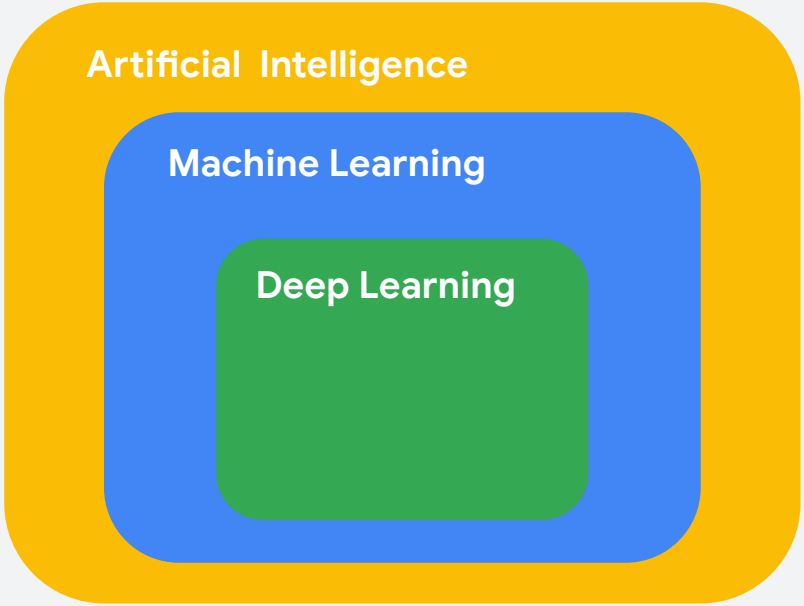
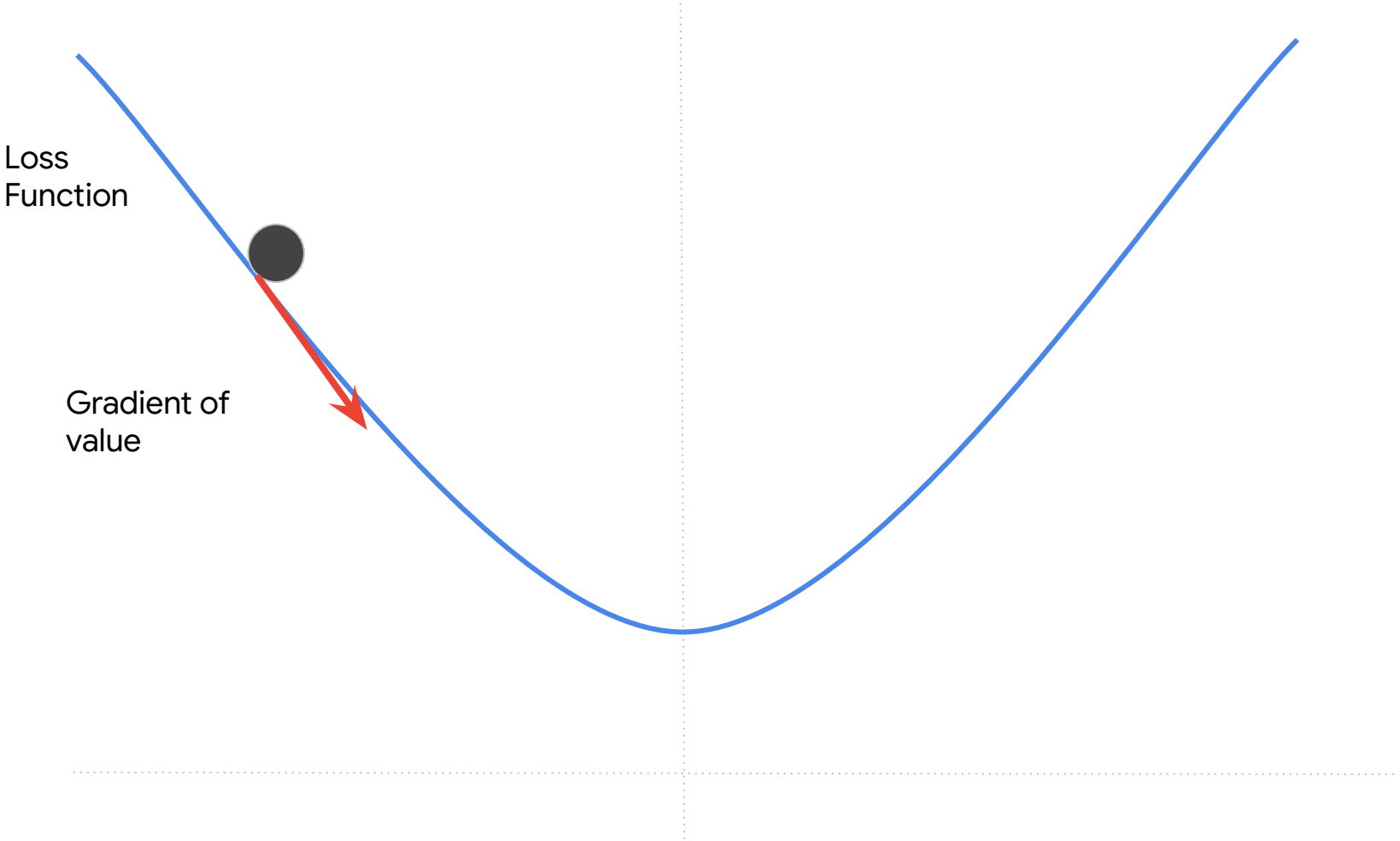
What We Have Learned Thus Far

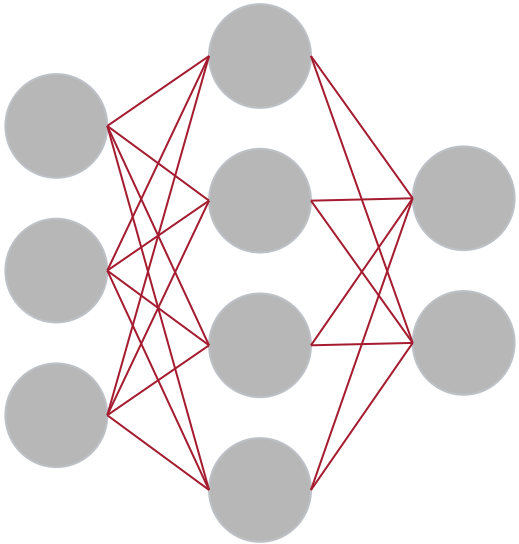
What is (Deep) Machine Learning?

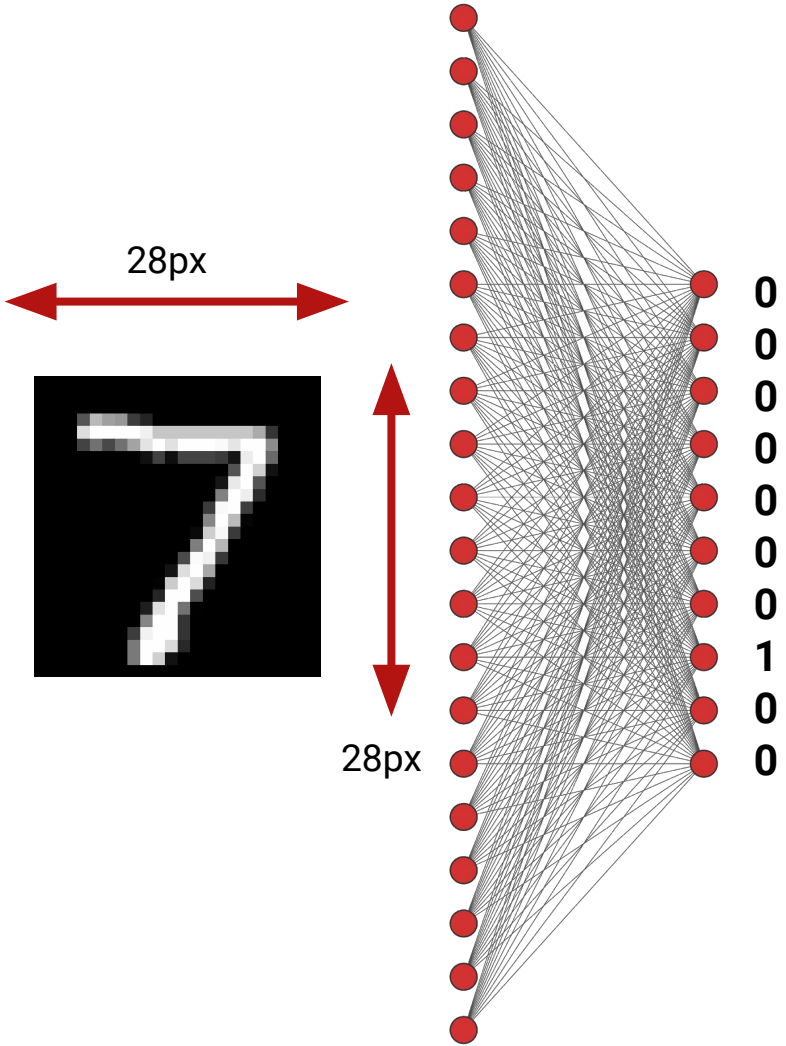
1. **Machine Learning** is a subfield of **Artificial Intelligence** focused on developing algorithms that learn to **solve problems by analyzing data for patterns**
2. **Deep Learning** is a type of Machine Learning that leverages **Neural Networks** and **Big Data**



The Machine Learning Paradigm


Loss
Function
Gradient of
value





0 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

1 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]

2 [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

3 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

4 [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

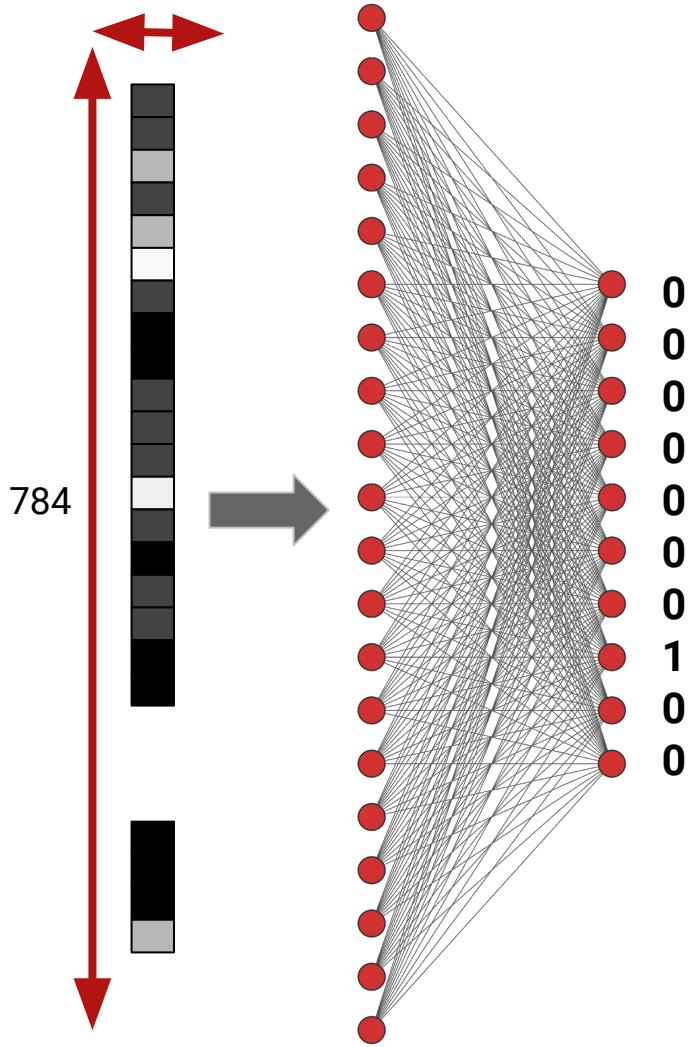
5 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

6 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

7 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

8 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]

9 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]



0 [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

1 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]

2 [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

3 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

4 [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

5 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

6 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

7 [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

8 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]

9 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

The Machine Learning Paradigm

-1	0	1
-2	0	2
-1	0	1

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu',
                           input_shape=(150, 150, 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.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

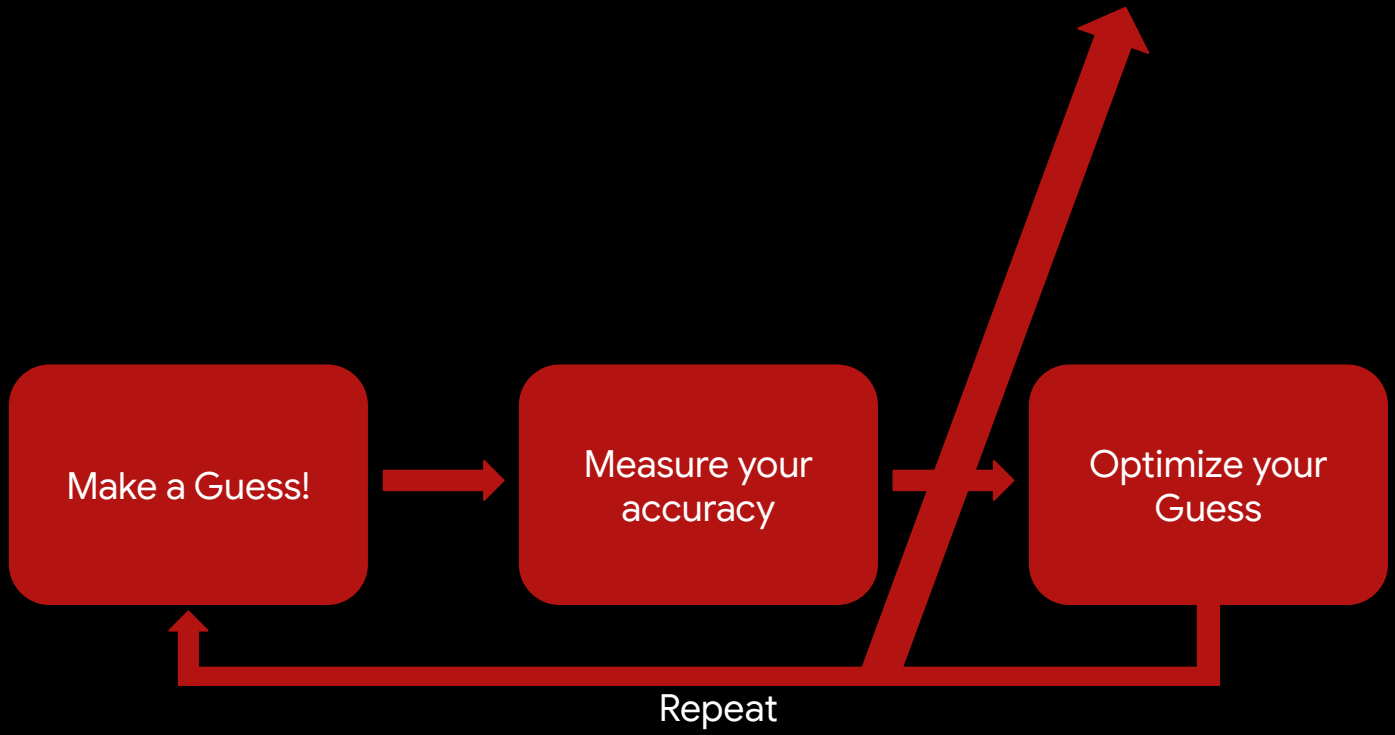
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu',
            input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64,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

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

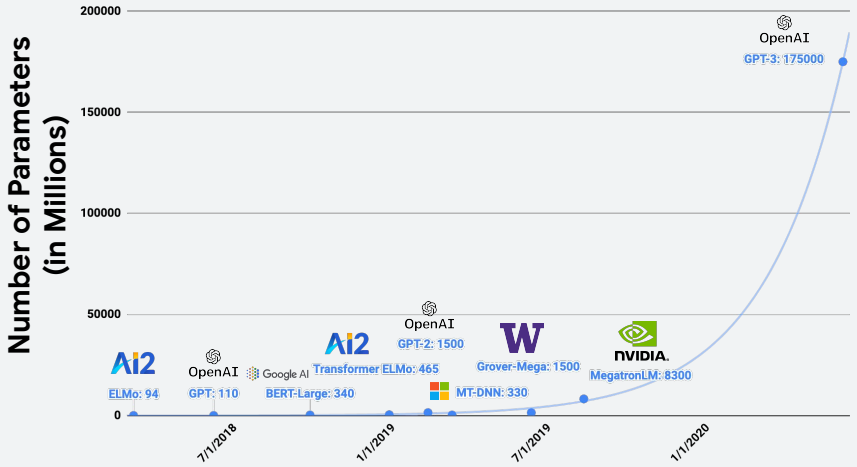


Cloud TPU

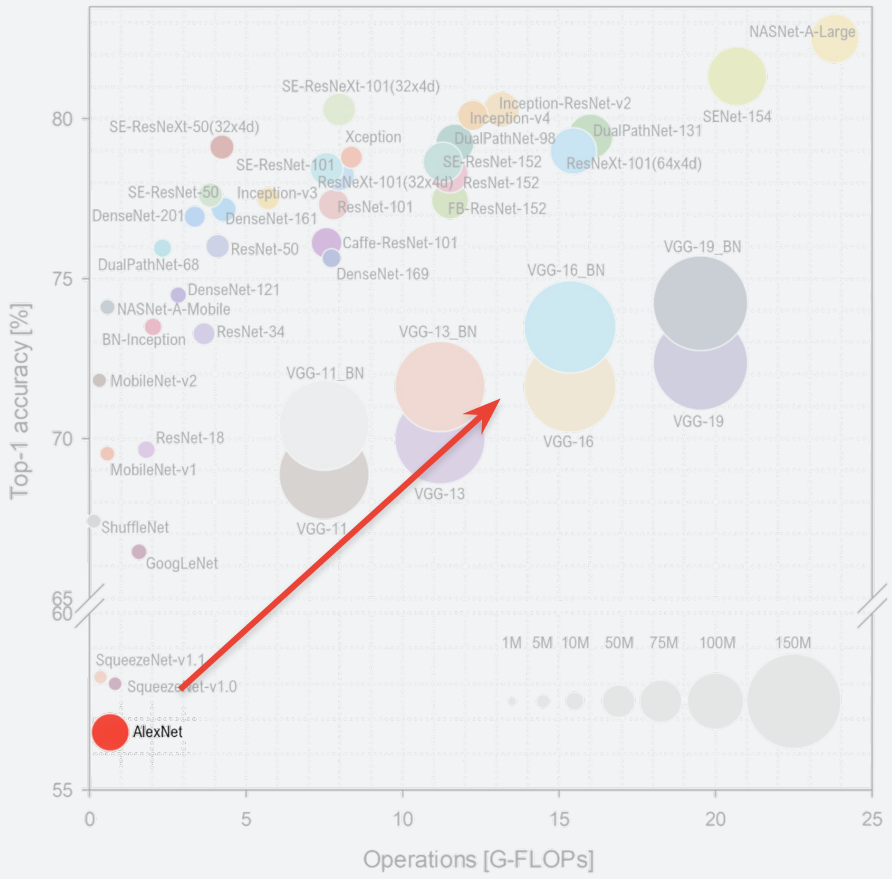
Source: Google

Source: Google

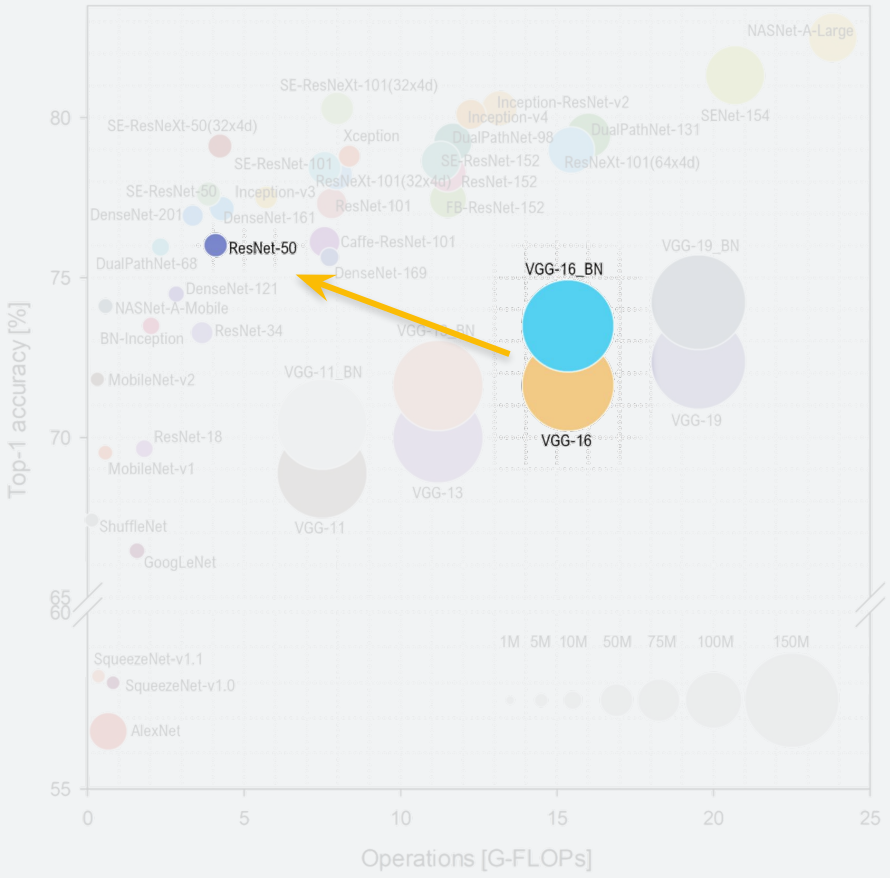
Machine Learning **Models**



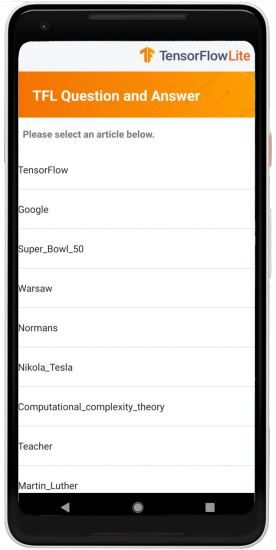
Machine Learning **Models**

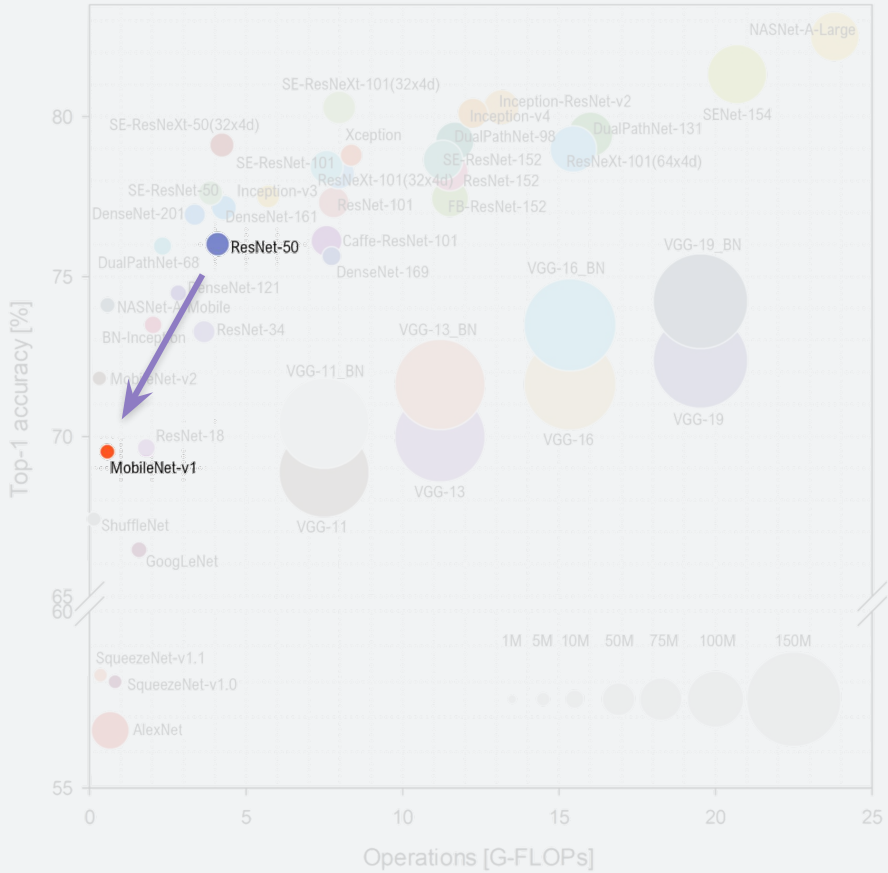


Machine Learning **Models**

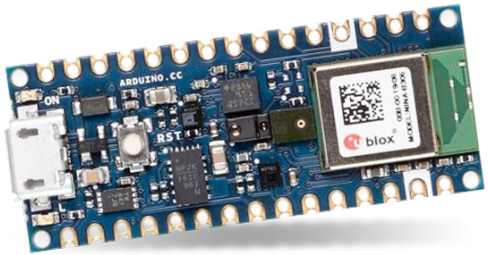


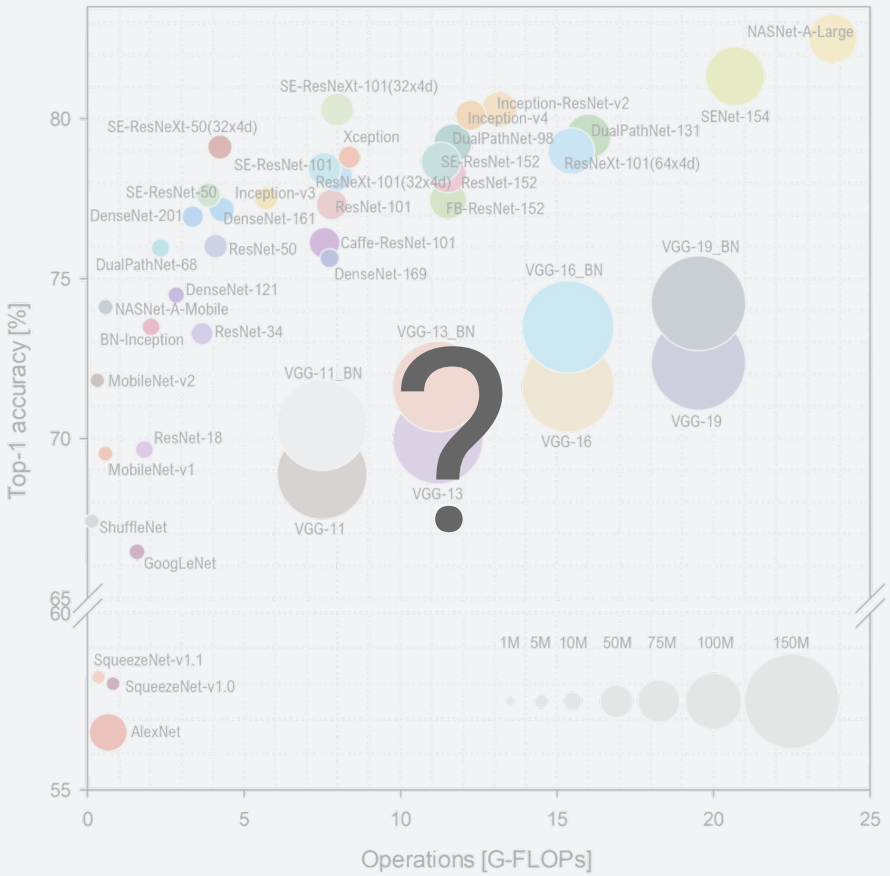
Machine Learning **Models**



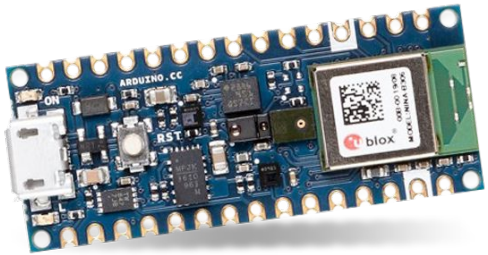


Machine Learning **Models**





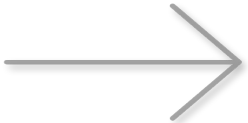
Machine Learning
Models



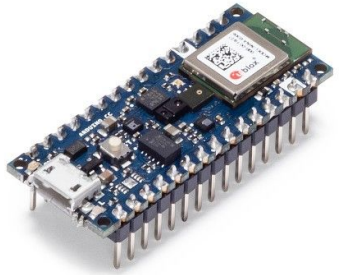
Problem:

Our board (in your kit for Course 3) only has **256KB** of RAM (memory) yet ***MobileNetv1*** needs **16.9MB**!

Cloud TPU



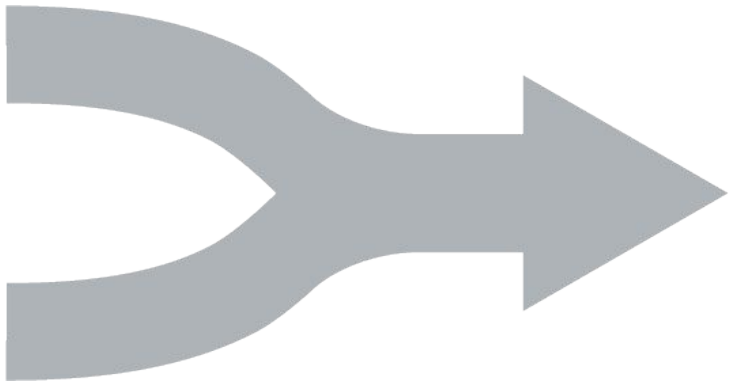
TinyML



What Makes **TinyML**?

Machine Learning

Embedded System



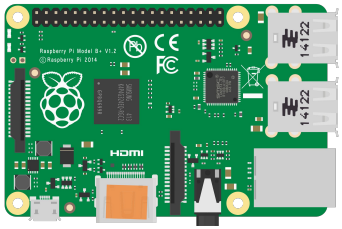
TinyML

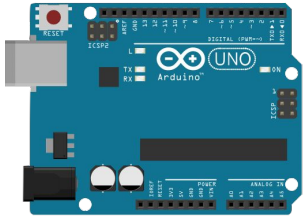
Orders of Magnitude Difference

	Microprocessor	>	Microcontroller
Platform			
Compute	1GHz–4GHz	**~10X**	1MHz–400MHz
Memory	512MB–64GB	**~10000X**	2KB–512KB
Storage	64GB–4TB	**~100000X**	32KB–2MB
Power	30W–100W	**~1000X**	150µW–23.5mW

Portability **Trade-offs**

Sacrifice portability across systems for efficiency in system performance and power efficiency



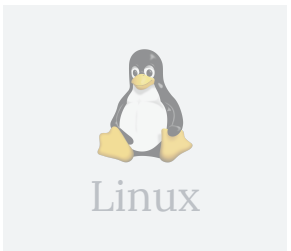


X	✓
Specific HW Implementation of a Library	

Lower Code Portability	X
Cost ($)	✓
Power (W)	✓
Eng. Effort	✓

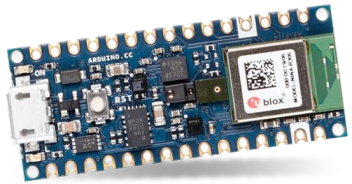
Widely Used Operating Systems

Widely Used Operating Systems

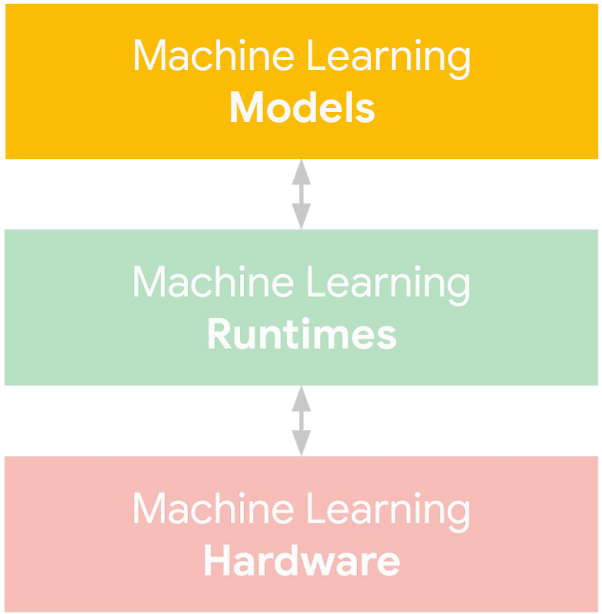


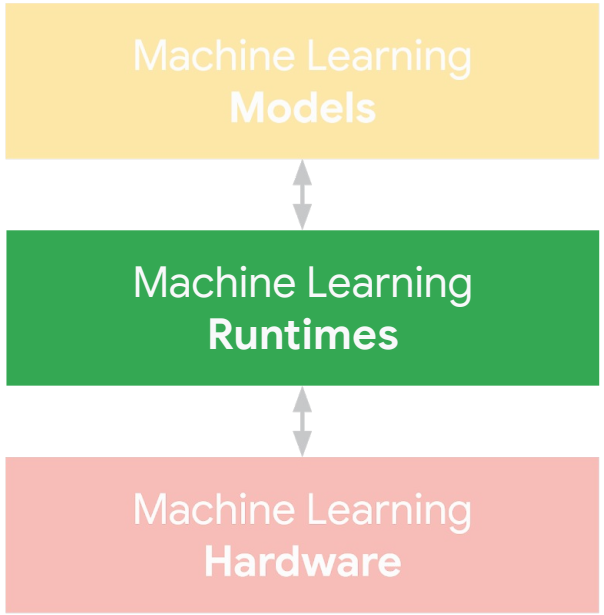
Mobile OS

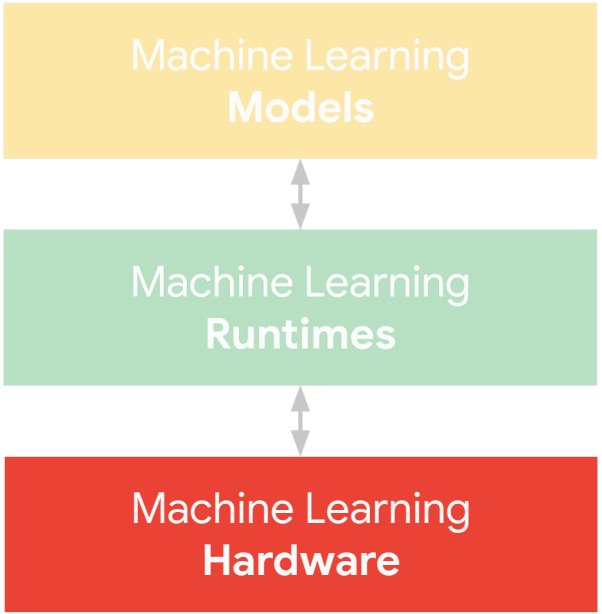
Embedded Systems



?


Machine Learning
Models
Machine Learning
Runtimes
Machine Learning
Hardware


Machine Learning
Models
Machine Learning
Runtimes
Machine Learning
Hardware


Machine Learning
Models
Machine Learning
Runtimes
Machine Learning
Hardware