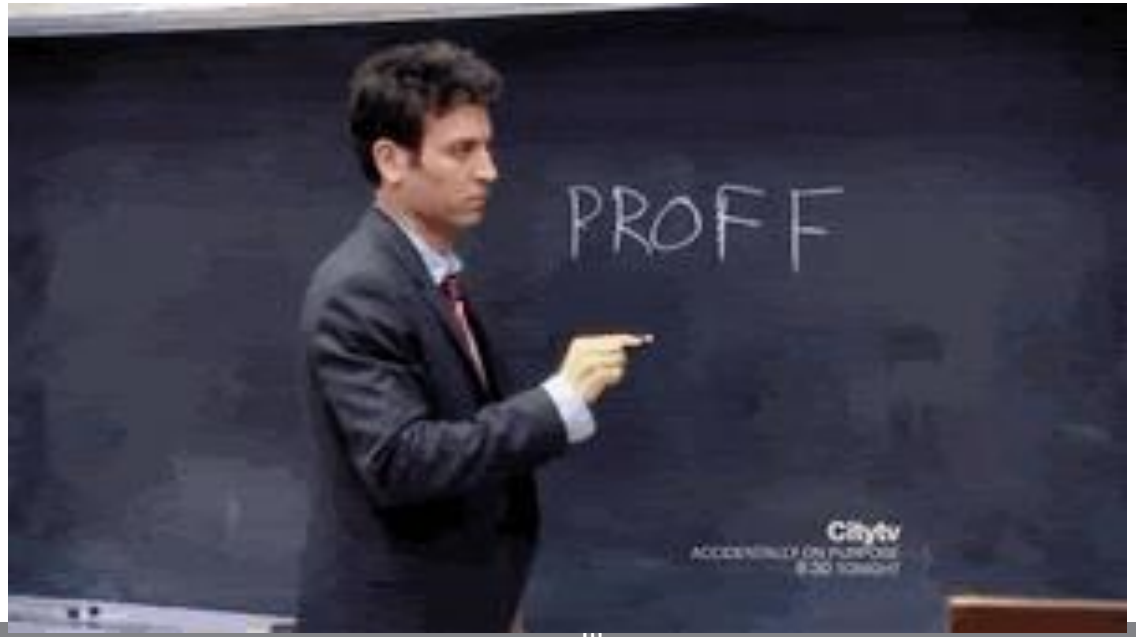
New Weight = Weight - Weight Delta

Deeper Dive in Gradient Descent

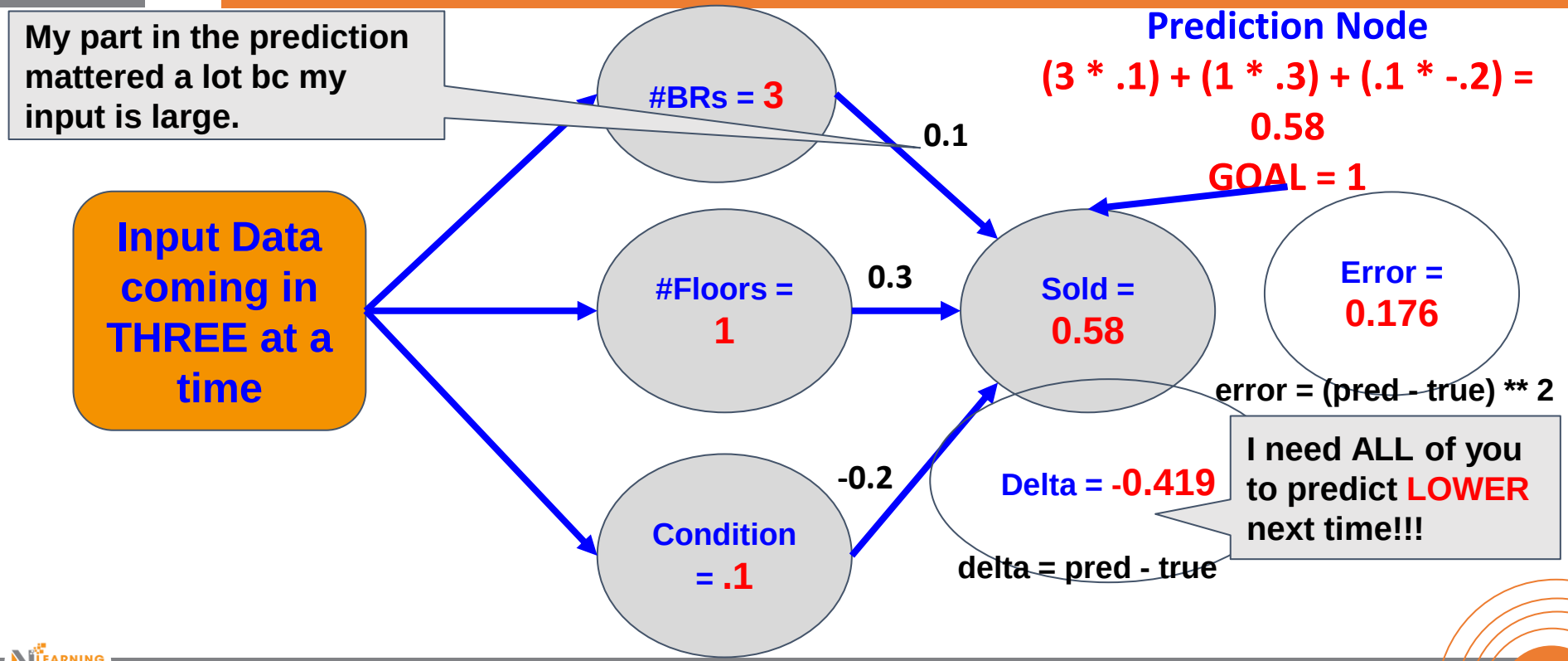


Deeper Dive in Gradient Descent

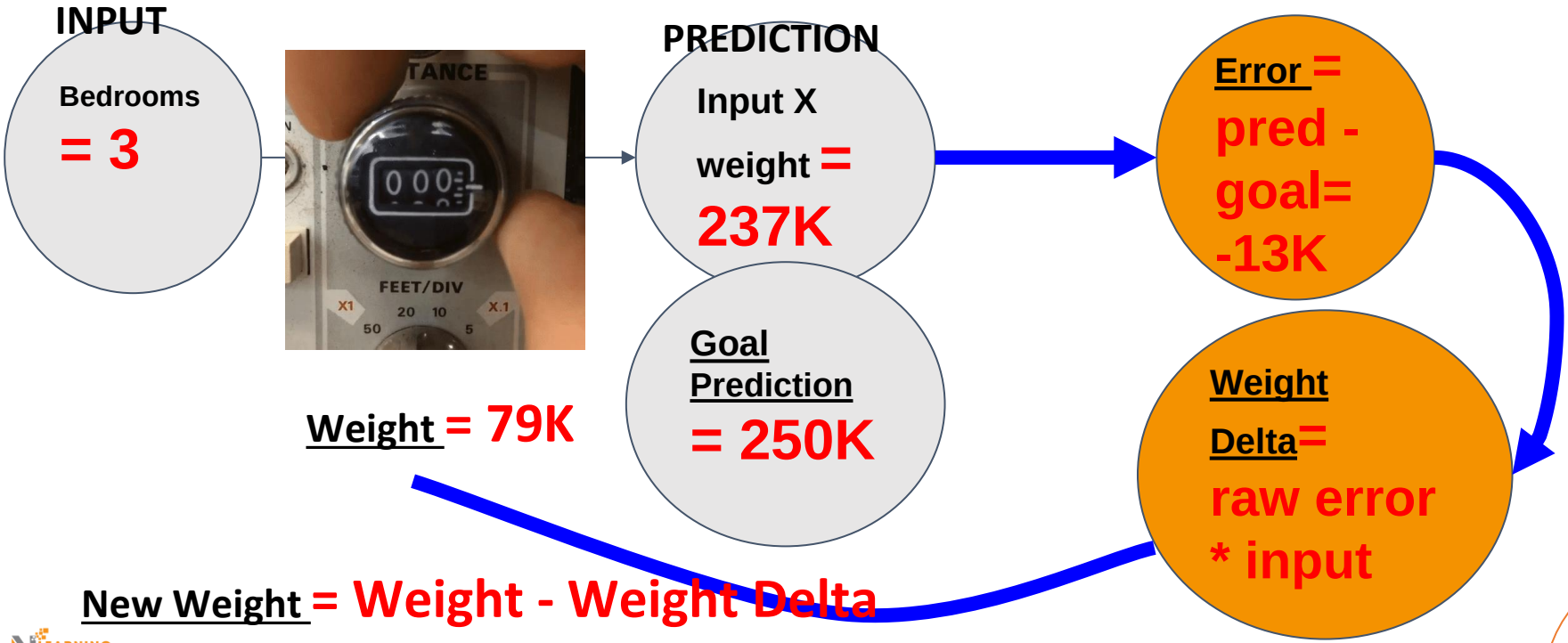
How do you turn a single delta from the output prediction into three weight_delta values?

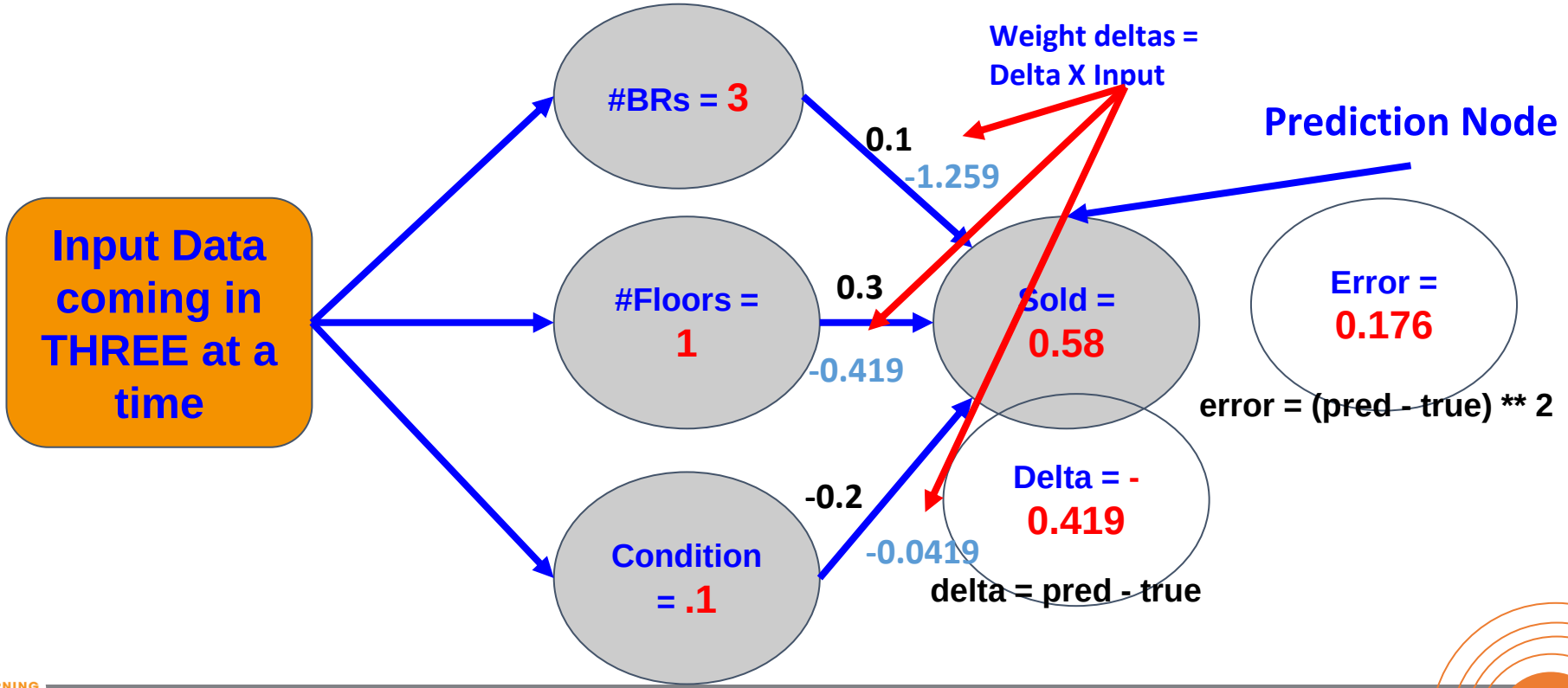


Deeper Dive in Gradient Descent



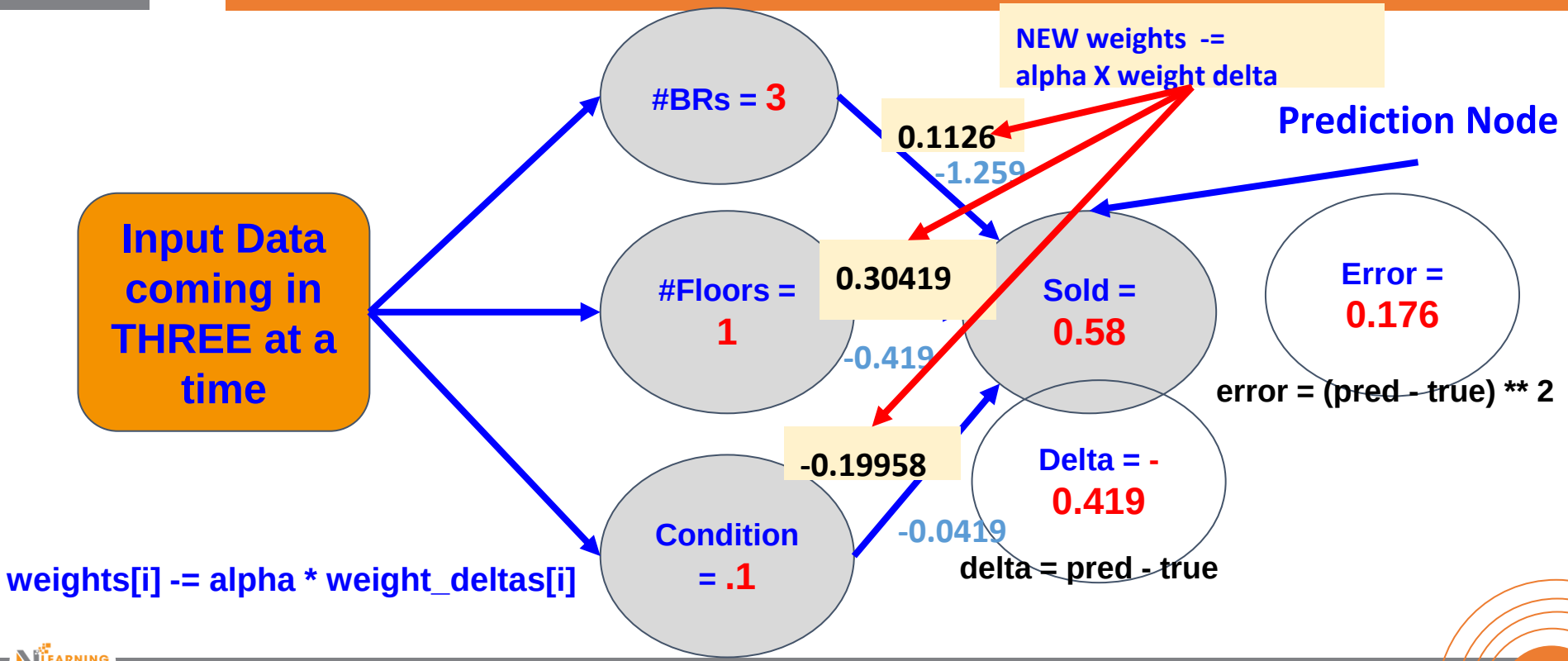
Deeper Dive in Gradient Descent



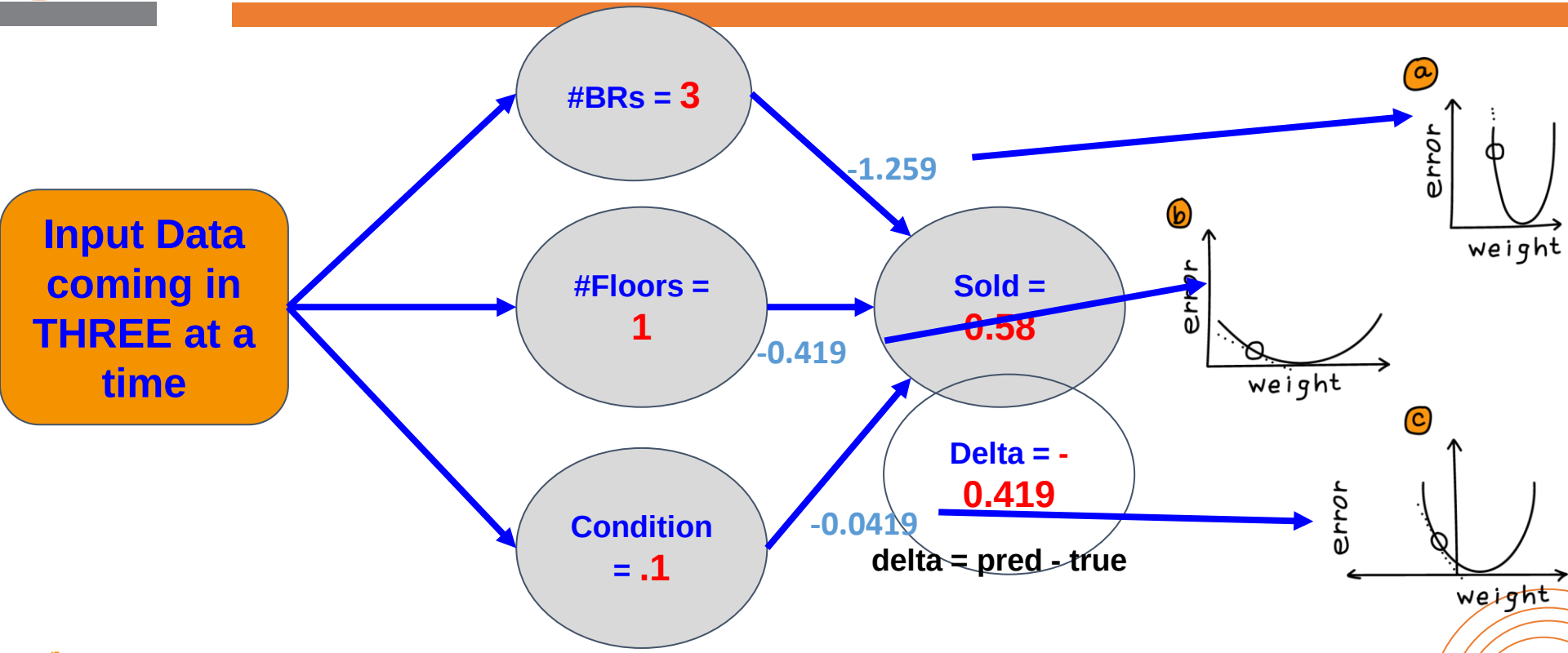

Input Data coming in THREE at a time
#BRs = 3
#Floors = 1
Condition = .1
0.1
-1.259
0.3
-0.419
-0.2
-0.0419
Weight deltas = Delta X Input
Prediction Node
Sold = 0.58
Error = 0.176
error = (pred - true) ** 2
Delta = - 0.419
delta = pred - true

Deeper Dive in Gradient Descent

Deeper Dive in Gradient Descent

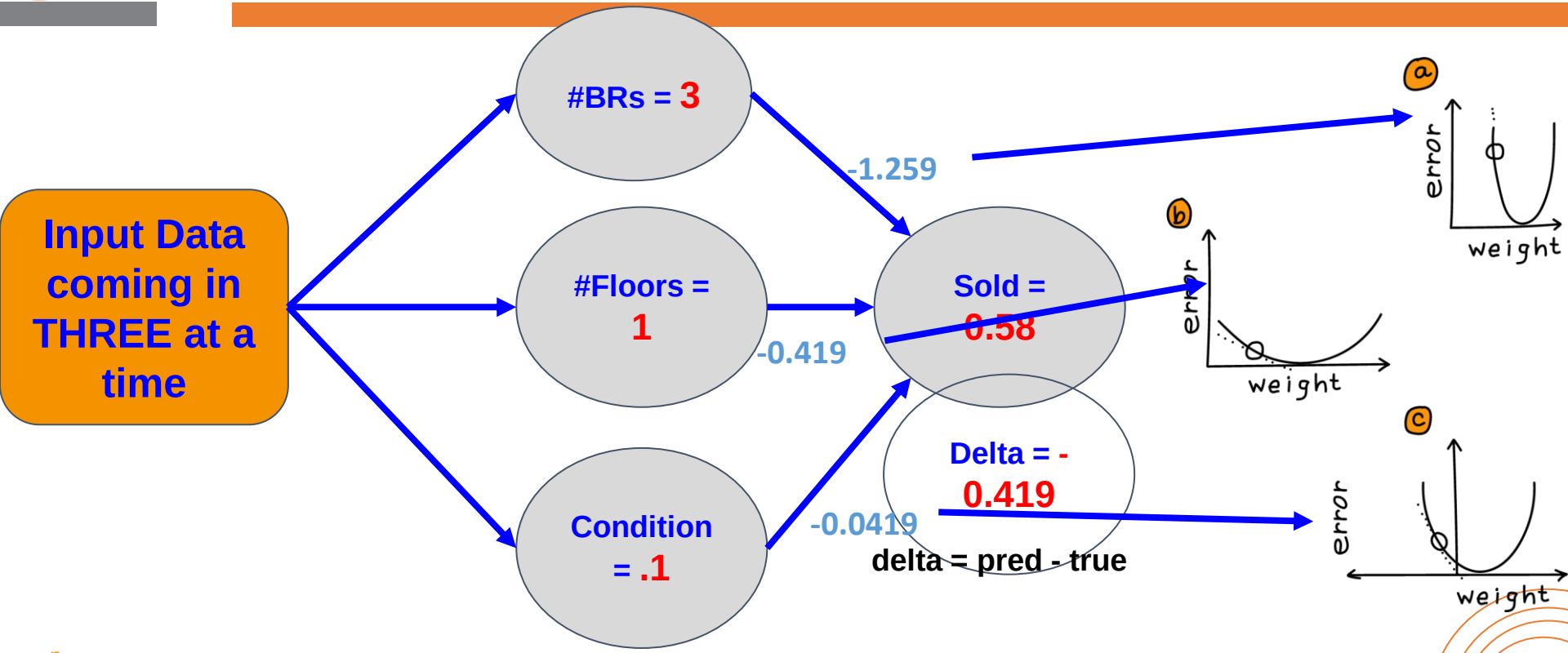


Iterations of Gradient Descent

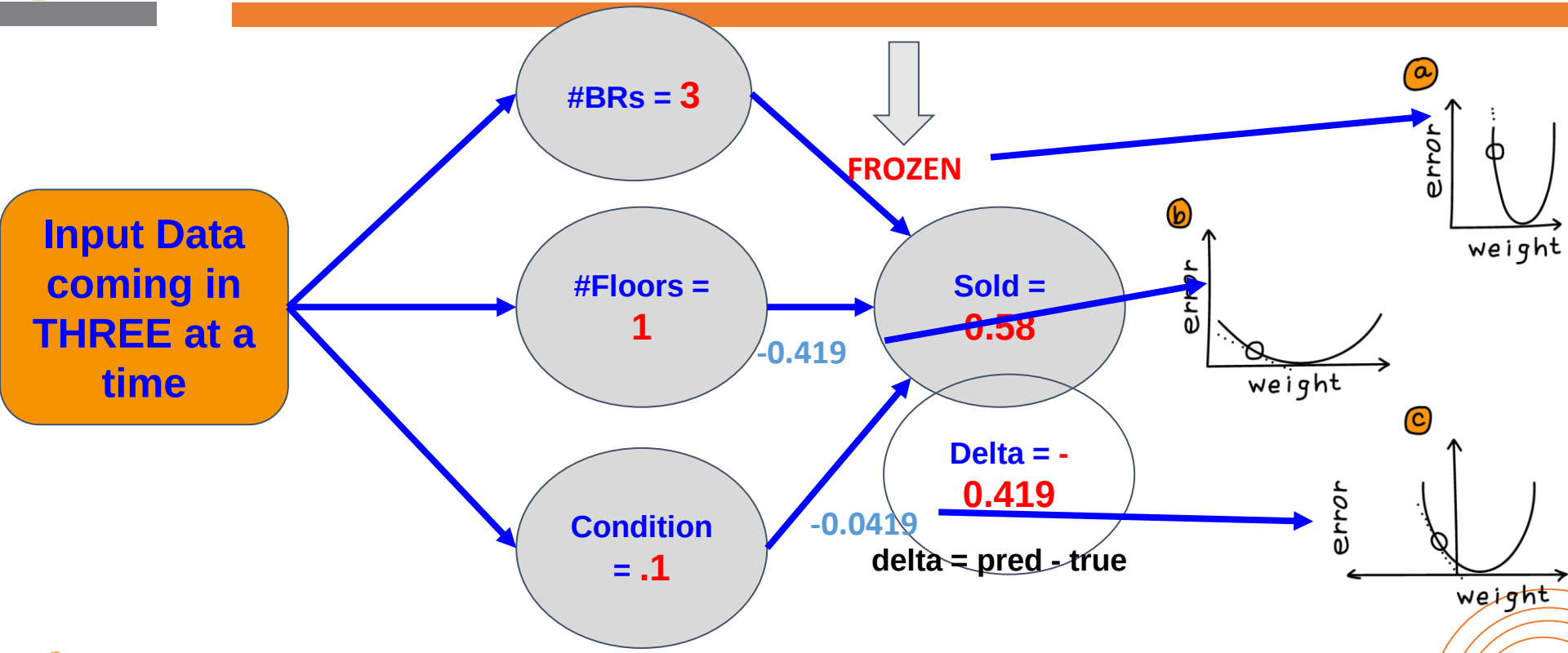


Iterations of Gradient Descent

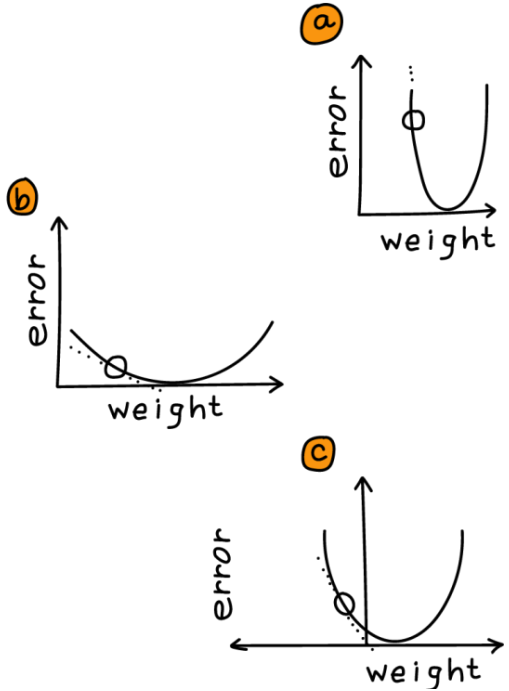
Iterations of Gradient Descent



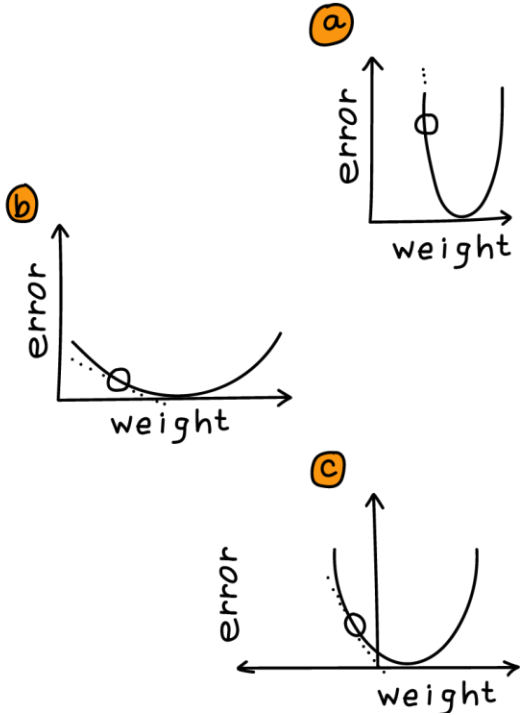
Learning with incomplete data

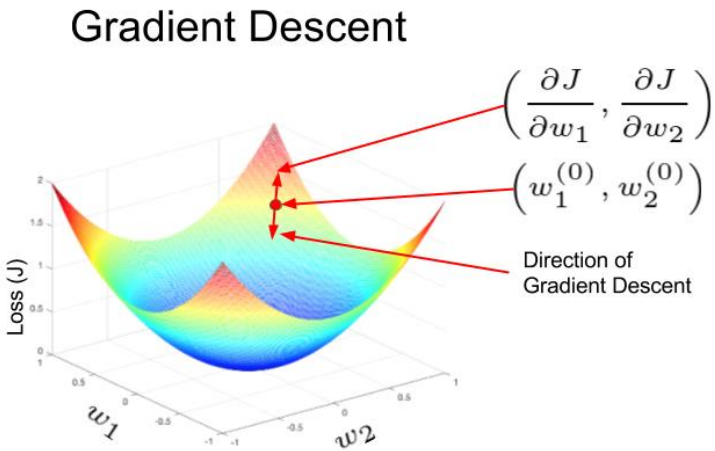


Learning with incomplete data



Learning with incomplete data

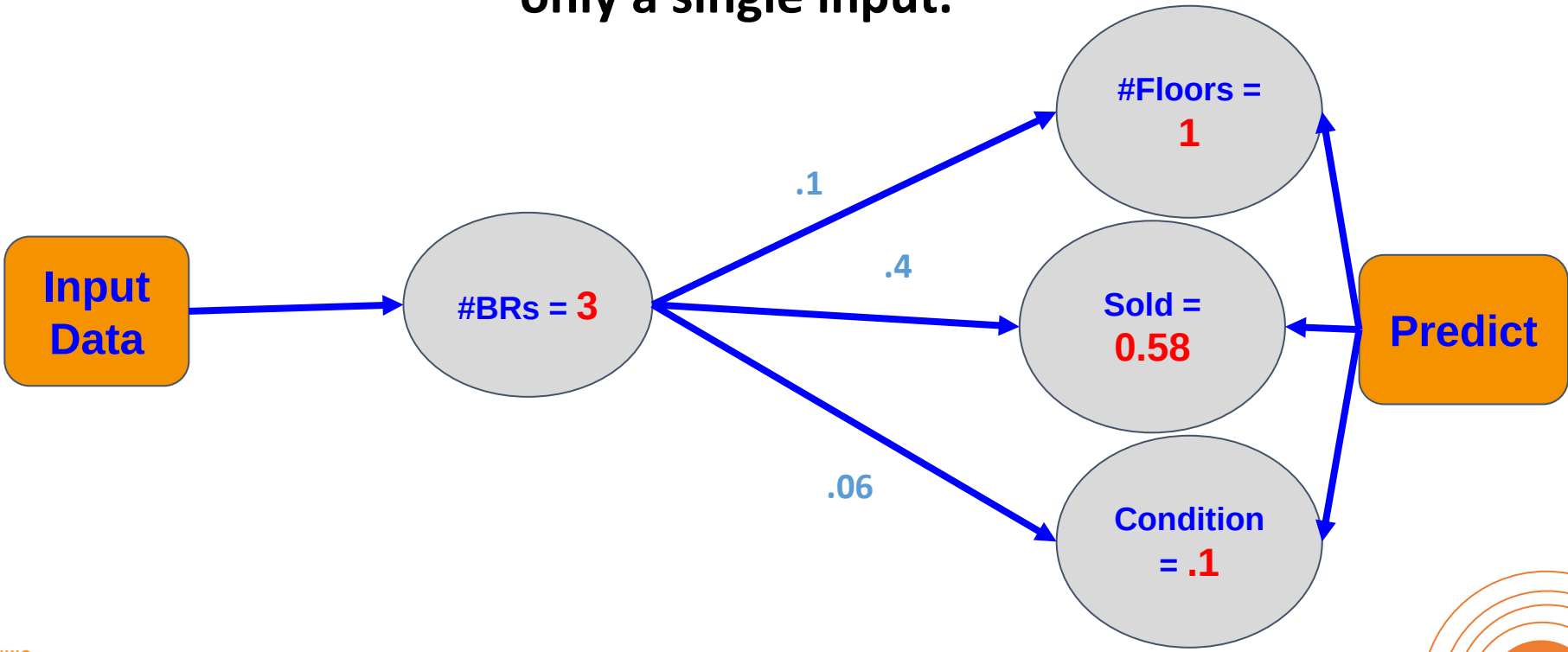




3D View b/c I can't show 4D

Gradient descent learning with multiple outputs

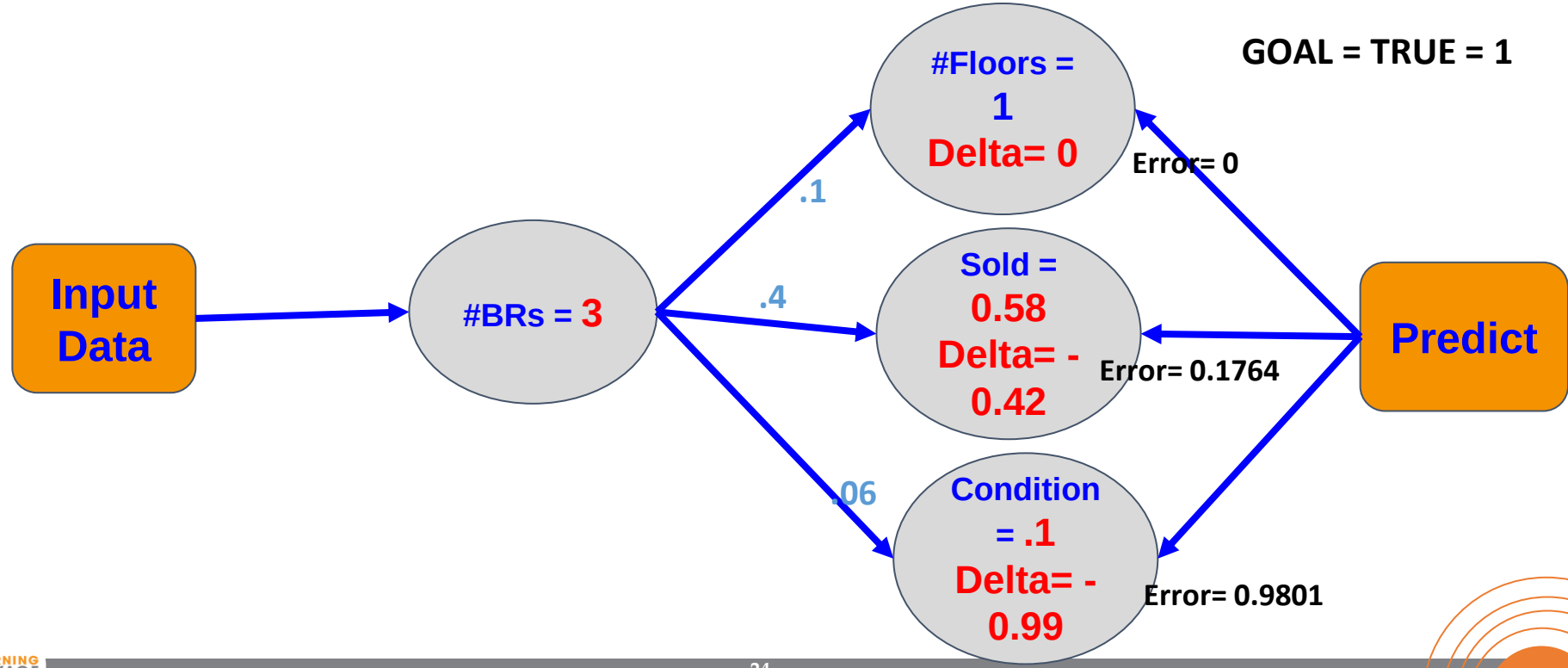
Neural networks can also make multiple predictions using only a single input.



Gradient descent learning with multiple outputs

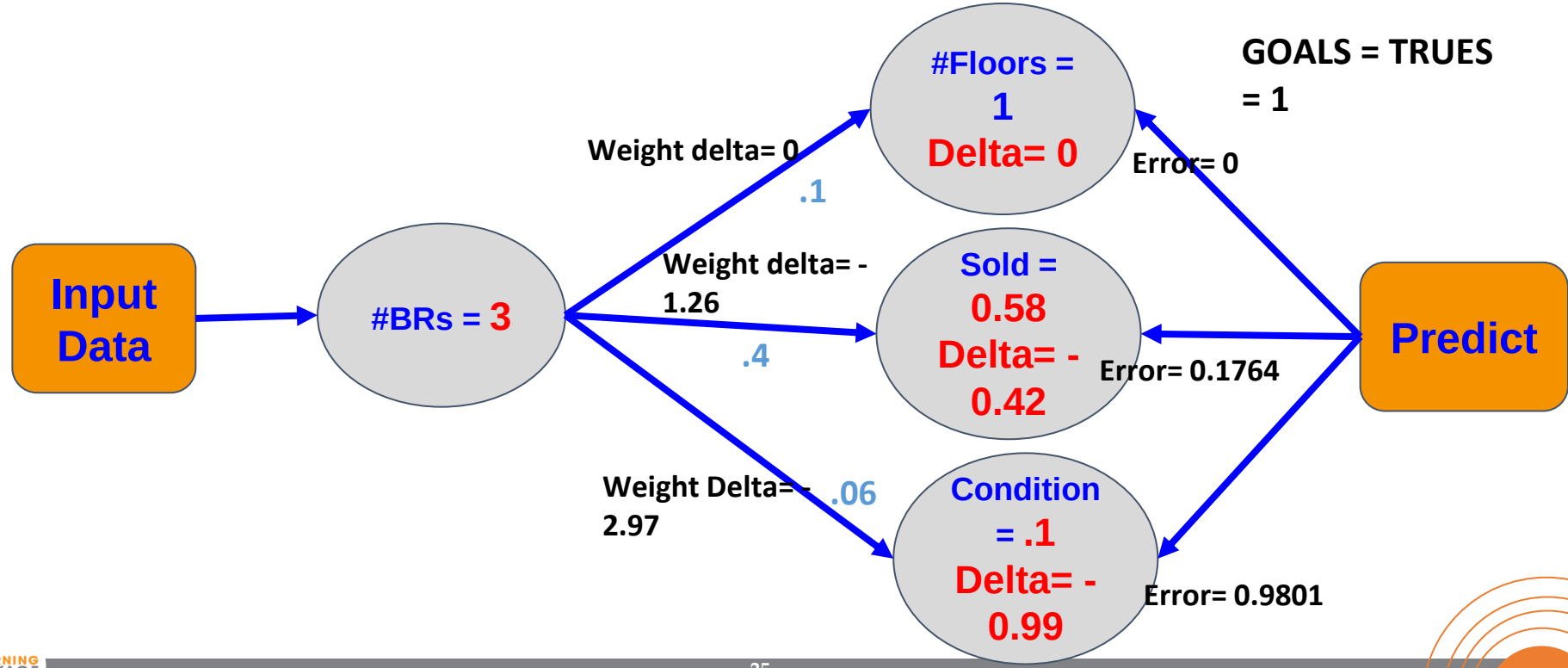
Gradient descent learning with multiple outputs

Computer Error and Delta



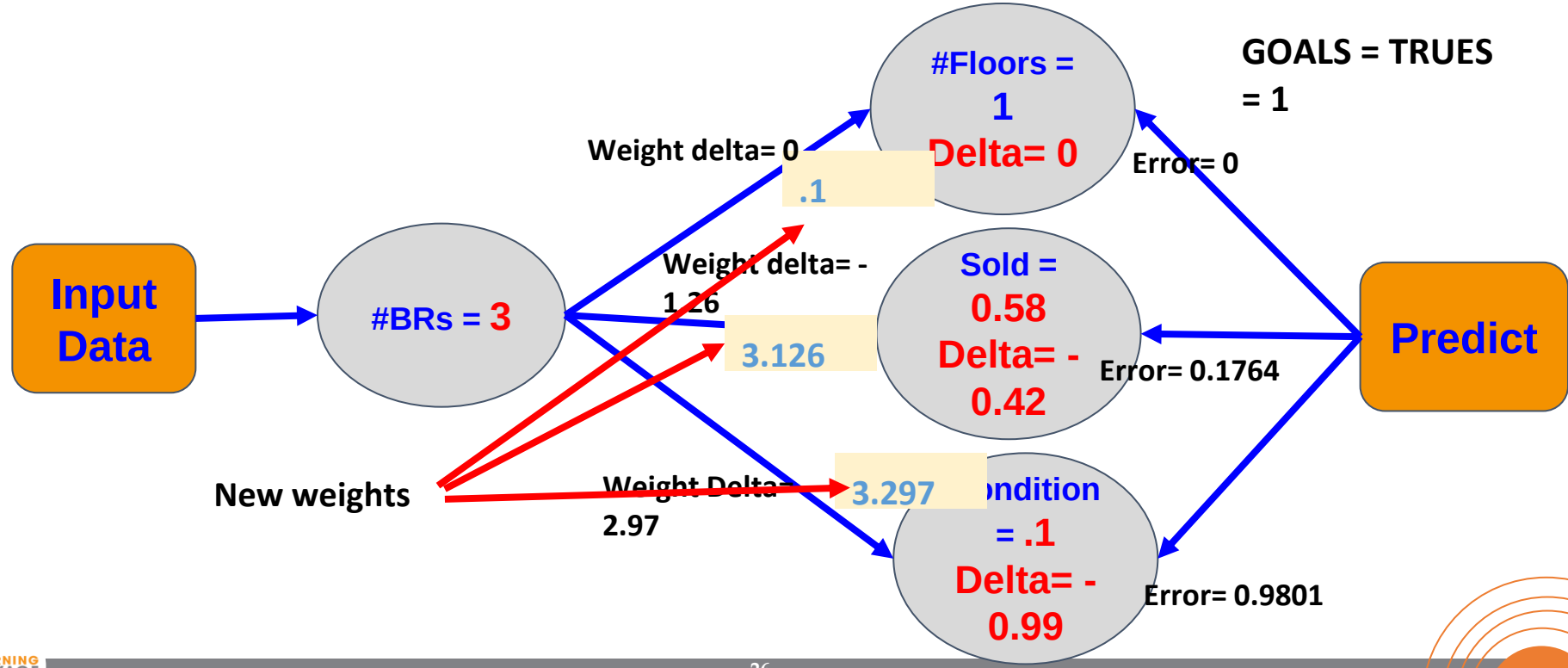
Gradient descent learning with multiple outputs

Computer Error and Delta

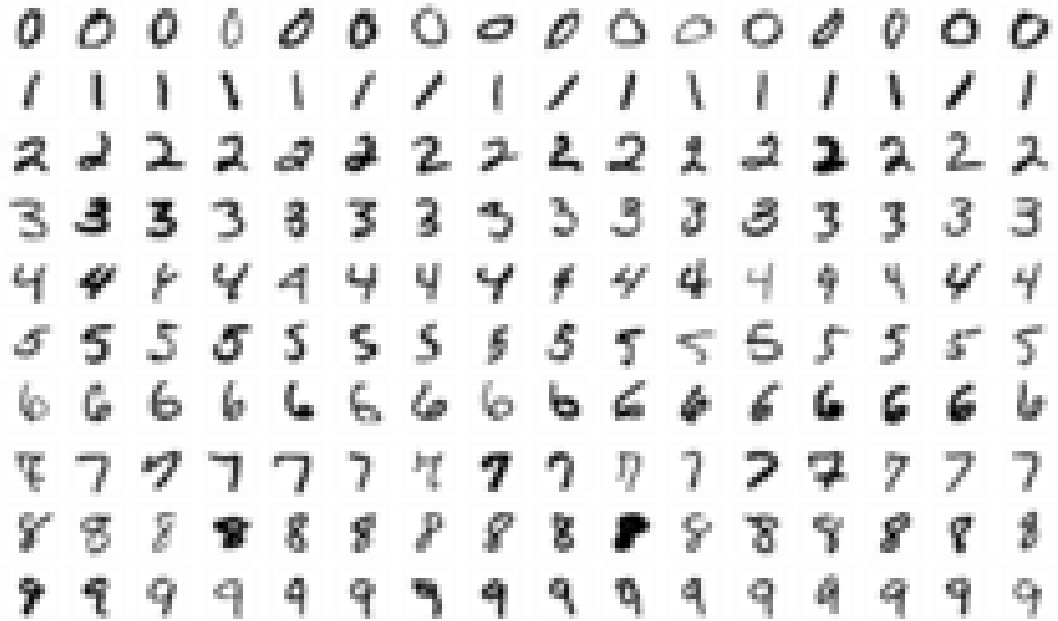


Gradient descent learning with multiple outputs

Computer Error and Delta

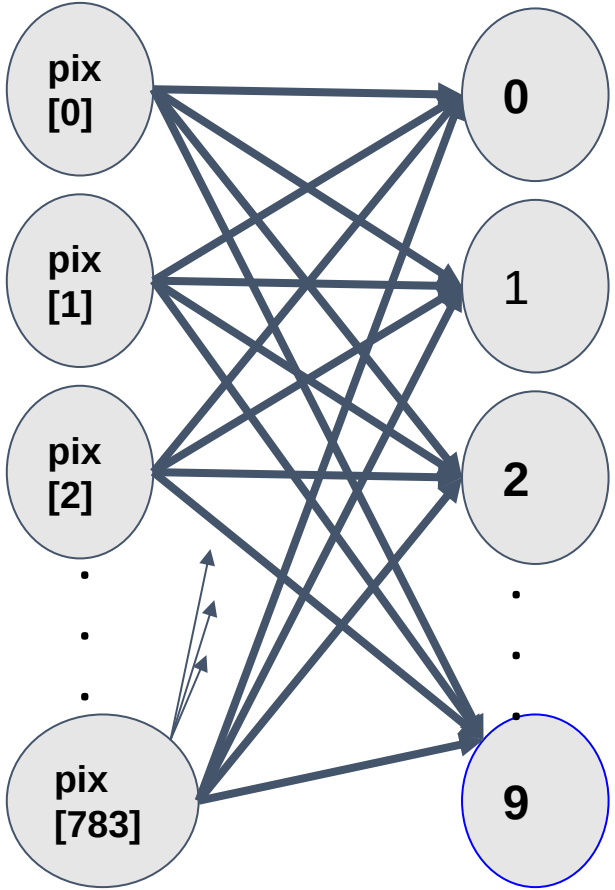


Real World Use Case

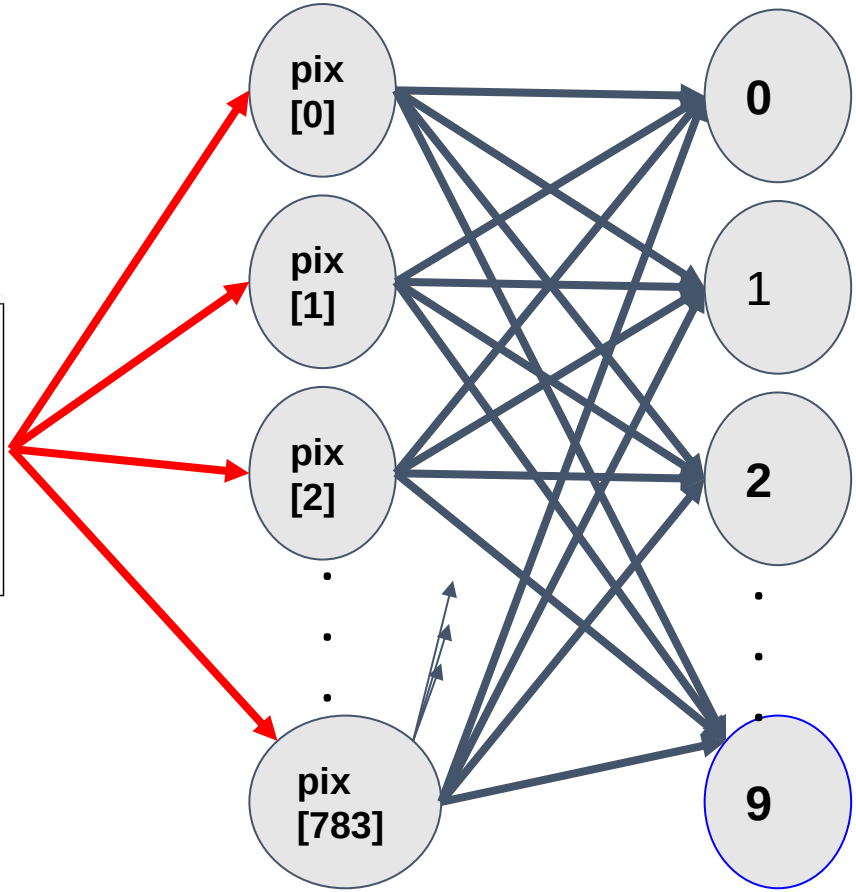


Real World Use Case

- So this is what your NN looks like now. It is a multi-input / multi-output NN with just many more inputs and outputs than we've seen before.



Input



0

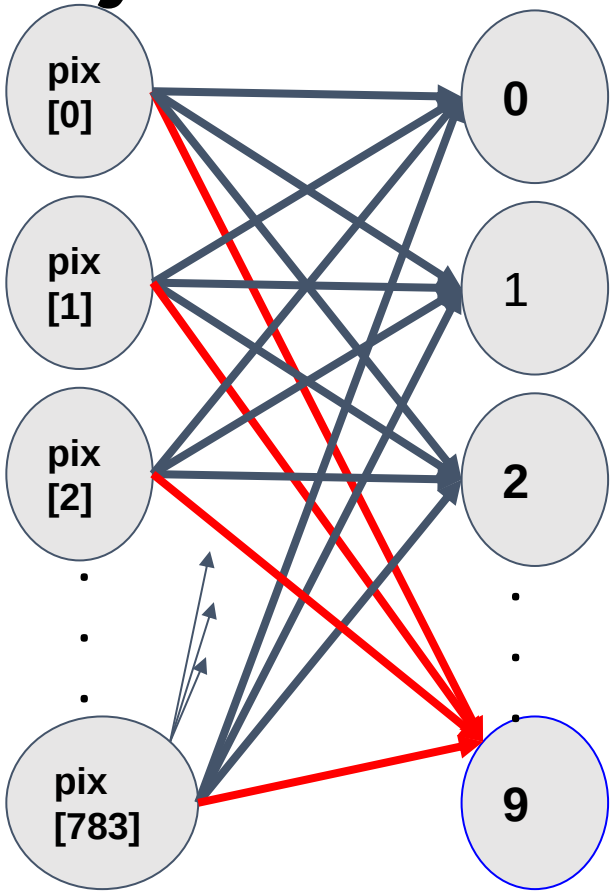
0.05

0

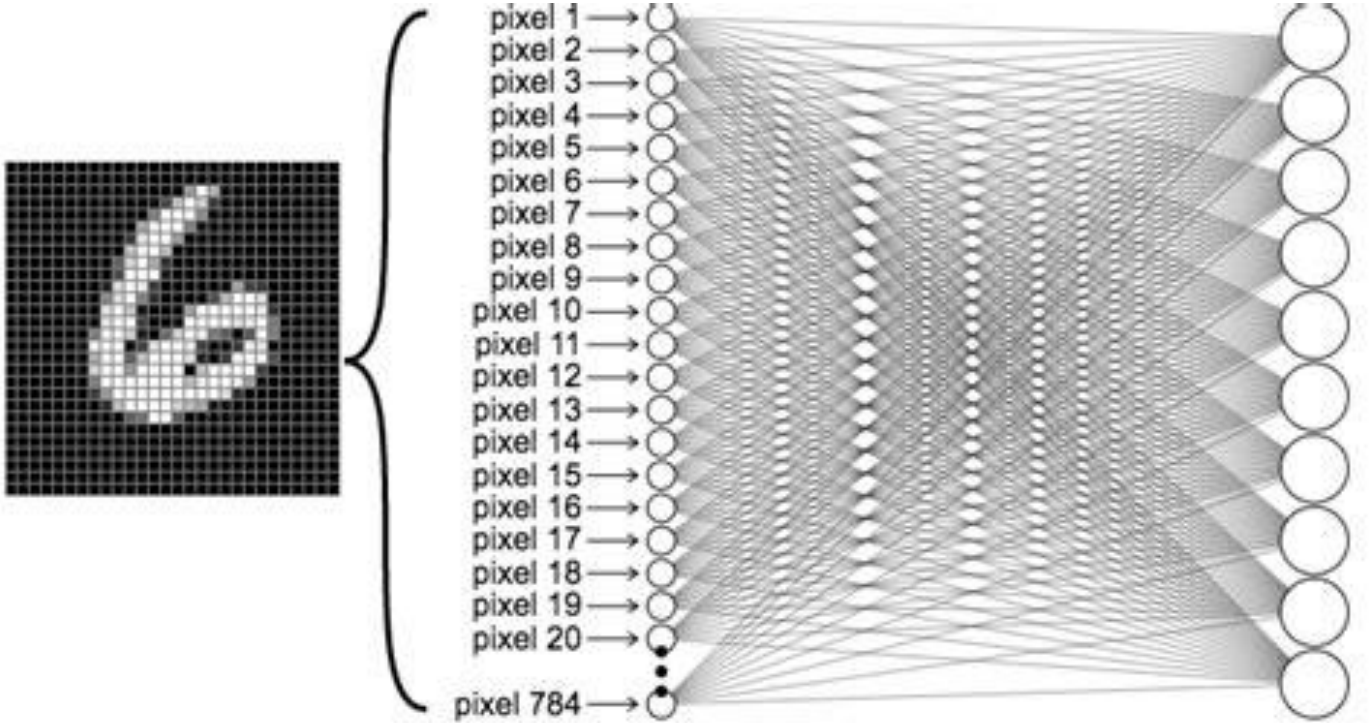
0.95

The network is 95% sure that the image is a 9 and 5% sure that it is a 1.

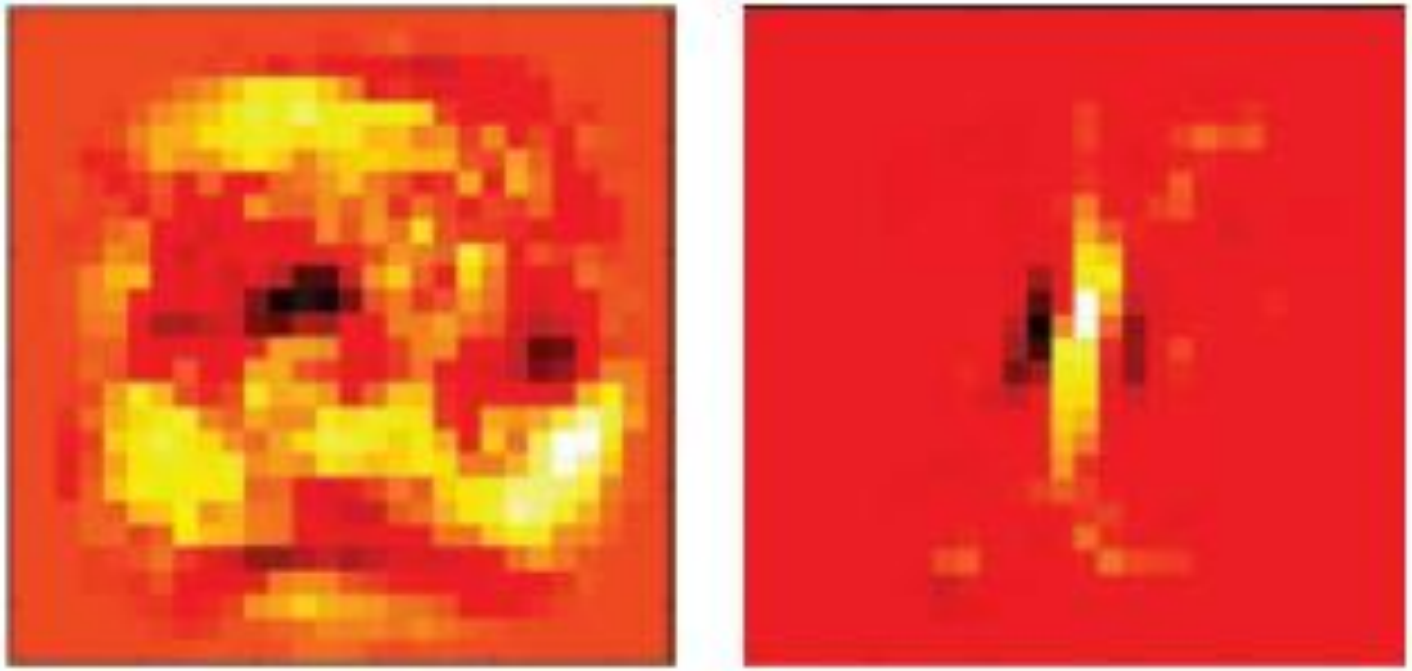
What does your network look like?



What does the machine see



What does the machine see



Linear Algebra - Dot Product

- Dot Products
- They take two vectors
  - multiply them together (element by element)
  - finally sum each to get the final score.

```
a = [ 0, 1, 0, 1]
b = [ 1, 0, 1, 0]

    [ 0, 0, 0, 0] -> 0
```

Dot Products / Weighted Sums

```
c = [ 0, 1, 1, 0]
d = [.5, 0,.5, 0]
```

```
b = [ 1, 0, 1, 0]
c = [ 0, 1, 1, 0]
```

Dot Products / Weighted Sums