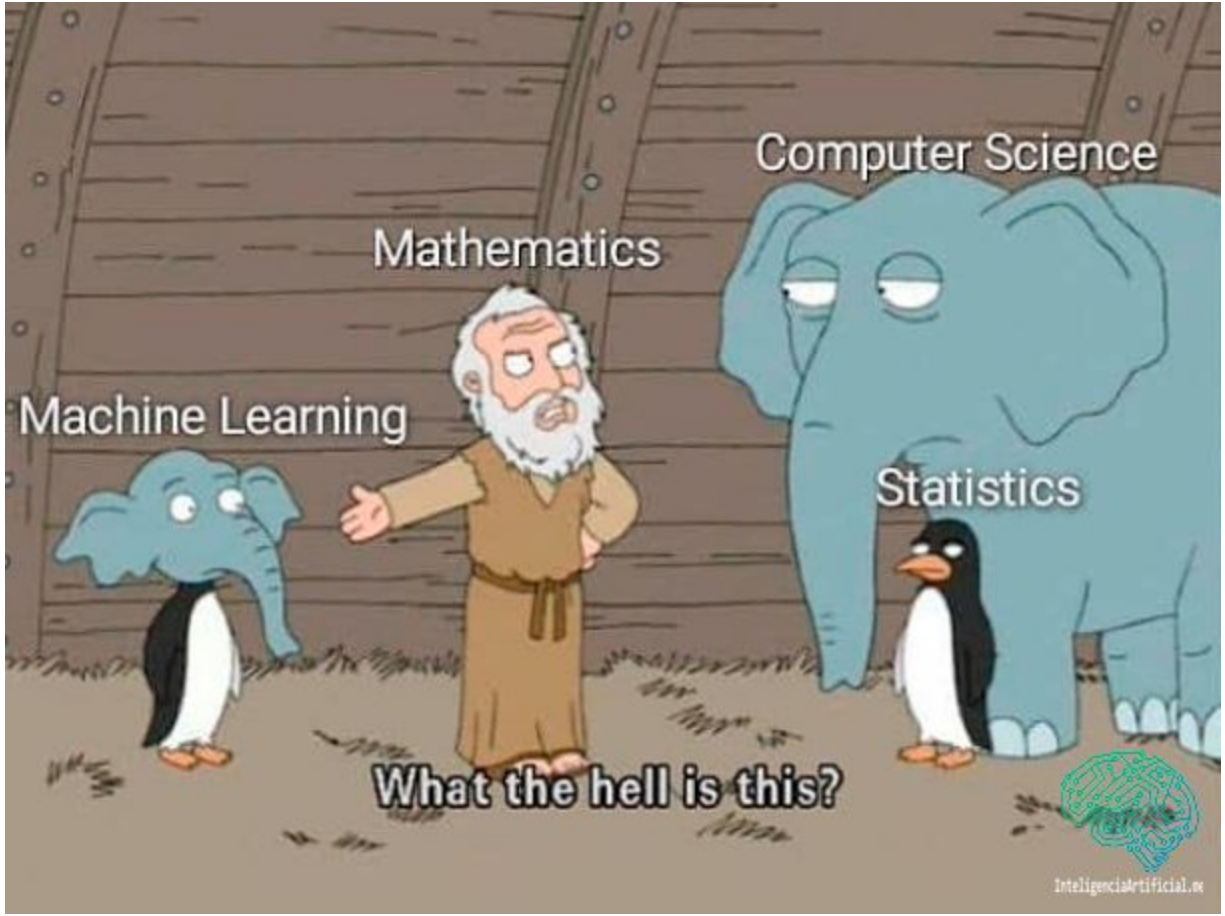
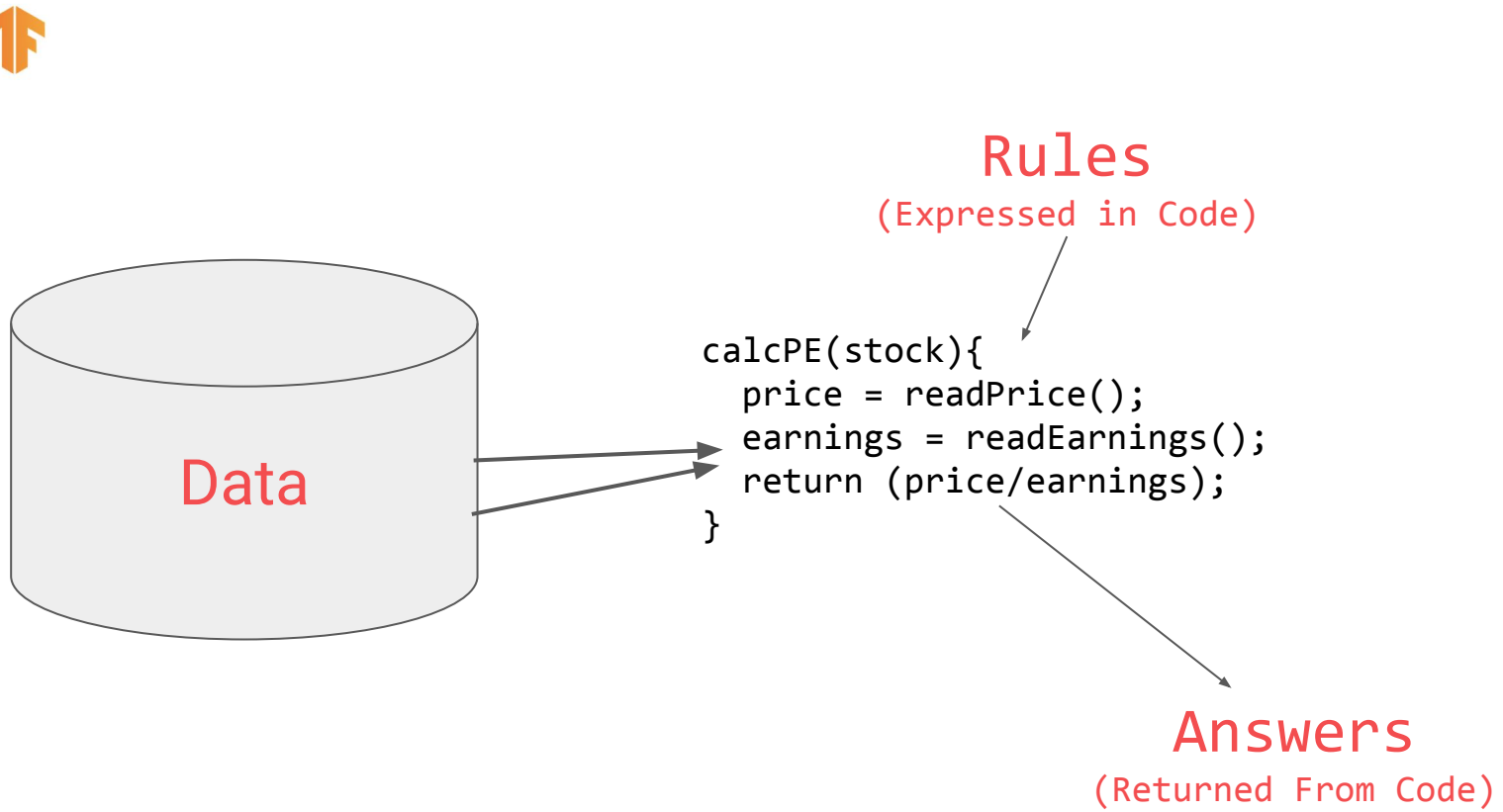
TensorFlow

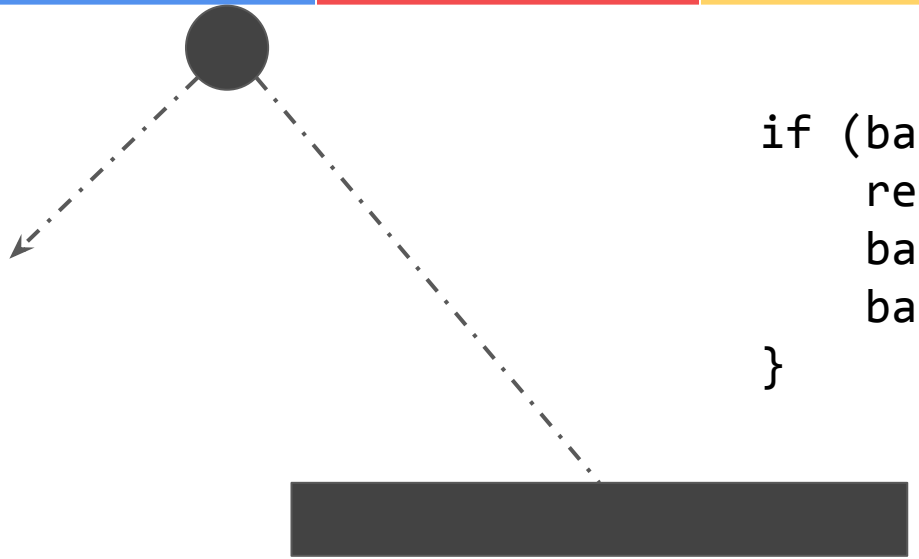
Zero to ~~Hero~~ Hinton with TensorFlow

Arnaldo Gualberto & Mikaeri Ohana

Google Developer Experts in ML

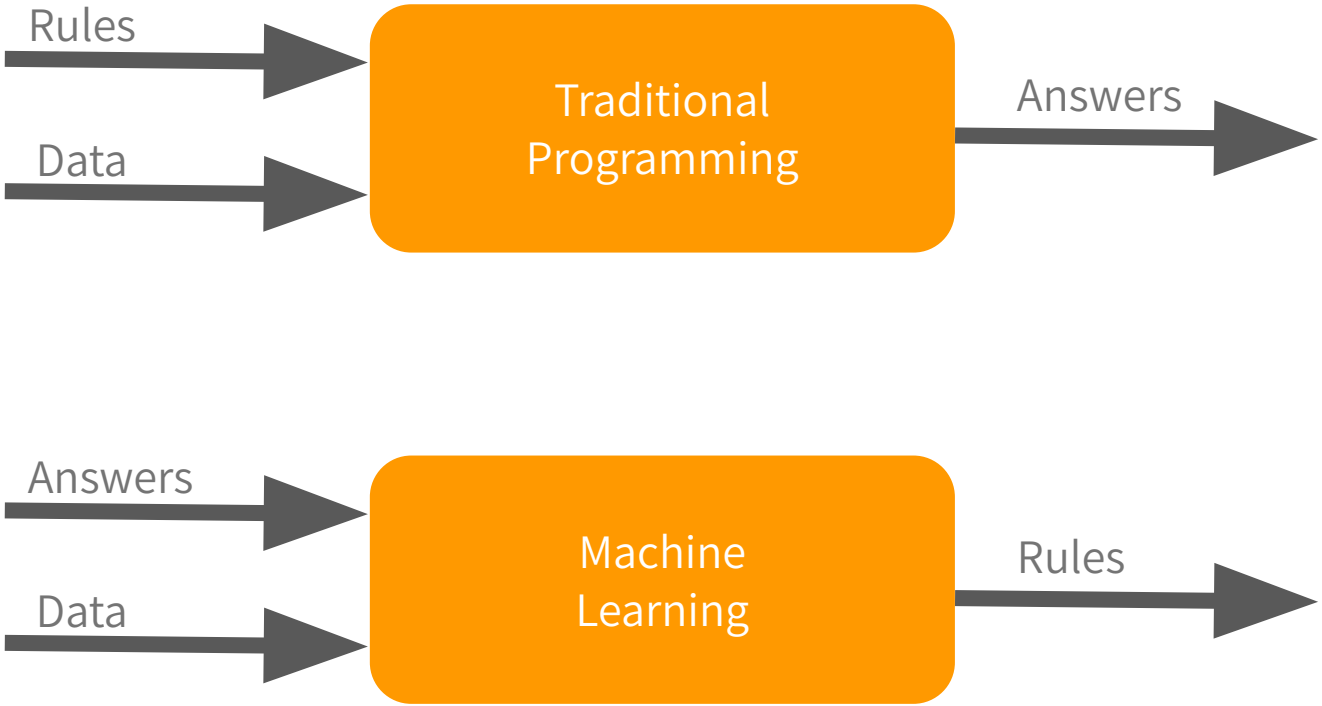

Computer Science
Mathematics
Machine Learning
Statistics
What the hell is this?
InteligenciaArtificial.mx


Rules
(Expressed in Code)
Data
calcPE(stock){
price = readPrice();
earnings = readEarnings();
return (price/earnings);
}
Answers
(Returned From Code)



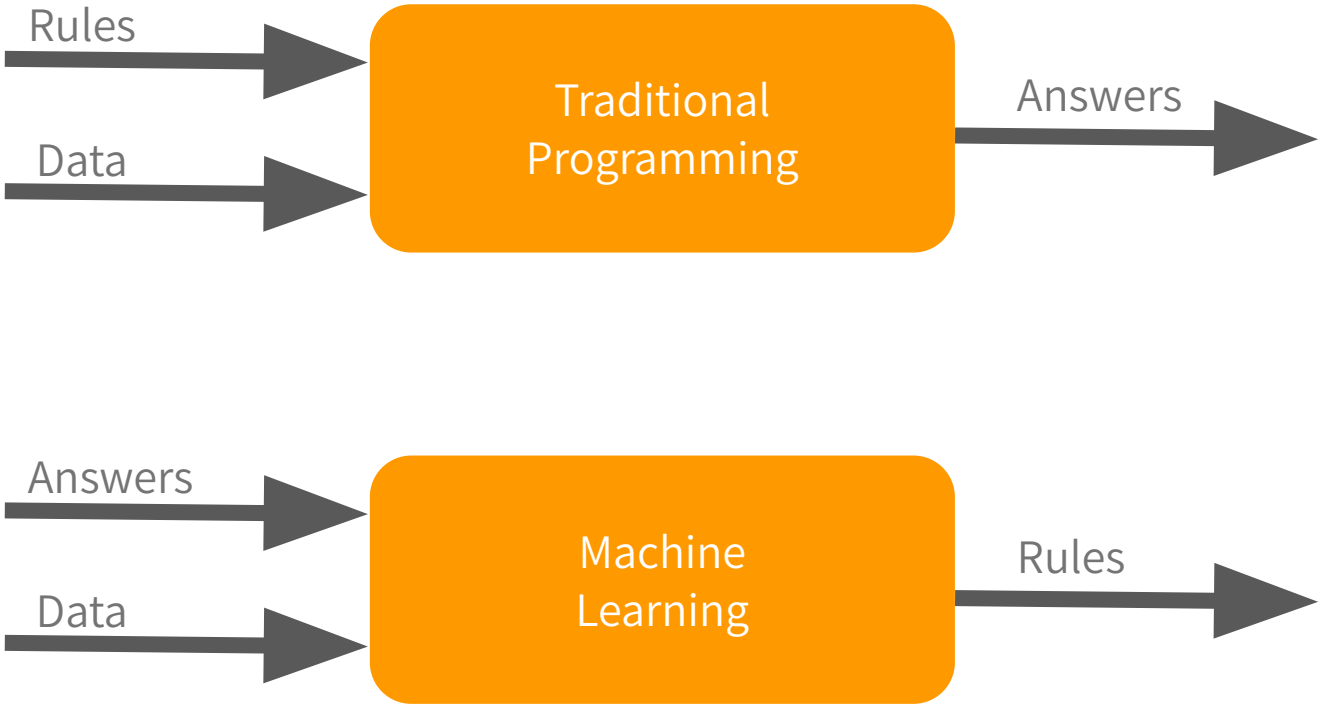
```
if (ball.collide(brick)){
    removeBrick();
    ball.dx=-1*(ball.dx);
    ball.dy=-1*(ball.dy);
}
```


Rules
Data
Traditional Programming
Answers


Rules
Data
Traditional Programming
Answers
Answers
Data
Machine Learning
Rules

Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```


Rules
Data
Traditional Programming
Answers
Answers
Data
Machine Learning
Rules

Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// ????
```

Activity Recognition

```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING

```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING

```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

Label = BIKING

```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING
(Sort of)

Activity Recognition

```
01011001010100101010
10010101010011011101
01001010100101011001
01010010101001010101
```

Label = WALKING

```
10101001010010101010
01010100100100010001
00100111110101011111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011100
10101011110101010111
11111100011110101010
```

Label = BIKING

```
11111111110100111010
00111110101111101010
01011101010101011100
10101010101001111100
```

Label = GOLFING
(Sort of)

Activity Recognition

```
01010010101010010101010
10010101010010111010
01001010100101010010
01010010101001010101
```

Label = WALKING

```
10101001010010101010
01010100100100100001
00100111110101011111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011100
10101011110101010111
11111100011110101010
```

Label = BIKING

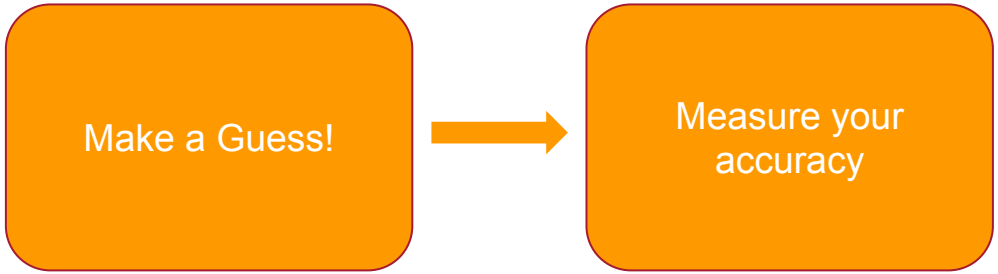
```
11111111110100111010
00111111010111110101
01011101010101011110
10101010101001111100
```

Label = GOLFING
(Sort of)

The Machine Learning Paradigm

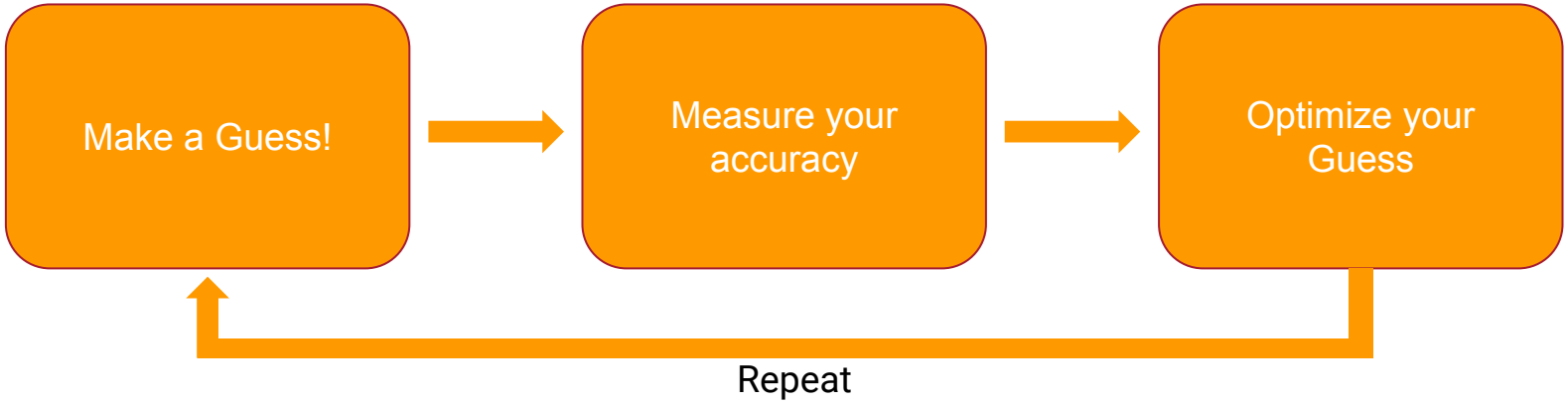
Make a Guess!

The Machine Learning Paradigm

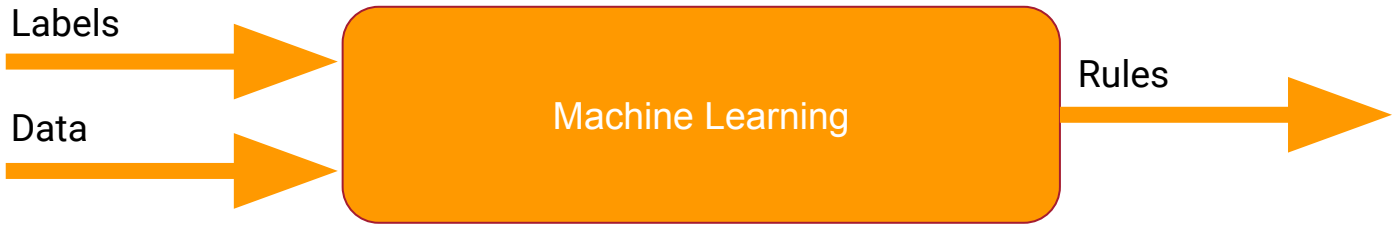


The Machine Learning Paradigm

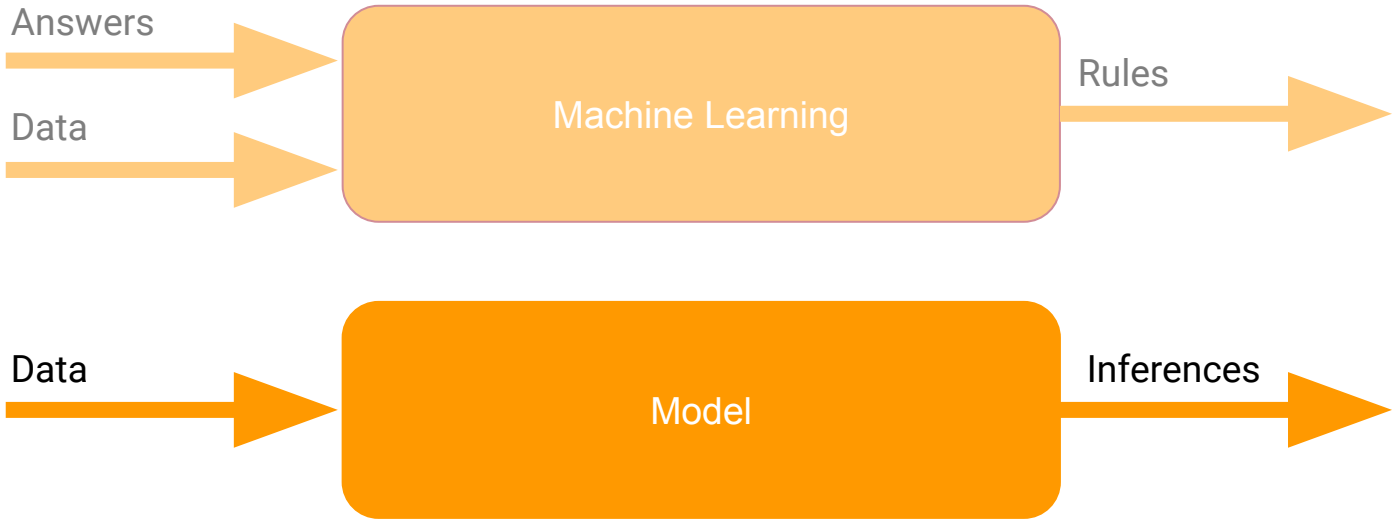
The Machine Learning Paradigm



The Machine Learning Paradigm



The Machine Learning Paradigm



TensorFlow

Computer Vision

Using Machine Learning to understand images

hit the button

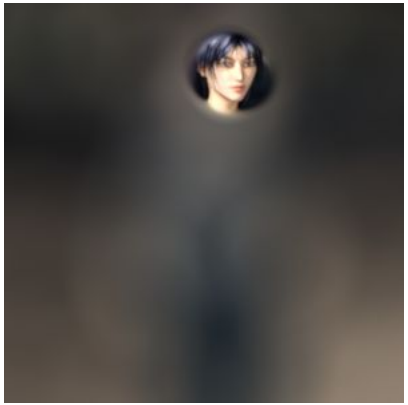

WHEN DID YOU BECOME AN EXPERT IN COMPUTER VISION?
LAST NIGHT
memegenerator.net

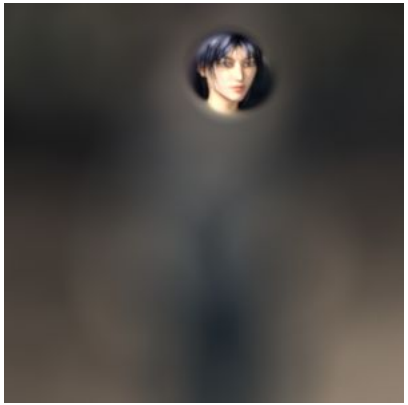

WHEN DID YOU BECOME AN EXPERT IN COMPUTER VISION?
LAST NIGHT
memegenerator.net

TODAY

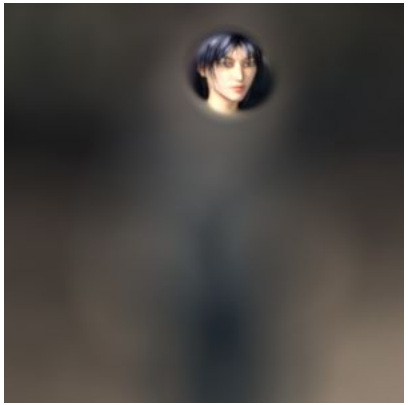
+

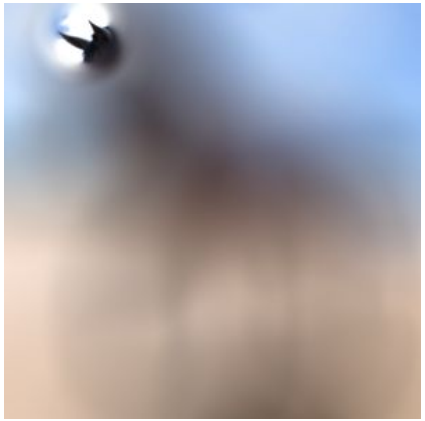
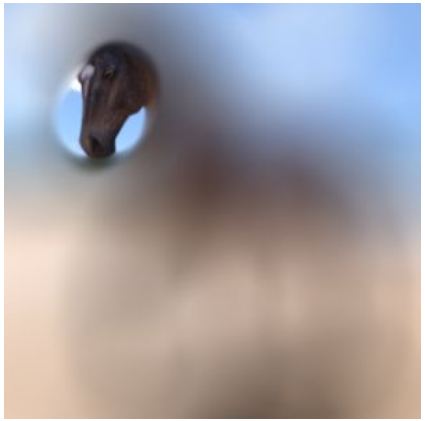

+

+


 +  +  = HUMAN

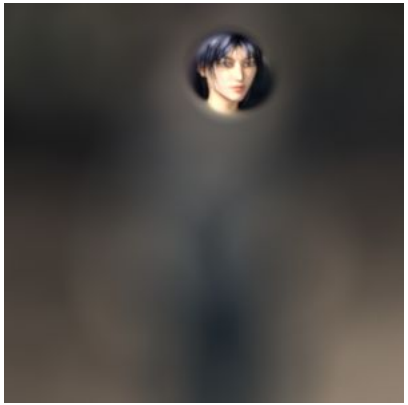

+

+

=
HUMAN


+

+

=
HORSE

+

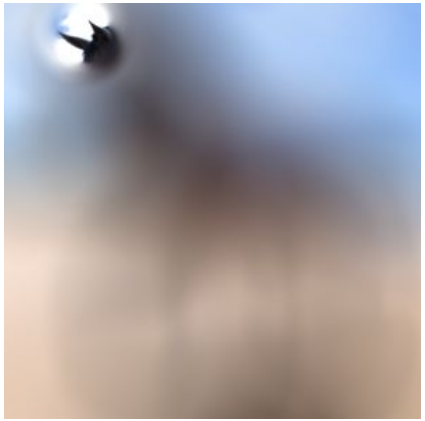
+



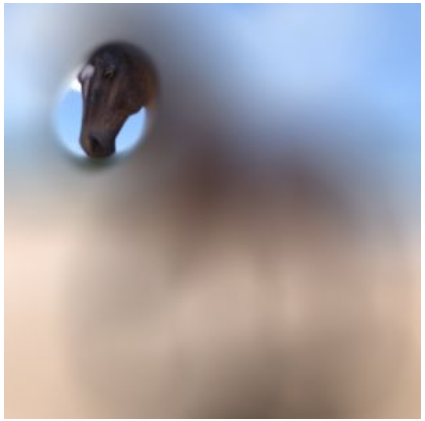
=

HUMAN

+



+

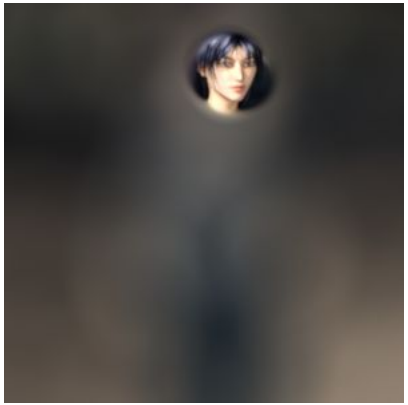


=

HORSE

+

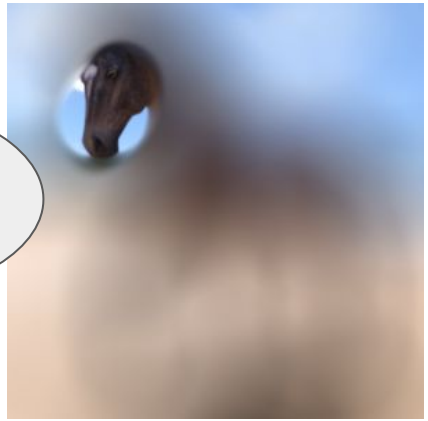
+



=

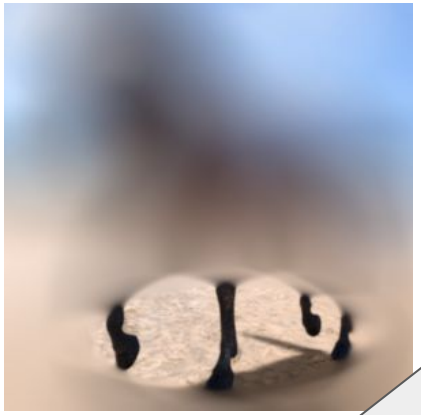
HUMAN

+

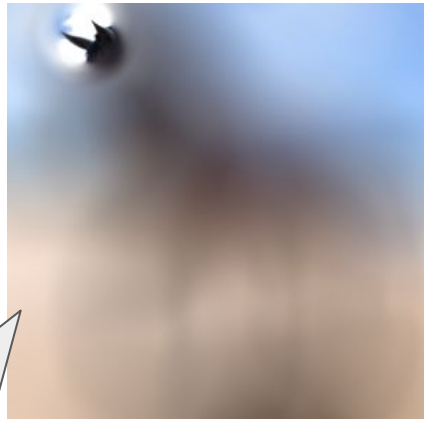


=

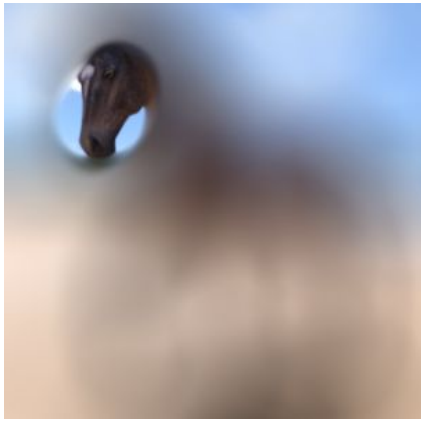
HORSE



+



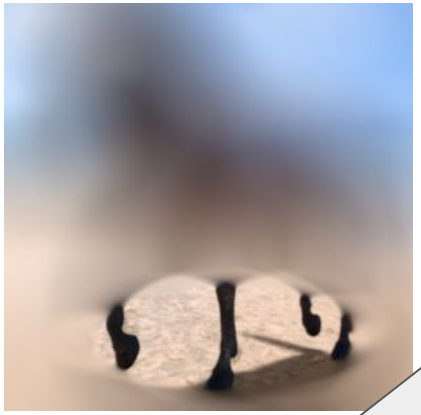
+



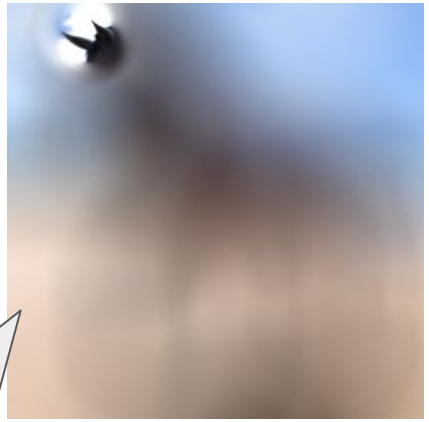
=

HORSE

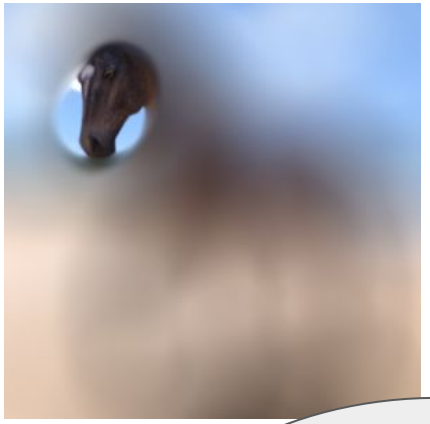
Filters can then be combined with labels to make a prediction of the image contents...



+



+



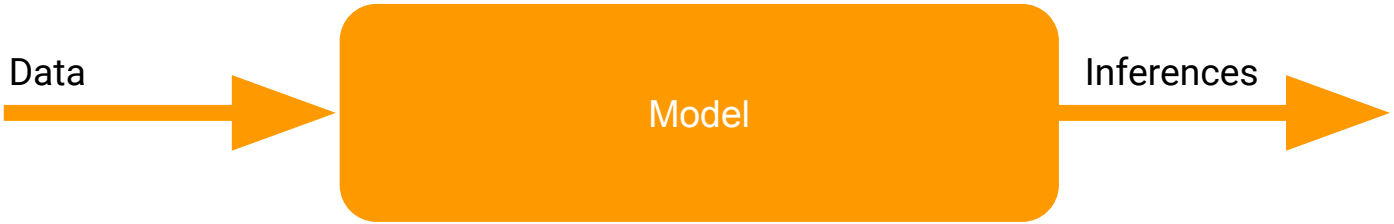
=

HORSE

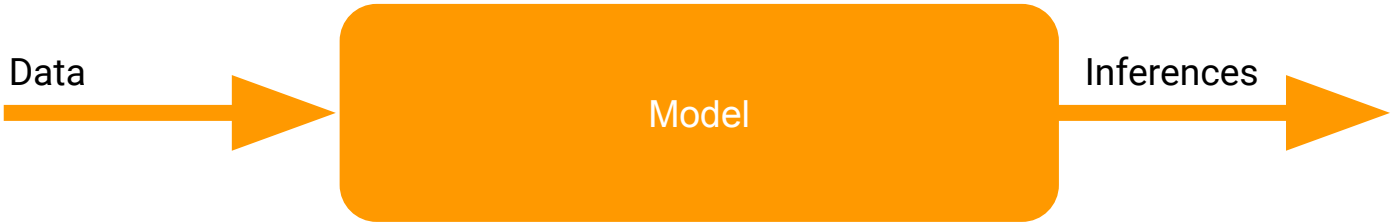
Filters can then be combined with labels to make a prediction of the image contents...

The filters that match the label are learned over time!


+
+
=
?

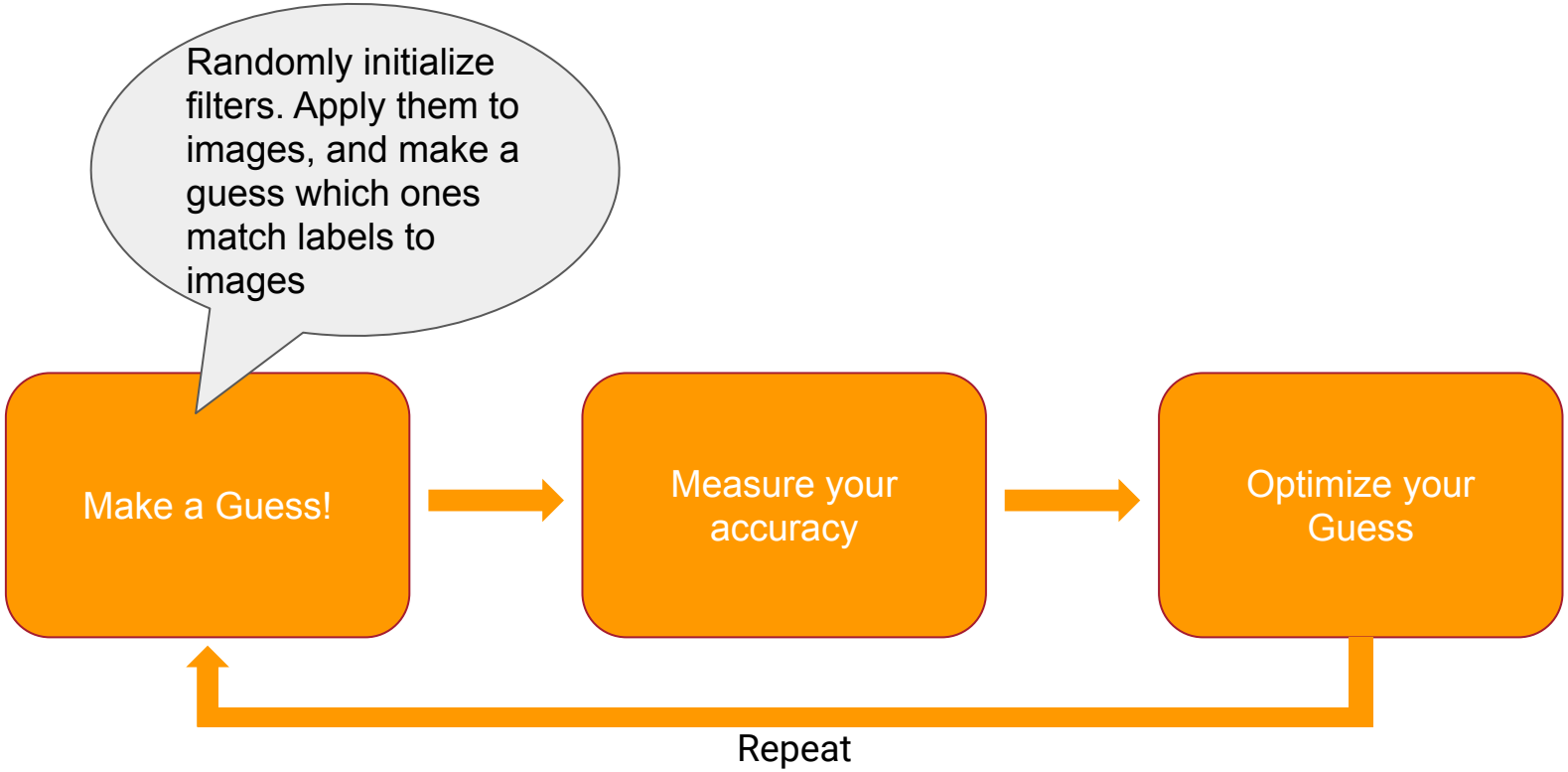

Data
Model
Inferences


+
+
=
HUMAN

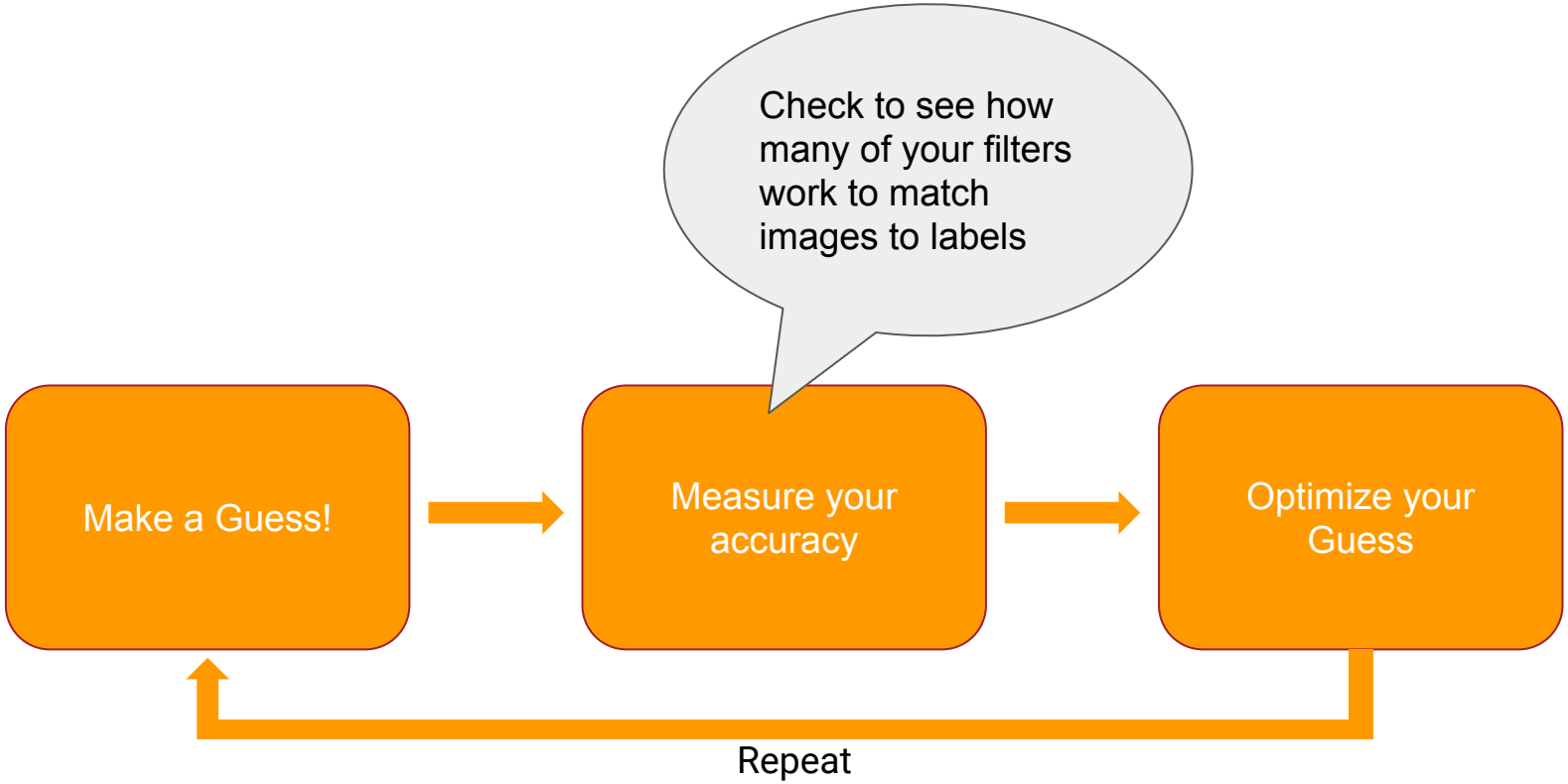

Data
Model
Inferences

The Machine Learning Paradigm

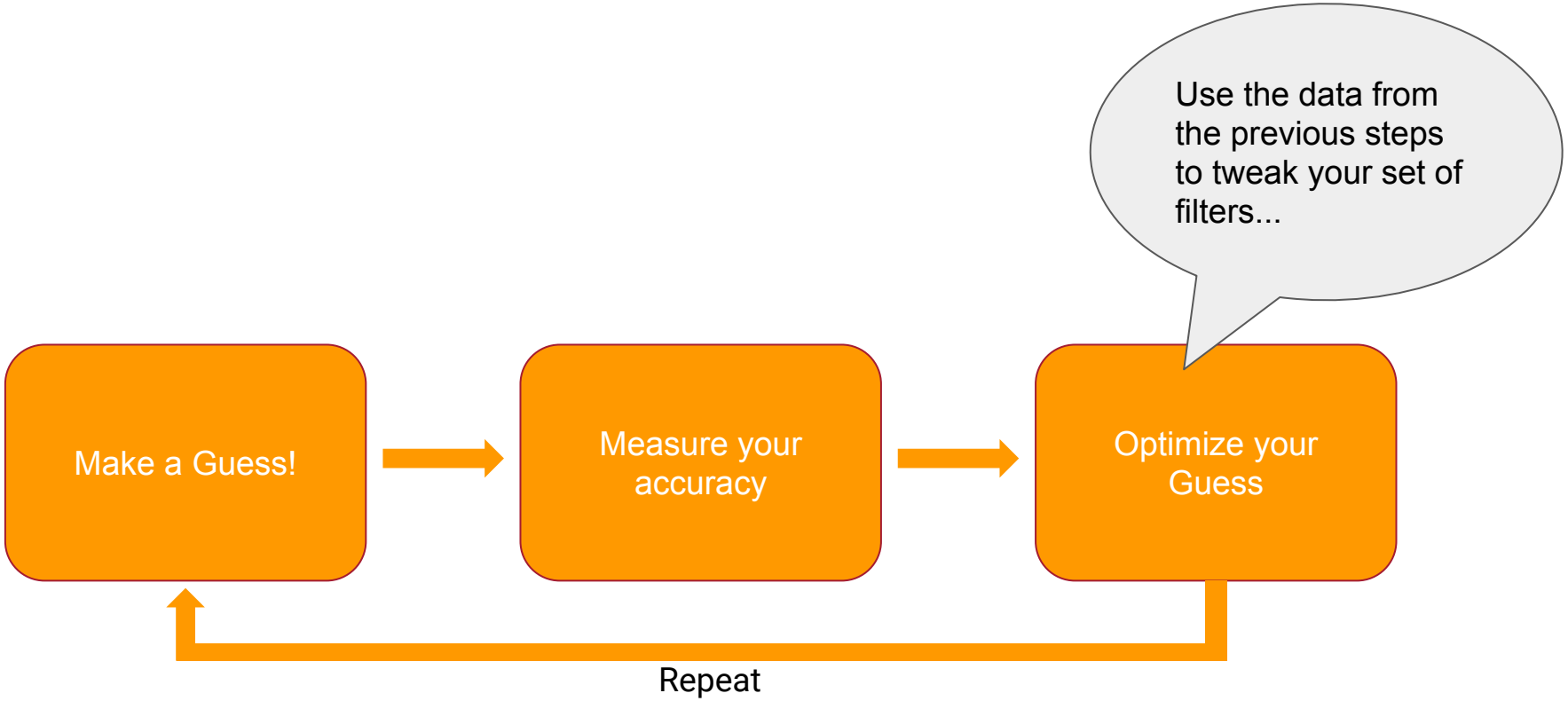
The Machine Learning Paradigm



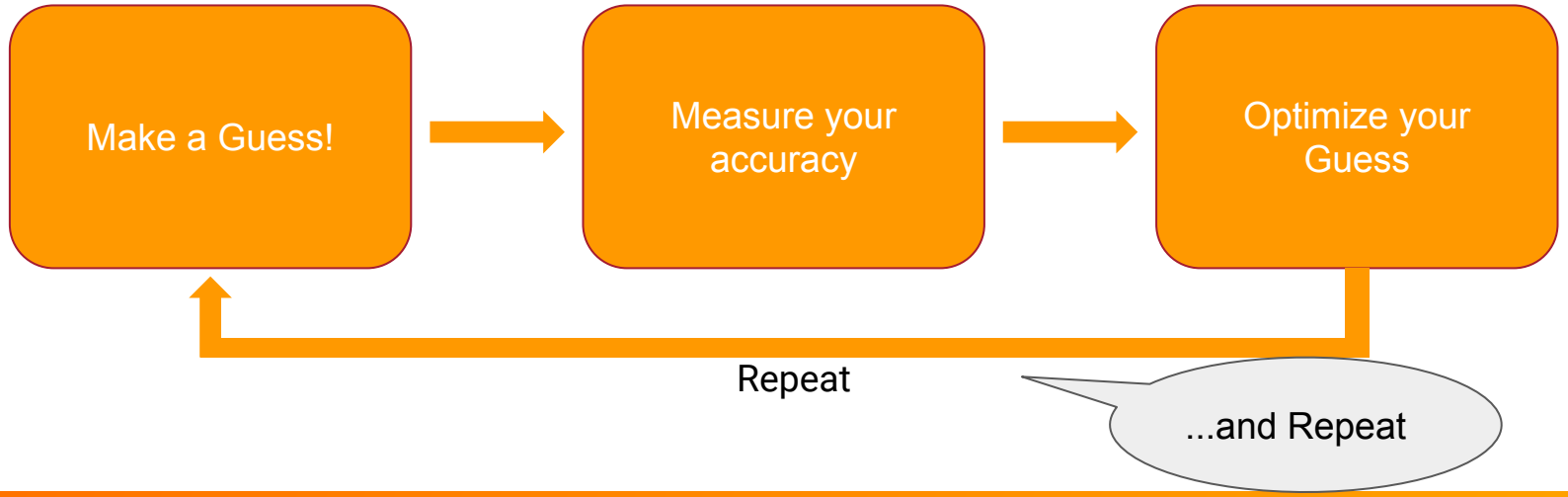
The Machine Learning Paradigm

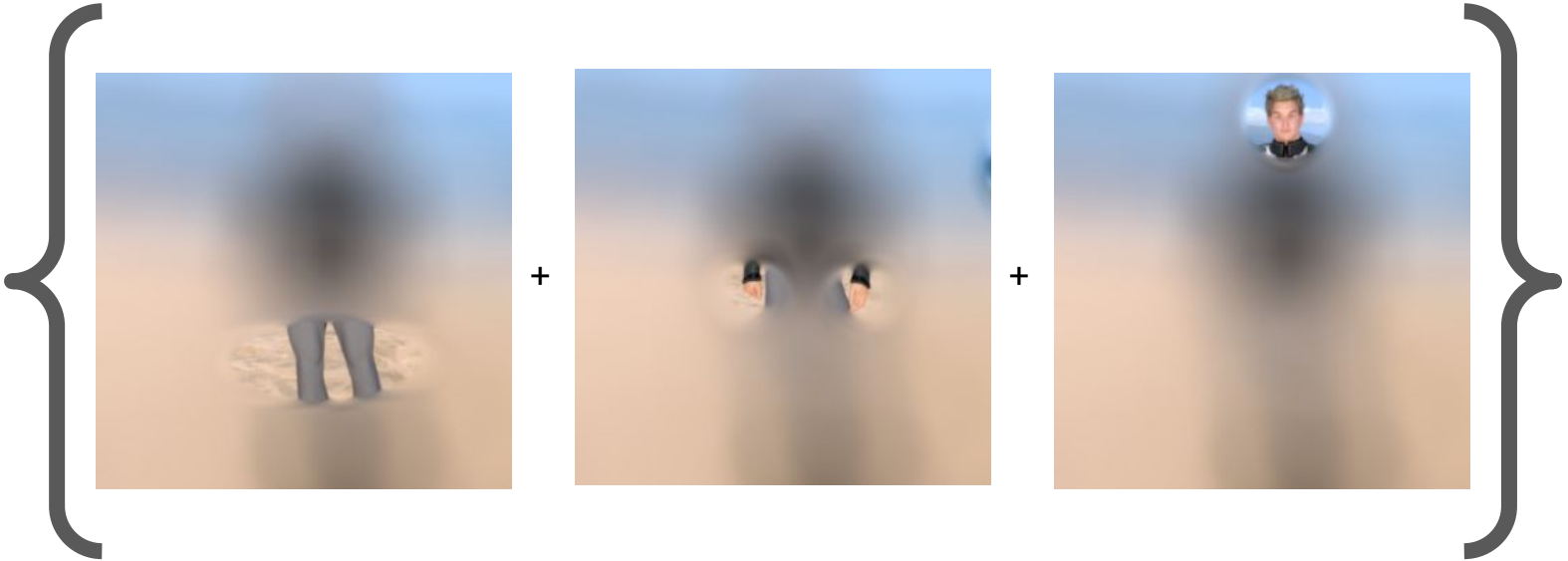


The Machine Learning Paradigm

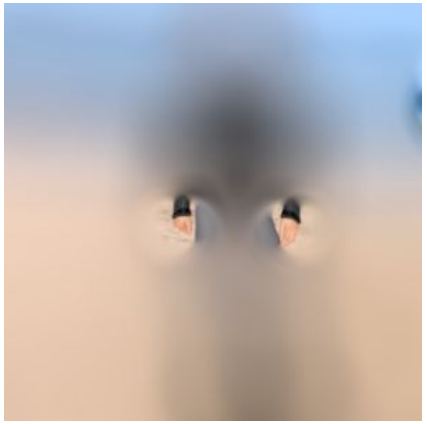


The Machine Learning Paradigm

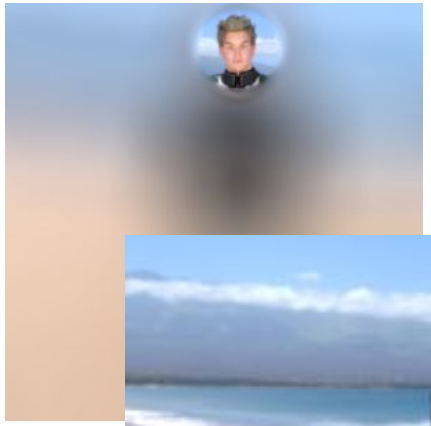



+
+
=
HUMAN

+



+



=

HUMAN

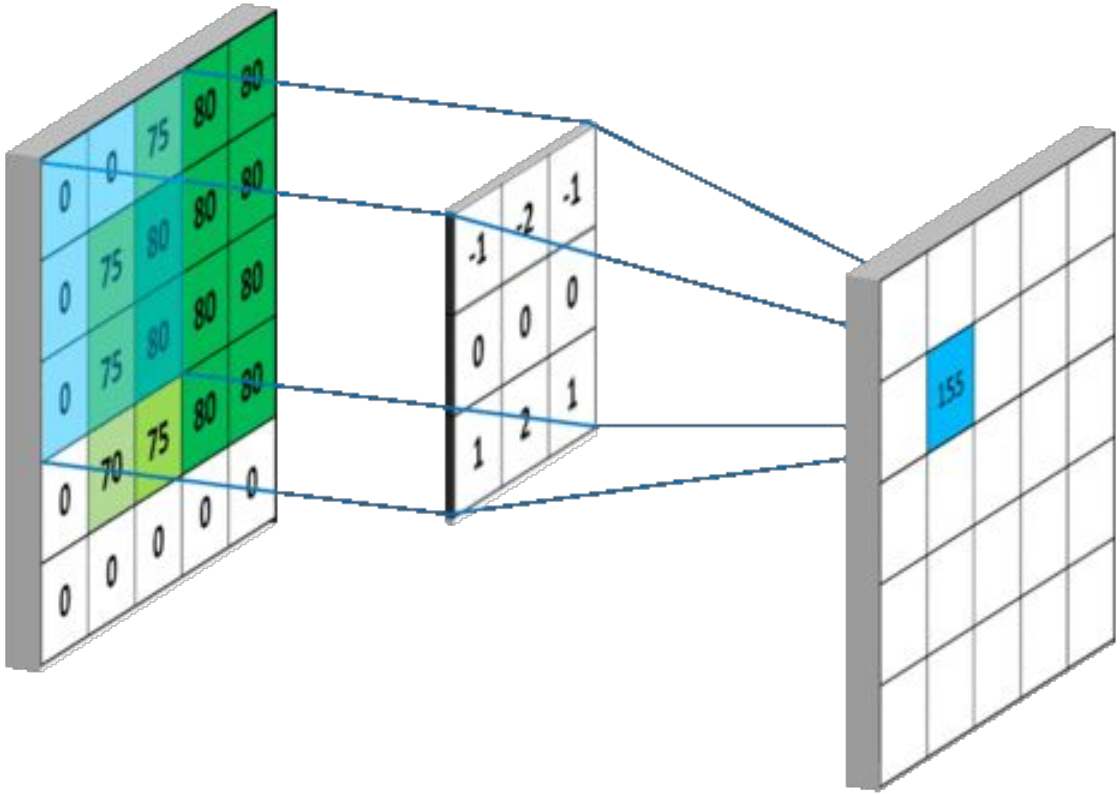

●●●○○ Verizon
4:20 PM
76%
Albums
chihuahua or muffin
Select
@teenybiscuit


●●●●○ Verizon LTE
3:02 PM
42%
Albums
sheepdog or mop
Select

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

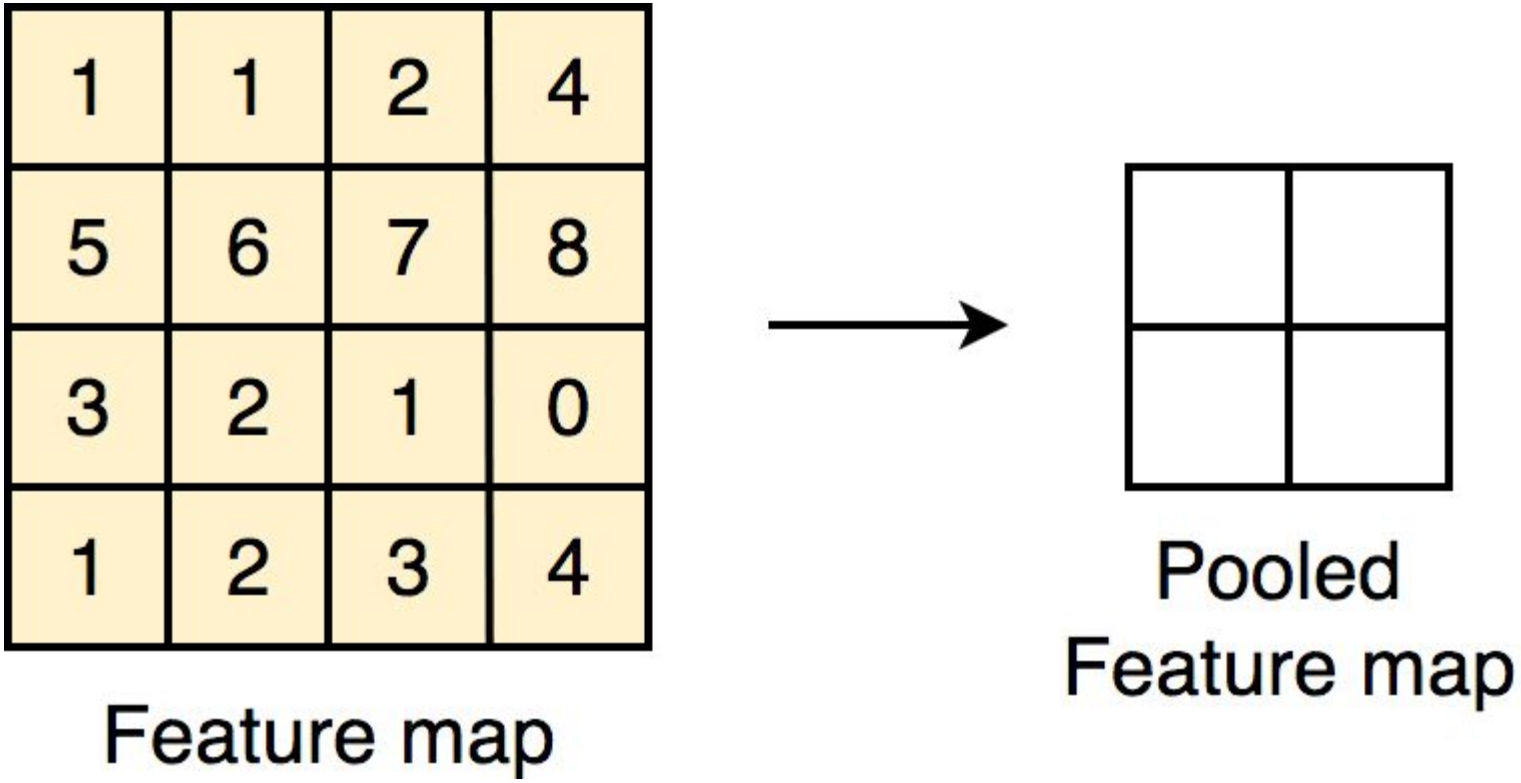
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Conv2D stands for 2D Convolution -- another word for a filter


0 0 75 80 80
0 75 80 80 80
0 75 80 80 80
0 70 75 80 80
0 0 0 0 0
-1 -2 -1
0 0 0
1 2 1
155

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

MaxPooling is a way of compressing the image while enhancing features


1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
Feature map
Pooled
Feature map

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

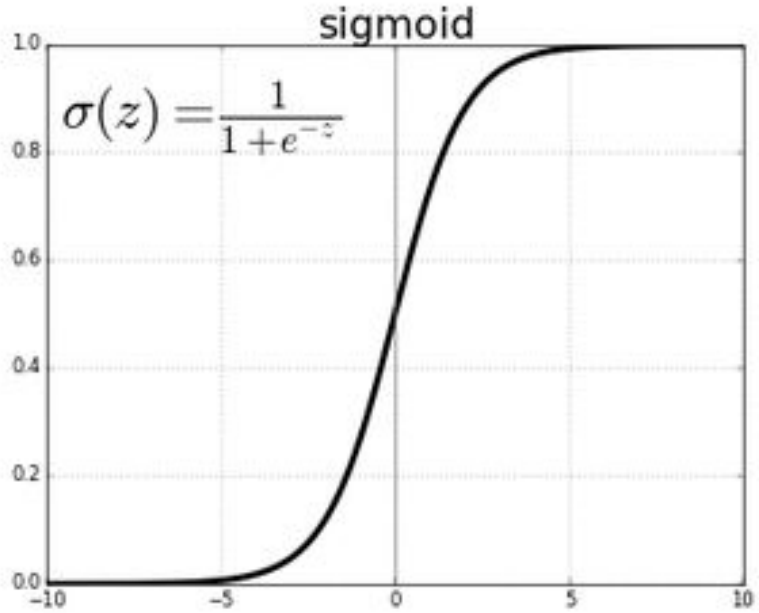
Dense is a neural network that matches the filters to the labels

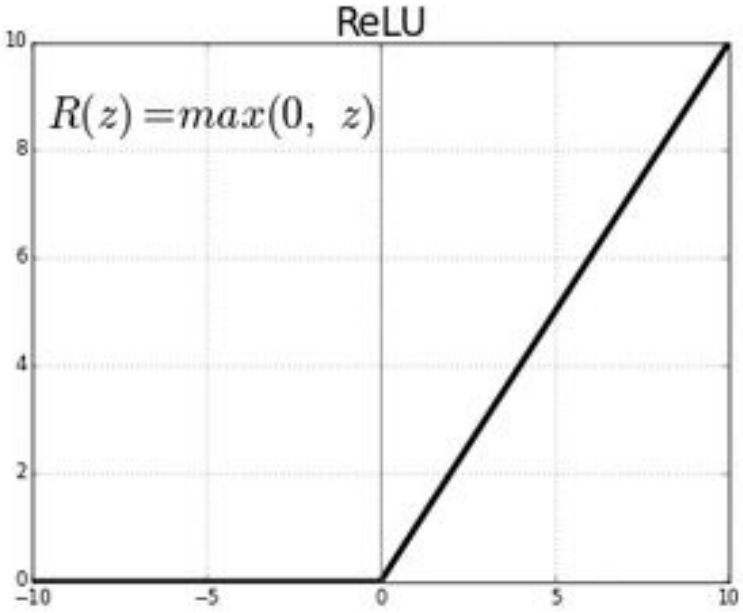
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```

The final Dense represents the labels:
Horse = 0,
Human = 1

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1,   activation='sigmoid')
])
```

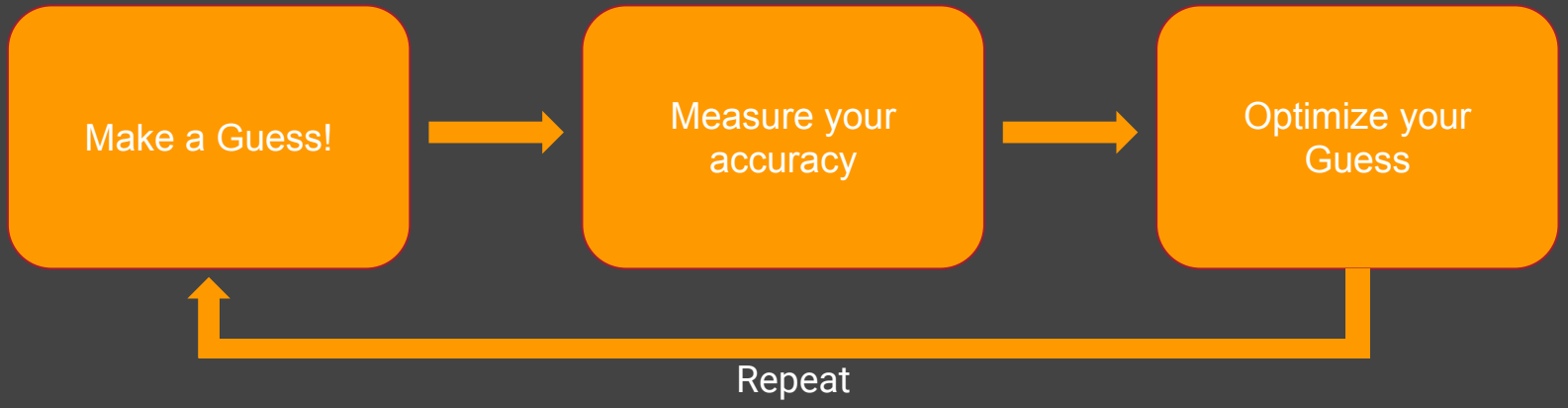
ReLU and Sigmoid are non-linear activation functions!


sigmoid
$\sigma(z) = \frac{1}{1+e^{-z}}$
1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10

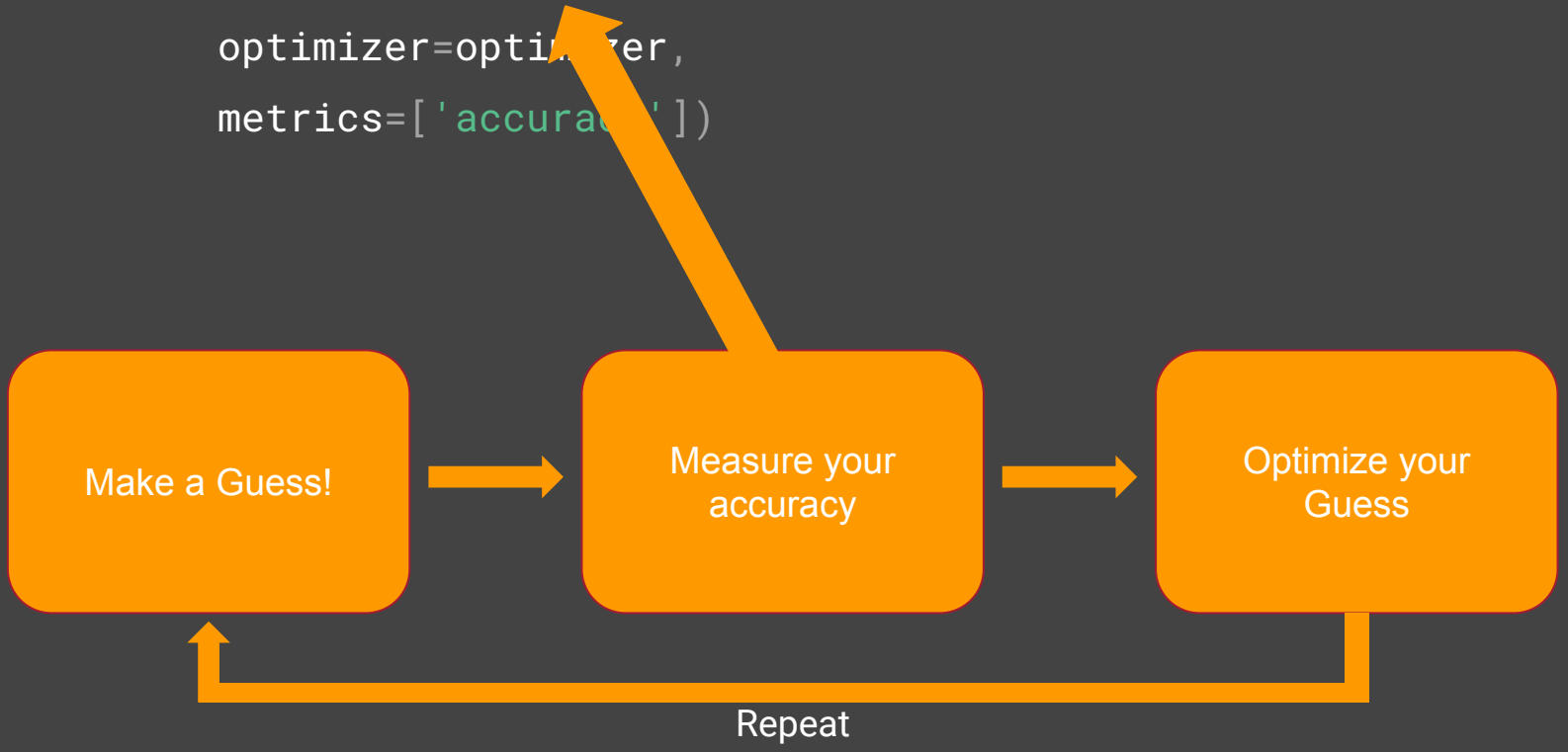

ReLU
$R(z) = max(0,\ z)$
10
8
6
4
2
0
−10
−5
0
5
10

```
optimizer = tf.keras.optimizers.RMSprop(lr=0.001)
model.compile(loss='binary_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```

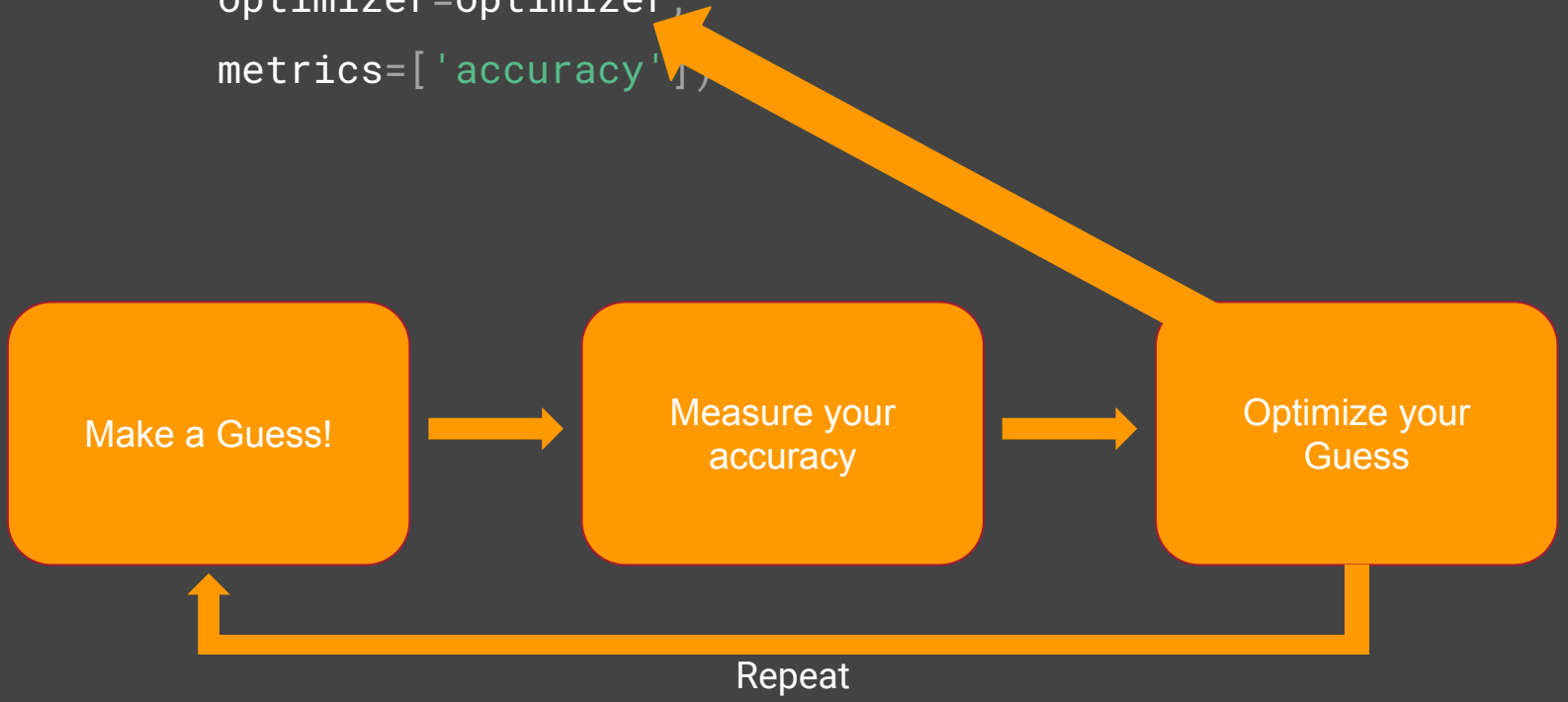
```
optimizer = tf.keras.optimizers.RMSprop(lr=0.001)
model.compile(loss='binary_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```



```
optimizer = tf.keras.optimizers.RMSprop(lr=0.001)
model.compile(loss='binary_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```

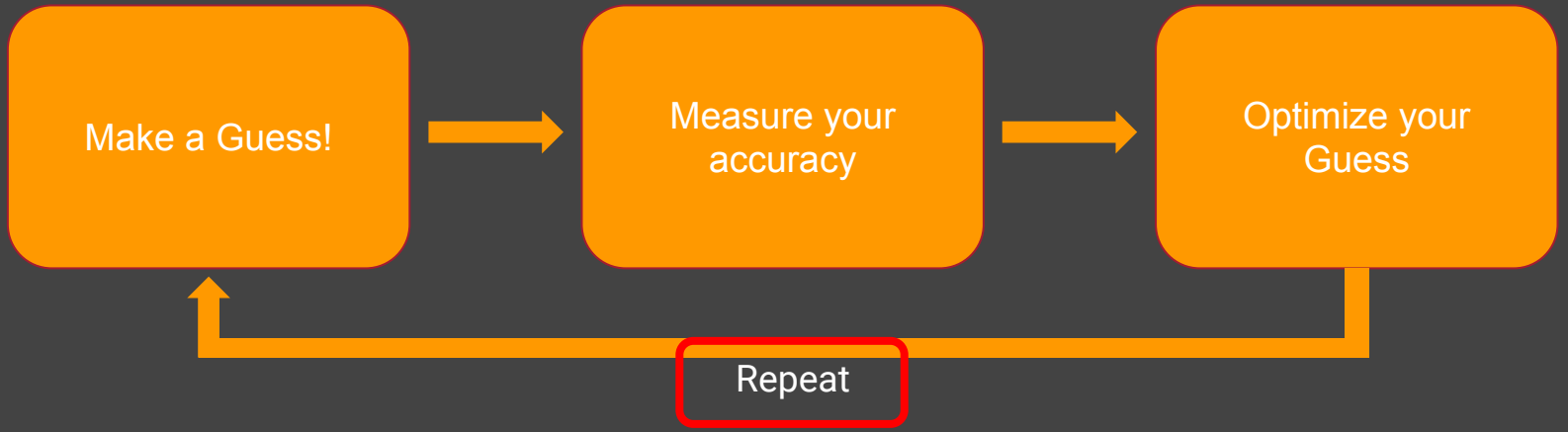


```
optimizer = tf.keras.optimizers.RMSprop(lr=0.001)
model.compile(loss='binary_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```



```
model.fit(train_generator, epochs=12, ...)
```

```
model.fit(train_generator, epochs=12, ...)
```



TensorFlow

Demo

Training a computer to recognize Horses or Humans

TensorFlow

Thank you!

@miohana

@miohana

@mikaeriohana

@mikaeriohana

@arnaldo.g12

@arnaldog12

@arnaldog12_

@arnaldog12