

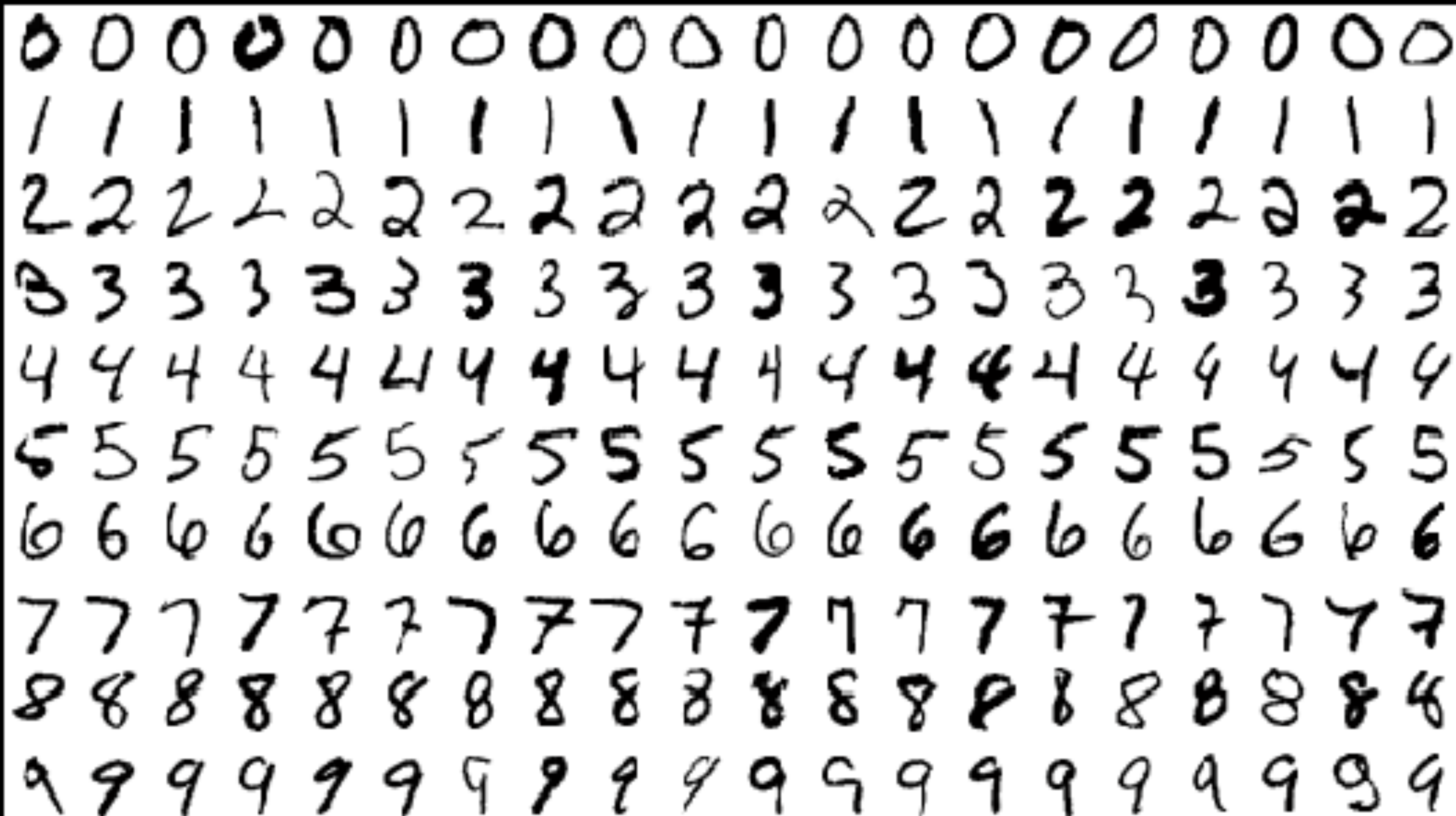
# Gjenkjenning av tall

En liten tå ned i TensorFlow

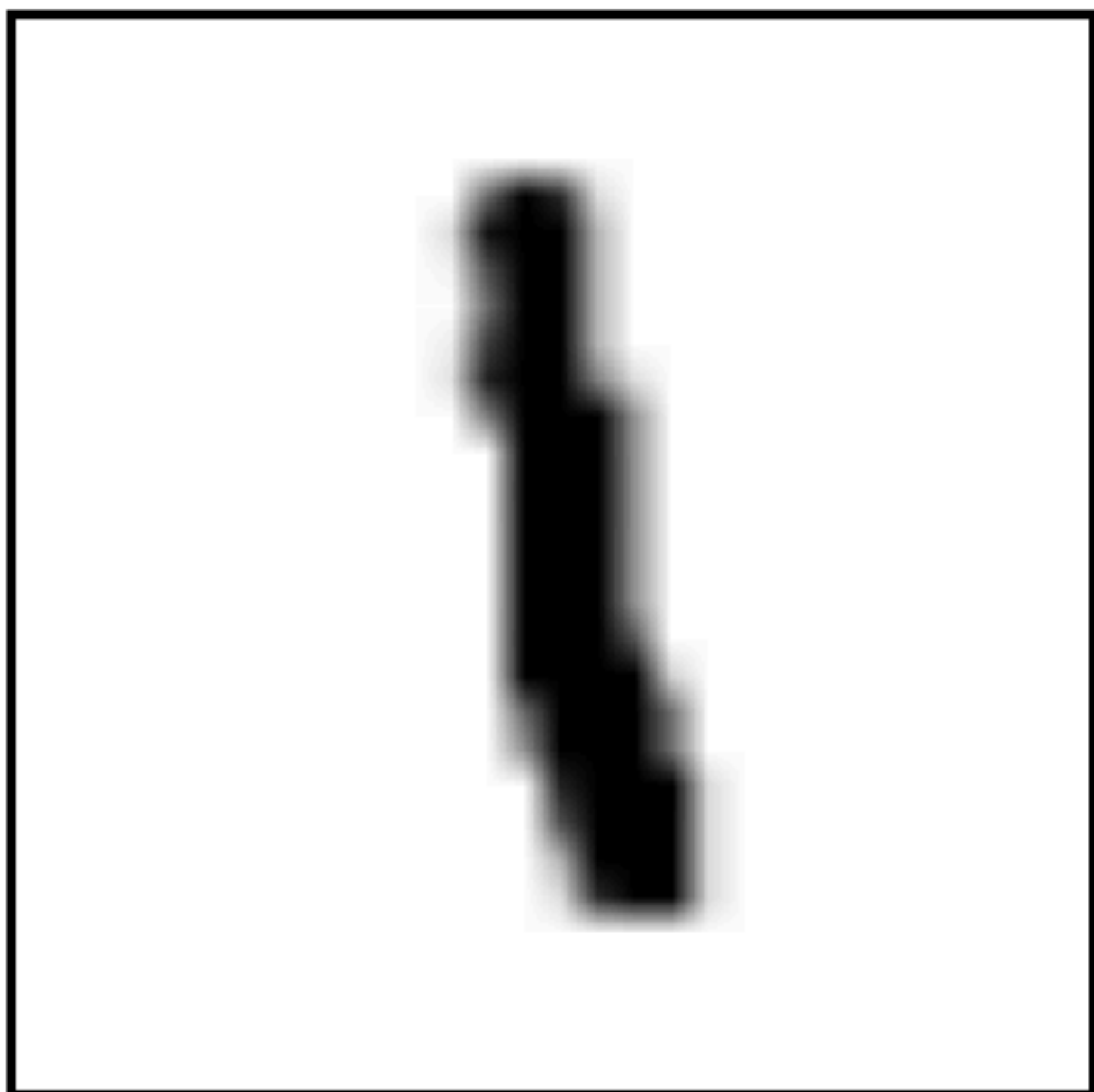
Is this your phone number? **5906868637**

Yes

No

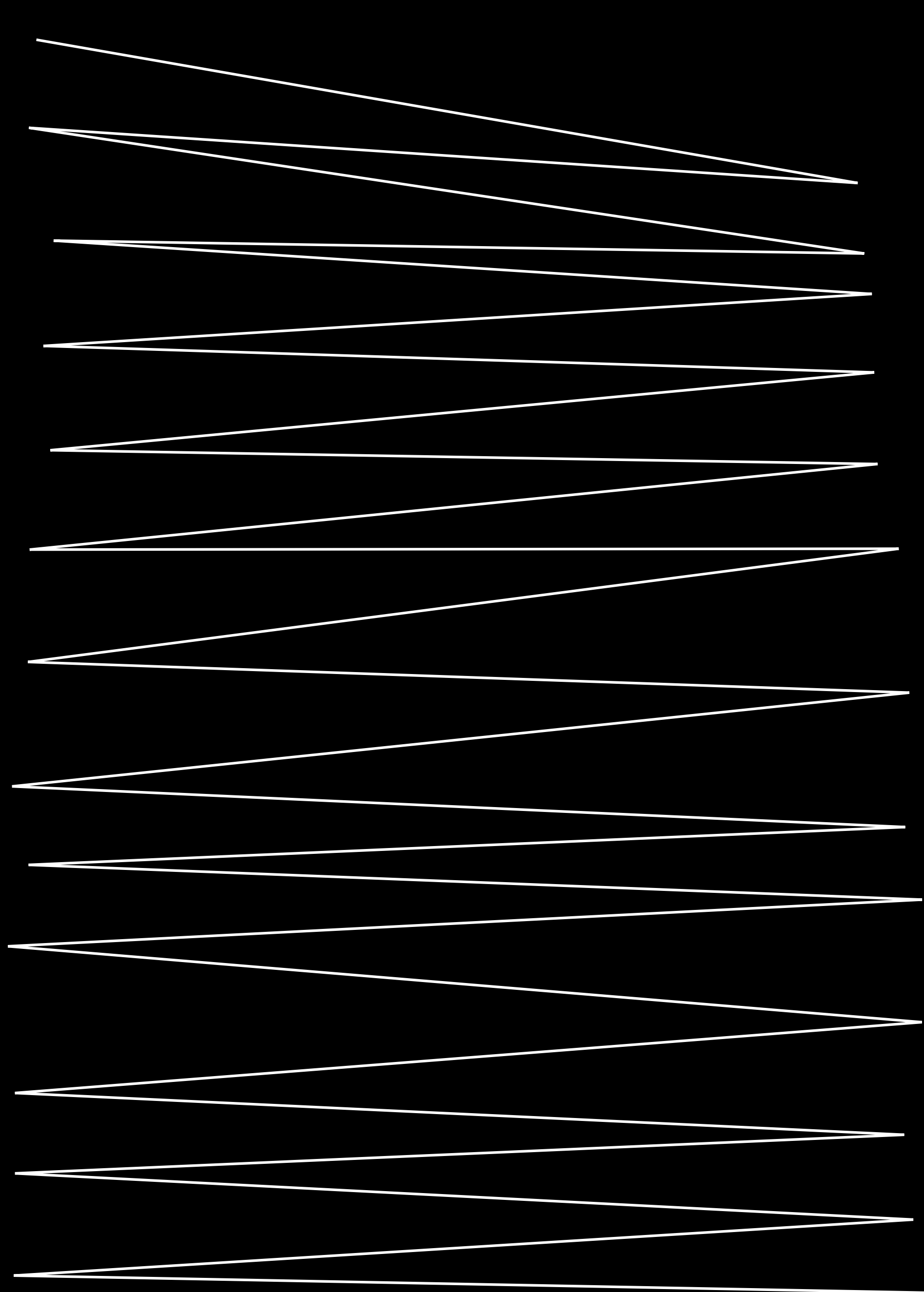
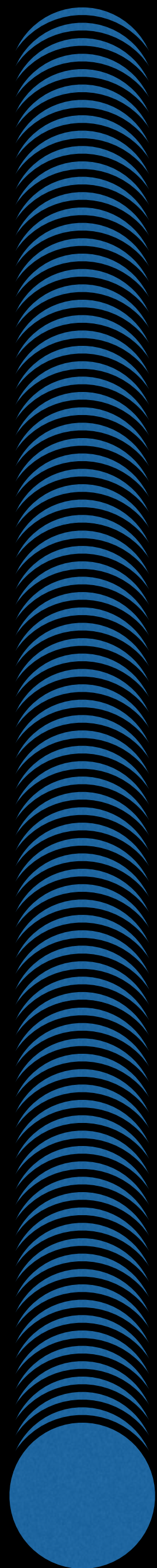






21

[illegible]



- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Sette opp nettverket

```
# Input tensors. 784 wide, and unlimited number of items
x = tf.placeholder("float", [None, 784])

# Weights, 784 input x 10 output
W = tf.Variable(tf.random_uniform([784, 10], -.01, .01))

# Bias
b = tf.Variable(tf.zeros([10]))

# Output tensors
y = tf.nn.softmax(tf.matmul(x, W) + b)

# Expected output
y_ = tf.placeholder("float", [None, 10])
```

```
# The entropy cross all output and  
# expected output  
cross_entropy = -tf.reduce_sum(y_*tf.log(y))  
  
# Training algorithm that should try to  
# minimize cross_entropy. 0.01 is the learning  
# rate. Increase to increase the changing of weights  
train_step = tf.train.GradientDescentOptimizer(0.01)\  
    .minimize(cross_entropy)
```



Trene nettverket

```
print("Starting session")
sess = tf.Session()
sess.run(tf.initialize_all_variables())
```

```
print("Training")
for i in range(1000):
    # Retrieving training set
    batch_xs, batch_ys = mnist.train.next_batch(100)
    # Run training of batch
    sess.run(train_step,
              feed_dict={x: batch_xs, y_: batch_ys})
```

Teste nettverket

```
# argmax returns the tensor with the highest value.
# For input 0 we want the first tensor to be returned from y.
# correct_prediction will be a list of booleans
# indicating if we got the correct number or not.
correct_prediction = tf.equal(
    tf.argmax(y,1), tf.argmax(y_,1))

# Casting correct_predictions to float, True=1.0 and False=0.0.
# Calculating the mean of these values. If we
# hit 90% correct, accuracy will be 0.9
accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))

#Testing network and returning the accuracy of the network
print sess.run(accuracy,
    feed_dict={x: mnist.test.images, y_: mnist.test.labels})
```



Lykke til!

