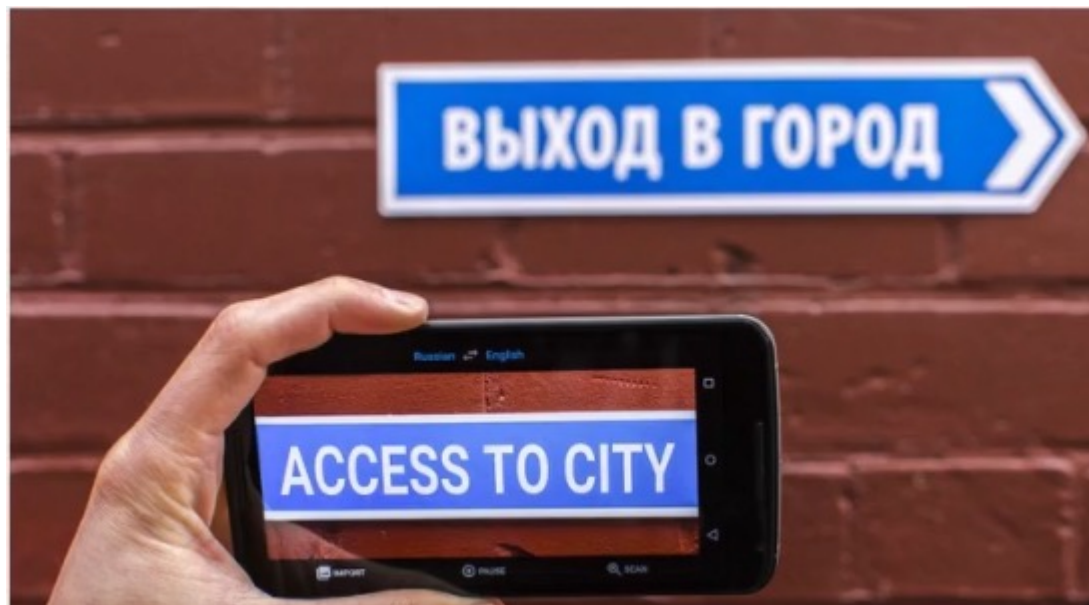




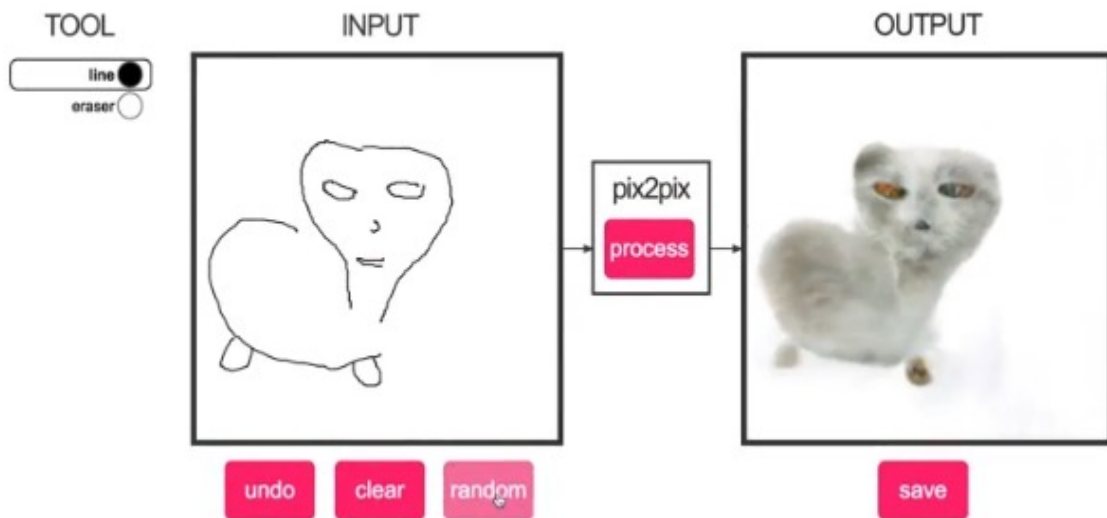
Deep Learning in Motion

Unit 1 – Module 2 Getting Started

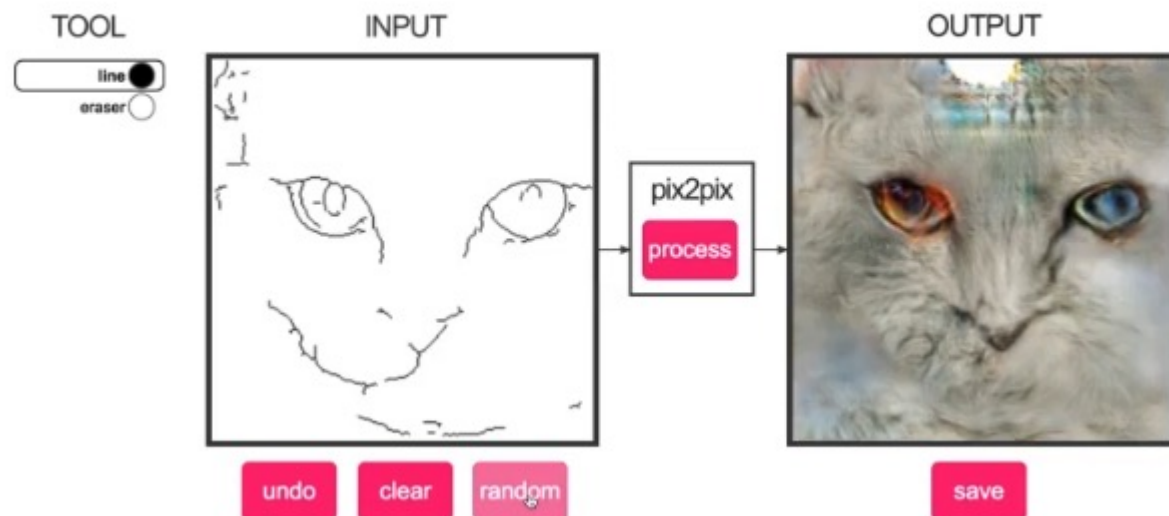
Beau Carnes



edges2cats



edges2cats

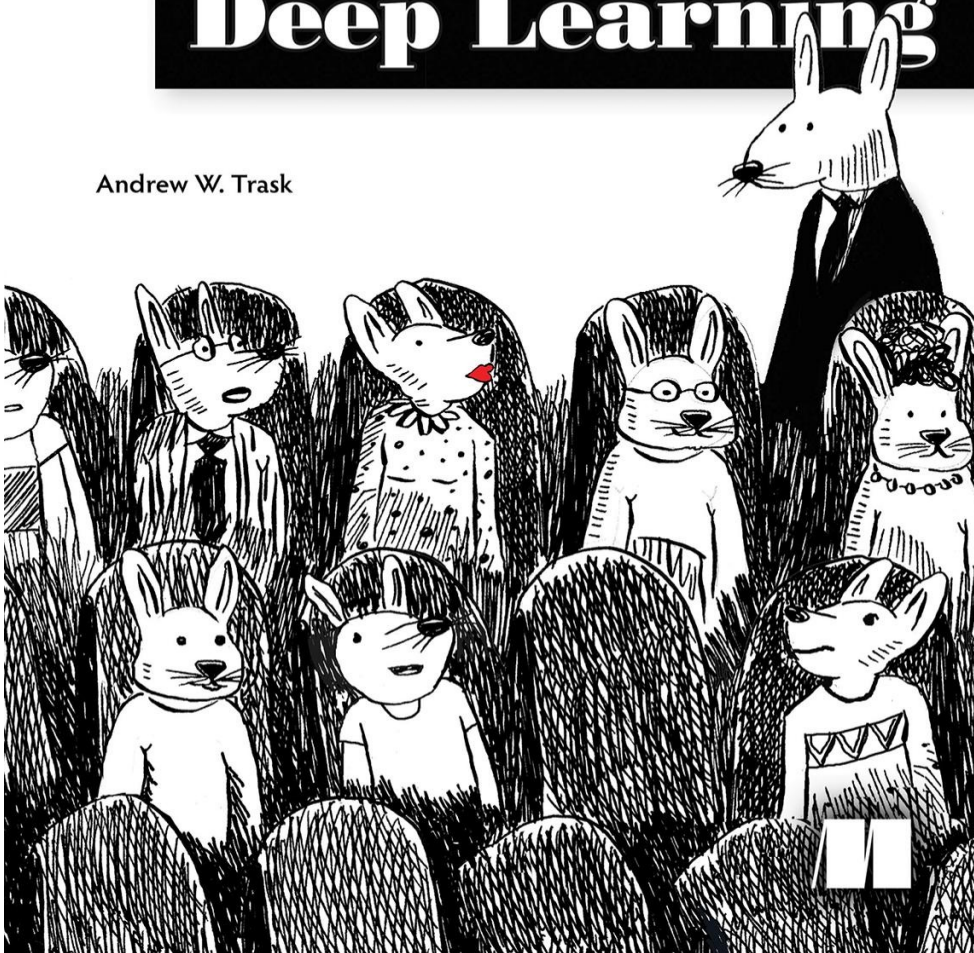


<https://www.youtube.com/watch?v=AmUC4m6w1wo>

grokking


Deep Learning

Andrew W. Trask



Google Scholar

SIGN IN

 **Andrew Trask** [FOLLOW](#)

DeepMind, University of Oxford, and OpenMined
Verified email at google.com - [Homepage](#)

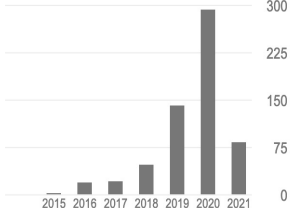
[Deep Learning](#) [Differential Privacy](#) [Secure Multi-Party Comput...](#) [Federated Learning](#)
[Natural Language Processing](#)

[GET MY OWN PROFILE](#)

	All	Since 2016
Citations	629	620
h-index	9	9
i10-index	9	9

TITLE	CITED BY	YEAR
A generic framework for privacy preserving deep learning T Ryffel, A Trask, M Dahl, B Wagner, J Mancuso, D Rueckert, ... arXiv preprint arXiv:1811.04017	121	2018
sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings A Trask, P Michalak, J Liu arXiv preprint arXiv:1511.06388	121	2015
Neural arithmetic logic units A Trask, F Hill, S Reed, J Rae, C Dyer, P Blunsom arXiv preprint arXiv:1808.00508	105	2018
Systems and methods for neural language modeling A Trask, D Gilmore, M Russell US Patent 10.339.440	80	2019

Cited by



Year	Citations
2015	10
2016	20
2017	30
2018	50
2019	150
2020	280
2021	100

<https://www.manning.com/books/grokking-deep-learning>


<https://github.com/iamtrask/Grokking-Deep-Learning>

<https://scholar.google.com/citations?user=2Ajxf1sAAAAJ&hl=en>

Data science technology for groundbreaking research.



```
saqibali — -bash — 74x6
[(base) Saqibs-MacBook-Pro:~ saqibali$ python --version
Python 3.8.5
(base) Saqibs-MacBook-Pro:~ saqibali$ Jupyter notebook]
```

 jupyter


QuitLogout

FilesRunningClusters










Select items to perform actions on them.

UploadNew ↕

☐ 0 ▾

 /

Name ▾Last ModifiedFile size

<input type="checkbox"/>  Desktop	25 minutes ago	
<input type="checkbox"/>  Documents	21 hours ago	
<input type="checkbox"/>  Downloads	4 hours ago	
<input type="checkbox"/>  Dropbox	38 minutes ago	
<input type="checkbox"/>  Movies	4 years ago	
<input type="checkbox"/>  Music	2 days ago	
<input type="checkbox"/>  opt	a day ago	
<input type="checkbox"/>  Pictures	4 years ago	
<input type="checkbox"/>  Public	4 years ago	

Select items to perform actions on them.

<input type="checkbox"/> 0 ▾	 /	Name ▾	
<input type="checkbox"/>	 Desktop		
<input type="checkbox"/>	 Documents		
<input type="checkbox"/>	 Downloads		
<input type="checkbox"/>	 Dropbox		
<input type="checkbox"/>	 Movies		4 years ago
<input type="checkbox"/>	 Music		2 days ago
<input type="checkbox"/>	 opt		a day ago
<input type="checkbox"/>	 Pictures		4 years ago
<input type="checkbox"/>	 Public		4 years ago

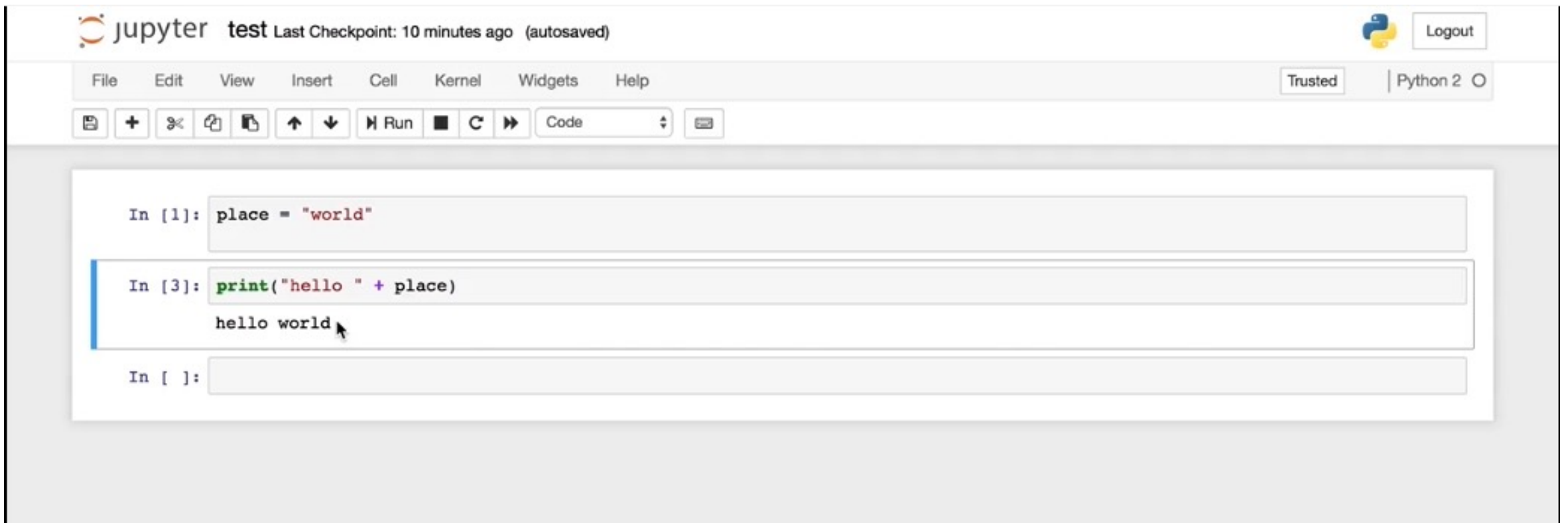
Notebook:

Other:

Text File

Folder

Terminal



One of the most important parts of learning deep learning is the ability to stop a network while it's training and tear apart absolutely every piece to see what it looks like. This is something Jupyter Notebook is incredibly useful for.

What are Lists?

- Lists are used to store multiple items in a single variable.
- Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are [Tuple](#), [Set](#), and [Dictionary](#), all with different qualities and usage.
- Lists are created using square brackets:
- For Example:

```
thislist = ["apple", "banana", "cherry"]  
print(thislist)
```

List Items

- List items are ordered, changeable, and allow duplicate values.
- List items are indexed, the first item has index [0], the second item has index [1] etc.

Ordered

- When we say that lists are ordered, it means that the items have a defined order, and that order will not change.
- If you add new items to a list, the new items will be placed at the end of the list.

Changeable

- The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.

Allow Duplicates

- Since lists are indexed, lists can have items with the same value.
- **For Example** - Lists allow duplicate values:

```
thislist = ["apple", "banana", "cherry", "apple", "cherry"]  
print(thislist)
```


List Length

- To determine how many items a list has, use the len() function.
- For Example - Print the number of items in the list.

```
thislist = ["apple", "banana", "cherry"]  
print(len(thislist))
```

List Items - Data Types

- List items can be of any data type.
- **For Example** - String, int and boolean data types.

```
list1 = ["apple", "banana", "cherry"]
```

```
list2 = [1, 5, 7, 9, 3]
```

```
list3 = [True, False, False]
```

A list with strings, integers and boolean values:

```
list1 = ["abc", 34, True, 40, "male"]
```

ObjectType

- **type()**
- From Python's perspective, lists are defined as objects with the data type 'list'.

<class 'list'>

What is the data type of a list?

```
mylist = ["apple", "banana", "cherry"]  
print(type(mylist))
```

The list() Constructor

- It is also possible to use the list() constructor when creating a new list.
- **For Example** - Using the list() constructor to make a List:

```
thislist = list(("apple", "banana", "cherry")) # note the double  
round-brackets  
print(thislist)
```

List	Tuples	Dictionary
A list is mutable	A tuple is immutable	A dictionary is mutable
Lists are dynamic	Tuples are fixed size in nature	In values can be of any data type and can repeat, keys must be of immutable type
List are enclosed in brackets[] and their elements and size can be changed	Tuples are enclosed in parenthesis () and cannot be updated	Tuples are enclosed in curly braces { } and consist of key:value
Homogenous	Heterogeneous	Homogenous
Example: List = [10, 12, 15]	Example: Words = ("spam", "eggs") Or Words = "spam", "eggs"	Example: Dict = {"ram": 26, "abi": 24}
<u>Access:</u> print(list[0])	<u>Access:</u> print(words[0])	<u>Access:</u> print(dict["ram"])
Can contain duplicate elements	Can contain duplicate elements. Faster compared to lists	Cant contain duplicate keys, but can contain duplicate values
Slicing can be done	Slicing can be done	Slicing can't be done
<u>Usage:</u> ❖ List is used if a collection of data that doesnt need random access. ❖ List is used when data can be modified frequently	<u>Usage:</u> ❖ Tuple can be used when data cannot be changed. ❖ A tuple is used in combination with a dictionary i.e.a tuple might represent a key.	<u>Usage:</u> ❖ Dictionary is used when a logical association between key:value pair. ❖ When in need of fast lookup for data, based on a custom key. ❖ Dictionary is used when data is being constantly modified.

NumPy Introduction

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

Why is NumPy Faster Than Lists?

- NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.
- This behaviour is called locality of reference in computer science.
- This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

Which Language is NumPy written in?

- NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

Create a NumPy ndarray Object

- NumPy is used to work with arrays. The array object in NumPy is called ndarray.
- We can create a NumPy ndarray object by using the array() function.
- For Example.

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

To create an ndarray, we can pass a list, tuple or any array-like object into the array() method, and it will be converted into an ndarray:

```
import numpy as np
arr = np.array((1, 2, 3, 4, 5))
print(arr)
```




The diagram consists of three concentric circles. The outermost circle is blue and represents Artificial Intelligence. Inside it is a teal circle representing Machine Learning. Inside the teal circle is a smaller orange circle representing Deep Learning. This visualizes that Deep Learning is a subset of Machine Learning, which is a subset of Artificial Intelligence.

ARTIFICIAL INTELLIGENCE

Programs with the ability to learn and reason like humans

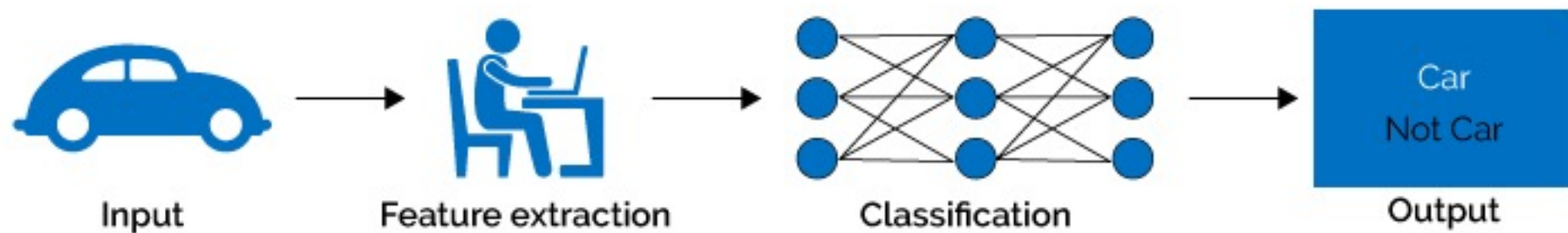
MACHINE LEARNING

Algorithms with the ability to learn without being explicitly programmed

DEEP LEARNING

Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Machine Learning



Deep Learning

