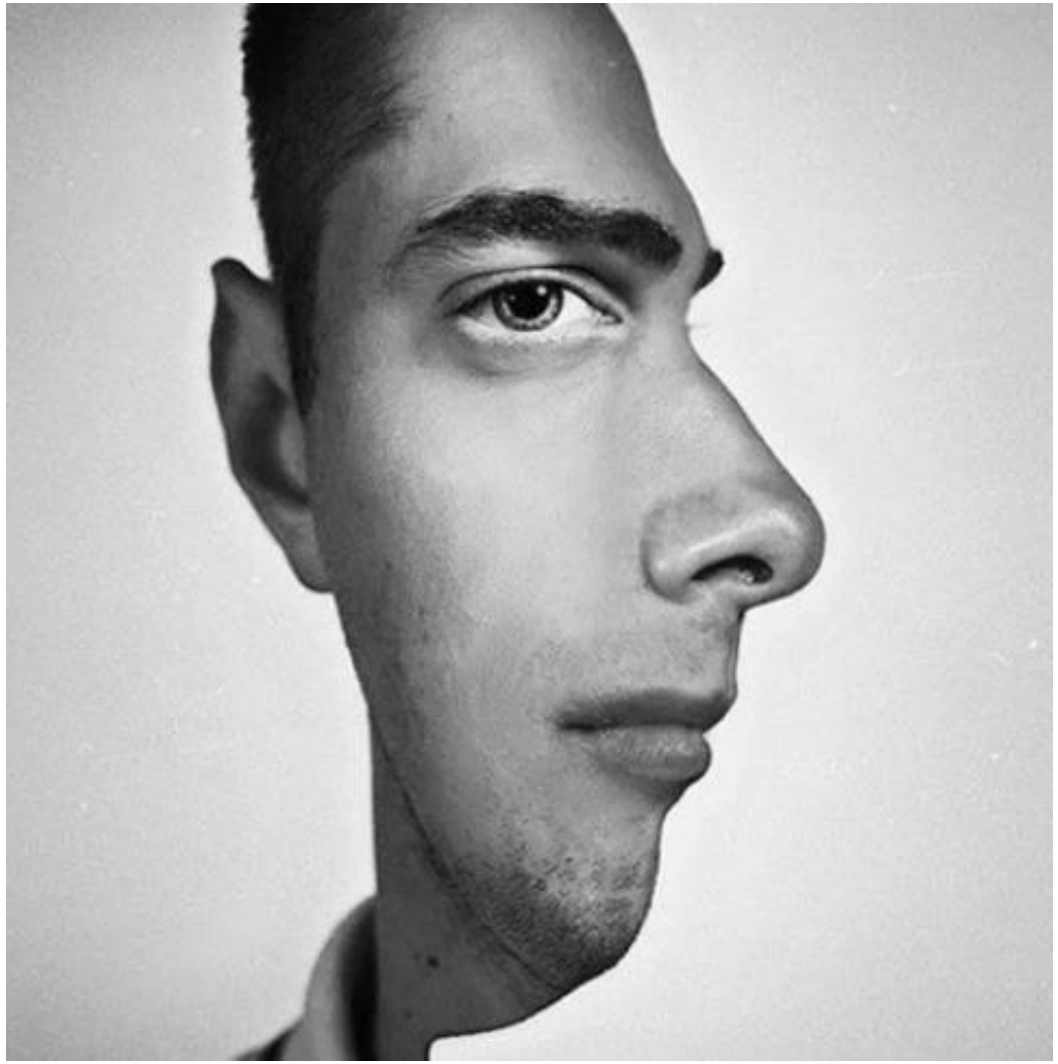
9.04 Convolutional Neural Networks

How does our brain classify this image?



How does our brain classify this image?

- In the previous image, if we look to the right side of the image, we will see a person looking towards the right side while, if we look in the center, we will perceive that the person is looking towards us.

- Does our brain struggle a lot in identifying these different scenarios and is confused if the person is looking in right or towards us?
  - This happens **because our brain studies for the features in the image** and then assume what to take.

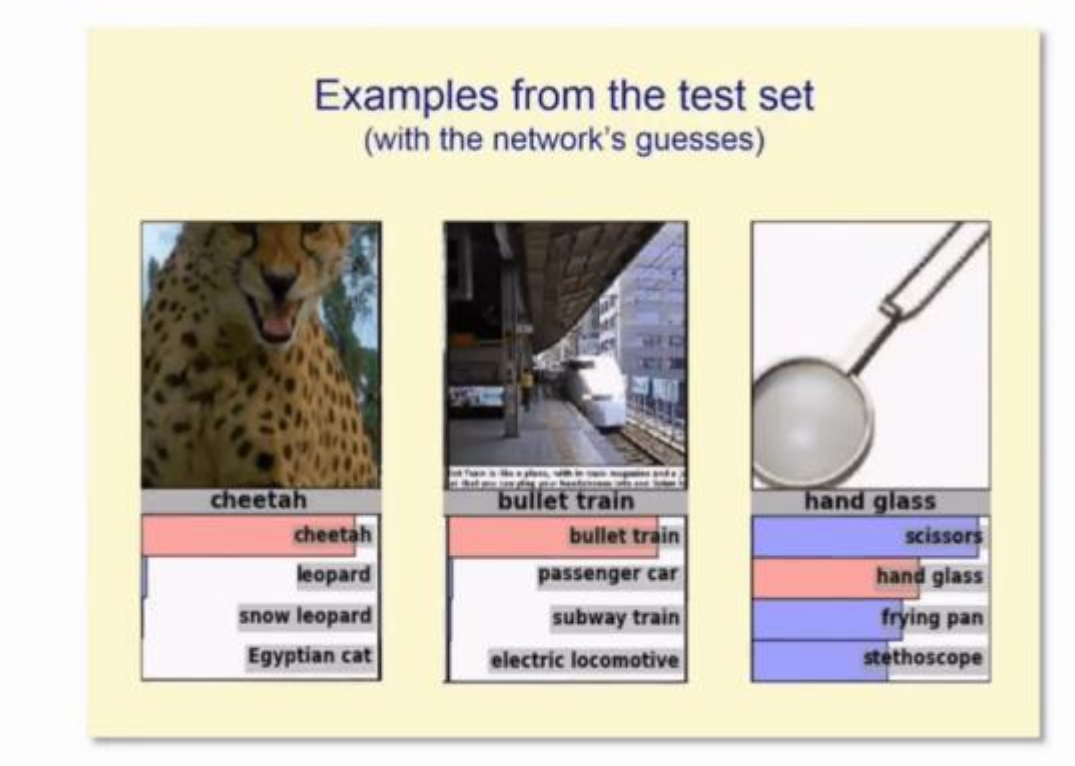
What if the features were not clear, like this?

Is your brain able to decide what is correct?

No! That’s because the features depicted are inadequate to assist the brain in classifying them.

All images are addressed to understand that our brain functions on the features of the image it sees and then classifies it accordingly.

Neural Networks



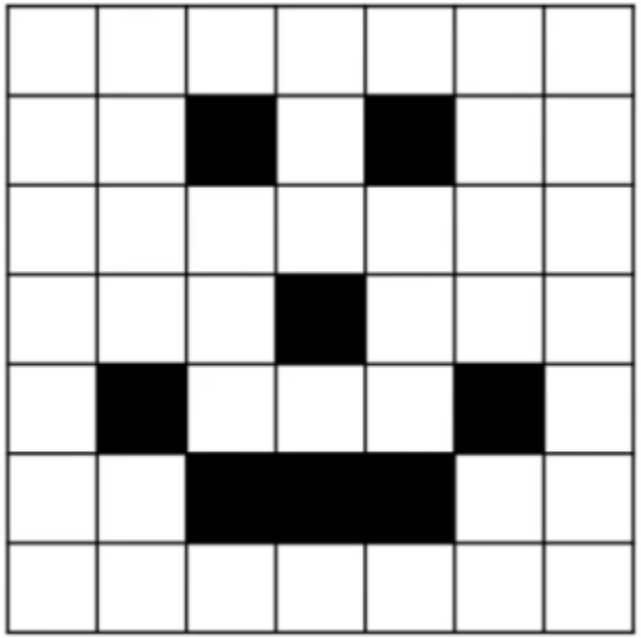
In a similar manner, neural networks work.

We can see in the image, the neural network has successfully classified cheetah and bullet train but was unsuccessful in predicting hand glass.

This is because of the unclear features in the image.

In summary, Neural Networks work exactly like a human mind

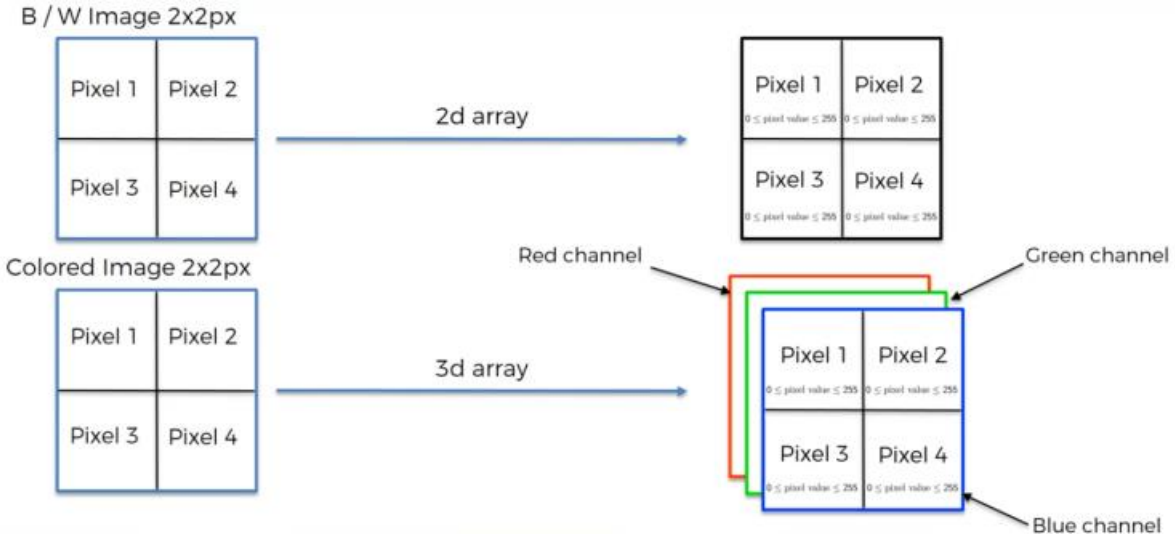
How the Computer sees an image?



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

How the Computer sees an image?

- An image is a matrix of pixels. If there is a black and white image, then we will get a 2D array. While, if we passed a colored image, then we will get a 3D array, which means it has an extra parameter for depth which is RGB channel as shown below. The pixel value lies between 0 and 255, and images are stored in bytes (0 to 255).



Convolutional Neural Network (CNN)

- CNN is a specific type of artificial neural network that uses perceptrons, a ML unit algo, for supervised learning, to analyse data
- CNNs are used for Image Recognition, Computer Vision, NLP, and other kinds of cognitive tasks
- A CNN is also known as a ConvNet

Convolutional Neural Network (CNN)

- Like other kinds of artificial neural networks, a CNN has an input layer, an output layer and various hidden layers.

- Some of these layers are convolutional, using a mathematical model to pass on results to successive layers. This mirrors some of the actions in the human visual cortex.

- CNNs are a core example of Deep Learning, where a more sophisticated model pushes the evolution of AI by offering systems that mirror different types of biological human brain activity.

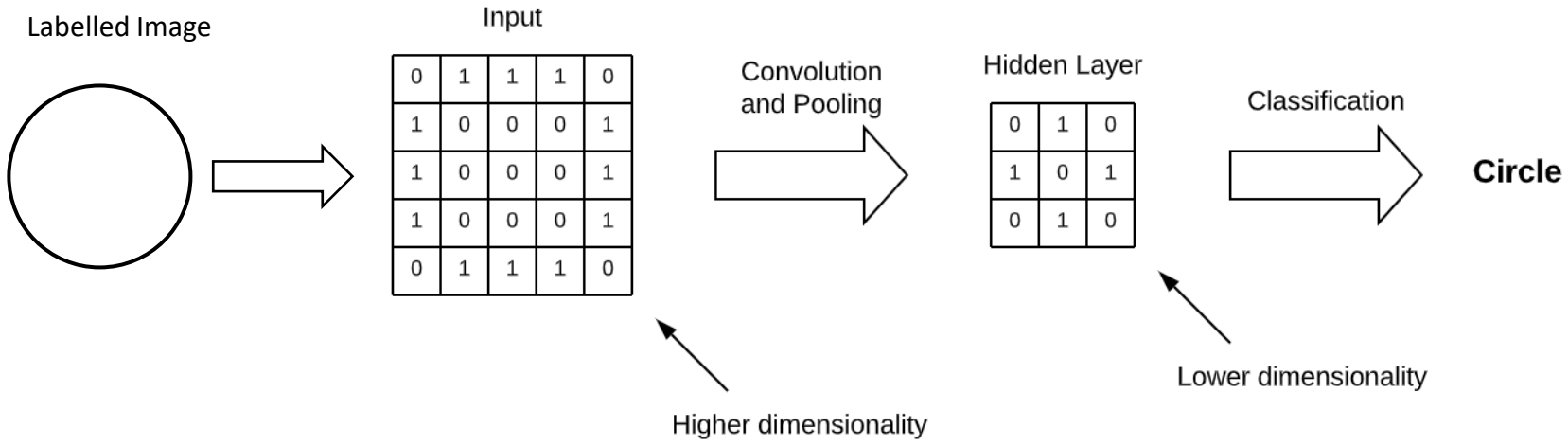
Neural Network Assumptions

- There are 2 key assumptions when building any Neural Network:

1. Feature Independence
2. Low Dimensionality

The data we typically process with CNNs (text, image, audio, and video) does not usually satisfy either of these assumptions, and this is exactly why we use CNNs **(explained in next slide)**

CNN Dimensionality Reduction



During CNN model training, by performing convolution and pooling, neurons of the hidden layers learn possible abstract representations over their input, which typically decrease its dimensionality.

The CNN assumes that these abstract representations, and not the underlying input features, are independent of one another.

These abstract representations are normally contained in the hidden layer of a CNN and tend to possess a lower dimensionality than that of the original input features

What are the different types of CNN Layers?

1. **Convolution Layer**
   - Uses filters that perform convolution operations as it is scanning the input with respect to its dimensions
   - Hyperparameters include number of filters and filter size
   - The resulting output is called feature map

2. **Pooling Layer**
   - A downsampling operation, typically applied after a convolution layer
   - Include max / average pooling where the max / average value is taken

What is a Convolution?

- A convolution is how the input is modified by a filter.

- In convolutional networks, multiple filters are taken to slice through the image and map them one by one and learn different portions of an input image.

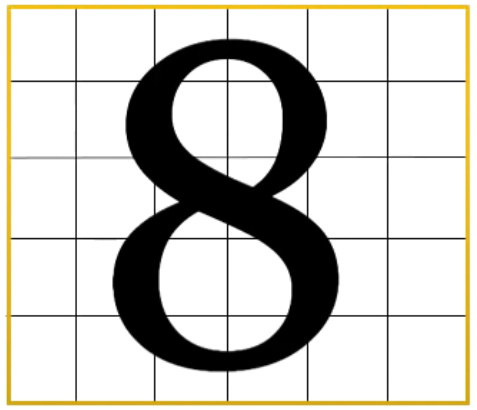
- Imagine a small filter sliding left to right across the image from top to bottom and that moving filter is looking for, say, a dark edge. Each time a match is found, it is mapped out onto an output image.

What is a Convolution?

Convolution operation forms the basis of any Convolution Neural Network

In CNN, every image is represented in the form of arrays of pixel values

Real Image of the digit 8

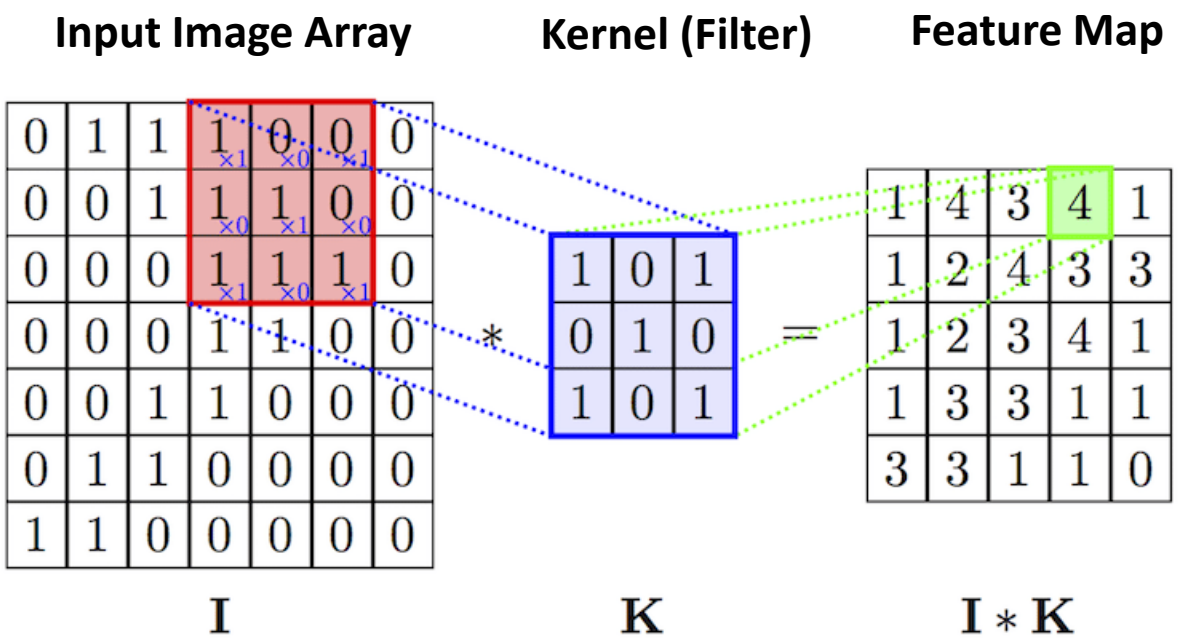


Represented in the form of an array

0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	1	0	0

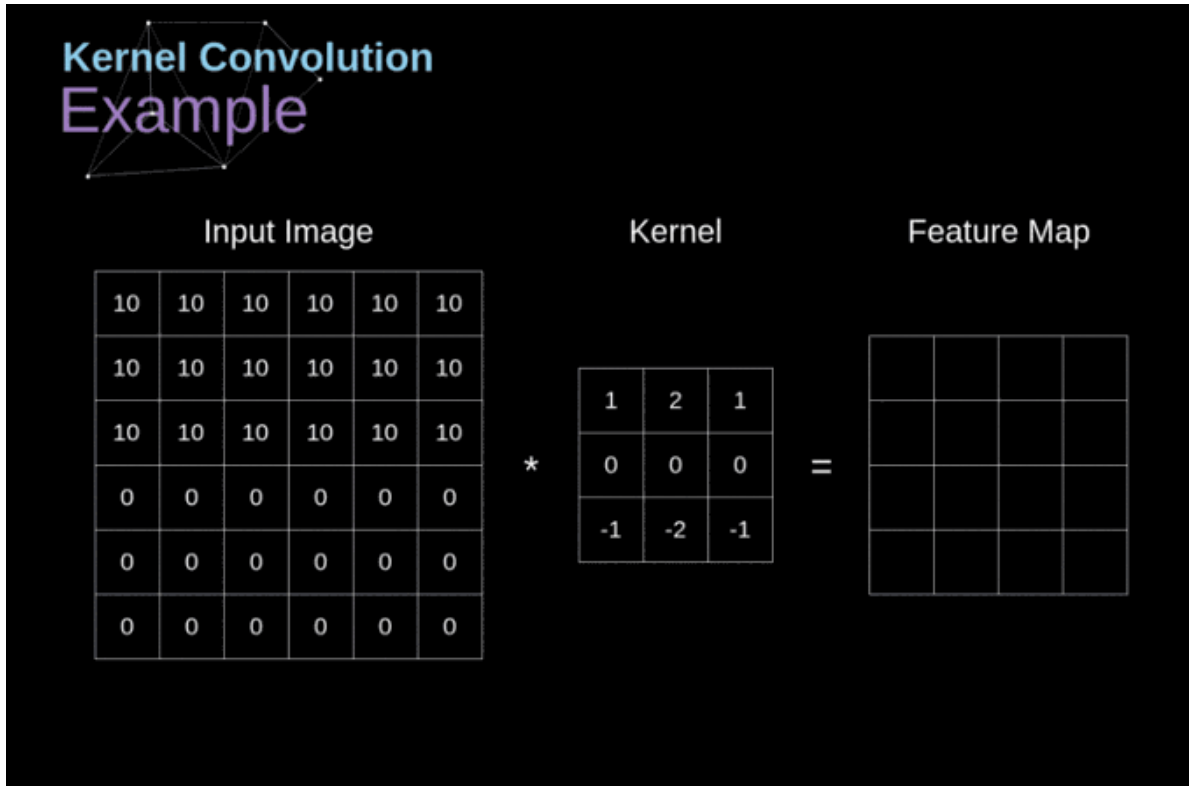
Digit 8 represented in the form of pixels of 0's and 1's

What is a Convolution?



- The first parameter required by a Conv layer is the **number of filters** that the convolutional layer will learn.
- Layers early in the network (closer to input image) learn fewer convolutional filters while layers deeper in network (closer to output predictions) will learn more filters.
- These filters emulate edge detectors and other feature extraction methods.
- **Number of the filters** should be tuned, but common practice to use powers of 2, starting with 32 and increasing to 64, 128 256, 512, 1024, etc.

What is a Convolution?



- The second parameter required by the convolutional layer is the **kernel_size**
- **kernel_size** is a 2-tuple specifying the width and height of the 2D convolution window.
- The kernel_size must be an odd integer, with typical values of (1, 1), (3, 3), (5, 5)
- It is rare to see kernel sizes larger than 7×7. If the input images are greater than 128×128, it is recommended to test a kernel size > 3 to help learn larger spatial filters and to help reduce volume size.

Pooling

- Note that pooling is a separate step from convolution.
- Pooling is used to reduce the image size of width and height.
- Note that the depth is determined by the number of channels.
- As the name suggests, all it does is it picks the maximum value in a certain size of the window.
- Although it’s usually applied spatially to reduce the x, y dimensions of an image.

Max-Pooling

- Max pooling is used to reduce the image size by mapping the size of a given window into a single result by taking the maximum value of the elements in the window.

2	0	1	1
0	1	0	0
0	0	1	0
0	3	0	0

Max pooling with a 2x2 window and stride 2

2	1
3	1

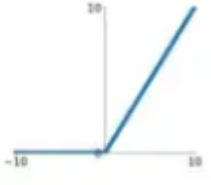
Average-Pooling

- It’s same as max-pooling except that it averages the windows instead of picking the maximum value.

Activation Function

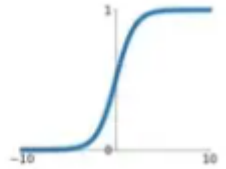
ReLU

$\max(0, x)$



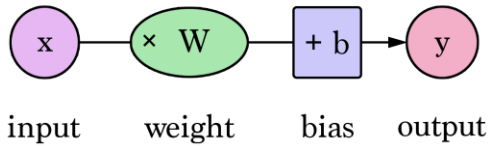
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



- The activation function is a node that is put at the end of Neural Networks. It helps to **decide if the neuron would fire or not**.
- ReLU is the most widely used activation function in neural networks.
- Key advantage ReLU has over other activation functions is that it does not activate all neurons at the same time.
- ReLU converts all negative inputs to zero and the neuron does not get activated. This makes it very computational efficient as few neurons are activated per time.
- In practice, ReLU converges six times faster than sigmoid activation functions.

What are weights and biases?



- Weights and bias are learnable params which are assigned initial random values
- During NN training, both parameters are adjusted toward desired values to get closer to the correct output
- Bias refers to the constant that is introduced to reduce errors in prediction
  - Bias can be added to the output value for obtaining the intended (true) value
  - A low bias refers to a lower error term (constant)
  - A high bias refers to a higher error term (constant)
- Weights can be interpreted as the strength of the input
  - Weight affects the amount of influence a change in the input will have on the output
  - A low weight value will have no change on the output
  - A high weight value will more impact on the output