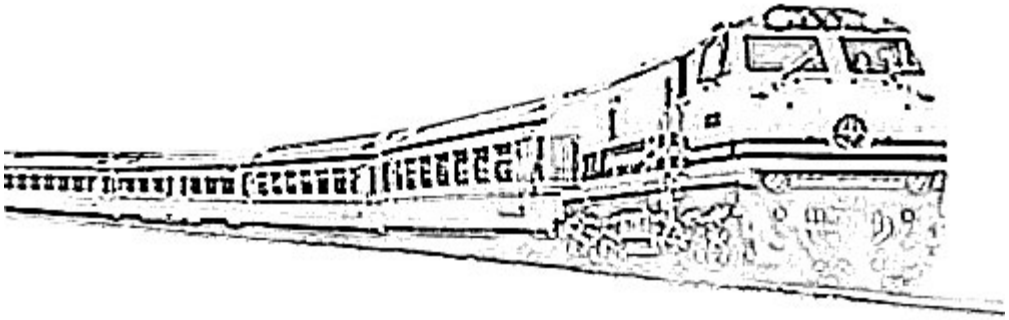
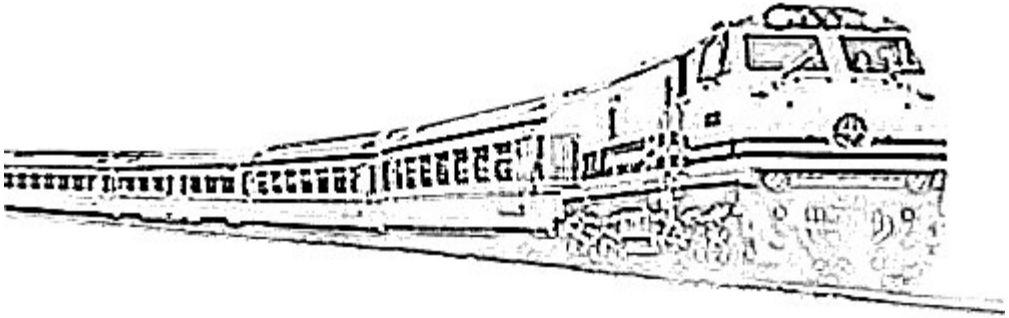
Neural Networks - I

Classification



How long the train can be late?

Will the train be late?



What grade the student is in?

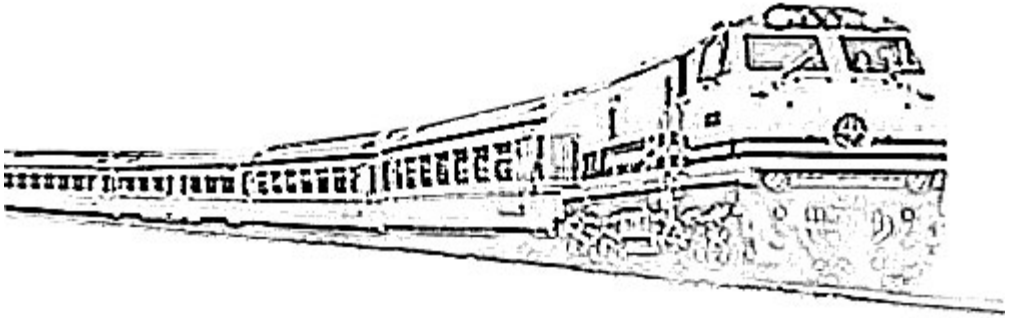
Math in Classification

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

Equation in Regression

How does it change?

Will the train be late?



What should be the value of prediction?

What grade the student is in?

What should be the value of prediction?

Multiple answers possible

Number of Predictions = Number of Possible Outputs

Grade 1	2%
Grade 2	5%
Grade 3	3%
Grade 4	6%
Grade 5	4%
Grade 6	15%
Grade 7	22%
Grade 8	36%
Grade 9	6%
Grade 10	1%
Total	**100%**

Probability of Student being in a particular Grade

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

How do we get multiple predictions?

$$y_1 = w_{11}x_1 + w_{12}x_2 + w_{13} + b_1$$

$$y_2 = w_{21}x_1 + w_{22}x_2 + w_{23} + b_2$$

$$y_3 = w_{31}x_1 + w_{32}x_2 + w_{33} + b_3$$

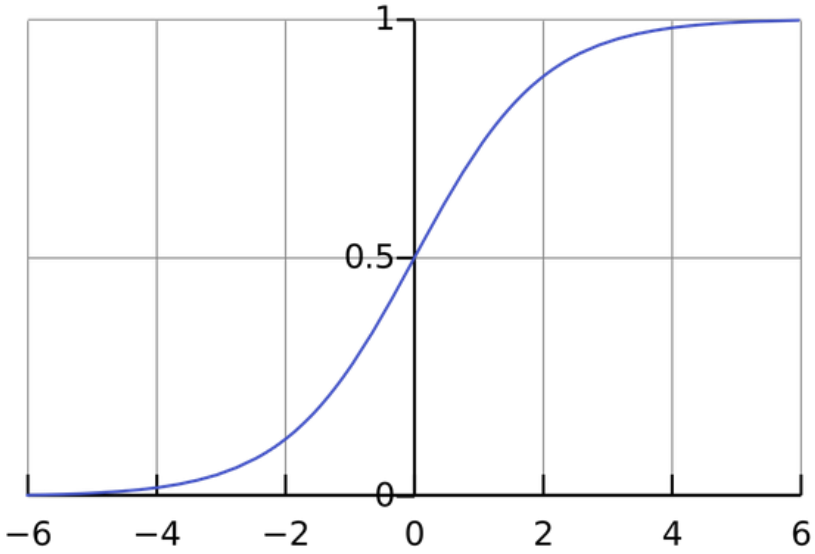
....

$$y_{10} = w_{101}x_1 + w_{32}x_2 + w_{103} + b_3$$

Using multiple equations

How do we get predictions between 0% and 100%?

Softmax



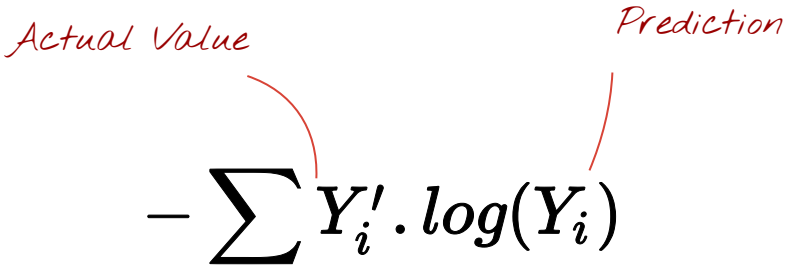
$$\sigma(y_i) = \frac{e^{y_i}}{\sum_{k=1}^{n} e^{y_k}}$$

- 'Squashes' each input between 0 and 1
- Sum of all outputs is 1 *i.e.* 100%

Prediction

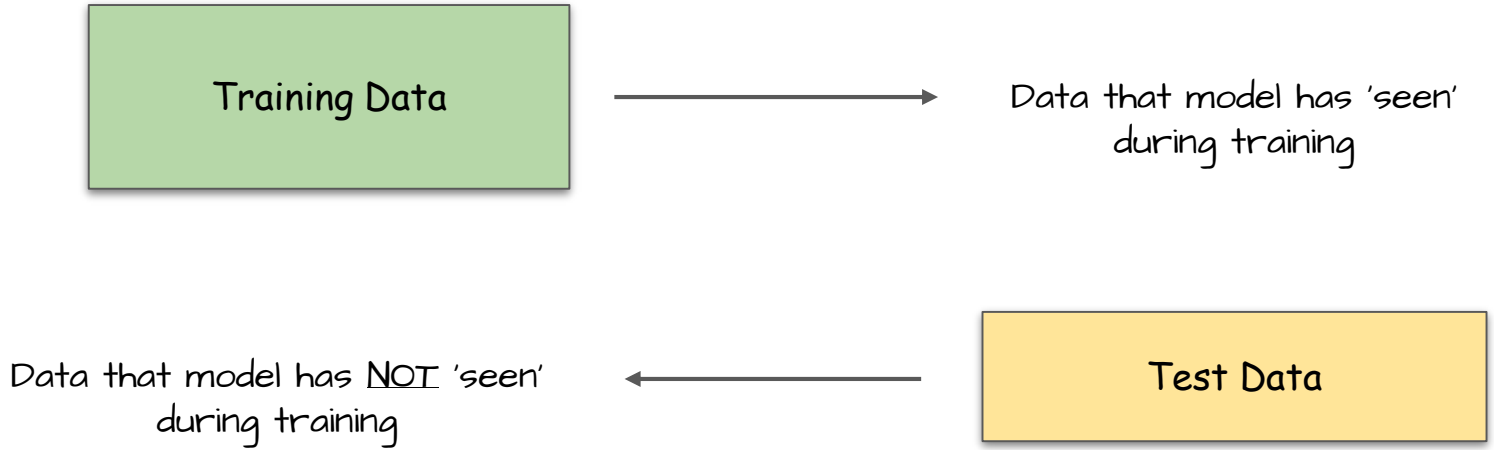
$$Y = softmax(X.W + b)$$

What about LOSS function?



Cross Entropy Loss

How do we assess ML model performance?


Training Data
Data that model has 'seen' during training
Test Data
Data that model has NOT 'seen' during training

Calculating Accuracy of the Model

Grade 1	2%
Grade 2	5%
Grade 3	3%
Grade 4	6%
Grade 5	4%
Grade 6	15%
Grade 7	22%
Grade 8	36%
Grade 9	6%
Grade 10	1%
Total	**100%**

Model Prediction = Grade 8

Student #	Model Prediction	Actual Grade	Accurate?
1	7	7	Yes
2	3	2	No
3	9	9	Yes
4	10	10	Yes
5	5	5	Yes

Accuracy = Number of correct predictions / Number of Predictions

Accuracy = 4 / 5 = 80%

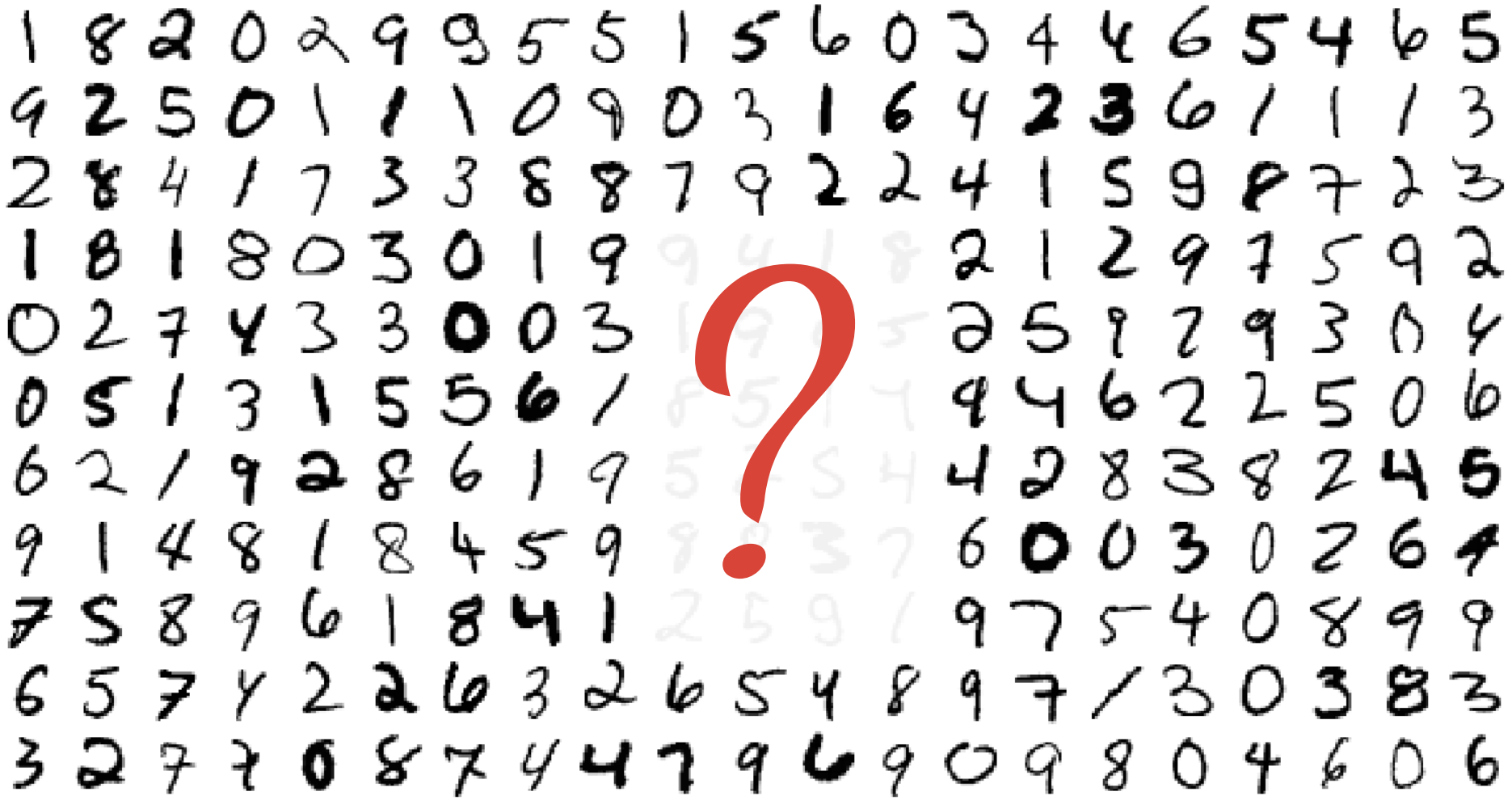
Classification Exercise

Handwritten digit recognition

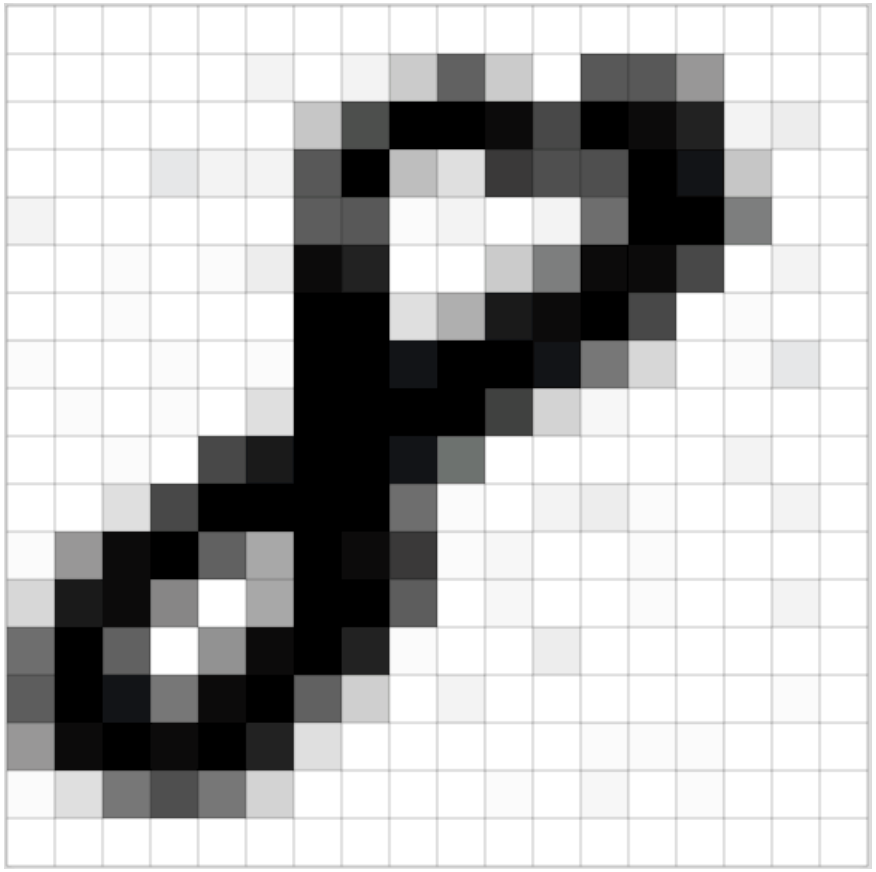
Scenario

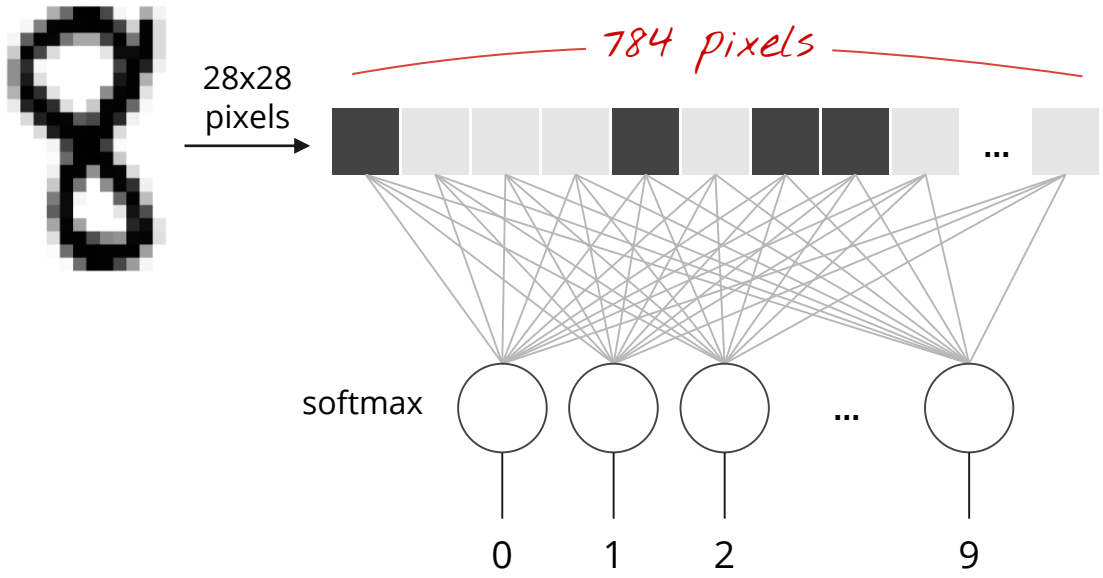
- **What needs to be done**
  - Build a Classifier to predict Handwritten numbers
  - Use Tensorflow to build the model

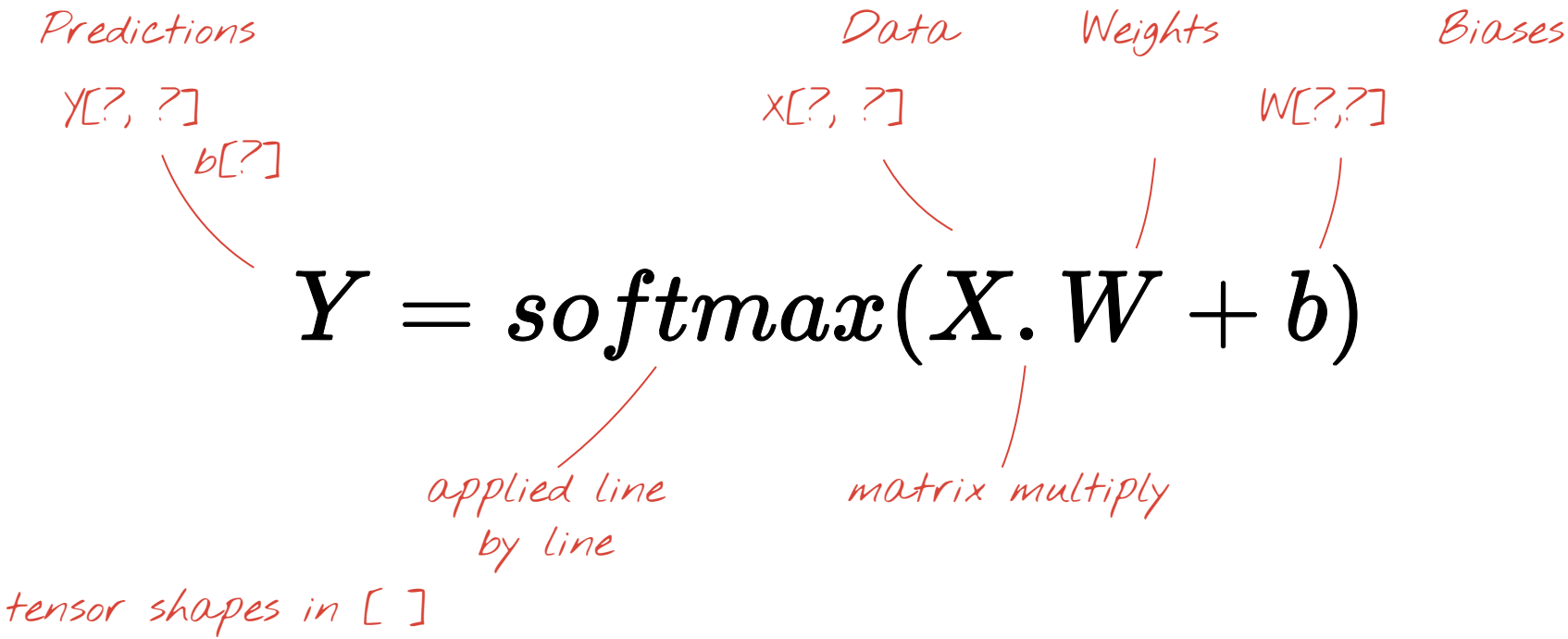
- **What is given**
  - Handwritten data (60,000 training examples, 10,000 test examples)
  - 10 classes



MNIST = Mixed National Institute of Standards and Technology - Download the dataset at http://yann.lecun.com/exdb/mnist/




784 pixels
28x28
pixels
...
softmax
...
0
1
2
9


Predictions
Data
Weights
Biases
Y[?, ?]
b[?]
x[?, ?]
W[?,?]
$Y = softmax(X.W + b)$
applied line by line
matrix multiply
tensor shapes in [ ]

Let’s build the classifier in TensorFlow

How many calculations at once?

7840 multiplications

7840 additions

Softmax + Loss + Gradient Descent

MNIST input features = 784

60,000 * 7840 multiplications

+

60,000 * 7840 additions

+

60,000 * Softmax

+

60,000 * Loss

+

Gradient Descent

MNIST Examples = 60,000

What if we had 1,000,000 records?

Memory Error!!!

How do we handle this?

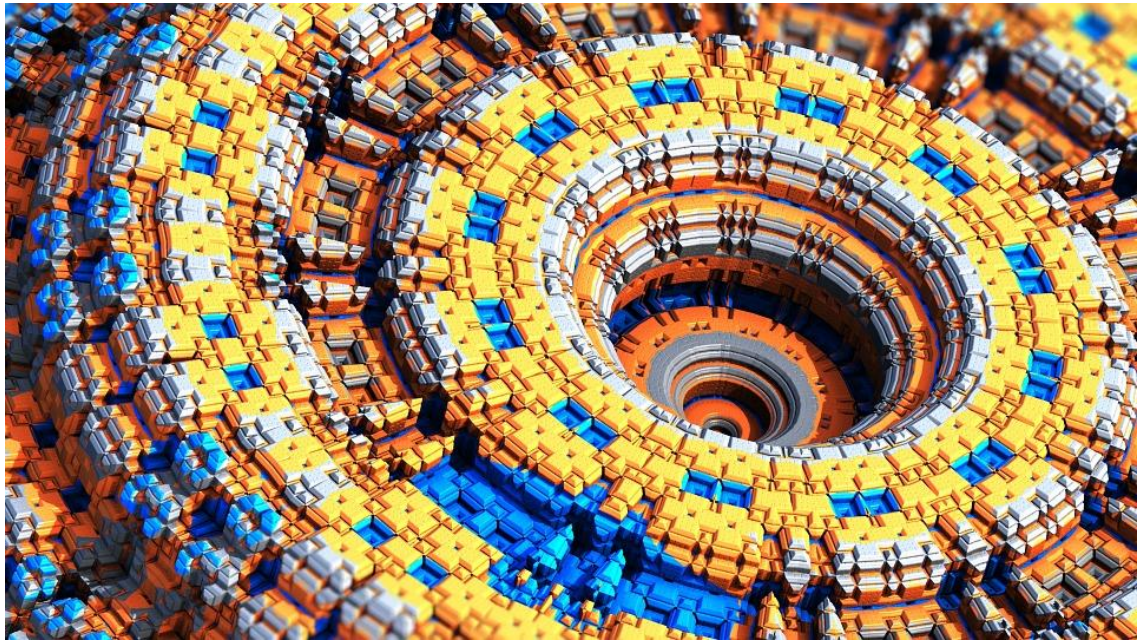
Small batches
or mini-batch

Exercise

Implementing mini-batch in TensorFlow Keras

Improving the model

Hyperparameters



Number of iterations

Batch Size


Learning rate

Using default learning rate

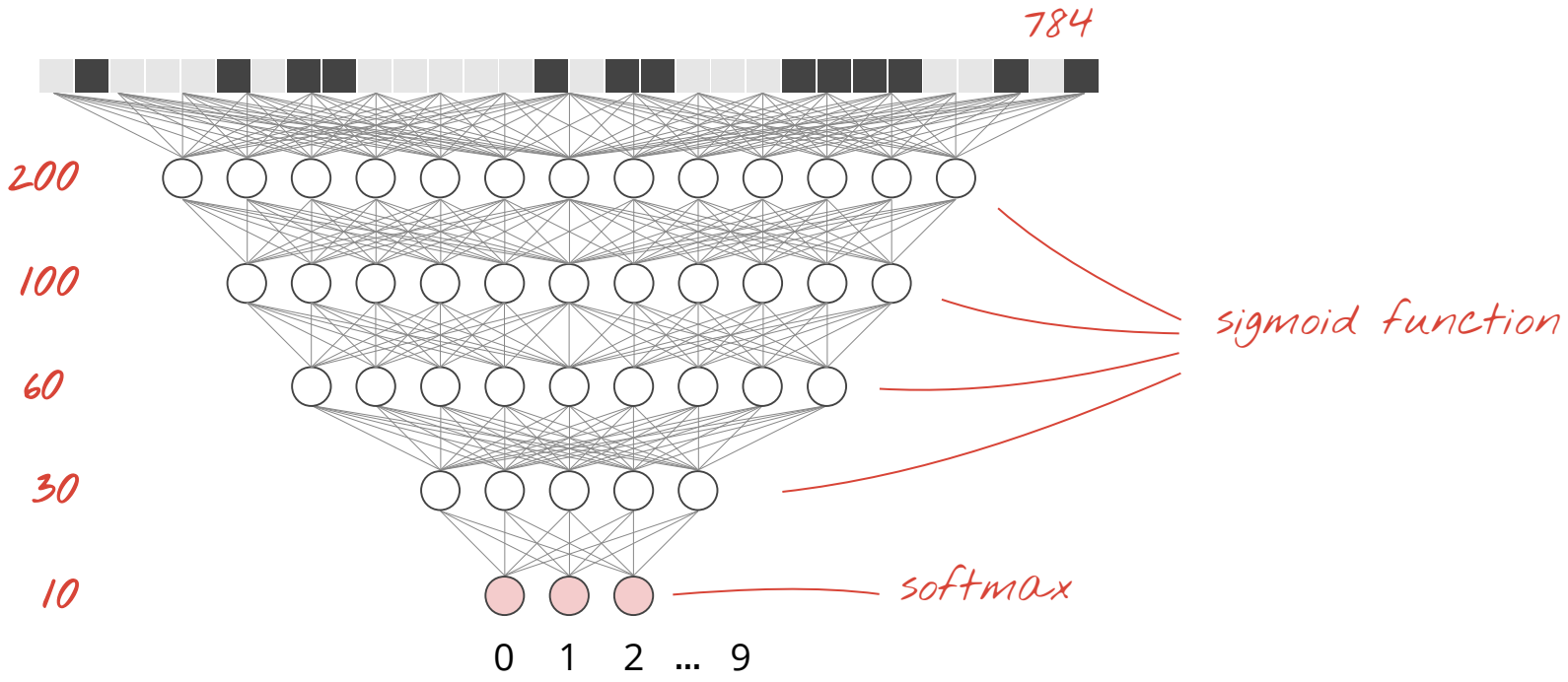
```
model.compile(optimizer='sgd', loss='mse’)
```

Setting a specific learning rate

```
sgd_optimizer = tf.keras.optimizers.SGD(lr=0.03)

model.compile(optimizer=sgd_optimiser, loss='mse’)
```

Applying
Deep
Learning(Multiple
Layers)
on MNIST


784
200
100
60
30
10
sigmoid function
softmax
0 1 2 ... 9

Exercise

MNIST Classification using Deep Neural Network

Exercise

Model Prediction