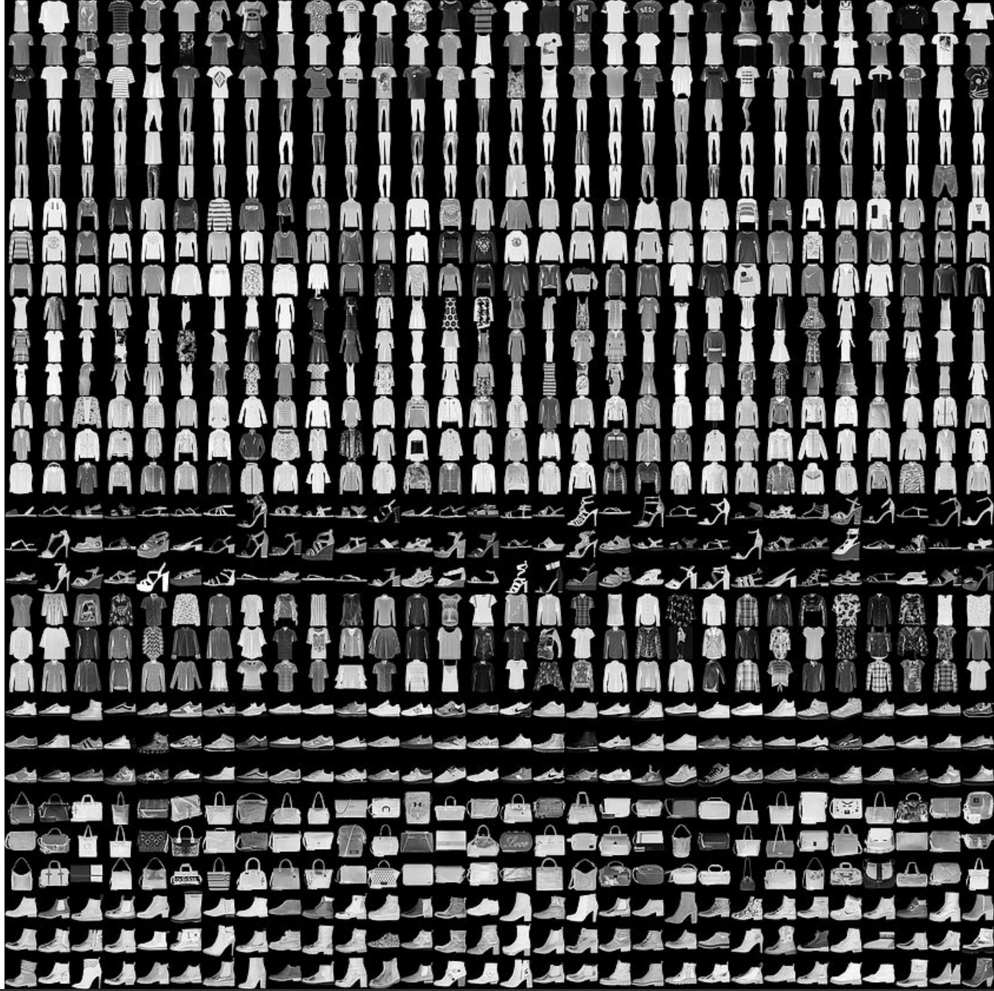
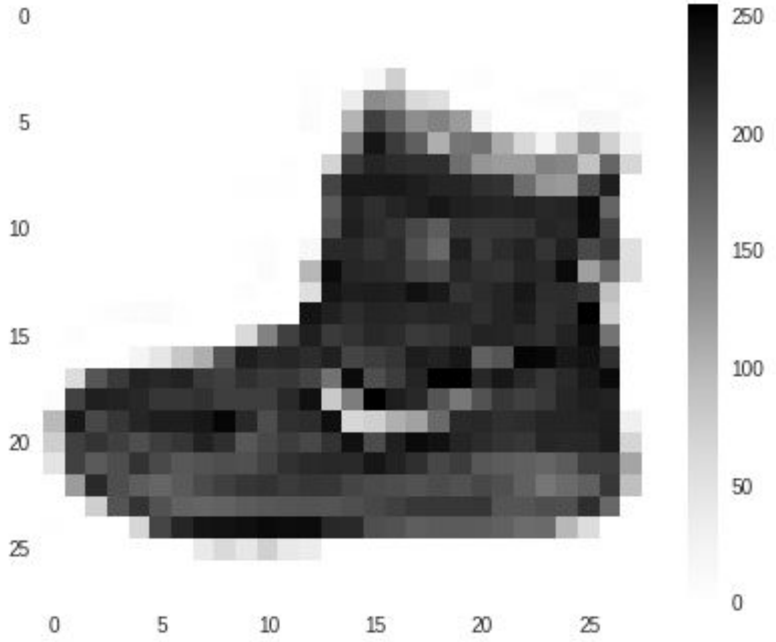
Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



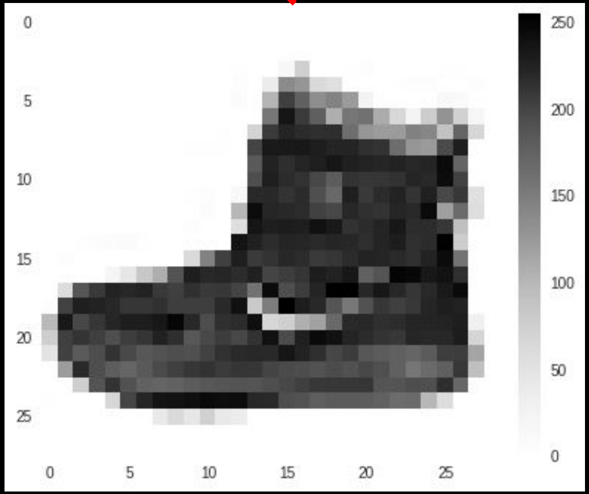
Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



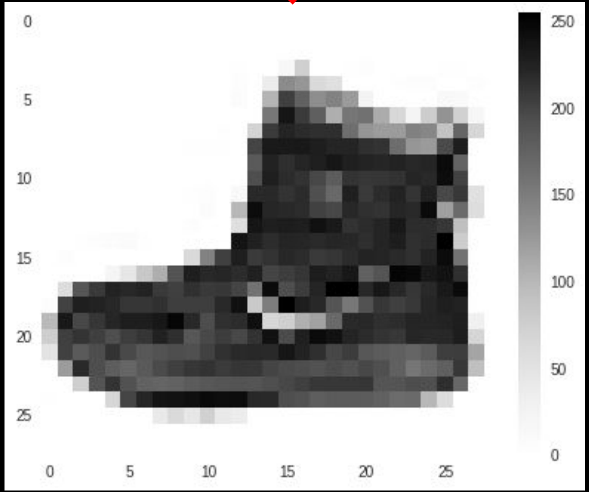
```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```



09

```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```



09

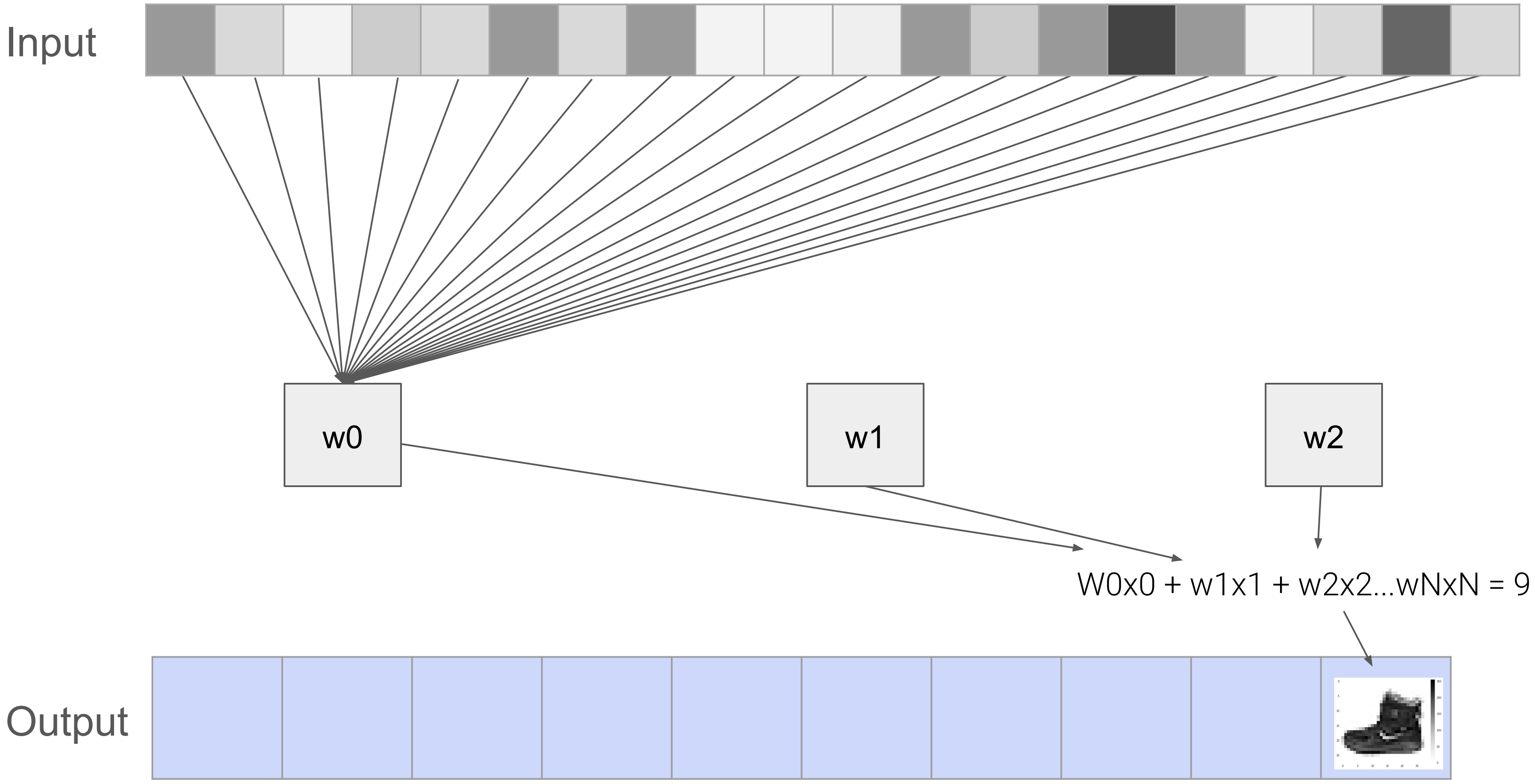
09 = ankle boot;
踝靴;
アンクルブーツ;
Bróg rúitín

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

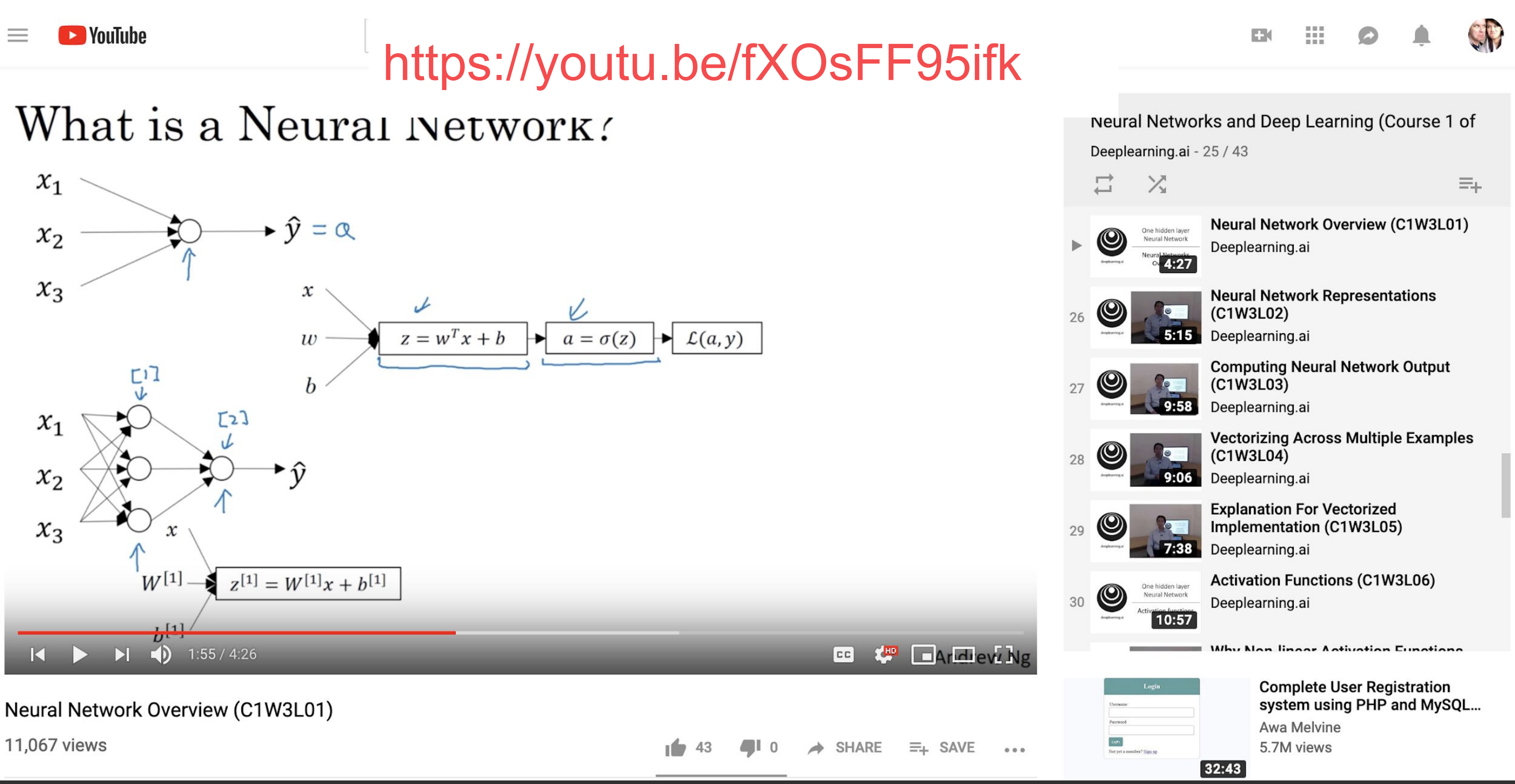
```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```


Input
w0
w1
w2
W0x0 + w1x1 + w2x2...wNxN = 9
Output

https://youtu.be/fXOsFF95ifk



What is a Neural Network?

$$\hat{y} = a$$

$$\underbrace{z = w^T x + b}\ \rightarrow\ \underbrace{a = \sigma(z)}\ \rightarrow\ \mathcal{L}(a, y)$$

$$z^{[1]} = W^{[1]}x + b^{[1]}$$

Neural Network Overview (C1W3L01)

11,067 views

43 0 SHARE SAVE

Neural Networks and Deep Learning (Course 1 of

Deeplearning.ai - 25 / 43

▶	4:27	Neural Network Overview (C1W3L01)	Deeplearning.ai
26	5:15	Neural Network Representations (C1W3L02)	Deeplearning.ai
27	9:58	Computing Neural Network Output (C1W3L03)	Deeplearning.ai
28	9:06	Vectorizing Across Multiple Examples (C1W3L04)	Deeplearning.ai
29	7:38	Explanation For Vectorized Implementation (C1W3L05)	Deeplearning.ai
30	10:57	Activation Functions (C1W3L06)	Deeplearning.ai

Complete User Registration system using PHP and MySQL...
Awa Melvine
5.7M views
32:43

deeplearning.ai

```python
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
training_images = training_images / 255.0
test_images = test_images / 255.0
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer=tf.optimizers.Adam(), loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5)
```

```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
training_images = training_images / 255.0
test_images = test_images / 255.0
model = tf.keras.Sequential([
    tf.keras.Input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer=tf.optimizers.Adam(), loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5)
```

```
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs=None):
    if logs['loss'] < 0.4:
      print('Loss is low so cancelling training!')
      self.model.stop_training = True
```

```
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs=None):
    if logs['loss'] < 0.4:
      print('Loss is low so cancelling training!')
      self.model.stop_training = True
```

```
class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs=None):
    if logs['loss'] < 0.4:
      print('Loss is low so cancelling training!')
      self.model.stop_training = True
```

```python
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
training_images = training_images / 255.0
test_images = test_images / 255.0
model = tf.keras.Sequential([
    tf.keras.input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer=tf.optimizers.Adam(), loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5)
```

```
fashion_mnist = tf.keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
training_images = training_images / 255.0
test_images = test_images / 255.0
model = tf.keras.Sequential([
    tf.keras.input(shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer=tf.optimizers.Adam(), loss='sparse_categorical_crossentropy')
model.fit(training_images, training_labels, epochs=5, callbacks=[myCallback()])
```