HOGESCHOOL UTRECHT

Webinar
Machine learning in practice

Bert Wassink
Joost Krapels
Stefan Leijnen
Sieuwert van Otterloo

July 17th, 2020

About the team

Dr. Stefan Leijnen , dr. Sieuwert van Otterloo

Utrecht University of Applied Sciences, Artificial Intelligence research group.

Applied research into responsible and ethical use of AI

Dr. Sieuwert van Otterloo, Joost Krapels MSc.

ICT Institute

Practical IT advice to large and medium-sized companies. Focus on software project management, security, privacy / personal data and AI.

Bert Wassink MSc.

Dataworkz – smart and fast data expert

Applied data science

Agenda

- Explanation of machine learning and neural networks
- Training a machine learning algorithm (Demonstration)
- Practical and ethical consequences of using machine learning

How to train a network yourself

1. Install python (https://www.python.org/downloads/) and the Jupyter toolbox:
2. Download the data set and python notebook at https://github.com/swzaken/cars-neuralnetwork
3. Install python packages. You can use the following commands

Library	Description	Command to install
Updated version of Pip	Installing packages	*python –m pip install –upgrade pip*
Numpy	Arrays and numbers	*pip install numpy*
MatPlotLib	Data visualization	*pip install matplotlib*
Pillow	Image processing	*pip install pillow*
JuPyter Notebooks	Running the code	*pip install jupyterlab*
Tensorflow	Machine learning algorithms	*pip install tensorflow*

If you do multiple python projects, we recommend an environment manager, such as Conda.
https://www.anaconda.com/products/individual

What is Artificial Intelligence?

“The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.”

Edsger W Dijkstra (computer scientist, 1930- 2002)

Artificial Intelligence is concerned with letting computers do tasks that humans would use intelligence for. There are many different artificial intelligence techniques, varying from rule based expert systems to biology-inspired methods.

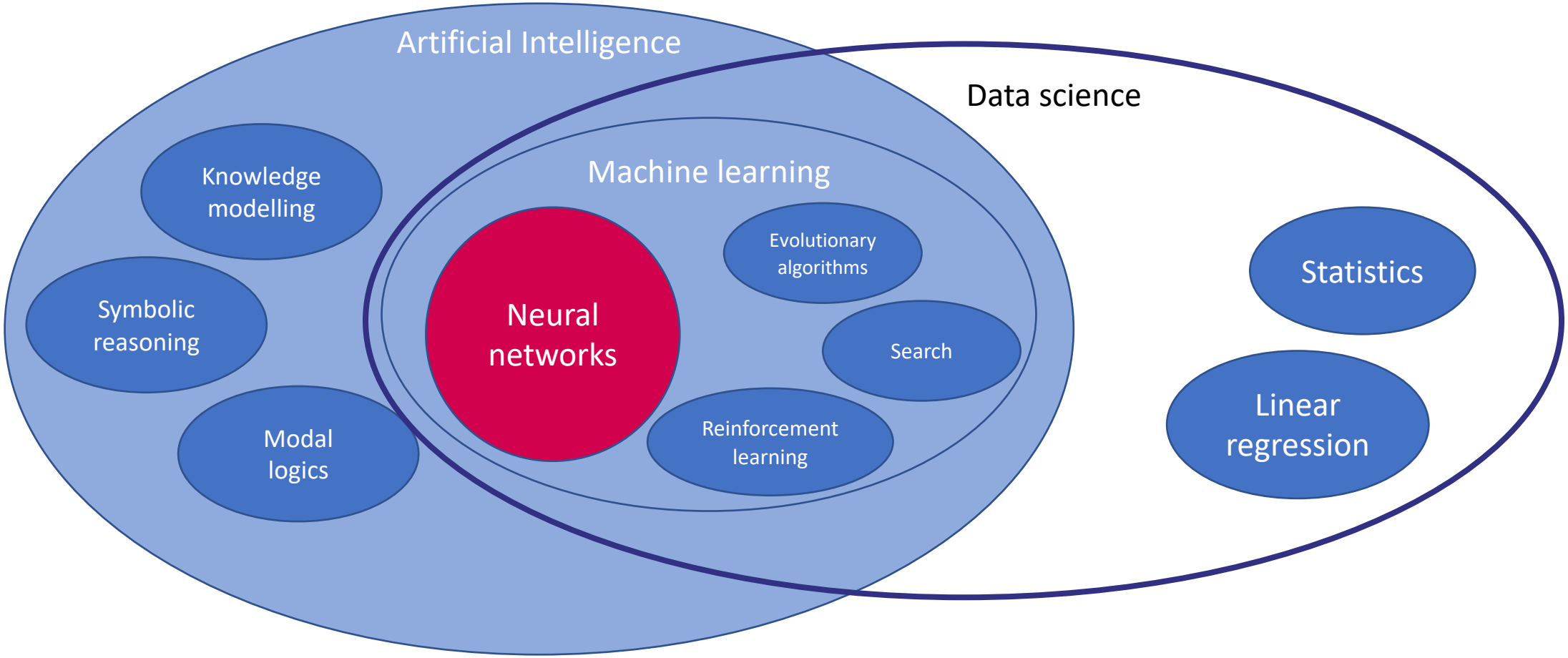
Why is Artificial Intelligence popular?

Discovery of new algorithms that can ‘learn’ more tasks with less assistance from human experts

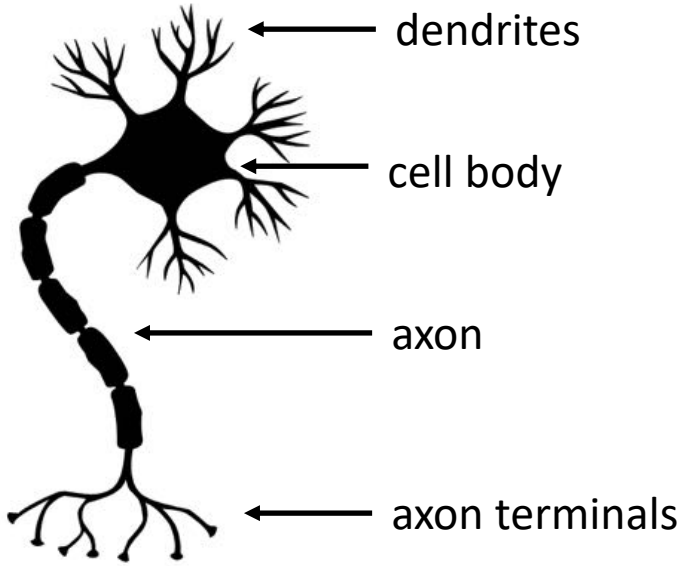
Digital innovation (e.g. search engines, e-commerce, smart meters, social media, digital camera’s) have made it much easier to collect and connects data sets

Improvements into parallel computing (driven by computer graphics chips) have made processing large data sets feasible

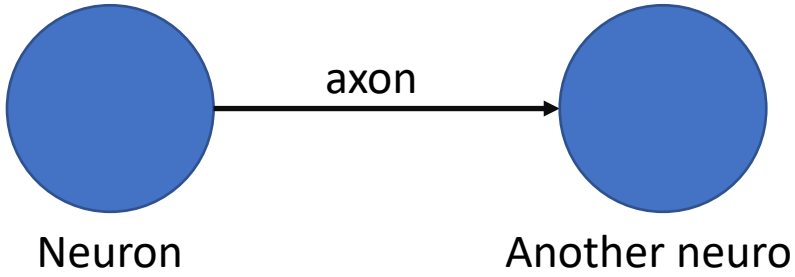
Neural networks is a subfield of AI



What are neural networks?



Symbolic illustration of actual neuron

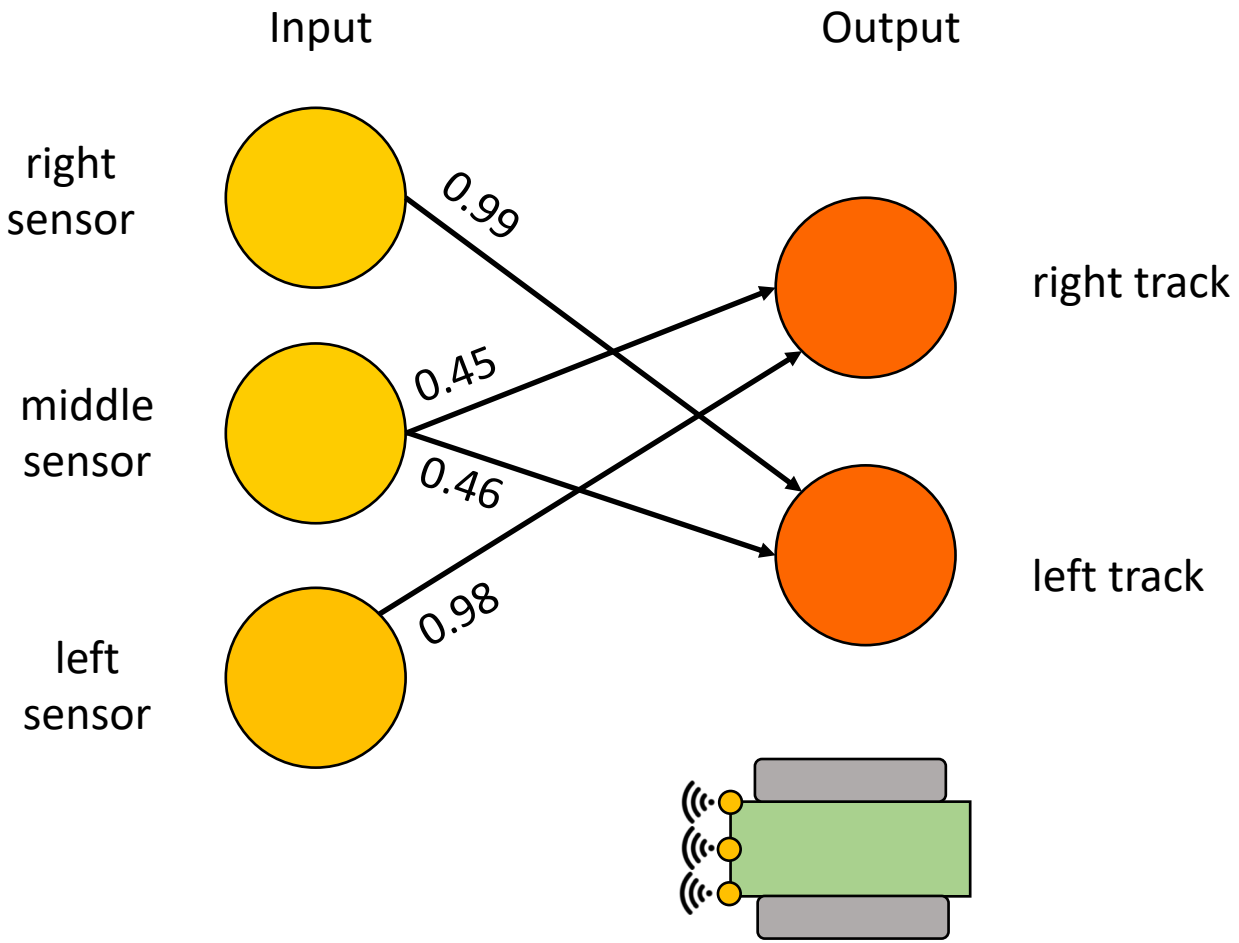


Neurons are cells that are part of the nervous system (brain, spinal cord, sensory organs, and nerves). They are often depicted as a core cell with many connections (axons/tendons) to other neurons.
The neurons thus form a neural network.

- Some neurons respond to external signals, e.g. light or sound
- Some neurons control muscle cells
- Other neurons respond to signals from other neurons. Once they are agitated by other signals, they will ‘fire’ and send a signal to other neurons

Illustration credit: Cenira Costa

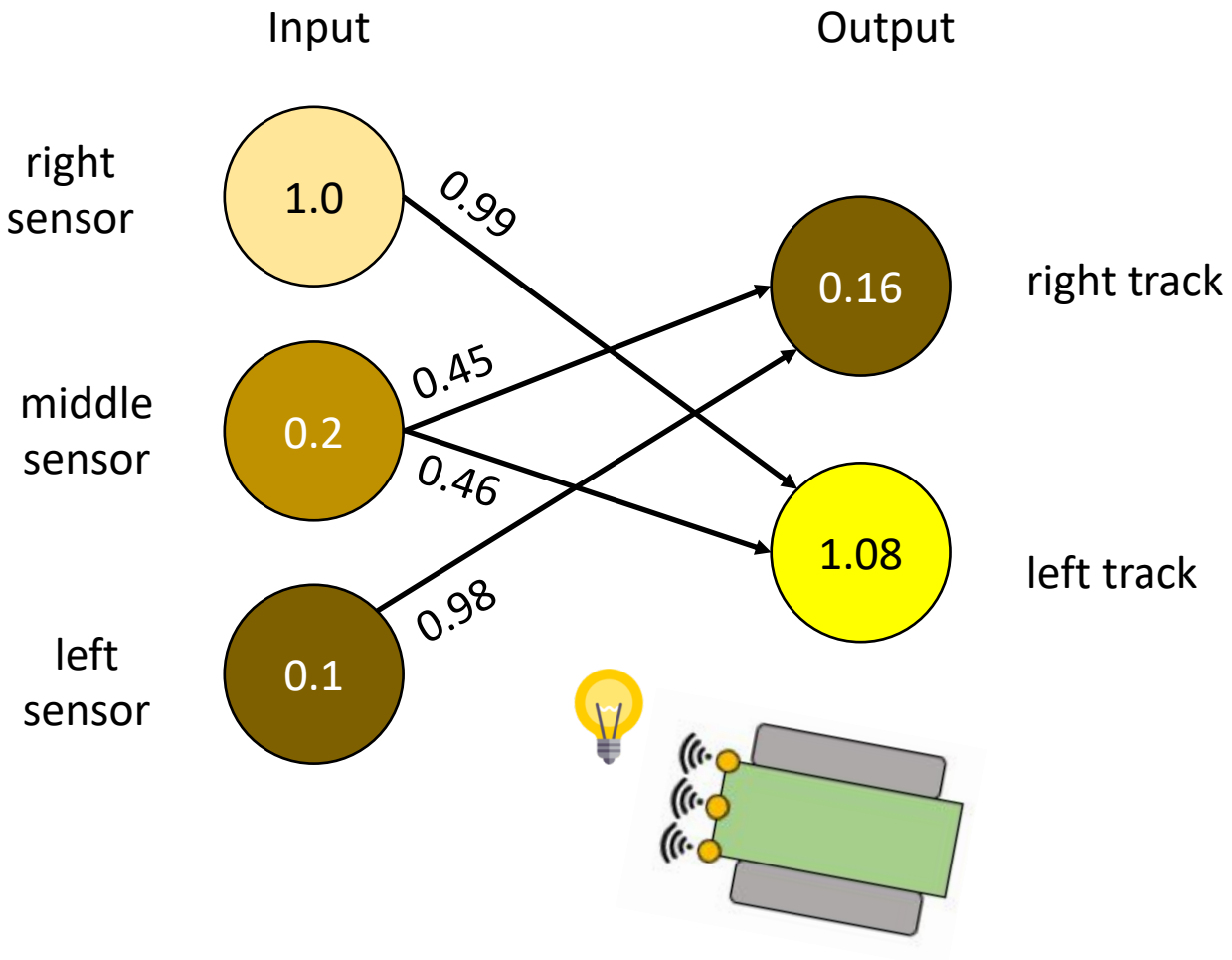
Artificial neural networks



The following simple network can be used for making tiny robots that either move towards or away from the light.

- Each neuron has an ‘activation’ (0, 1 or any value in between)
- Each link has a ‘weight’, typically between -1.0 and 1.0

Artificial neural networks - example



If there is a light on the right side of the robot, the left track gets the strongest signal. The robot will turn to the right, so towards the light.

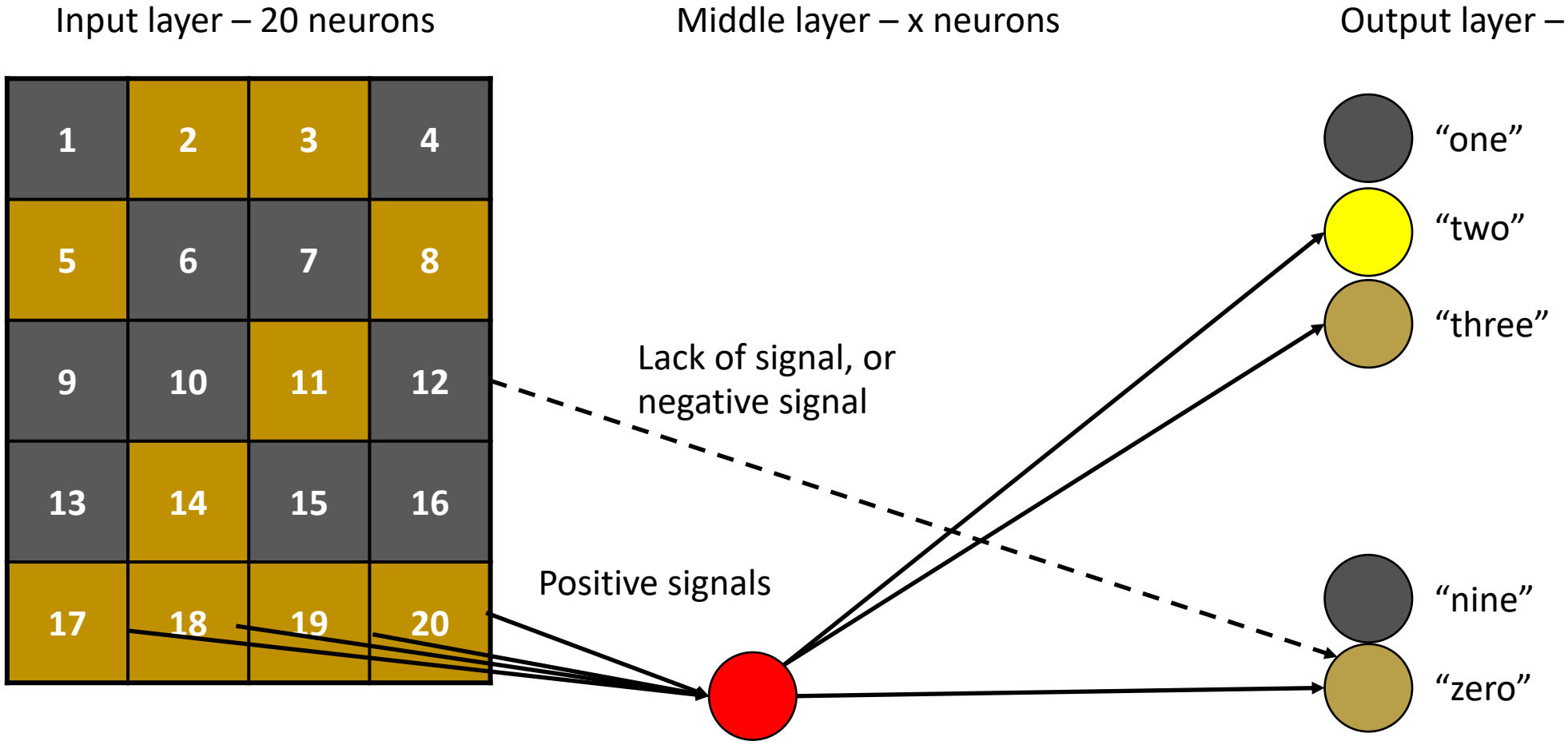
- For each neuron, you add up the sum of the input-neurons * weight
- If the sum is above the threshold (e.g. 1), the neuron is activated

Input Activation (left track) =

1.0 * 0.99. + 0.2 * 0.46 =

0.99 + 0.09 = 1.08

Artificial neural networks – image classification



Training steps

1. Each neural network has inputs and outputs that are determined by the problem to solve.
2. The researcher decides on the number of neurons and number of ‘hidden’ layers. More complex problems require more layers, or larger layers.
3. The weights are determined through ‘machine learning’. Using large amounts of example data, the computer determines a good set of weights.
4. The researcher must check the quality of the network at the end. Training algorithms are not guaranteed to work.

How to train?

Step 1: data collection

1. Select a domain and problem statement.
2. Collect a lot of data. We use a published training set with car photos.
3. Annotate each photo. You need, for training purposes, the correct answers to any questions you want to ask.

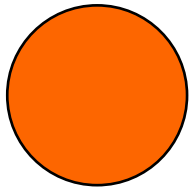
What questions could we ask for a car photo data set?

Source: https://ai.stanford.edu/~jkrause/cars/car_dataset.html

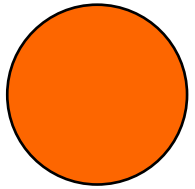
Possible questions

Question	ML-task	How hard is it for humans?
Is there a car in the picture?	Detection	Easy
Which side of the car is visible?	Classification	Easy
What is the color of the car?	Classification	Easy
What is the brand of the car?	Classification	Medium
What body style (minivan, coupe, sedan, convertible) does the car have?	Classification	Medium
What is the value of the car?	Prediction/ estimation	Hard
What is the build year of the car?	Prediction/ estimation	Hard/Impossible
Do two pictures show the same car?	Identification	Hard/Impossible

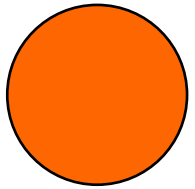
Example question 1: Car color



Red



Blue



Black

- For this task we need pictures of cars in different colors. We sort these images by color in different folders.
- The images must have a standard size, for example 256x256 pixels. You can resize the photos by hand or have your program do it for you.
- The required output of the neural network is an indication how likely the car is ‘red’, ‘blue’ or any color we will train.

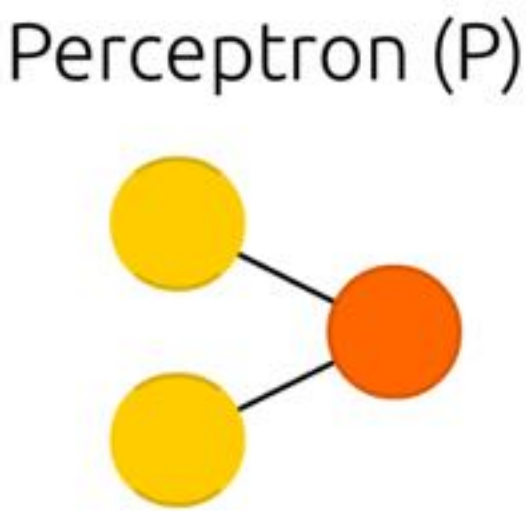
Example question 2: car rotation

CLASS_NAMES = [”Front", "Back", "Right", "Left", "Front right", "Front Left”]

In this webinar, we will use another property cars in pictures have: rotation.

- We sort these photos by angle in different folders.
- We wrote some code to resize all images to 256x256 pixels.
- The neural network will have six output neurons: one output neuron per angle.

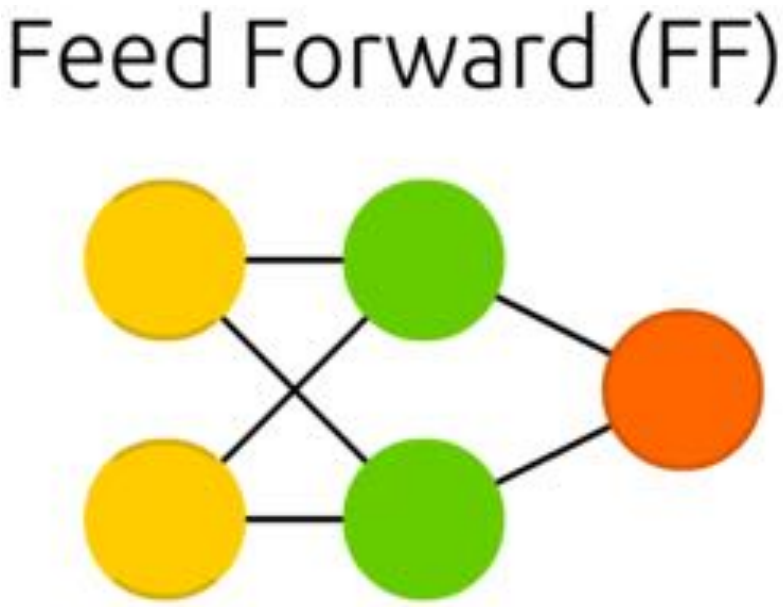
Simple topology – perceptron



- A perceptron has no hidden layers. The output neuron(s) is/are a linear combination of the input neurons.
- Training perceptrons is really fast.
- Many problems are not trainable for perceptrons.

Source: Neural Network Zoo – Asimov Institute

Feed Forward neural network

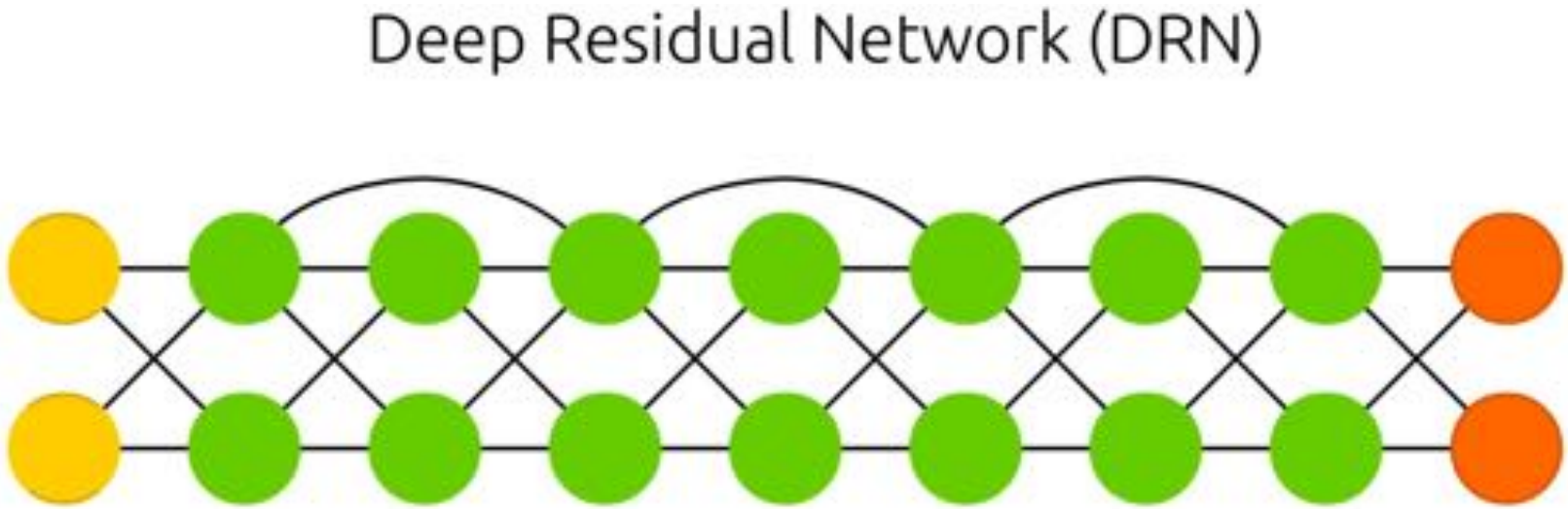


- A feed forward network only has connections from left to right. This makes it easier to develop training algorithms.
- You present the input on the left and get predictions/classifications on the right
- These algorithms can act as a black box. If there are enough hidden neurons, it can be hard to explain how a decision was made.

Today we will use feed forward networks.

Source: Neural Network Zoo – Asimov Institute

‘Deep’ networks

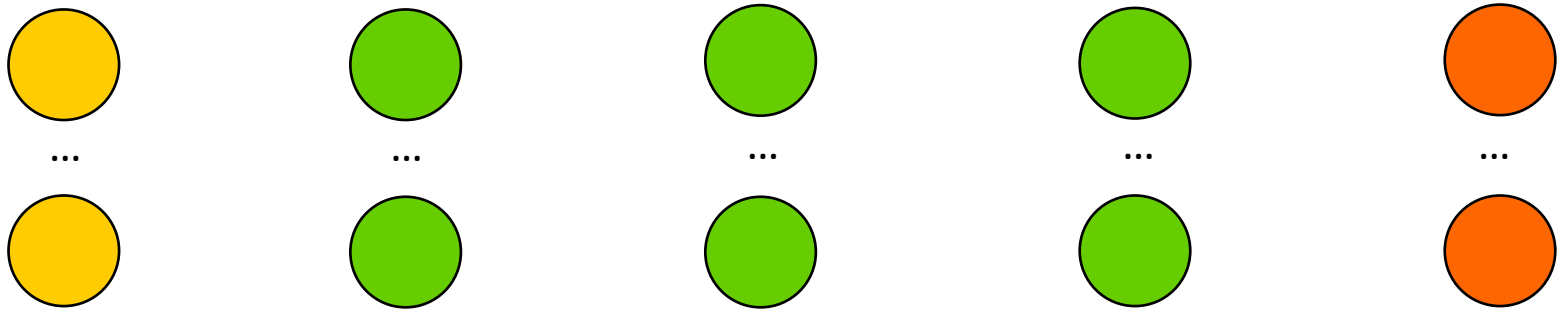


- ‘Deep’ networks have many hidden layers. They can learn more subtle, complex models
- Networks with fewer hidden layers (1 or 2) are easier to train than more complex models. They are good for simpler tasks

Source: Neural Network Zoo – Asimov Institute

Model 1: simple feed forward

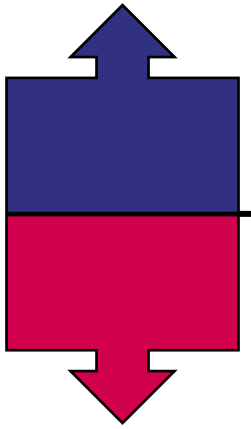
```
dense_model = keras.Sequential([
    keras.layers.Flatten(input_shape=(IMG_PIX, IMG_PIX, 3), name='flatten'),
    keras.layers.Dense(64, activation='relu', name='fc1'),
    keras.layers.Dense(6, name='fc2'),
])

probability_model = keras.Sequential([dense_model, tf.keras.layers.Softmax()])
```



Dataset 1: Training and testing

All images

336 images



Training images

Test images

Training and test data

Training images

270 images

- The algorithm is trained using the training data.
- Typically, you train until most are classified correctly.
- Some training algorithms separate the training data into training and ‘test’ data for testing during training

Test images

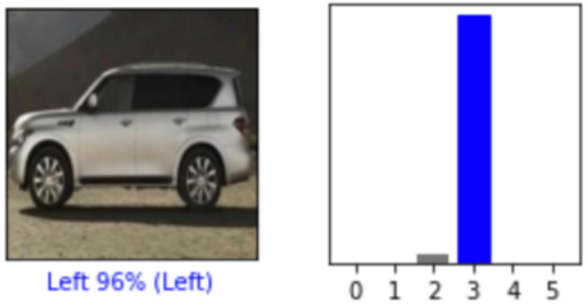
66 images

- The algorithm is tested on the remaining test set, consisting of images that the algorithm has never seen before.
- The performance on these new images is often lower, since these images might contain new challenges.

CLASS_NAMES = [”Front", "Back", "Right", "Left", "Front right", "Front Left”]

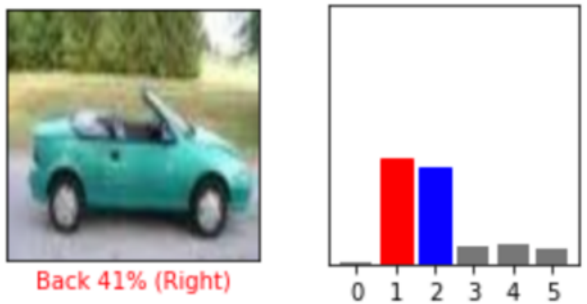
Example algorithm results

CORRECT RESULT



- One neuron is 96% activated. It is neuron 3, aka ‘Left’
- According to our label, this car is indeed facing left

INCORRECT RESULT



- No neuron has a more than 50% activation
- Two neurons have a 40%+ activation: ‘back’ and ‘right’. The highest activation is ‘back’
- According to our label, this car is facing right

By checking the neural network values for all training and all test images, you can compute the accuracy (percentage of correct answers) for both the training set and the test set.

Quiz 1

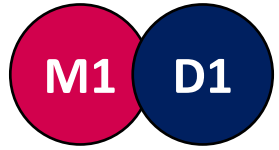
What percentage accuracy do you expect on the training data?

What percentage accuracy do you expect on the test data?

Which categories will be confused most often?

Let’s find out in the first demo!

Quiz 1 answers (simple model, small dataset)



61% accuracy on the training set (decent)

29% accuracy on the test set (very poor)

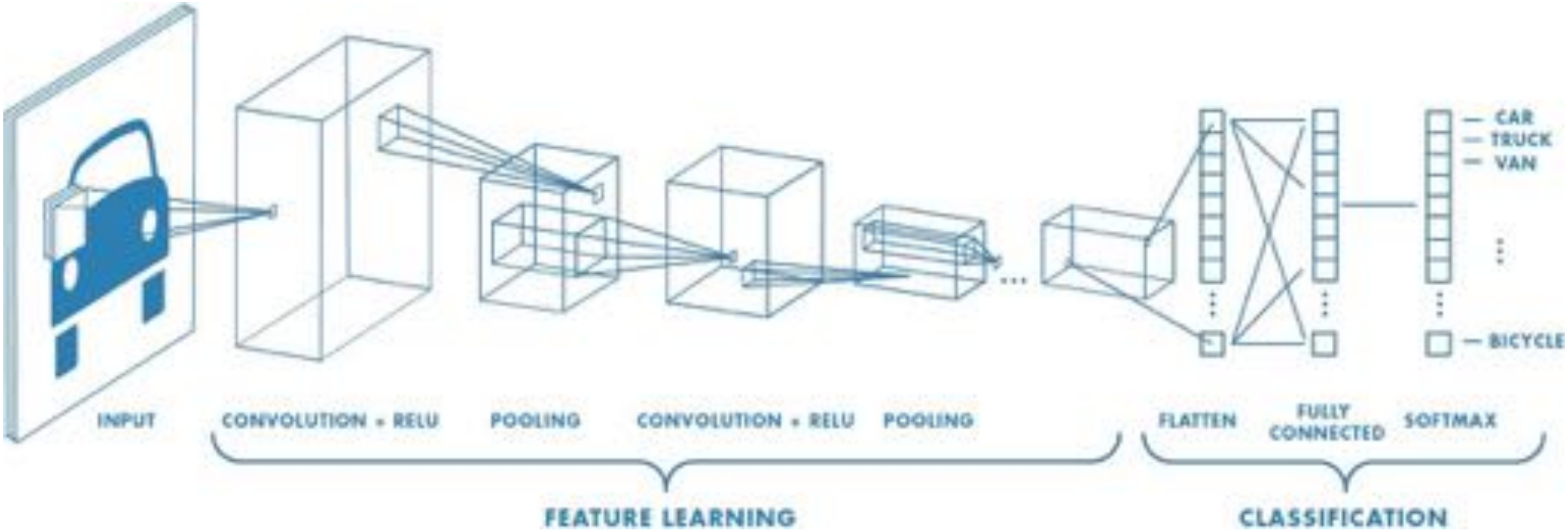
Nearly all categories are confused.

predicted actual	Front	Back	Right	Left	Front Right	Front Left
Front	4	7	0	1	0	0
Back	2	4	0	0	0	0
Right	3	2	2	5	0	0
Left	0	6	2	3	0	3
Front Right	2	6	0	0	1	1
Front Left	1	5	0	1	0	5

Common problems during training

Symptom	Possible root cause	Next step
Low accuracy on training set, low accuracy on test set	Not enough training	Increase the number of training rounds
Persistent low accuracy on training set, low accuracy on test set	**The network topology is too simple to learn this concept, or we have bad data**	**Change the topology by adding more layers, or clean up the data**
High accuracy on training set, low accuracy on test set	'Overfitting': the network has learning to recognize the training set	Simplify the topology, or add more data.
Training time too long, training accuracy does not converge	The network topology is too complex to learn this concept	Simplify the topology
High accuracy on training set, high accuracy on test set	It works!	Stop and save the trained network
Low accuracy on training set, high accuracy on test set	This is not possible.	Manually inspect your code and data

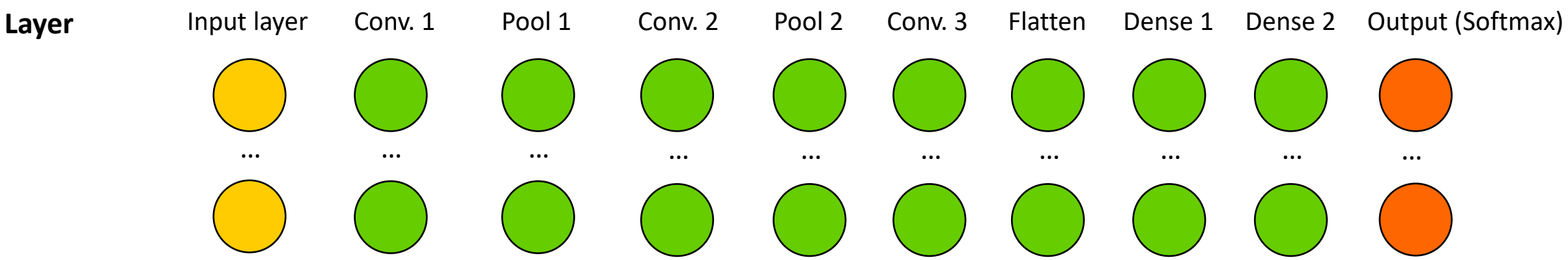
Model 2: Convolutional neural network



We chose this model based on recommendations in the literature. Apparently these first layers are good for learning similar visual concepts in different locations.

Model 2: Convolutional neural network

```
cv_model = keras.Sequential([
    keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_PIX, IMG_PIX, 3), name='block1_conv1'),
    keras.layers.MaxPooling2D((2, 2), name='block1_maxpool'),
    keras.layers.Conv2D(64, (3, 3), activation='relu', name='block2_conv1'),
    keras.layers.MaxPooling2D((2, 2), name='block2_maxpool'),
    keras.layers.Conv2D(32, (3, 3), activation='relu', name='block3_pool'),
    keras.layers.Flatten(name='flatten_6'),
    keras.layers.Dense(64, activation='relu', name='fc1'),
    keras.layers.Dense(6, name='fc2'),
])

probability_model = tf.keras.Sequential([cv_model, tf.keras.layers.Softmax(name='softmax_9')])
```



Quiz 2

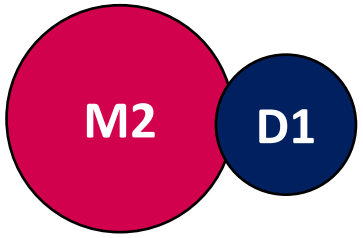
What percentage accuracy do you expect on the training data?

What percentage accuracy do you expect on the test data?

Which categories will be confused most often?

Let’s find out in the second demo!

Quiz 2 answers (complex model, small dataset)



94% accuracy on the training set (very good)

61% accuracy on the test set (decent)

Front and Back are often confused

predicted / actual	Front	Back	Right	Left	Front Right	Front Left
Front	6	1	1	0	2	2
Back	2	4	0	0	0	0
Right	1	0	9	0	1	1
Left	1	2	6	2	3	0
Front Right	0	1	0	0	9	0
Front Left	0	0	0	0	2	10

Common problems during training

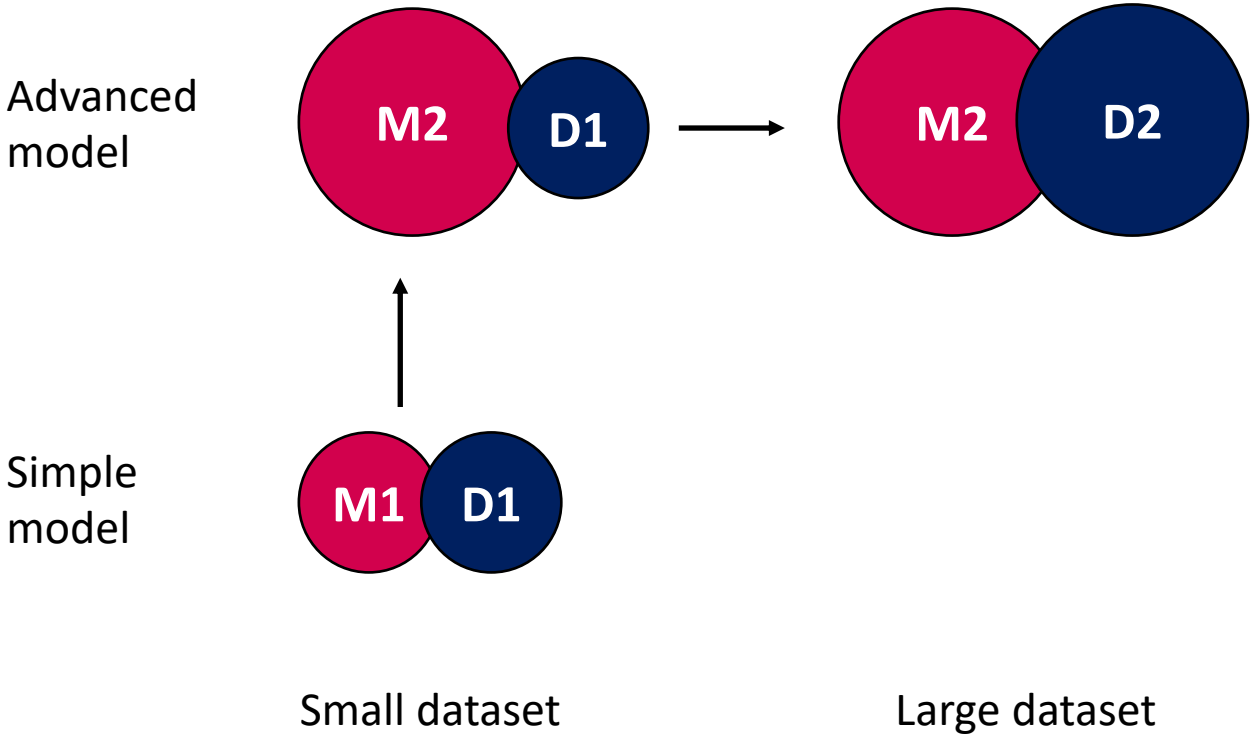
Symptom	Possible root cause	Next step
Low accuracy on training set, low accuracy on test set	Not enough training	Increase the number of training rounds
Persistent low accuracy on training set, low accuracy on test set	The network topology is too simple to learn this concept, or we have bad data	Change the topology by adding more layers, or clean up the data
High accuracy on training set, low accuracy on test set	**'Overfitting': the network has learning to recognize the training set**	**Simplify the topology, or add more data.**
Training time too long, training accuracy does not converge	The network topology is too complex to learn this concept	Simplify the topology
High accuracy on training set, high accuracy on test set	It works!	Stop and save the trained network
Low accuracy on training set, high accuracy on test set	This is not possible.	Manually inspect your code and data

Getting more data: 3 options

1. The ’honorable’ way is to get out there and make new pictures. This is however costly and time consuming
2. A poor way of getting data is to search for existing large data sets with data similar to what is needed. A major challenge is that you often have to manually add the property you want to train according to your classification
3. Technical trickery: using fabricated data that may or may not trick your algorithm. We would never do this, except when it is really convenient.

Src: andrew-draper via unsplash

Model number 2 + more data



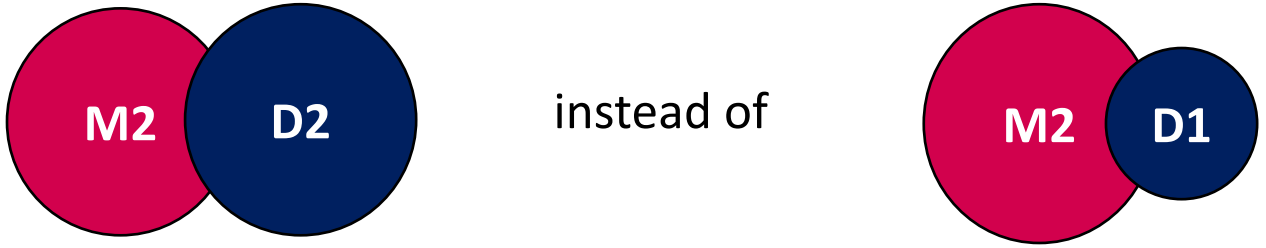
- Our neural network is quite large because we have many input neurons.
- We therefore expect that more data will do more for the performance than an even more advanced model.
- This is just a hunch: in practice, the search for better data and better model go hand in hand.

Quiz 3

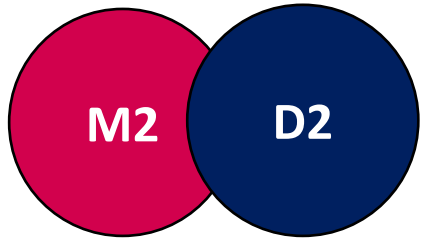
What percentage accuracy do you expect on the training data?

What percentage accuracy do you expect on the test data?

Which categories will be confused most often?



Quiz 3 answers (complex model, large data)

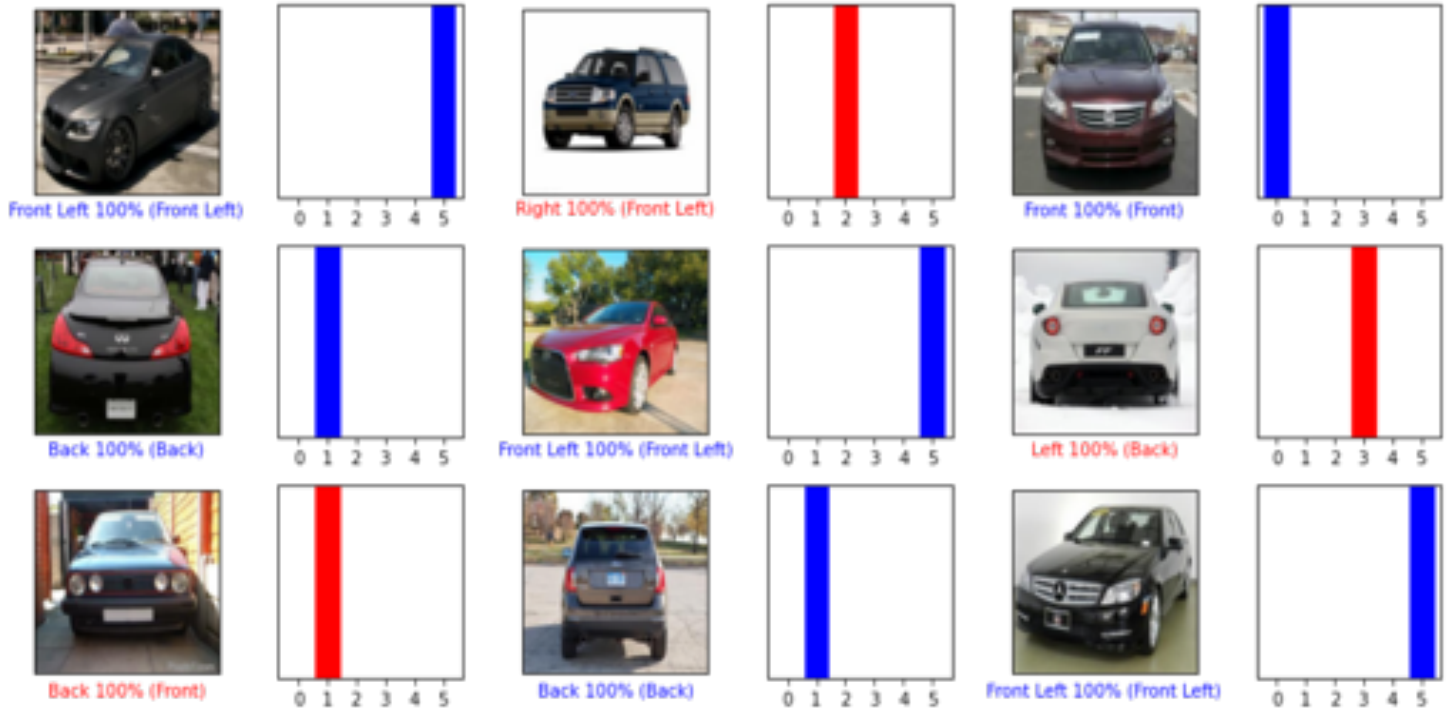


99% accuracy on the training set (excellent)

65% accuracy on the test set (poor)

Front is not predicted well, right and left often confused

predicted / actual	Front	Back	Right	Left	Front Right	Front Left
Front	6	2	0	1	3	0
Back	1	6	1	3	0	3
Right	0	0	11	10	4	1
Left	0	0	5	29	1	1
Front Right	0	0	1	1	15	0
Front Left	0	3	1	2	1	17



Common problems during training

Symptom	Possible root cause	Next step
Low accuracy on training set, low accuracy on test set	Not enough training	Increase the number of training rounds
Persistent low accuracy on training set, low accuracy on test set	The network topology is too simple to learn this concept, or we have bad data	Change the topology by adding more layers, or clean up the data
High accuracy on training set, low accuracy on test set	**'Overfitting': the network has learning to recognize the training set**	**Simplify the topology, or add more data.**
Training time too long, training accuracy does not converge	The network topology is too complex to learn this concept	Simplify the topology
High accuracy on training set, high accuracy on test set	It works!	Stop and save the trained network
Low accuracy on training set, high accuracy on test set	This is not possible.	Manually inspect your code and data

Final step: technical trickery

- Technical trickery is not a silver bullet. By expanding existing data, you could be expanding existing bias. It can lead to an illusion of accuracy, especially when you confuse training and test accuracy.

- Having said that, in some cases algorithms (and people) learn better from seeing related examples

How we expand the data

Original

08562.jpg (‘right’)

All four images are included in the training set. The test set will only have original images

The right and left sides of cars are not truly identical, but very similar and would trick 99% of people

08562f.jpg (‘left’)

Color removed. This teaches the algorithm we do not care about color

08562c.jpg (‘right’)

Small translation or rotation. This teaches the algorithm we do not care too much about positions

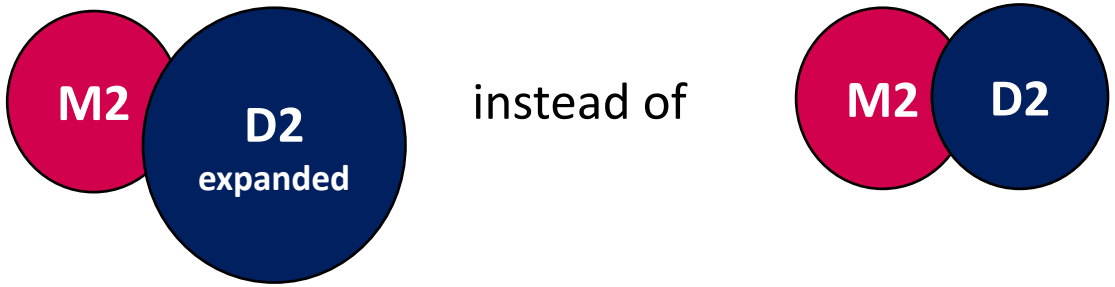
08562p.jpg (‘left’)

Quiz 4

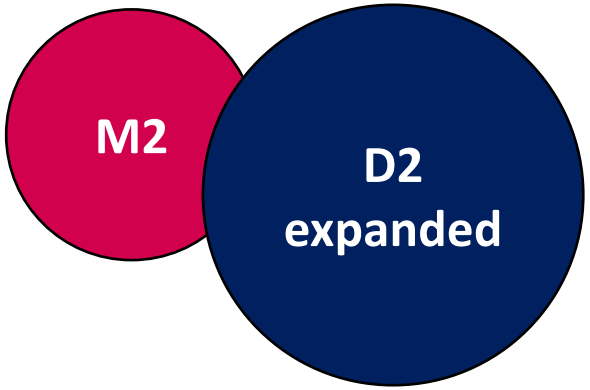
What percentage accuracy do you expect on the training data?

What percentage accuracy do you expect on the test data?

Which categories will be confused most often?



Results

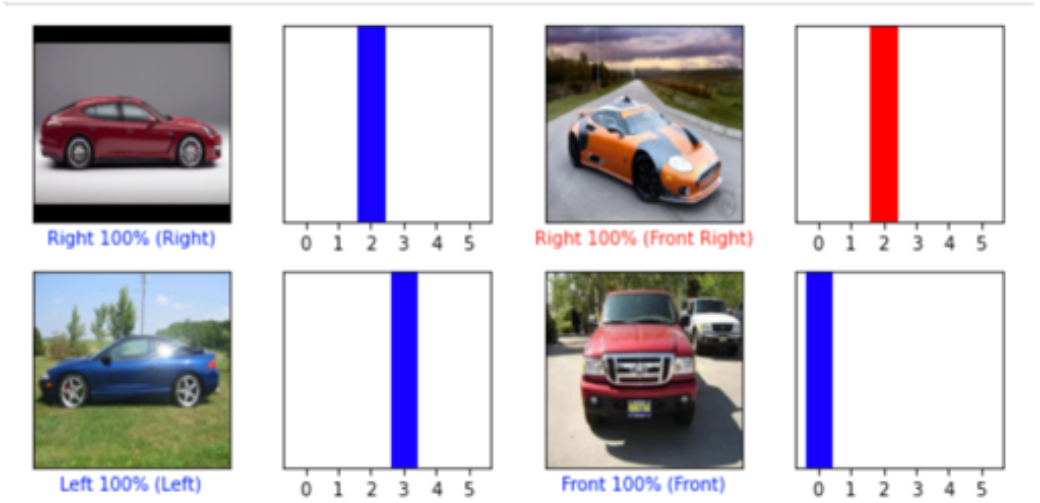


65% accuracy on the training set

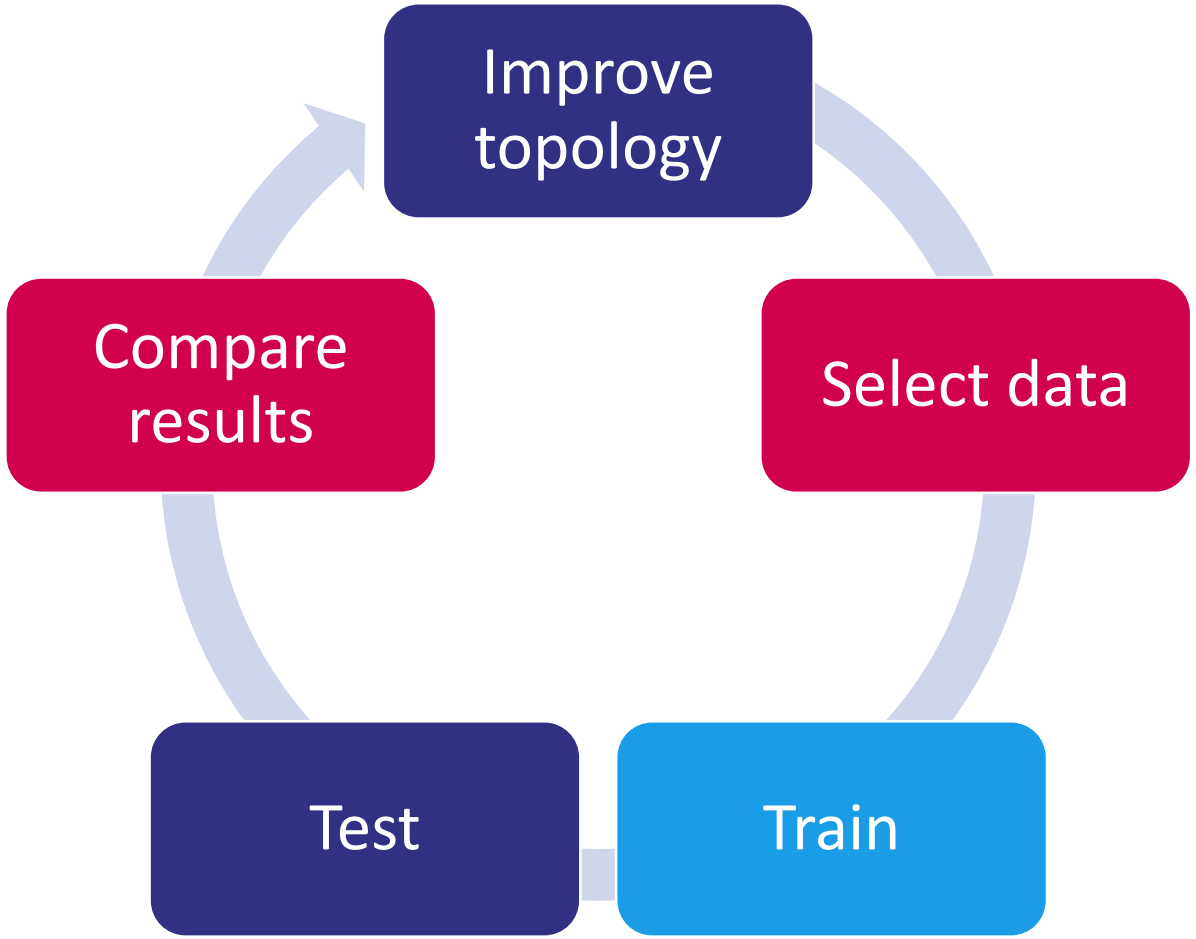
71% accuracy on the test set

Left and right most often confused

predicted / actual	Front	Back	Right	Left	Front Right	Front Left
Front	3	0	1	1	0	0
Back	1	19	0	3	5	2
Right	0	0	14	9	1	0
Left	0	0	7	32	0	0
Front Right	0	0	1	1	11	1
Front Left	2	0	0	1	6	13

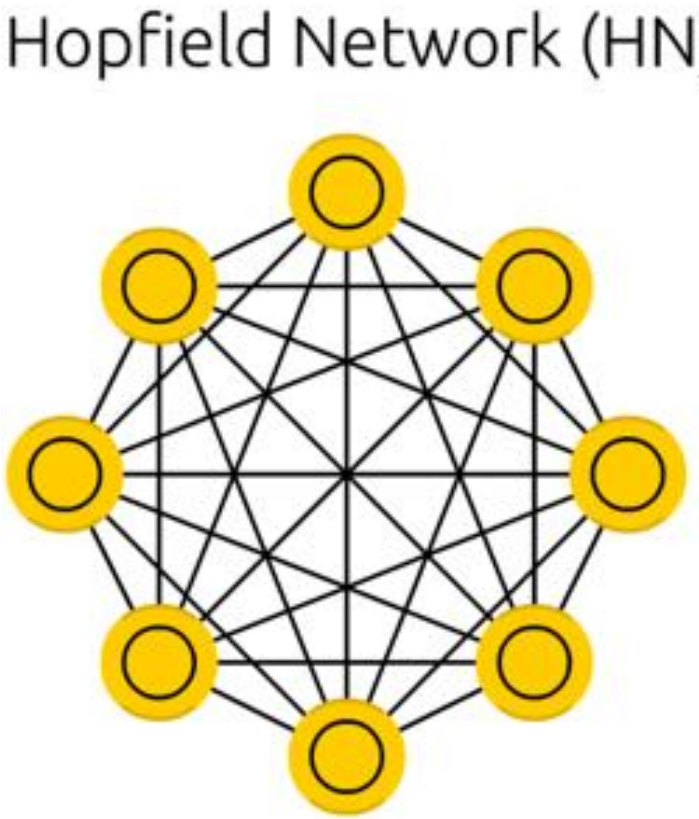


Conclusion



- Training neural networks is an iterative process, that requires human judgment and experiment
- The best algorithm will not work if you do not have enough data
- Algorithms must be independently verified. Accuracy number on training data are often way too optimistic

Hopfield network



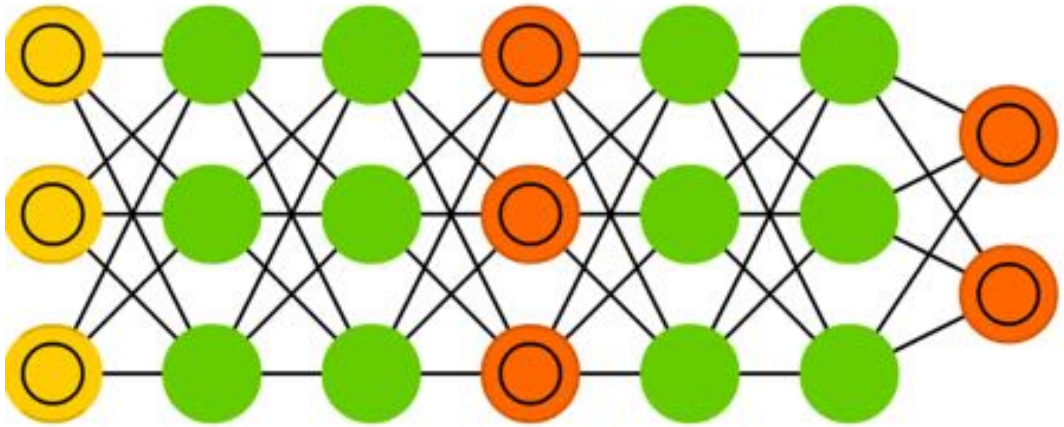
- In the Hopfield network, every neuron is an input neuron.
- Each neurons learns which of their neighbours often have a similar activation. If so, their connection is strengthened.
- You can use the network to complete partial data (e.g. in character recognition). The network can will try to reconstruct a learned state from new input.

These more complex topologies are more similar to real brain cells, but harder to use for prediction.

Source: Neural Network Zoo – Asimov Institute

Other uses

Generative Adversarial Network (GAN)



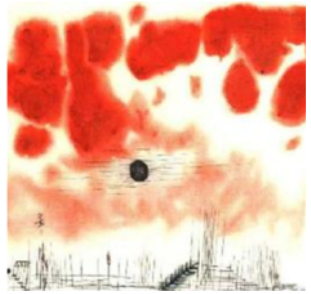
Input image

new image

Target function

GANs can be used to generate new images from existing images.

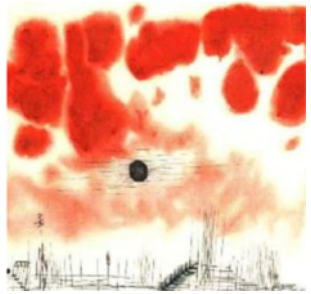
- One example is style transfer: how would an image look in a different art style?



- Another example is deep fakes: generate photos and videos that look real but are fictional

Neural network Style transfer

=



=

=

Generated by:
https://reiinakano.com/arbitrary-image-stylization-tfjs/

Agenda

- Explanation of machine learning and neural networks
- Training a machine learning algorithm (Demonstration)
- **Practical and ethical consequences of using machine learning**

Problem: Neural networks are not perfect

- Many neural networks receive less than perfect scores
- Even if the average score is fine, it can do poorly an individual cases
- The result of tested neural networks can go down

Solution: Monitoring and evaluation

- One should monitor automated decision making by checking the decisions
- Data sets should be regularly expanded with new cases

Problem: Neural networks propagate bias


TECH | AMAZON | ARTIFICIAL INTELLIGENCE

Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word "women's"

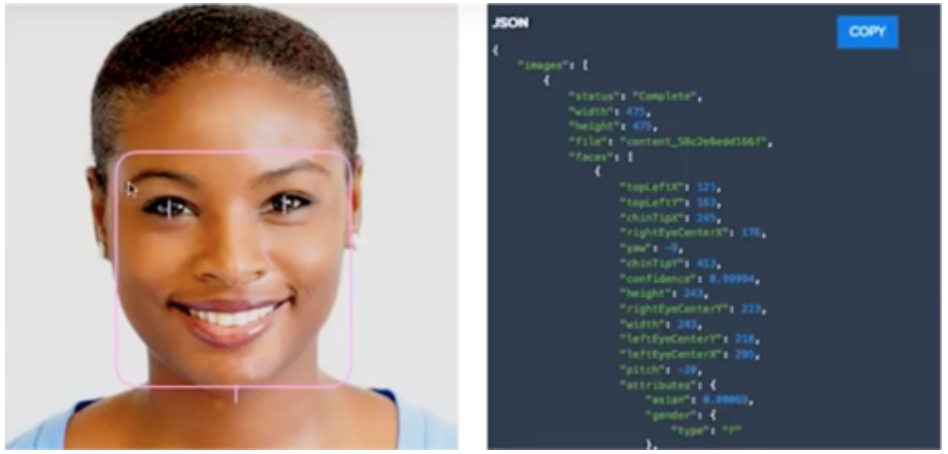
By James Vincent | Oct 10, 2018, 7:09am EDT

- Many data sets have implicit bias from human decision making
- Neural networks will learn any mistakes in the data set

Solution: Accurate data annotation

- Make sure that input data is correctly annotated
- Hire experts to validate data sets, especially if you cannot validate the annotation yourself

Problem: Algorithmic bias



- Most data scientists are rich-country, young, abled, white, male people with interest in computer science
- Algorithms and datasets are over-tested on people with the same characteristics, often in good circumstances (high-def camera's, perfect lighting)
- Algorithms are often used in completely different circumstances

https://ictinstitute.nl/algorithmic-bias-project-management/

Solution: responsible use of AI measures

- **Carefully phrase detailed quality criteria**.
- **Make sure training sets are diverse** and unbiased.
- **Test the resulting system on a diverse group of subjects**, both in lab and in the field.
- **Do not implement automated decision making** without human safeguarding. This is not just sound advice, this is a legal requirement under EU privacy law
- **Provide transparency on algorithms and test data**. People have a right to know how decisions are made
- **Ask consent for use of AI** and provide a mechanism for questions and complaints.

Problem: repeatability

- The names ‘machine learning’ and ‘data science’ might lead one to believe that there is a set procedure that guarantees a correct result. This is not the case
- A lot of AI and machine learning applications are based on trial and error using insufficient and imperfect data.

Solution?

- Educate decision makers on AI. Involve them in the training and testing process.
- Ask questions when AI is mentioned, such as:
  - How was data obtained?
  - How was data annotated?
  - How often is the algorithm updated, recalibrated or retested?
  - Is there a human in the loop? What percentage of decisions are checked?
  - Are people affected by decision informed about the data used?

Problem: explainability

- Neural network decisions are often hard to explain. It is not clear what rules or what inputs are used to come to certain decisions.
- Neural network decisions may seem random, and could have been different in case of retraining

Solution?

- Not using neural networks?
- Publishing all data and the training algorithm?
- Re-assessment by human expert?

Some research is being done to add explainability rules to neural networks.

Conclusion

- Neural networks can be trained for many different tasks. You can do this yourself using python and a lot of patience.
- Neural networks are only as good as their input data and training process. The results are often ‘interesting’ but they do require supervision
- Algorithms do not just ‘steal jobs’ but also create many new jobs, such as testing, supervision and annotation.