MACHINE LEARNING AND NEURAL NETWORKS

By Haider

CONTENT

1 MACHINE LEARNING

Basic intro to machine learning and how it works

2 NEURAL NETWORKS

What are they and how they work

3 Algorithm Example

Real life example of neural networks

4 Code Example

Python code example of neural networks

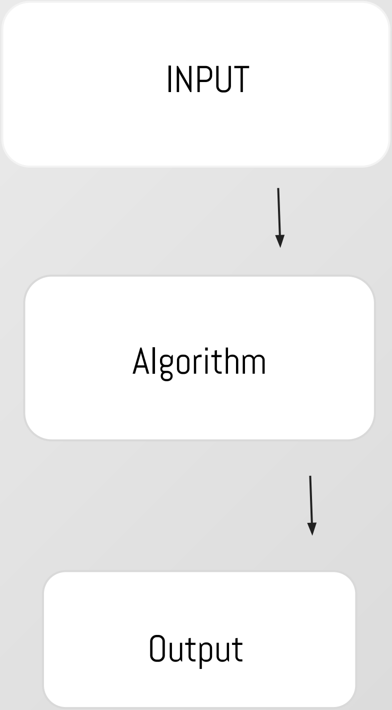
Machine learning

- Machine learning is a type of artificial intelligence that involves training computers to learn to perform a task without explicitly programing to perform that task

- machine learning works by taking in data as input, running it through an algorithm, then making predictions or decisions based on the output generated by the algorithm.

- A algorithm has to be trained in order to make these predictions. During training the algorithm learns from examples and uses it to identity patterns and gradually becomes smarter as more and more data it given.

- Applications range from predicting future outcomes like in the stock market or identifying the objects in a picture.

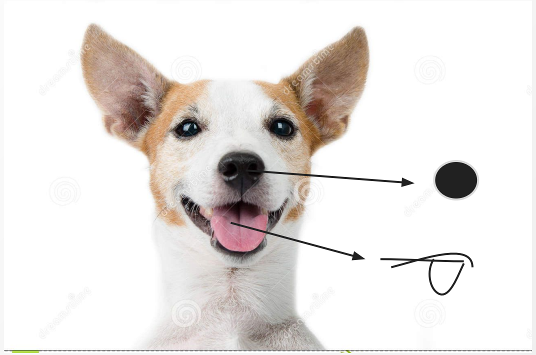


INPUT

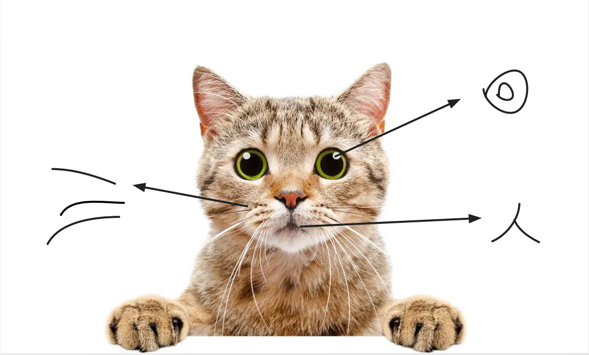
Algorithm

Output

How Machine learning Works (training)



VS



Patterns to identify:

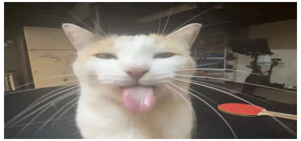
- Tongue out
- Eye shape
- Nose and mouth

Patterns to identify:

- Whiskers
- Eye shape
- Mouth shape

EXAMPLE

INPUT



Algorithm

Cat traits:
Whiskers, cat nose, cat mouth

Dog traits:
Tongue out

Output

CAT: 80% certainty

Neural networks and machine learning

- Machine learning has 3 main types supervised learning, unsupervised learning, and reinforcement learning. Neural networks fall under supervised learning category

- Supervised learning means that they are trained using “labeled data” meaning the algorithm is given input data along with corresponding labels

- You aren't telling the computer this is number one using a image you would do that using more complex ways we will learn that later

- the goal is to learn a function that can predict the output when given a input each time

Labeled data:

“ This is the number 1 “

“ This is the letter A “

“Normal arm” vs “broken arm”

“ This is a tree”

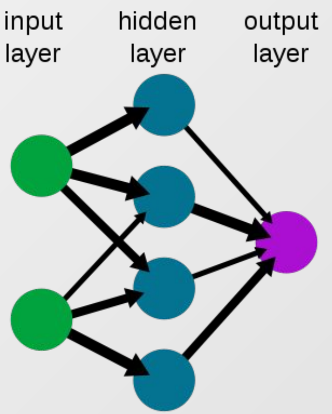
NEURAL NETWORKS

- a neural network works by inputting data into the first layer of neurons this data passes to the next layer of neurons, and so on until the output layer produces a prediction or decision.

- Each neuron in the network takes in a set of inputs and multiplies it by wights think of this as the influence the input has on the output

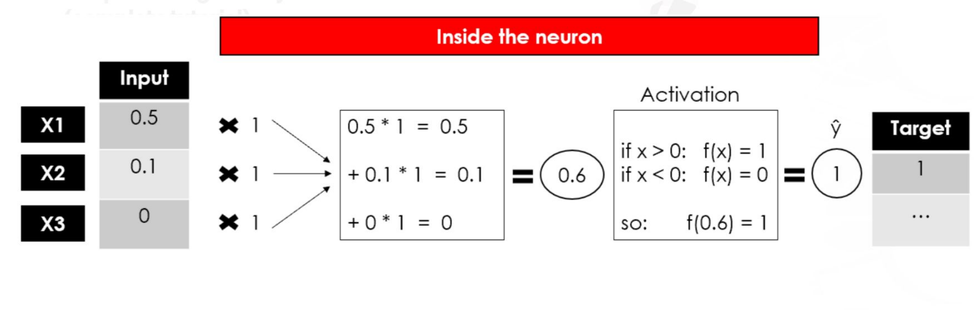
- neural networks are particularly useful in tasks like image recognition, speech recognition, and language processing.

A simple neural network



● = A neuron

Each neurons weight contributes to a activation in the next layer its weight it known as its “Activation”



Inside the neuron

	Input
X1	0.5
X2	0.1
X3	0

✖ 1
✖ 1
✖ 1

0.5 * 1 = 0.5
+ 0.1 * 1 = 0.1
+ 0 * 1 = 0

= 0.6

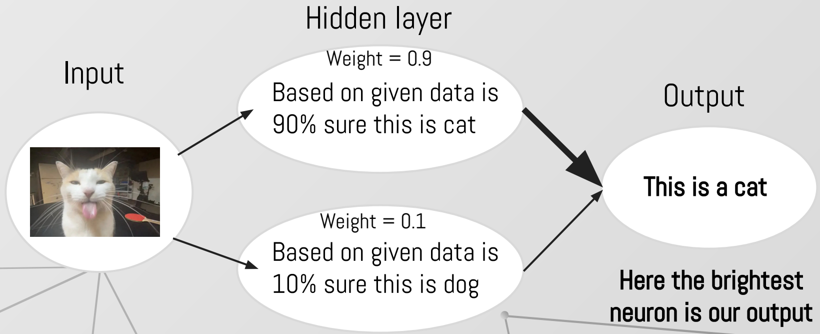
Activation

if x > 0: f(x) = 1
if x < 0: f(x) = 0
so: f(0.6) = 1

= ŷ 1

Target
1
...

Neuron activation example (cat or dog) ?



Input

Hidden layer

Weight = 0.9
Based on given data is 90% sure this is cat

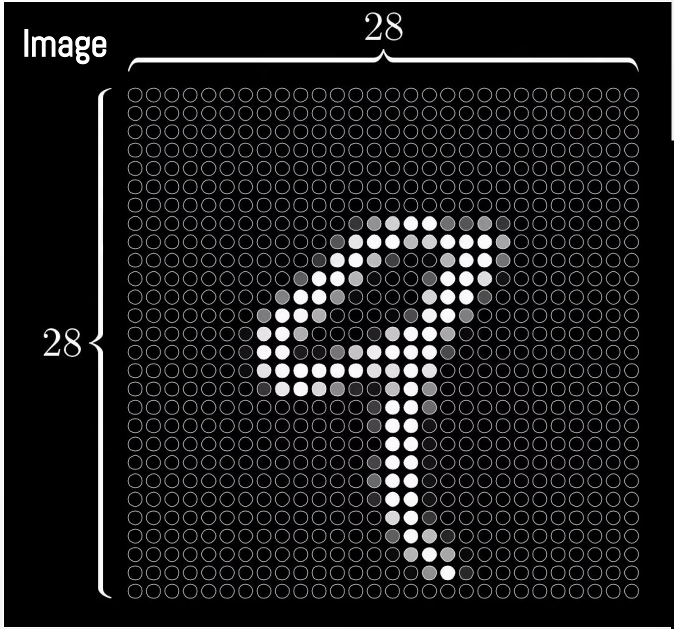
Weight = 0.1
Based on given data is 10% sure this is dog

Output

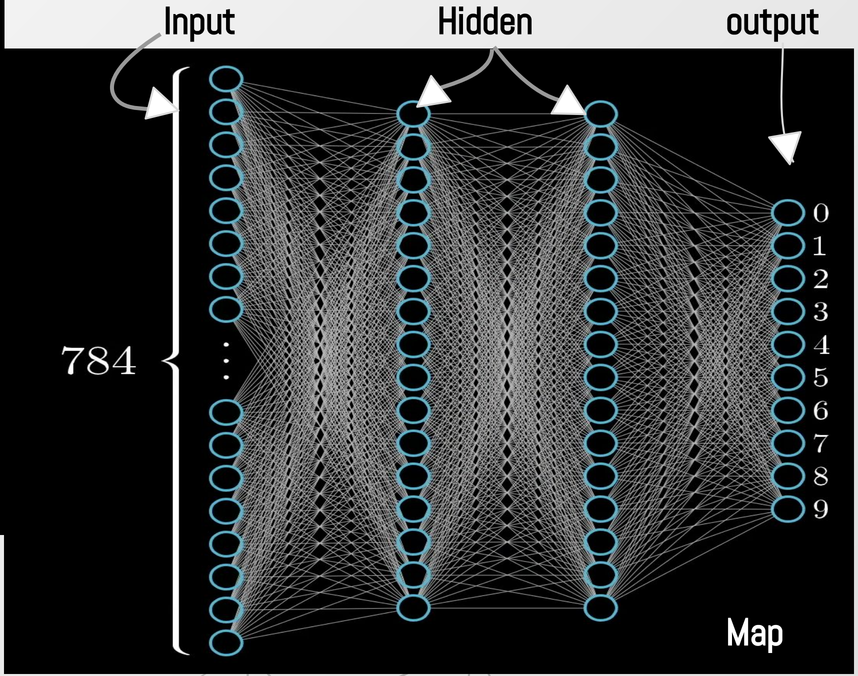
This is a cat

Here the brightest neuron is our output

Real life neural network Example reading number from picture



28 x 28 - 784 (# of neurons in layer 1)



Each pixel represents a neuron in the input layer **total pixels: 784**

● Circle (pixle) = single neuron

1

0

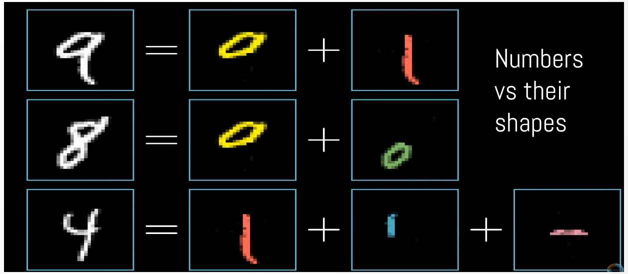
Neurons weight ranges from 0-1 depending of grayscale value

- This is a basics example of how a computer can detect a number 1-9 from a image

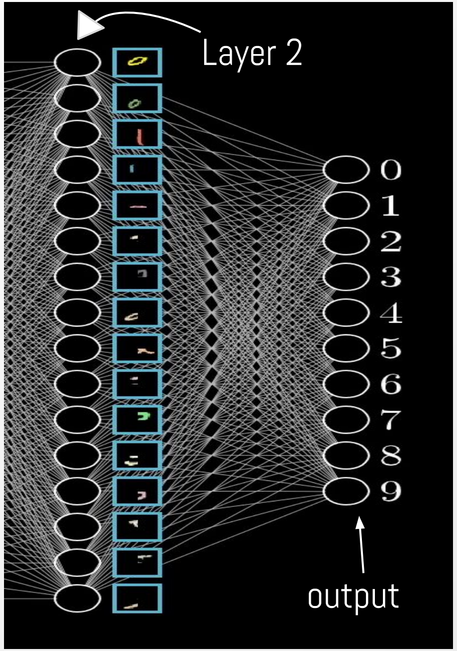
- This system uses a neural network made of 4 layers total

- By using the brightness of a pixel the network can make out shapes and patterns in the image to then pass it through a algorithm to them make a prediction based on the input

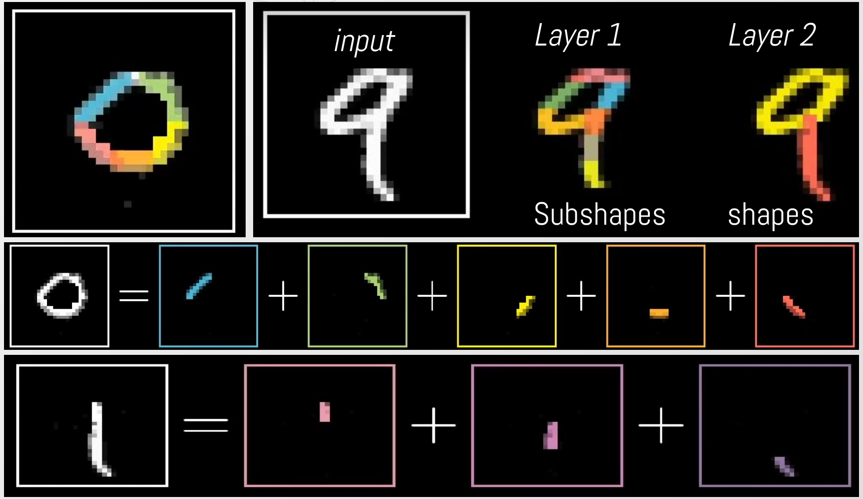
Neural network example continued



You can't just say loop plus stick = 9 its gets even more complex as you have to first explain what is a loop or a stick to the computer



To better demonstrate look at how layer 2, that makes out the circles and sticks will connect to the output numbers. Each neuron is connected to each output but each output will have a different wight (certainty). So if we had just a circle the output of 0 will be the brightest (most certain) while other numbers like 9 might light up the final answer will be based on the neuron with the most weight (most certain).



1. **First layer** in the hidden layer can make out the sub shapes from the images pixels →

2. The **second hidden layer** in the can find the loops or sticks using the sub shapes that make up the shapes →

Then the second layer can connect those loops and sticks shapes to the corresponding numbers in the **output layer** by knowing what combination of shapes make what numbers

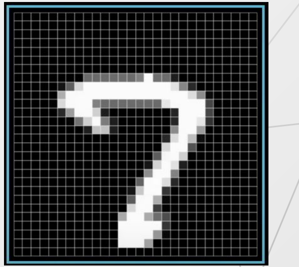
Neural network edge detection

We broke images down to shapes then to sub shapes but even subshapes break down to pixels

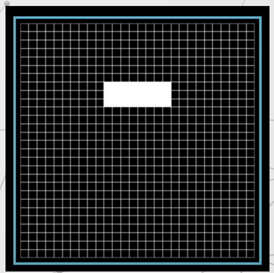
A image that has any object will have shapes but ultimately the pixels make up those shapes

To detect the sub shapes and pixel patterns we can use edge detection

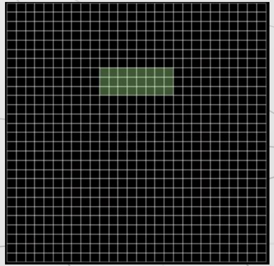
edge detection is a techniques that neural networks can use to extract features from image pixels. The network learns to detect edges and other patterns in the input image by analyzing the changes in intensity or color values between adjacent pixels.



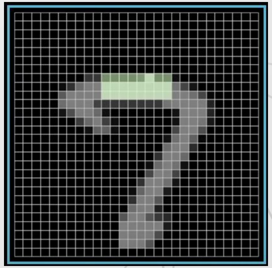
Given a image of number 7 we want to detect a edge



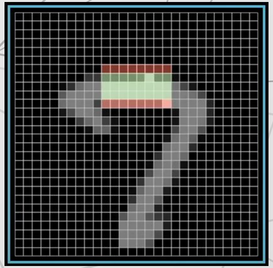
Pick out a edge from the shape we want to analyze from pixels



On the same grid give the edge a color to hold a value



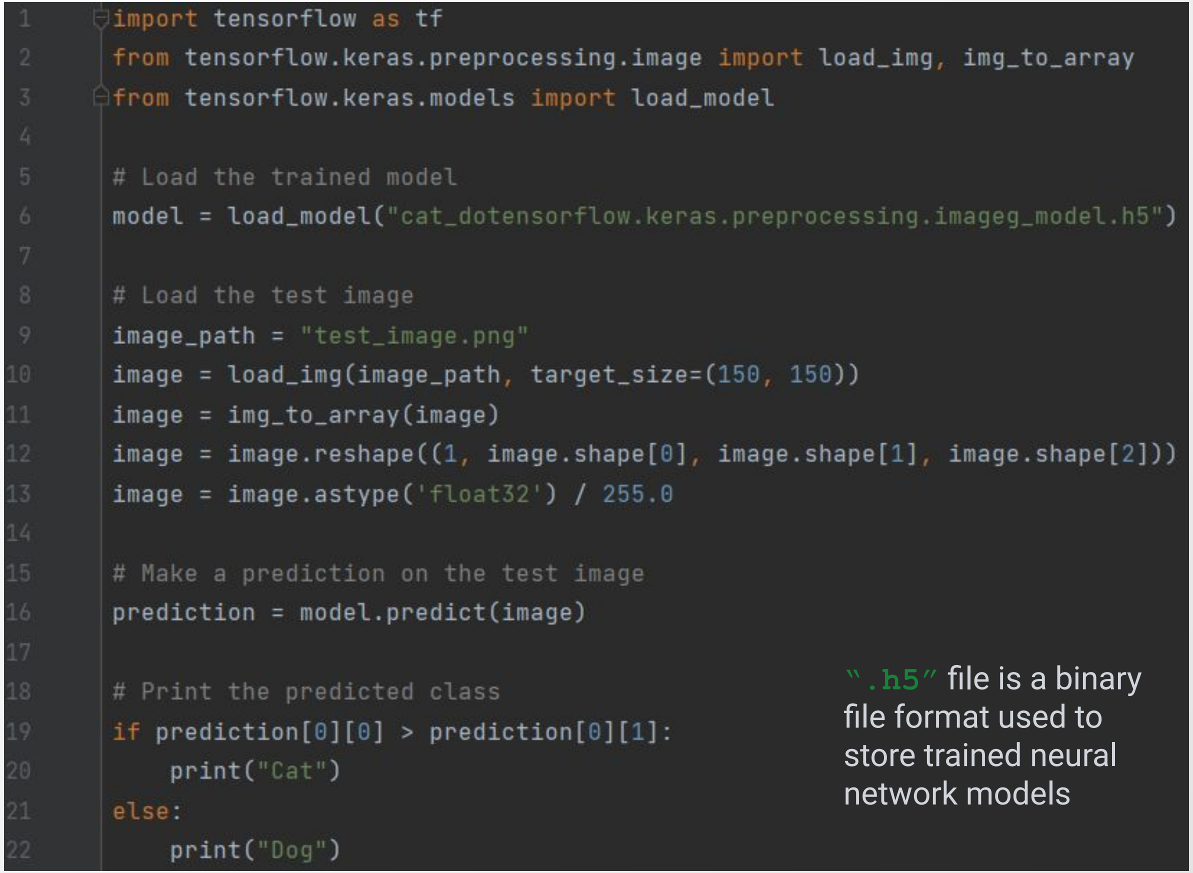
Putting the 2 images together we see the edge we are analyzing



Here green represents the lit up and red the surrounding pixels

To pick up a edge we can give weights to the lit up (green) and black surrounding (red) pixels and figure out if there is a edge from analyzing the the bright pixels vs dark pixels

Neural network code example python



```python
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model("cat_dotensorflow.keras.preprocessing.imageg_model.h5")

# Load the test image
image_path = "test_image.png"
image = load_img(image_path, target_size=(150, 150))
image = img_to_array(image)
image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
image = image.astype('float32') / 255.0

# Make a prediction on the test image
prediction = model.predict(image)

# Print the predicted class
if prediction[0][0] > prediction[0][1]:
    print("Cat")
else:
    print("Dog")
```

".h5" file is a binary file format used to store trained neural network models

This simple neural network can take a image and decide whether its a dog or a cat this program relies on pythons many machine learning imports. To run this you need 2 things. 1 the model this is your training data that has the info to depict a dog and cat this is can use a neural network. 2 the image you have to provide a image for the model to make a prediction. What this code and imports do is format everything and process it so that model works along with the image it passes the image through the model then prints a result in the terminal based on the output.

THANKS FOR LISTENING

Sources:
https://www.ibm.com/topics/machine-learning
https://www.youtube.com/watch?v=aircAruvnKk&ab_channel=3Blue1Brown
https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/
https://towardsdatascience.com/deep-learning-with-python-neural-networks-complete-tutorial-6b53c0b06af0
https://en.wikipedia.org/wiki/Neural_network