Machine Learning and Convolutional Neural Networks

(and why their beauty is their simplicity)

Nick Malleson

Machine Learning is Fun!

The world’s easiest introduction to Machine Learning

Machine Learning is Fun!

Fantastic series of blog posts on *medium*

- https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471

I'm basically just copying material from those blog posts

What I'm going to cover:

- **What is machine learning?** *(Everyone in this room probably knows this already, even if you don't realise it)*
- **What are neural networks?**
- **What are convolutional neural networks?**

Why?

- Convolutional neural networks are absolutely amazing!!!

What has this got to do with geography?

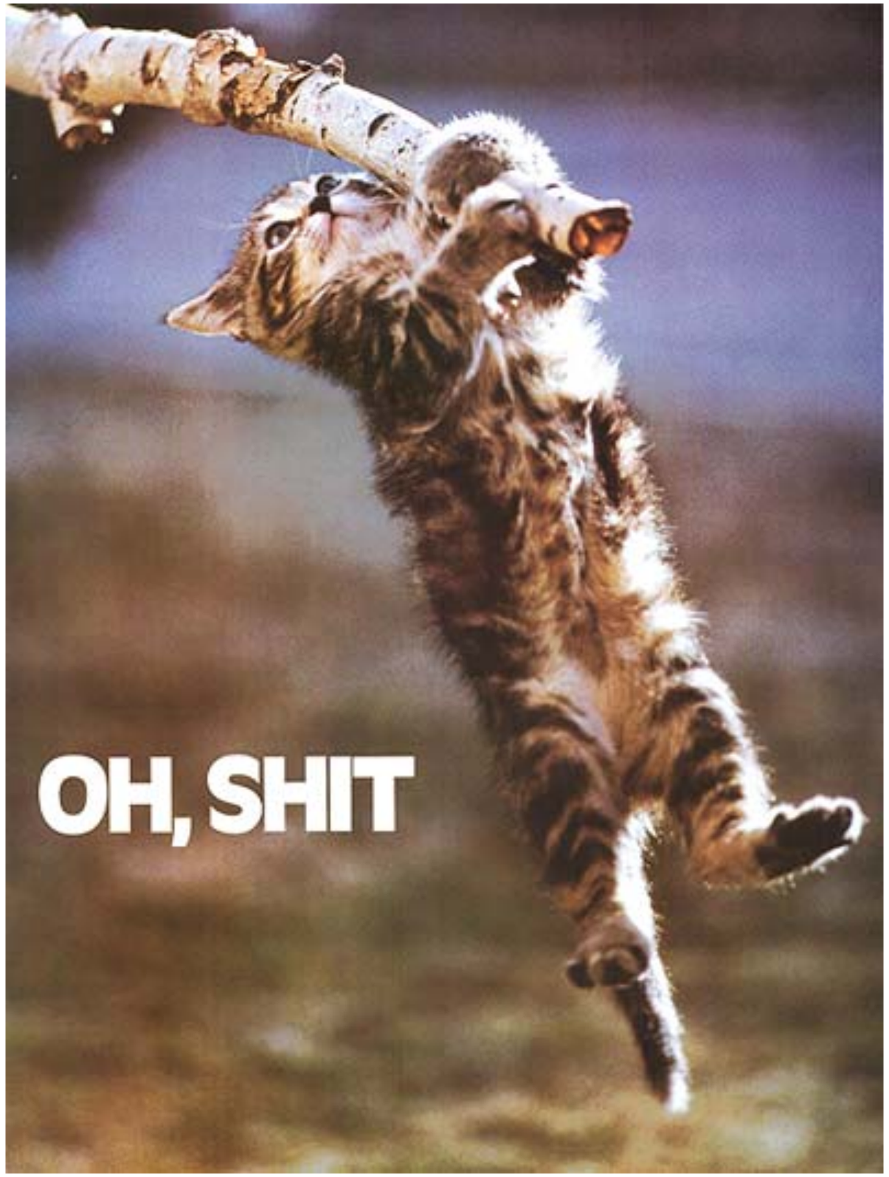
Why?

Convolutional neural networks are absolutely amazing!!!

They take a really simple idea (neural networks), make a couple of small adaptations, and then allow a computer to learn how to recognise complicated images.

The beauty is in their simplicity

The maths is pretty easy, but I’m not going to touch on it here


OH, SHIT

1. What is machine learning?

“*Machine learning is the idea that there are generic algorithms that can tell you something interesting about a set of data without you having to write any custom code specific to the problem. Instead of writing code, you feed data to the generic algorithm and it builds its own logic based on the data*.”

Question: name a machine learning algorithm that you have used

(One) Answer: regression!

Other machine learning algorithms

Regression Trees & Random Forest

Naive Bayes

K-Nearest Neighbors

Learning Vector Quantization

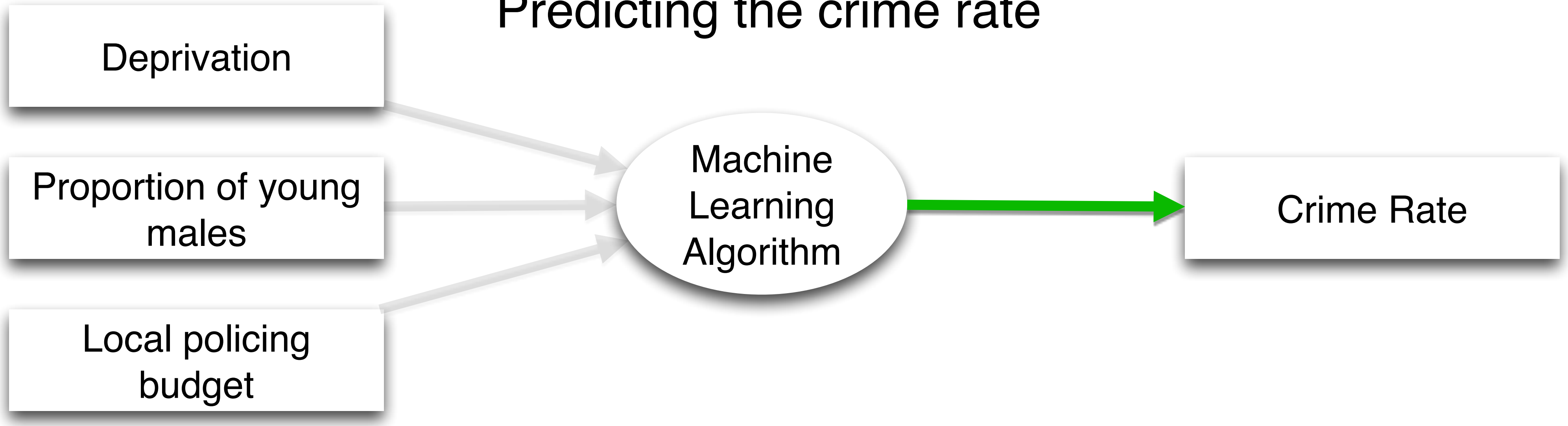
Support Vector Machines

https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11

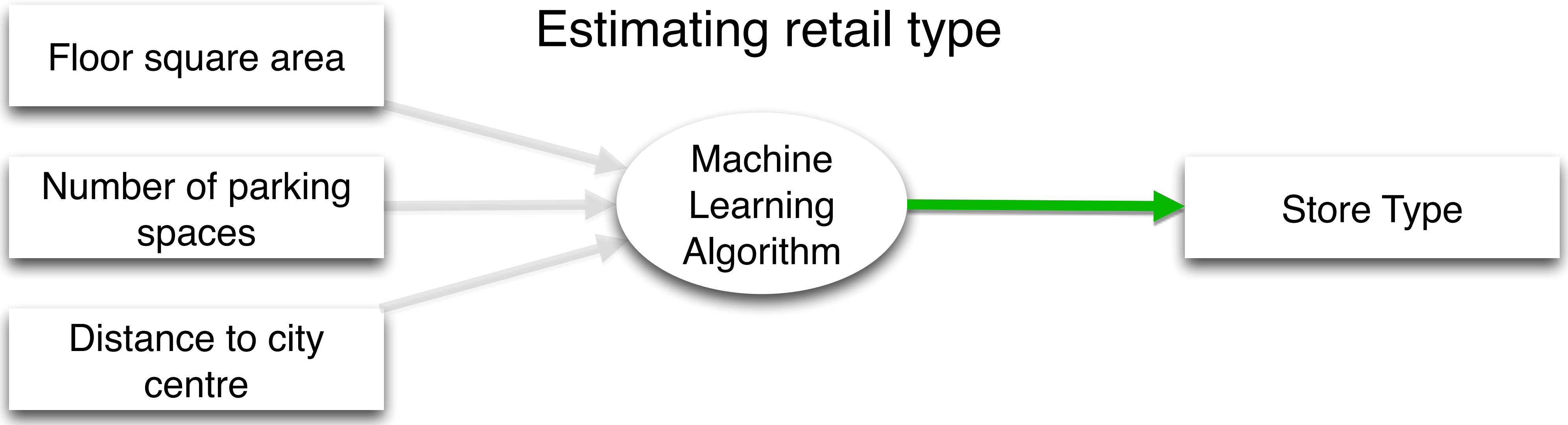
Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
$r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2}\sqrt{\sum(y_i-\bar{y})^2}}$
$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$

A Tour of The Top 10 Algorithms for Machine Learning Newbies

Predicting the crime rate



Estimating retail type



Two kinds of machine learning: **Supervised** and Unsupervised

With **supervised** learning, the algorithm uses *training* data to learn what to output given a certain input.

E.g. for the crime example, we have information on each neighbourhood which we can use to train our algorithm

Area	Deprivation	Proportion of young males	Policing budget	Crime Rate
A	15	0.6	500,000	5
B	190	0.5	350,000	80
C	60	0.7	500,000	70
D	20	0.3	550,000	15
E	120	0.5	120,000	30

Two kinds of machine learning: **Supervised** and Unsupervised

With **supervised** learning, the algorithm uses *training* data to learn what to output given a certain input.

E.g. for the crime example, we have information on each neighbourhood which we can use to train our algorithm

And can then use the algorithm to estimate the crime rate in unknown areas, or in areas where something changes (e.g. a decrease in the police budget)

Area	Deprivation	Proportion of young males	Policing budget	Crime Rate
F	20	0.55	400,000	??

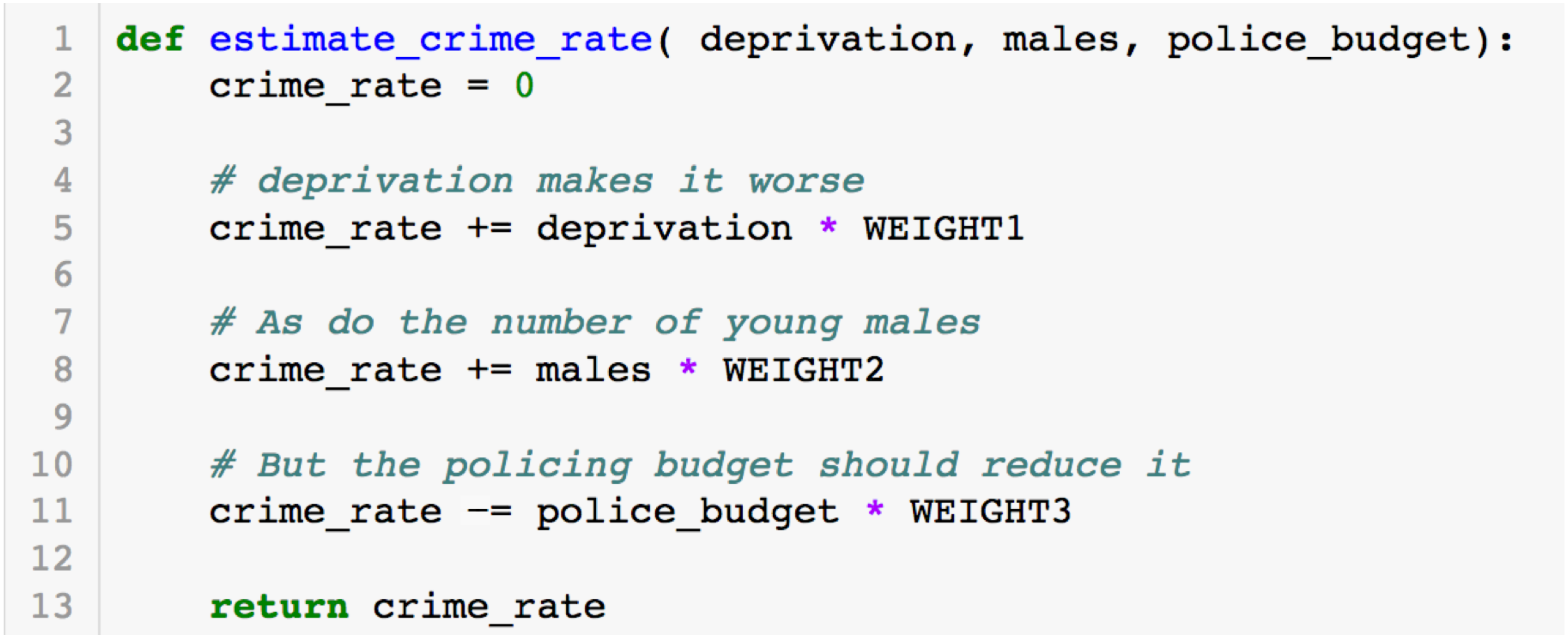
Two kinds of machine learning: Supervised and **Unsupervised**

With **unsupervised** learning, instead of trying to make predictions based on our data, we look for interesting patterns.

E.g. in the retail example, we might try to classify stores into particular types given their characteristics

Store	Floor square area	No. parking spaces	Distance to the city centre	Store category
A	15	10	500	?
B	190	500	350	?
C	60	210	5,000	?
D	20	0	10	?
E	120	90	1,500	?

Example code



```python
def estimate_crime_rate( deprivation, males, police_budget):
    crime_rate = 0

    # deprivation makes it worse
    crime_rate += deprivation * WEIGHT1

    # As do the number of young males
    crime_rate += males * WEIGHT2

    # But the policing budget should reduce it
    crime_rate -= police_budget * WEIGHT3

    return crime_rate
```

The process of ‘learning’

We can *train* the algorithm by:

Running all of our data through it and estimating crime rates

Comparing our estimates to the known rates to calculate error (or ‘cost’)

Repeat, but trying new combinations of WEIGHT values

Eventually, we should find WEIGHT values that predict crime rates in all areas well

This is basically all that machine learning algorithms do, although slightly more complicatedly in practice

But is this *learning*?

Problems

Overfitting

Is the problem solvable with the data?

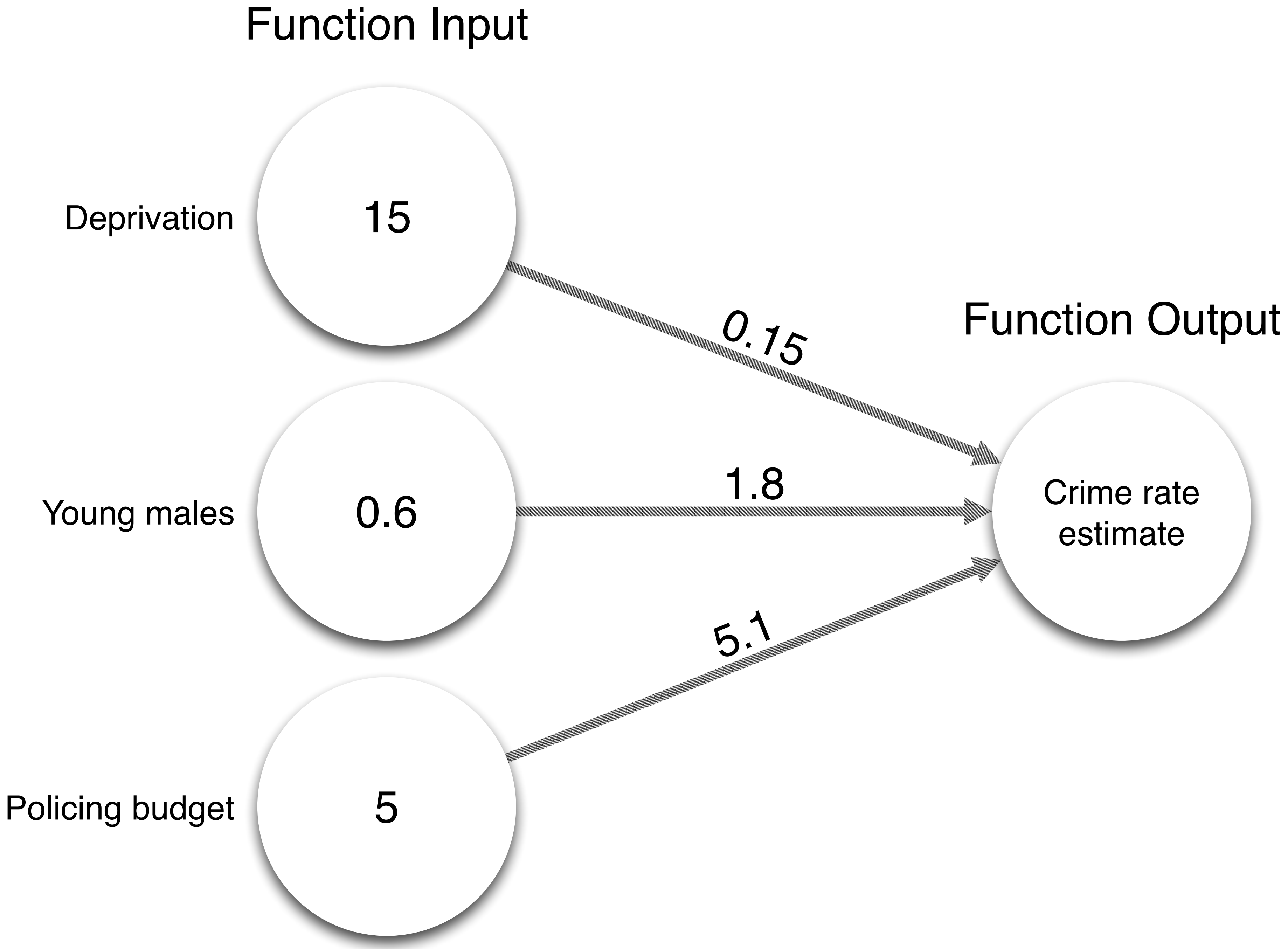
- If we had data on the number of people with first names beginning with ‘S’ then we could not predict crime!

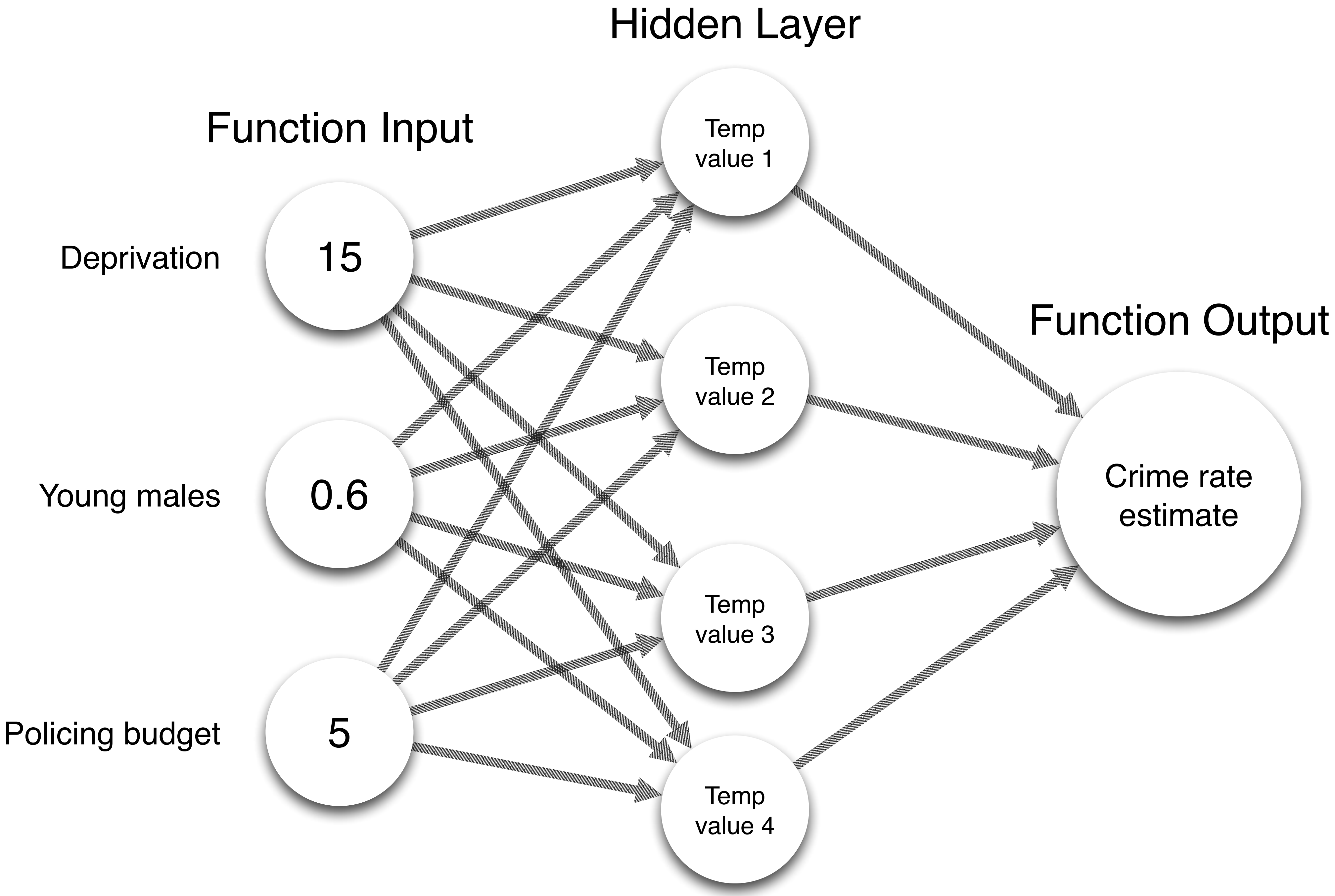
Is the model appropriate?

- E.g. the previous example (basically linear regression) assumes that the inputs have a linear affect on the outputs
  - If police budgets go up, the crime rate goes down.
- But what if areas with very high deprivation and lots of young men actually have higher community cohesion and lower crime?

2. What are neural networks?

https://medium.com/@ageitgey/machine-learning-is-fun-part-2-a26a10b68df3


Function Input
Deprivation
15
Young males
0.6
Policing budget
5
0.15
1.8
5.1
Function Output
Crime rate estimate


Hidden Layer
Function Input
Deprivation
15
Young males
0.6
Policing budget
5
Temp value 1
Temp value 2
Temp value 3
Temp value 4
Function Output
Crime rate estimate

2. What are neural networks?

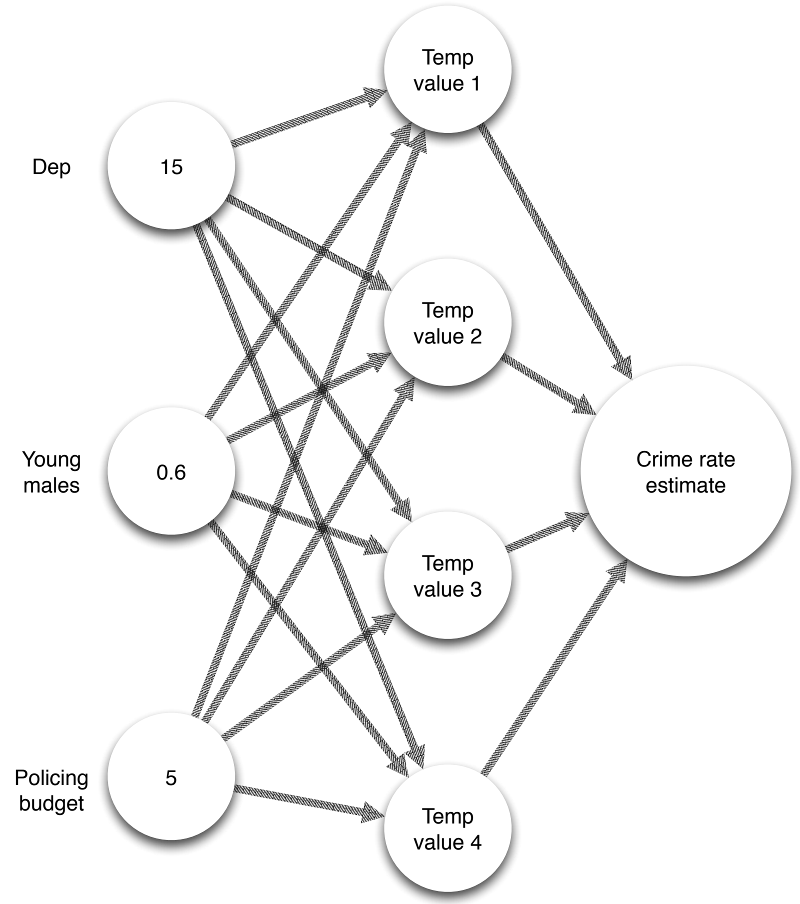
This is a simple neural network

We just add another layer of nodes in between the input and output

Then the task is to find values for all of the new weights, based on our input data.

You can do this with **backcasting** *(I wont try to explain it though!)*

And what about **deep learning**? You just add more hidden layers!



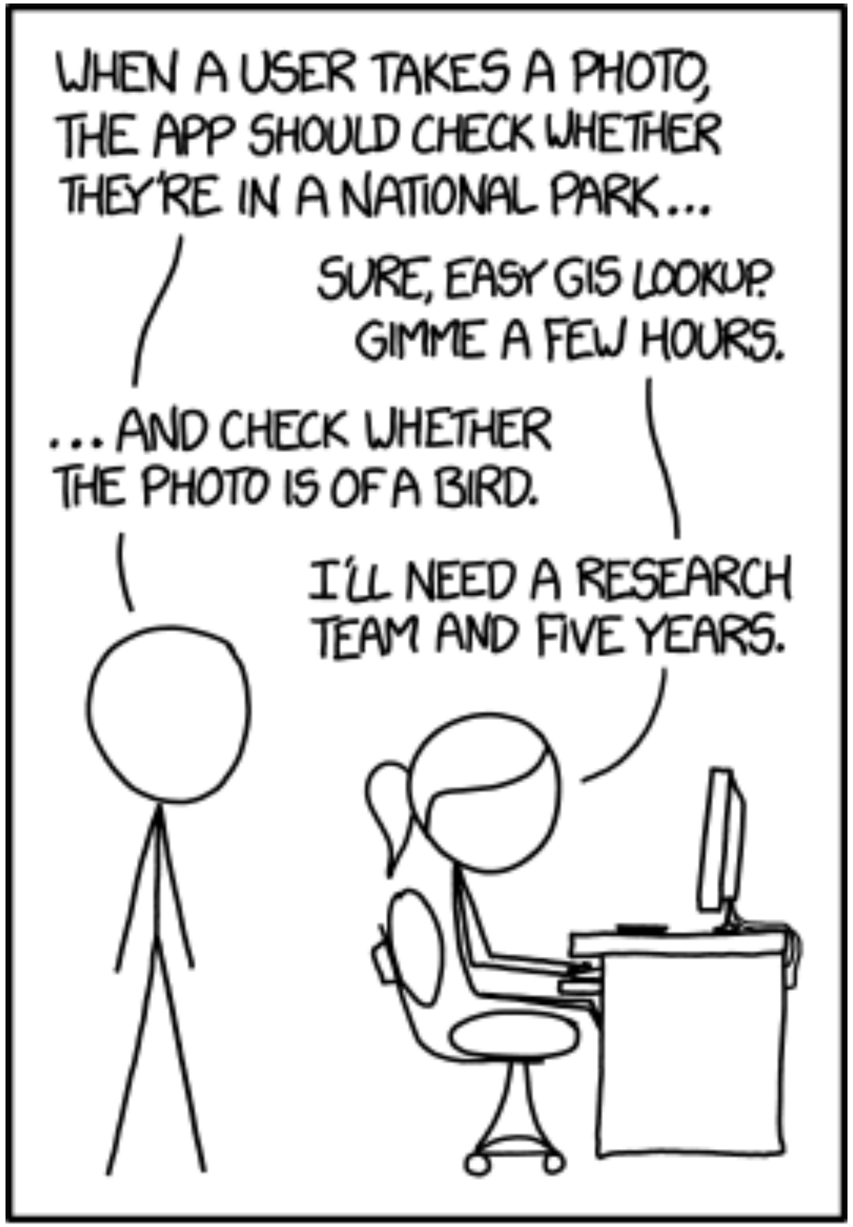
3. What are convolutional neural networks?

https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

(I also draw heavily on: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/)

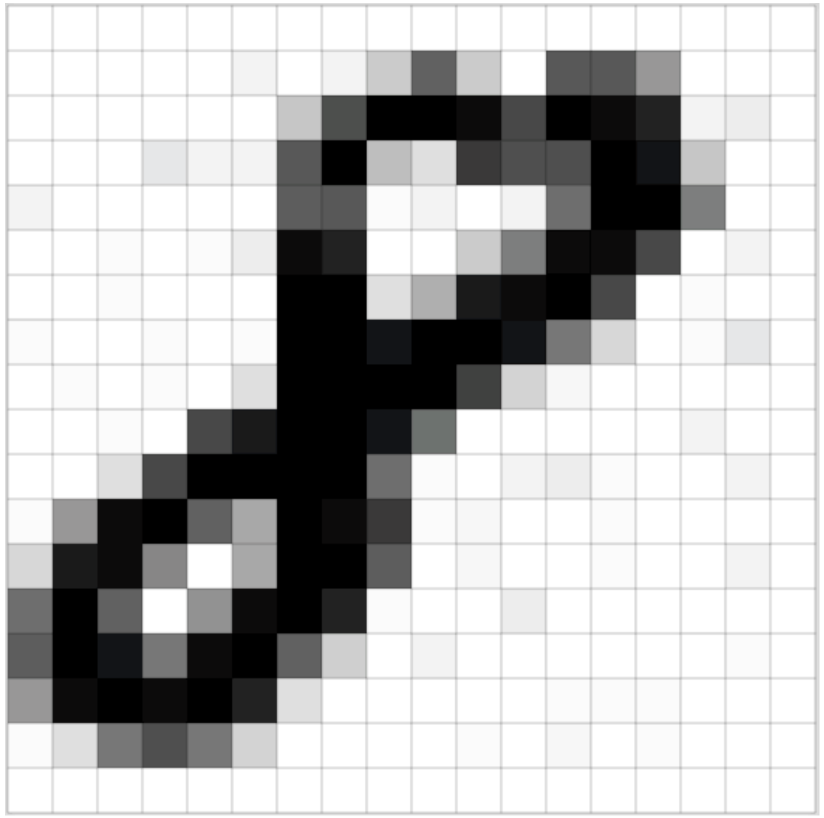
In short – they are deep neural networks with clever hidden layers

Feed them loads of images of certain objects, and they should be able to learn which images contain the object and which don’t https://xkcd.com/1425/


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

On a computer, images are stored as lists of numbers



=

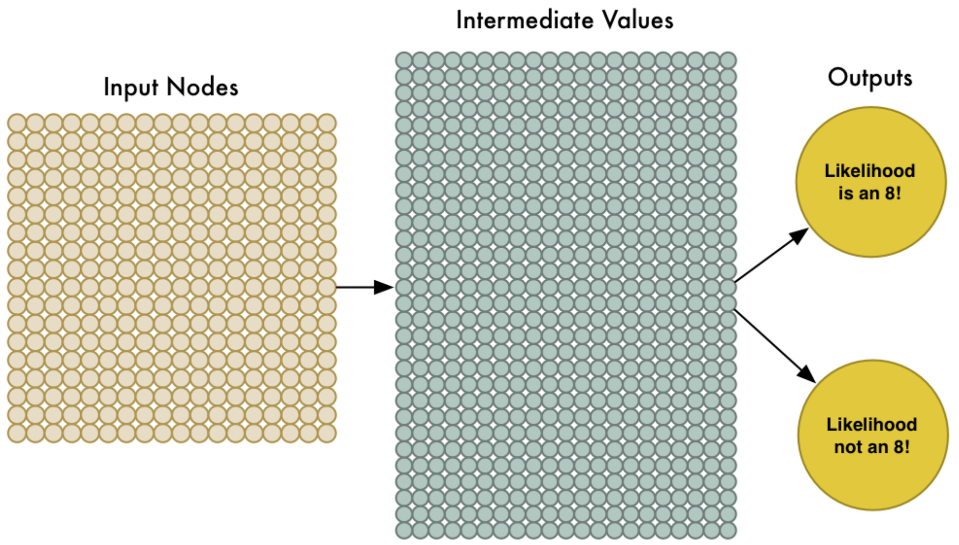
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
 0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

3. What are convolutional neural networks?

On a computer, images are stored as lists of numbers

> E.g. black and white picture with 18x18 pixels = 324 numbers

We can enlarge our neural network to have 324 input nodes; one for each pixel



To train the model: input loads of images and let the network try to work out what the image contains.

But, this only works on very simple images

It will break if the images are moved, rotated, etc.

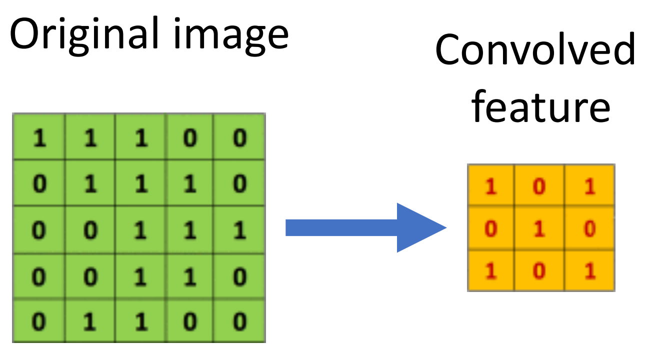
Solution: **convolution**

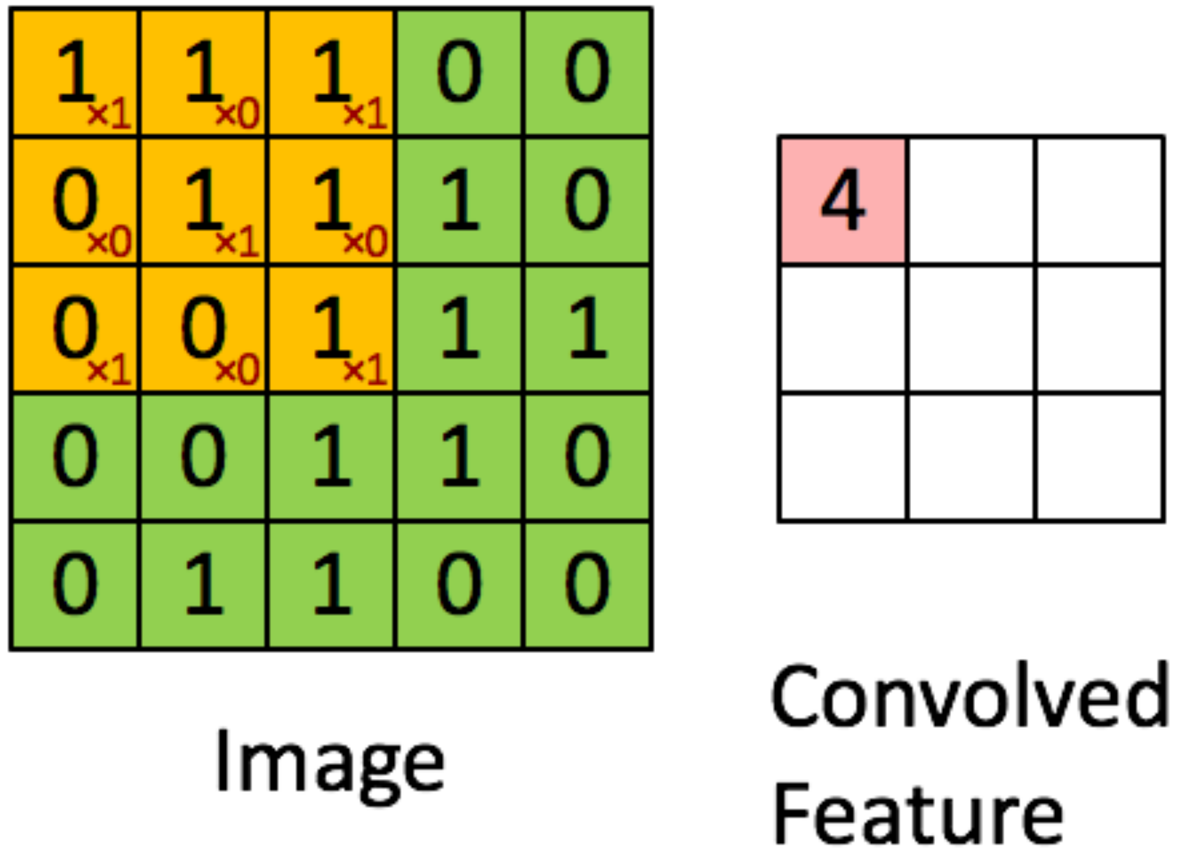
How convolution works

1. The image is broken into many overlapping tiles
2. Each tile is fed into a small neural network.
3. The small network (also called a ‘filter’) can be configured to pick out ‘interesting’ features.
4. The output of the filter is *convolved feature*.
5. Continue filtering, using different filters and some other clever image manipulations…

The really amazing thing: the algorithm identifies the features that it needs to look for!!

Creating the convolved feature





https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Create a pipeline

Convolution

Non Linearity (ReLU)

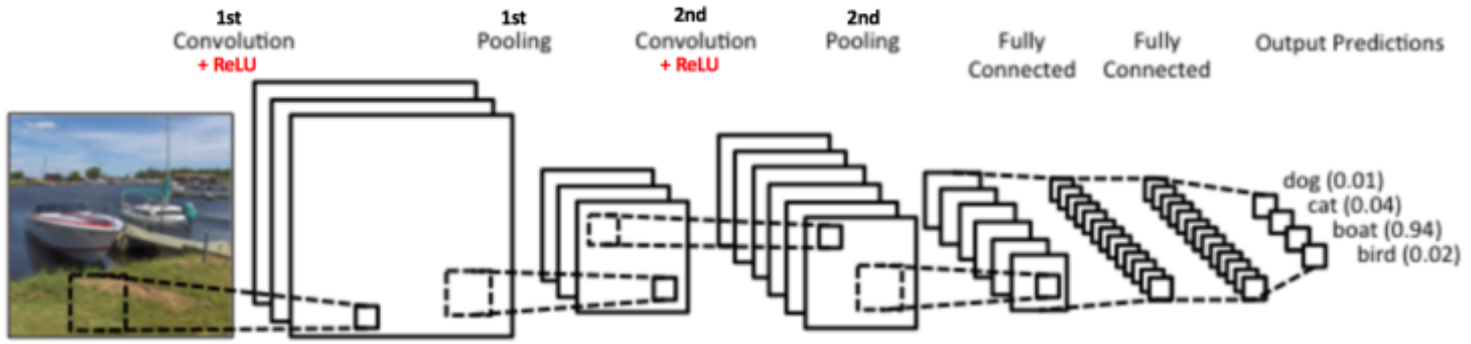
Introduce some non-linearity after convolution

Pooling

(down-sample, e.g. min, max, avg..)

Classification (fully connected layer)

use previously identified features to classify the image



https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

The really amazing thing: the algorithm identifies the features that it needs to look for!!

With a cleverly configured neural network, computers can pick objects out of images

Data

10,000s images might not be sufficient

Google et al use millions to train their networks.

“In machine learning, having more data is almost always more important that having better algorithms.”

This is why Google, Facebook, etc. now release lots of their algorithms publicly. The value is in the data, not the code.

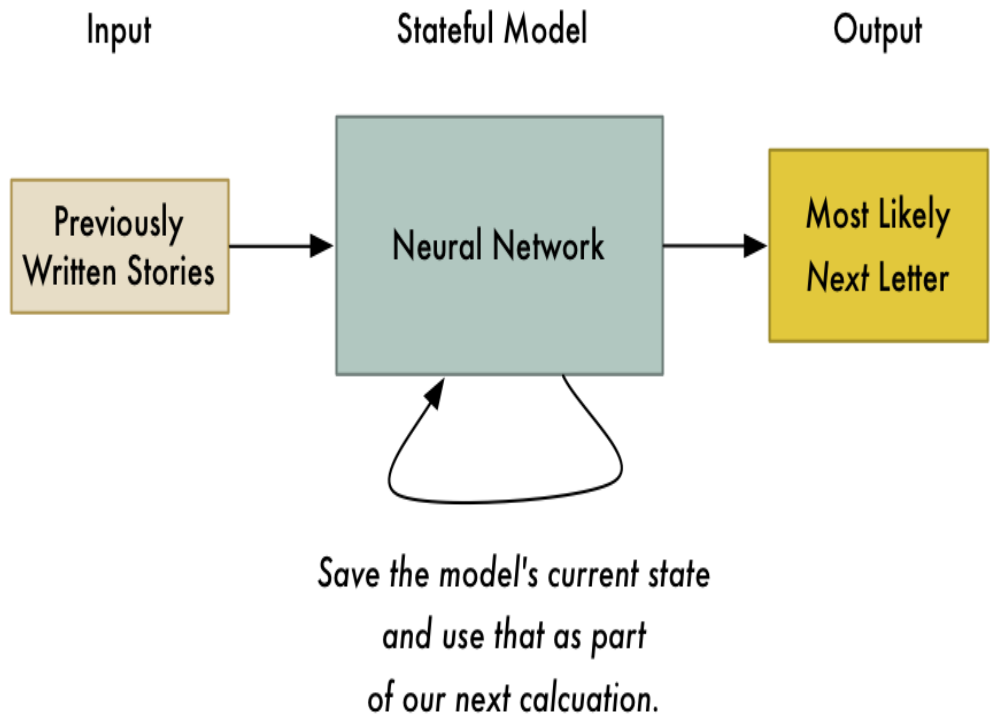
Another neat trick – adding memory

Recurrent neural networks keep track of their state

Output depends on current input, *and previous inputs*

Can be trained on published text and then write new text in the style of the author (kind of)

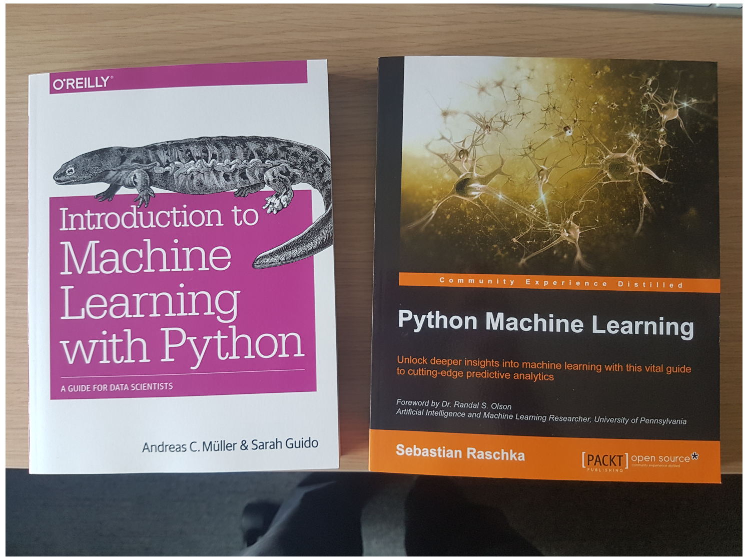
Possible application: predicting traffic?


Input
Stateful Model
Output
Previously Written Stories
Neural Network
Most Likely Next Letter
Save the model's current state and use that as part of our next calcuation.

Want more?

Read the blog on medium

(Search for 'Machine Learning is Fun')


O'REILLY
Introduction to Machine Learning with Python
A GUIDE FOR DATA SCIENTISTS
Andreas C. Müller & Sarah Guido
Community Experience Distilled
Python Machine Learning
Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics
Foreword by Dr. Randal S. Olson
Artificial Intelligence and Machine Learning Researcher, University of Pennsylvania
Sebastian Raschka
[PACKT] open source

Do Andrew Ng's Machine Learning course on Coursera *(it's really good)*
https://www.coursera.org/learn/machine-learning

I also took loads of material from this blog post:

https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

There are lots of practical text books

Machine Learning and Convolutional Neural Networks