Artificial Intelligence: Deep Learning

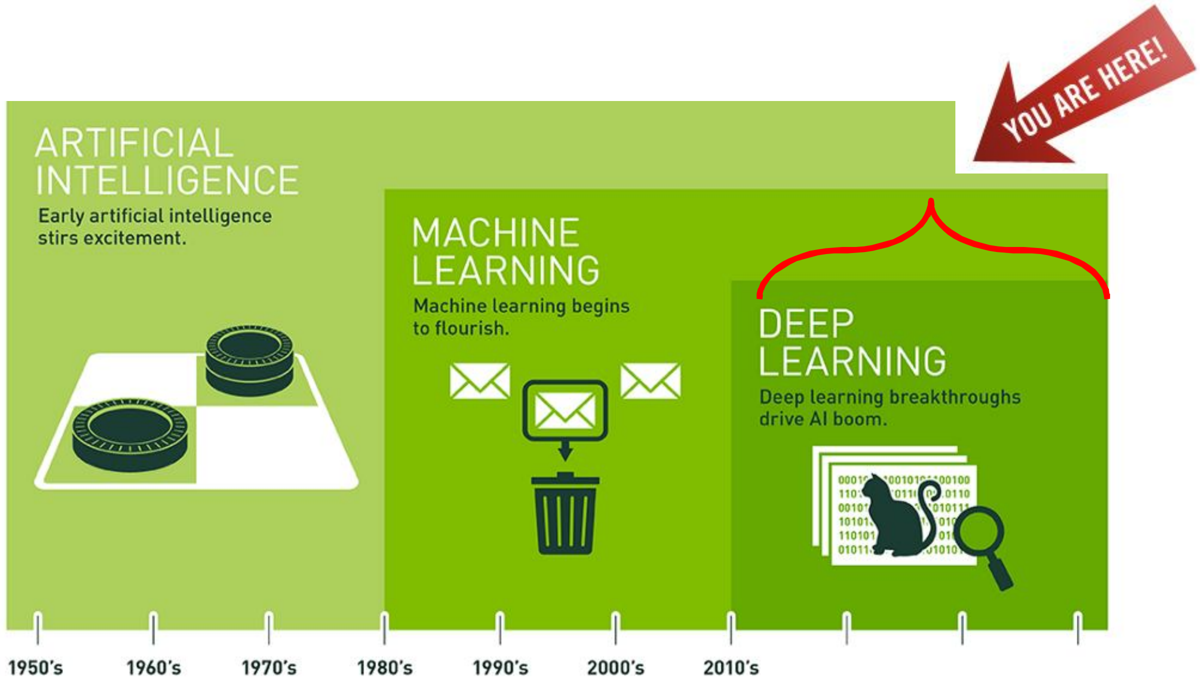
(too recent to be in the Russell & Norvig book)
many slides from: Y. Bengio, A. Ng and Y. LeCun

Today

1. Motivation 

2. Feature Learning
3. CNNs for Image Processing
4. Conclusion

History of AI

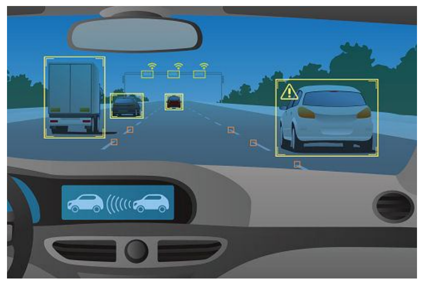


https://www.linkedin.com/pulse/goedels-incompleteness-theorem-emergence-ai-eberhard-schoeneburg/

Major Breakthroughs

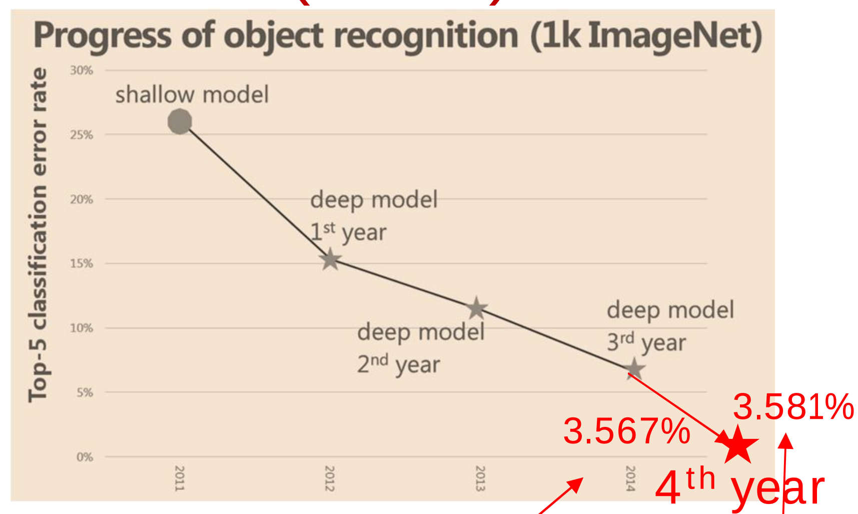
- Speech Recognition & Machine Translation (2010+)
- Image Recognition & Computer Vision (2012+)

Object recognition



Self driving cars

Google Photos



Microsoft Research Google

- Natural Language Processing (2014+)
- ...

Figure from Yoshua Bengio, 2015

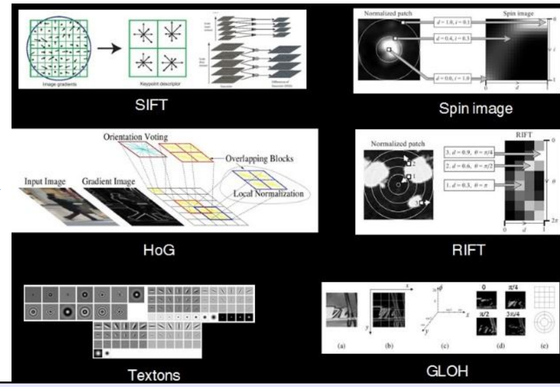
Today

1. Motivation
2. Feature Learning 

3. CNNs for Image Processing
4. Conclusion

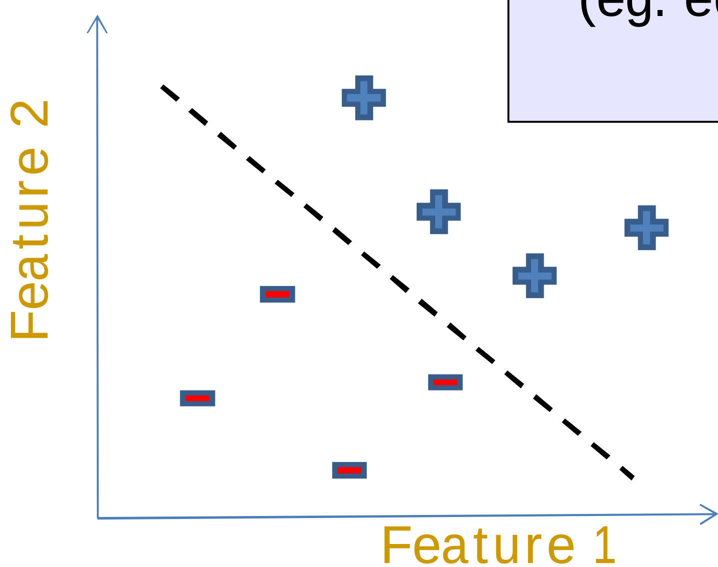
Classic ML

Input



Manual Extraction of Features
(eg. edge detection, colors, texture,...)

Learning algorithm



+ Motorbikes
– "Non"-Motorbikes

Classic ML, requires labeled data and hand-crafted features

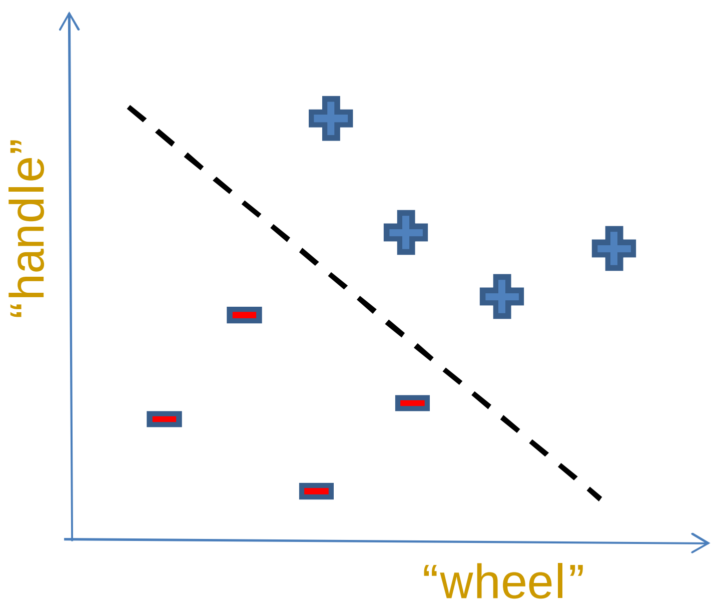
1. Needs expert knowledge
2. Time-consuming and expensive
3. Does not generalize to other domains

Automatic Feature Learning

Input

→ **Automatic Feature Representation** → **Learning algorithm**

eg. handle, wheel, ...



- Motorbikes
- "Non"-Motorbikes

With Automatic Feature Learning:

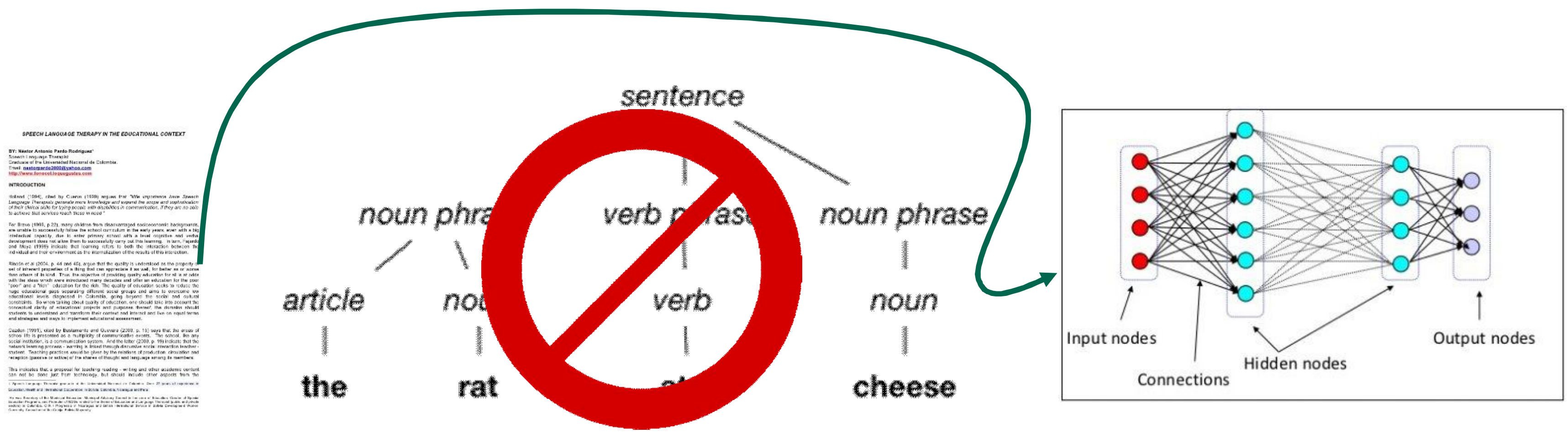
1. We feed the network the raw data (not feature-curated)

2. The features are learned by the network

3. Features learned can be re-used in similar tasks.

[redacted]

Advantages of Unsupervised Feature Learning



Automatic Feature Learning

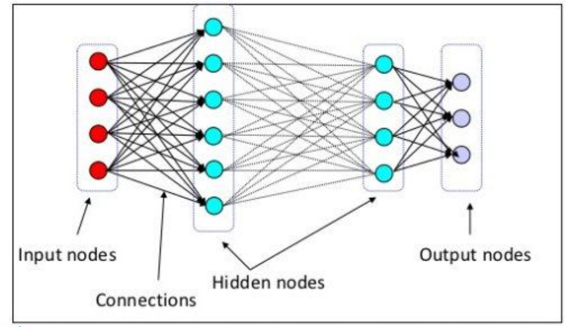
Deep Learning = Machine learning algorithms based on learning multiple levels of representation / abstraction.
– Y. Bengio

- Each layer learns more abstract features that are then combined / composed into higher-level features automatically

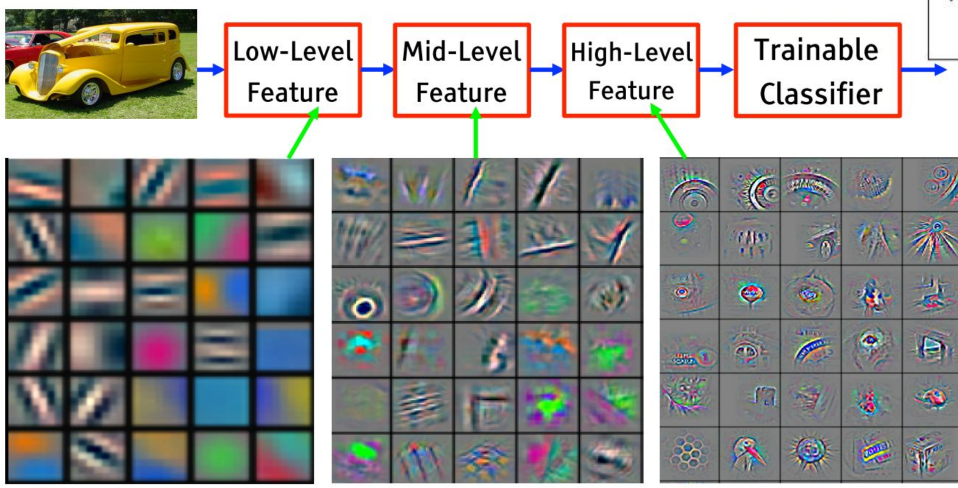
- Like the human brain …
  - has many layers of neurons which act as feature detectors
  - detecting more and more abstract features as you go up

- E.g. to classify an image of a cat:
  - Bottom Layers: Edge detectors, curves, corners straight lines
  - Middle Layers: Fur patterns, eyes, ears
  - Higher Layers: Body, head, legs
  - Top Layer: Cat or Dog

What Types of Features?



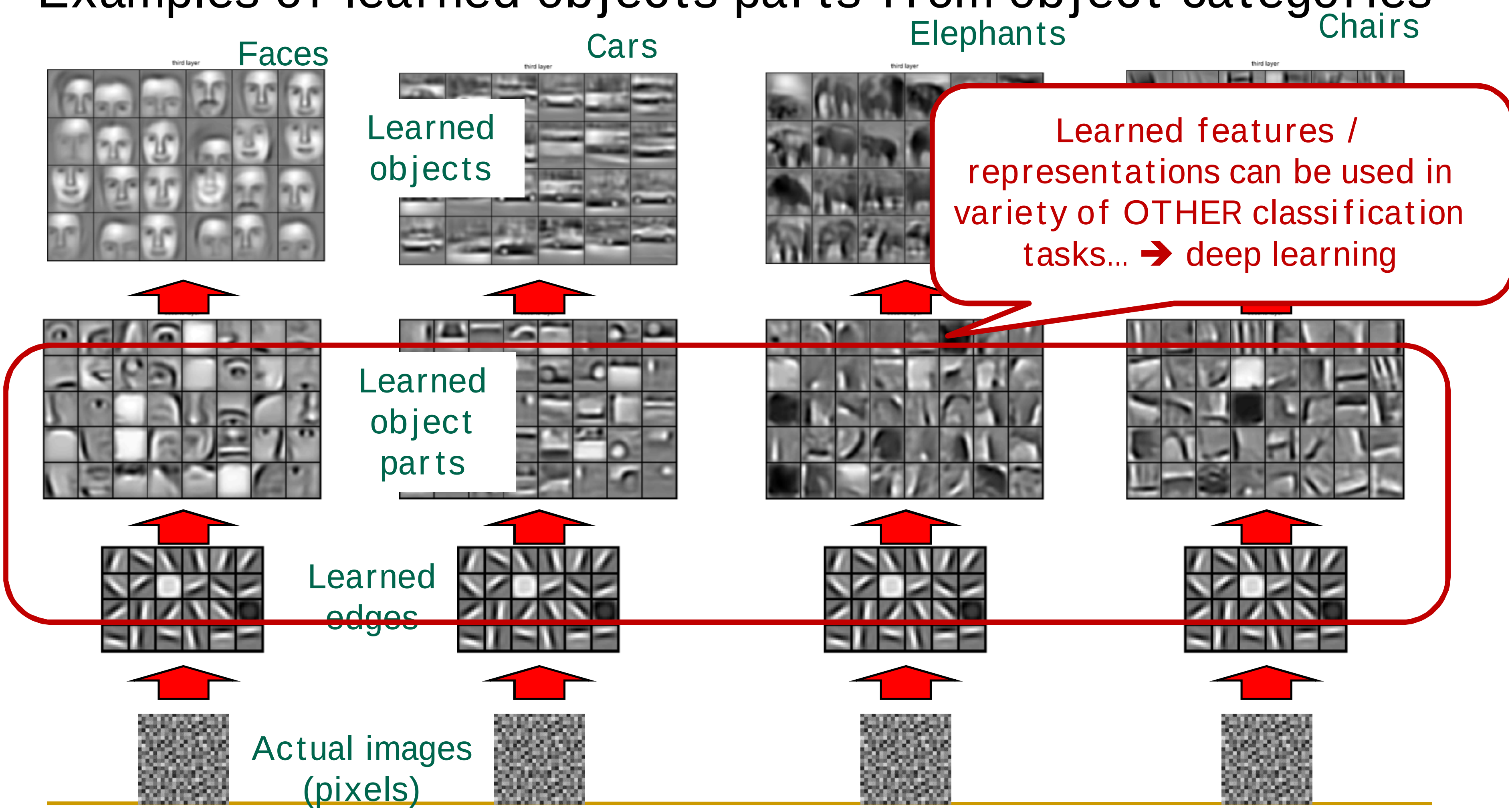
- For image recognition
  - pixel -> edge -> texton -> motif -> part -> object



- For NLP
  - character -> word -> constituents -> clause -> sentence -> discourse
- For speech:
  - sample ->spectral band -> sound -> ... phone -> phoneme -> word

Eg: Learning Image Features

Examples of learned objects parts from object categories

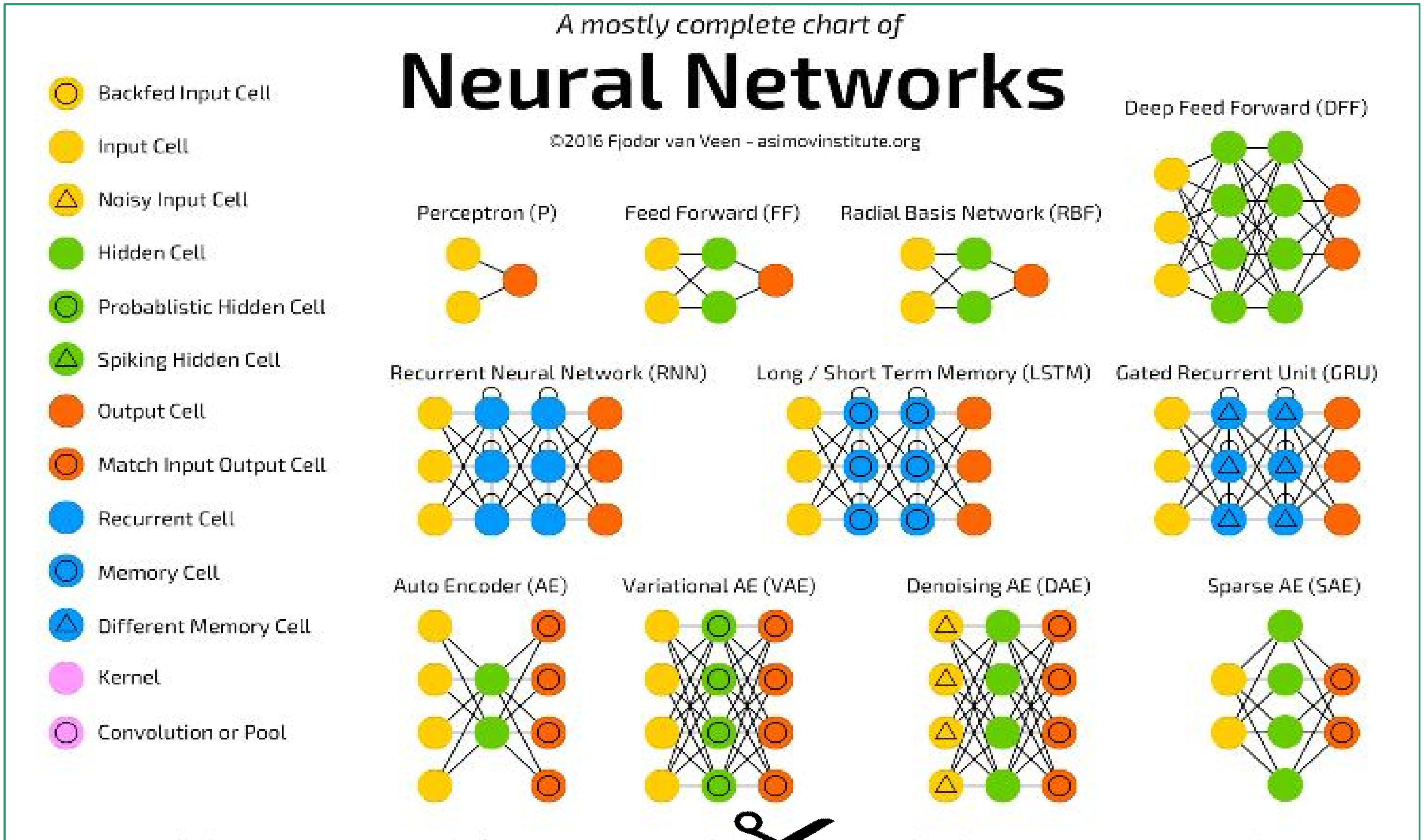


Today

1. Motivation
2. Feature Learning
3. CNNs for Image Processing 

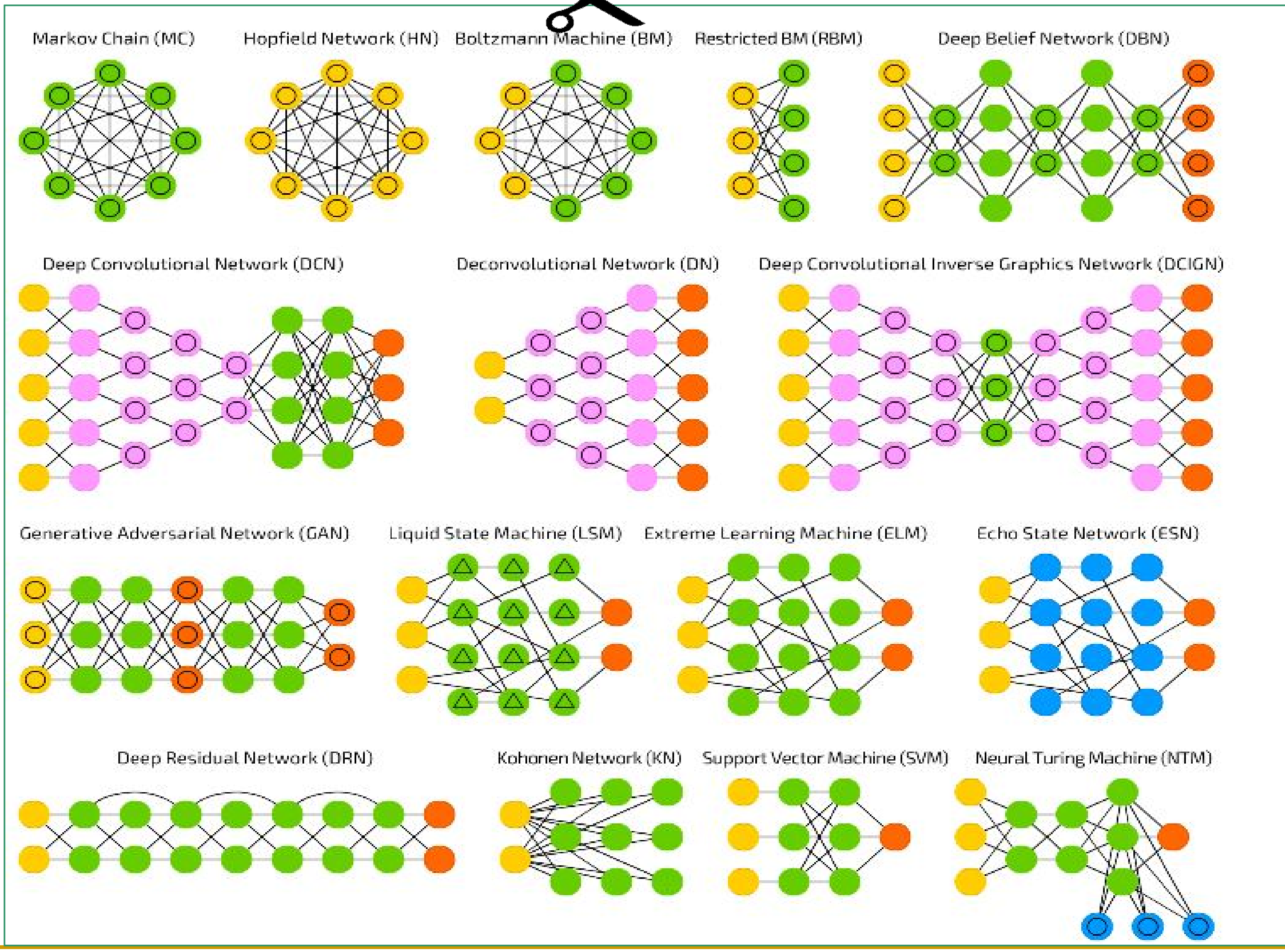
4. Conclusion

Many Types of Neural Networks



https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463

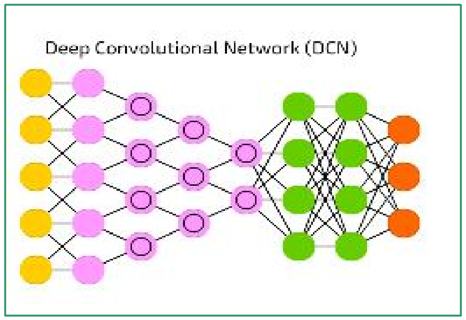
Many Types of Deep Networks (con't)

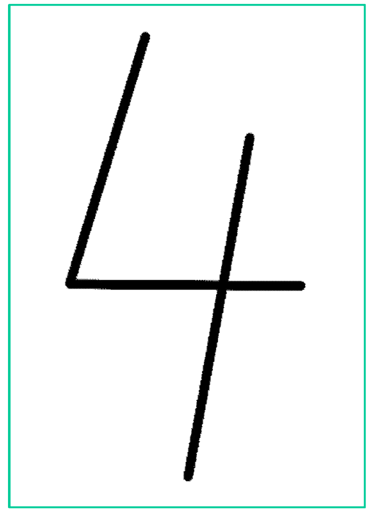


https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463

CNNs for Image Processing

CNNs = Convolutional Neural Networks





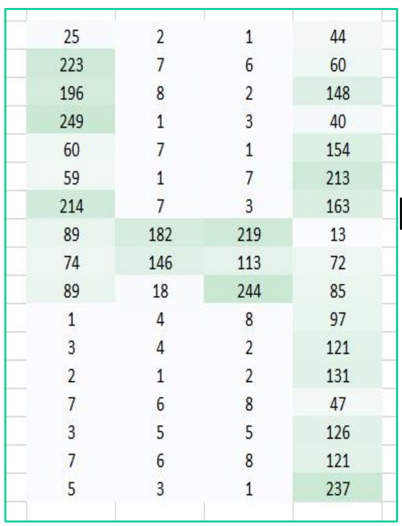
25	2	1	44
223	7	6	60
196	8	2	148
249	1	3	40
60	7	1	154
59	1	7	213
214	7	3	163
89	182	219	13
74	146	113	72
89	18	244	85
1	4	8	97
3	4	2	121
2	1	2	131
7	6	8	47
3	5	5	126
7	6	8	121
5	3	1	237

Image of a 4 in grey scale
Value = 0-> white .... 255->black

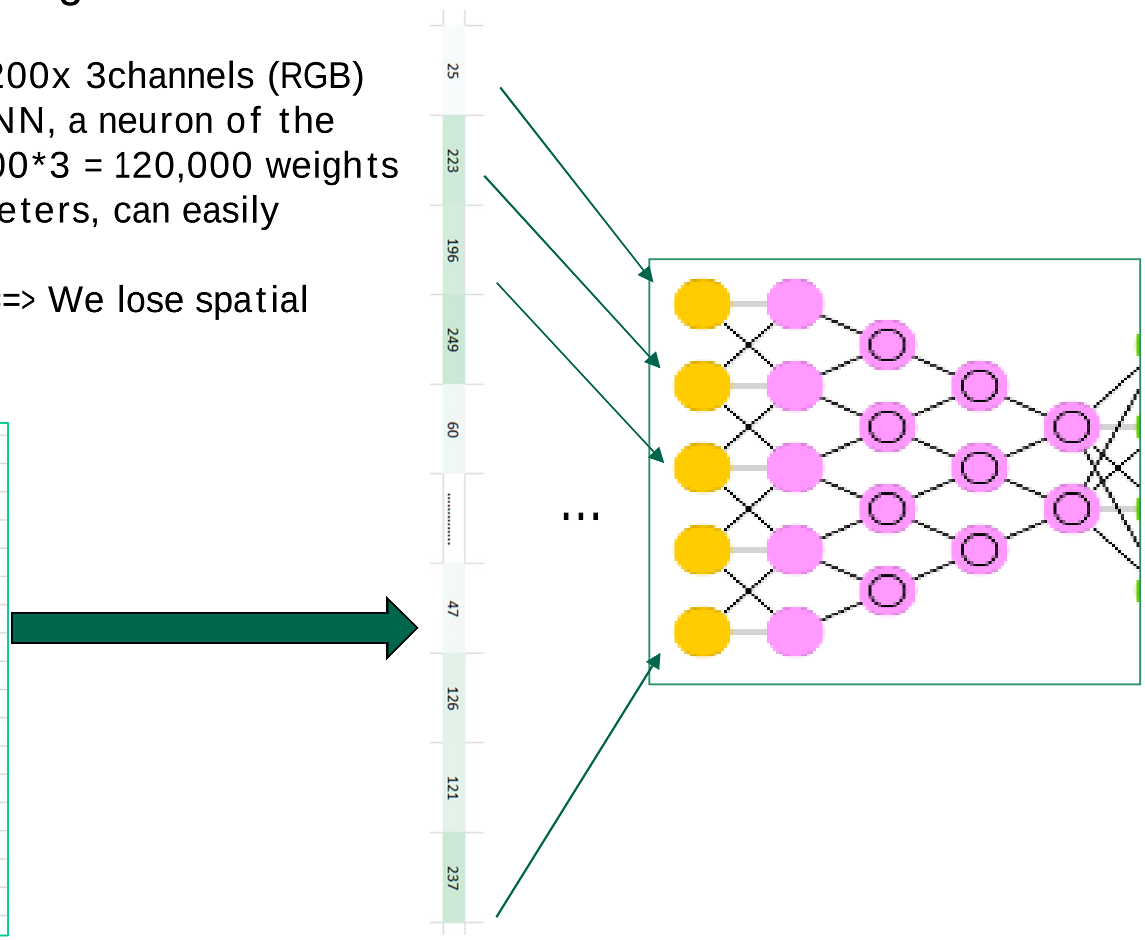
https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

CNNs for Image Processing

- Standard input of the image in the ANN:
  1. 1 pixel = 1 input.
  2. Eg. color image of 200x200x 3channels (RGB)
     --> in a fully connected ANN, a neuron of the input layer has 200*200*3 = 120,000 weights
     --> huge number of parameters, can easily overfit
  2. We linearize the image ==> We lose spatial information



25	2	1	44
223	7	6	60
196	8	2	148
249	1	3	40
60	7	1	154
59	1	7	213
214	7	3	163
89	182	219	13
74	146	113	72
89	18	244	85
1	4	8	97
3	4	2	121
2	1	2	131
7	6	8	47
3	5	5	126
7	6	8	121
5	3	1	237



https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

Convolutional Layer

- Use a filter (aka kernel) that "convolves" on the image
- Filter = small weight matrix to learn

86	4	8	184
252	3	8	40
34	7	7	163
105	2	3	69
56	3	8	175
126	1	2	178
163	8	4	142
22	222	74	180
163	158	204	253

I (original image)

1	0.3
0.5	2

W (filter)

219.2	23.9	147.2
283.9	22.9	349.5
92.6	16.1	195.4
139.6	20.4	377.7
121.9	9.9	417.5
223.8	13.6	341.4
620.4	268.2	443.6
486.1	731.2	736

C (activation map)

$$C_{11} = (I_{11} \times W_{11}) + (I_{12} \times W_{12}) + (I_{21} \times W_{21}) + (I_{22} \times W_{22})$$
$$= 86 \times 1 + 4 \times 0.3 + 252 \times 0.5 + 3 \times 2 = 219.2$$

Convolutional Layer

- Use a filter (aka kernel) that “convolves” on the image
- Filter = small weight matrix to learn

86	4	8	184
252	3	8	40
34	7	7	163
105	2	3	69
56	3	8	175
126	1	2	178
163	8	4	142
22	222	74	180
163	158	204	253

I (original image)

1	0.3
0.5	2

W (filter)

219.2	23.9	147.2
283.9	22.9	349.5
92.6	16.1	195.4
139.6	20.4	377.7
121.9	9.9	417.5
223.8	13.6	341.4
620.4	268.2	443.6
486.1	731.2	736

C (activation map)

$$C_{12} = (I_{12} \times W_{11}) + (I_{13} \times W_{12}) + (I_{22} \times W_{21}) + (I_{23} \times W_{22})$$
$$= 252 \times 1 + 3 \times 0.3 + 34 \times 0.5 + 7 \times 2 = 283.9$$

Convolutional Layer

- Use a filter (aka kernel) that “convolves” on the image
- Filter = small weight matrix to learn

86	4	8	184
252	3	8	40
34	7	7	163
105	2	3	69
56	3	8	175
126	1	2	178
163	8	4	142
22	222	74	180
163	158	204	253

I (original image)

1	0.3
0.5	2

W (filter)

219.2	23.9	147.2
283.9	22.9	349.5
92.6	16.1	195.4
139.6	20.4	377.7
121.9	9.9	417.5
223.8	13.6	341.4
620.4	268.2	443.6
486.1	731.2	736

C (activation map)

$$C_{13} = (I_{13} \times W_{11}) + (I_{14} \times W_{12}) + (I_{23} \times W_{21}) + (I_{24} \times W_{22})$$
$$= 34 \times 1 + 7 \times 0.3 + 105 \times 0.5 + 2 \times 2 = 92.6$$

Convolutional Layer

- Use a filter (aka kernel) that “convolves” on the image
- Filter = small weight matrix to learn

86	4	8	184
252	3	8	40
34	7	7	163
105	2	3	69
56	3	8	175
126	1	2	178
163	8	4	142
22	222	74	180
163	158	204	253

I (original image)

1	0.3
0.5	2

W (filter)

219.2	23.9	147.2
283.9	22.9	349.5
92.6	16.1	195.4
139.6	20.4	377.7
121.9	9.9	417.5
223.8	13.6	341.4
620.4	268.2	443.6
486.1	731.2	736

C (activation map)

$$
\begin{aligned}
C_{38} &= (I_{37} \times W_{11}) + (I_{47} \times W_{12}) + (I_{38} \times W_{21}) + (I_{37} \times W_{22}) \\
&= 74 \times 1 + 180 \times 0.3 + 204 \times 0.5 + 253 \times 2 = 736
\end{aligned}
$$

Learn the Filters

18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

$I\ (6\times 6)$

1	0	1
0	1	0
1	0	1

$W\ (3\times 3)$

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

$C\ (4\times 4)$

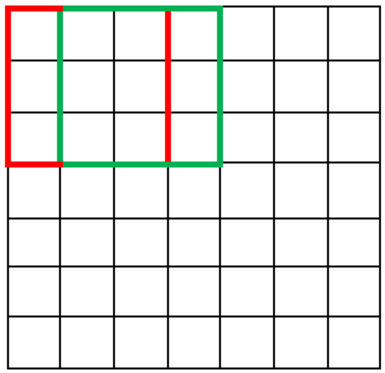
- The weight matrix (filter/kernel) behaves like a filter
- The network learns the values of the filter(s) that activate when they "see" some visual feature that is useful to identify the object (the final classification)
  - Ex. a horizontal line, a blotch of some color, a circle…

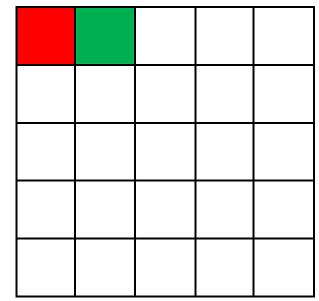
https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

Convolution Hyper-parameters

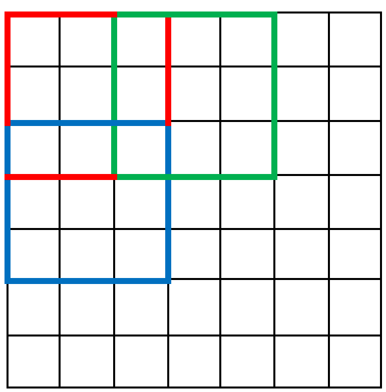
1. Stride
2. Padding

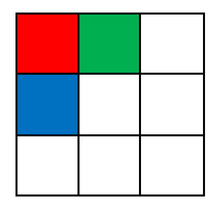
Stride





I (7×7) W (3×3) with stride =1 C (5×5)





I (7×7) W (3×3) with stride =2 C (3×3)

Padding

9	0	0	1
1	0	1	0
0	1	1	2
2	1	0	1

0	0	0
0	1	0
0	0	0

0	1
1	1

9 not picked up ;-(

filter should pick up high values surrounded by low values

0	0	0	0	0	0
0	9	0	0	1	0
0	1	0	1	0	0
0	0	1	1	2	0
0	2	1	0	1	0
0	0	0	0	0	0

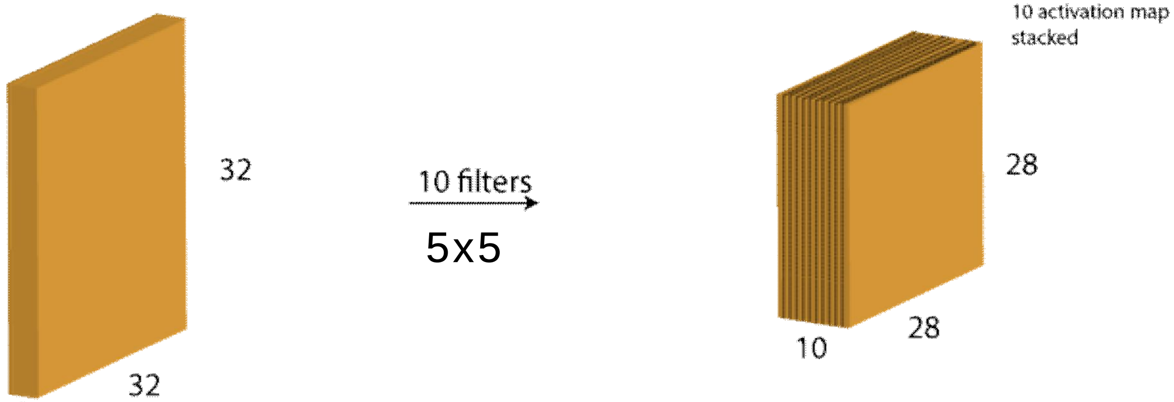
0	0	0
0	1	0
0	0	0

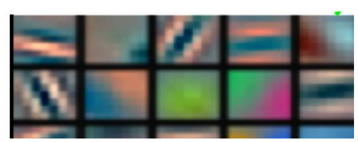
9	0	0	1
1	0	1	0
0	1	1	2
2	1	0	1

9 picked up ;-)

Learn Several Filters

- So we create 1 activation map per filter





https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

Pooling Layer

- Used to:
  - To reduce the size of the activation maps
  - So that we reduce the number of parameters of the network and hence avoid overfitting.
- Several strategies:
  - Max pooling

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

792	856
913	851

  - Average pooling

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

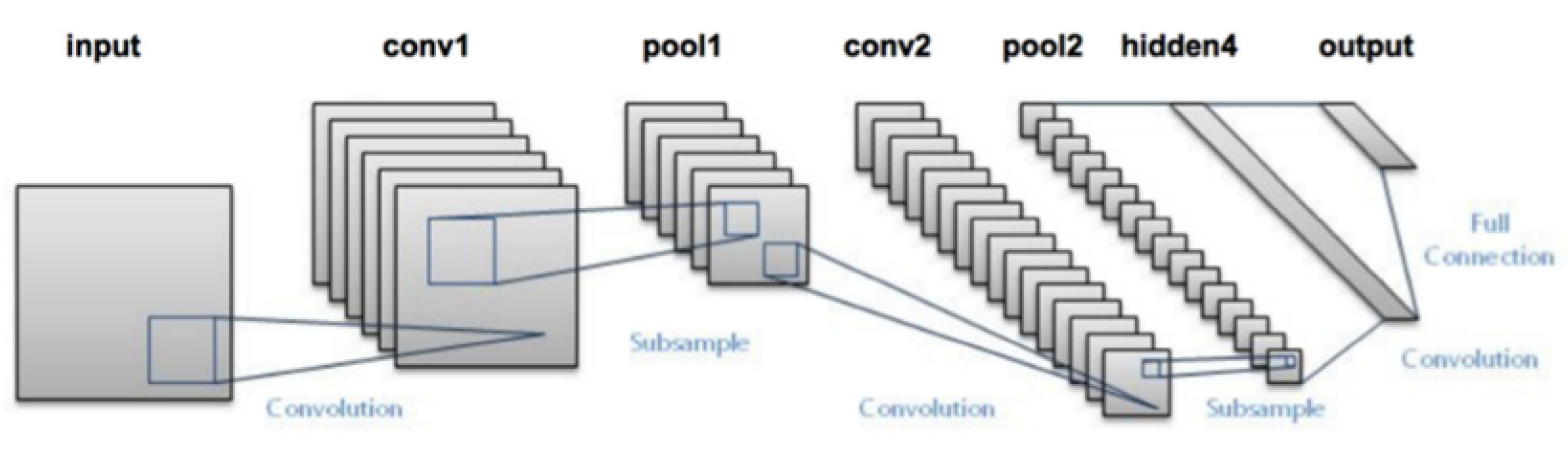
496.8	648.5
701.8	743

  - ...

https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

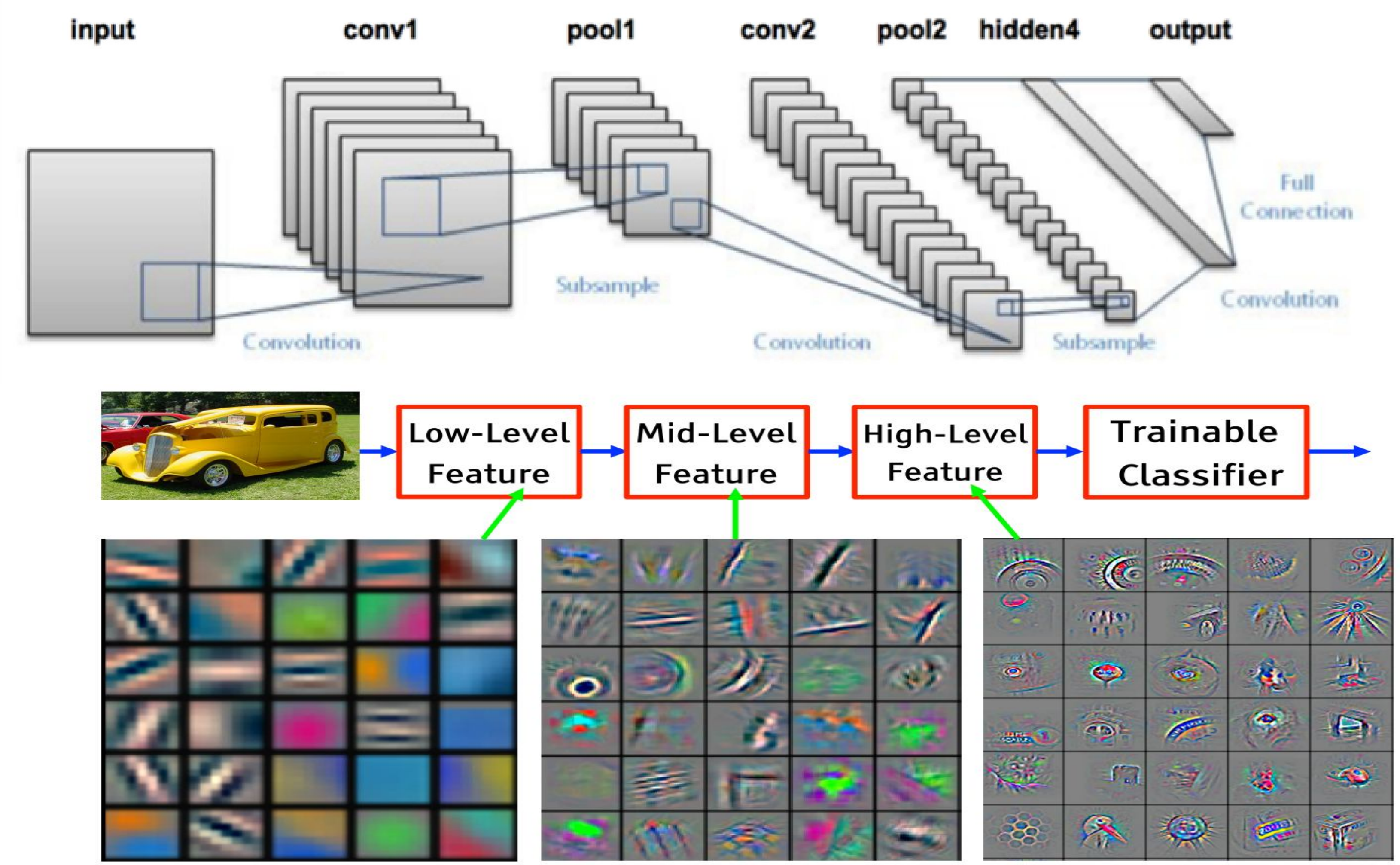
Architecture of a CNN

- Stack:
  - Convolutional Layers
  - Pooling Layers
- Finish off with:
  - A fully connected layer at the end for the final classification



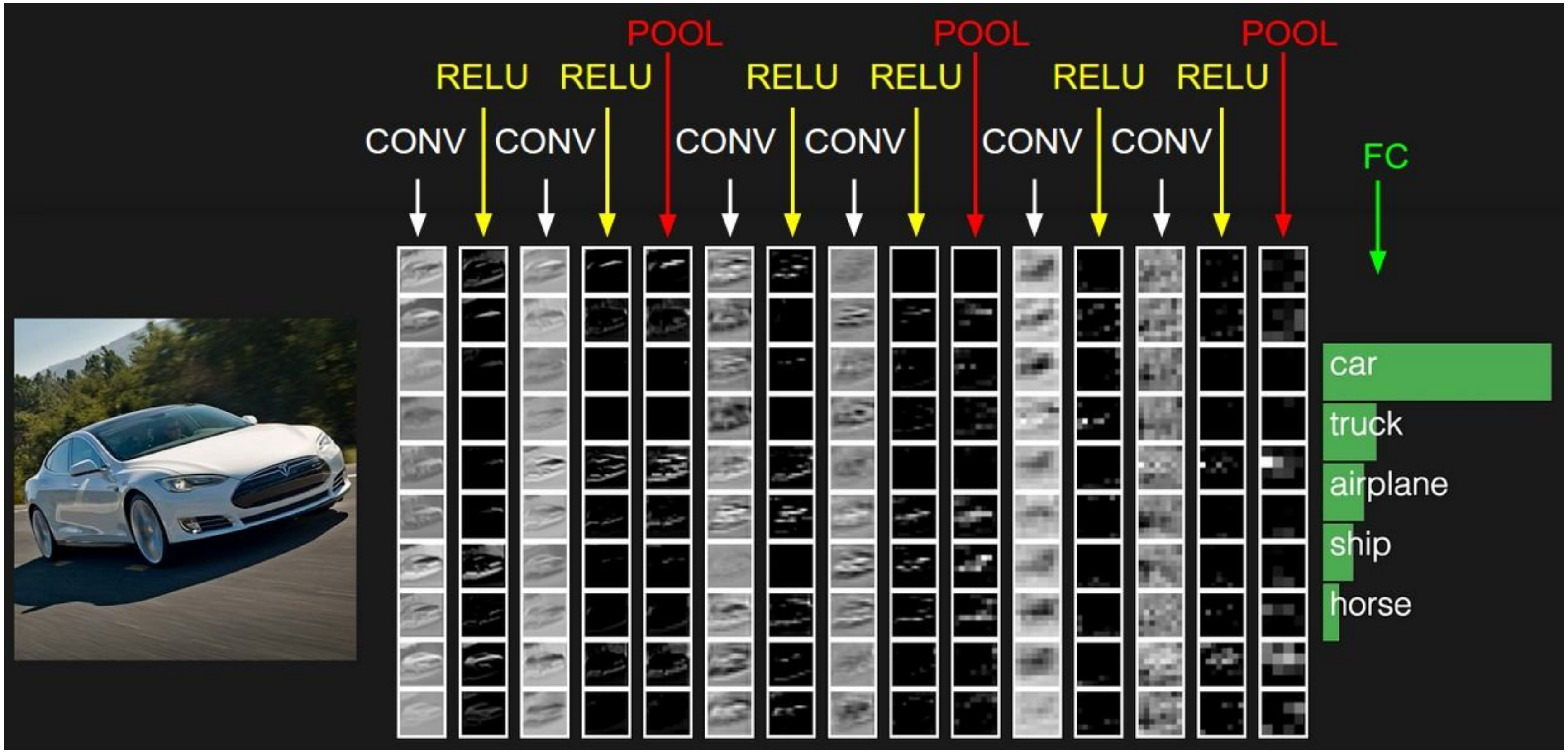
https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

Learning a Hierarchy of Features



https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/

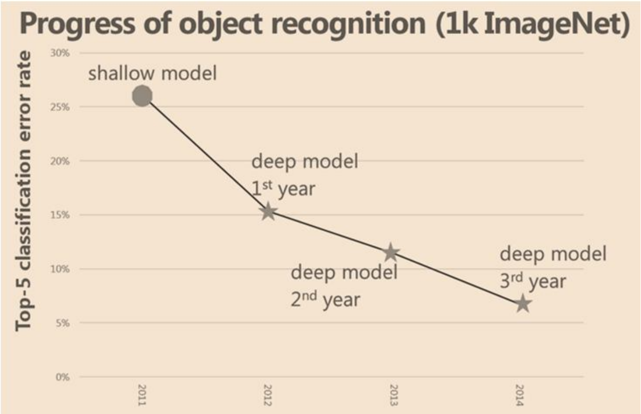
Example of a CNN



http://cs231n.github.io/convolutional-networks/

Successful CNN Networks

- LeNet
  - First successful applications of CNNs
  - Developed by Yann LeCun in the 1990's
  - used to read zip codes, digits, etc.
- AlexNet
  - First work that popularized CNNs for computer vision
  - developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton (U. of Toronto)
  - In 2012 significantly outperformed all teams at the ImageNet ILSVRC challenge



http://cs231n.github.io/convolutional-networks/

Today

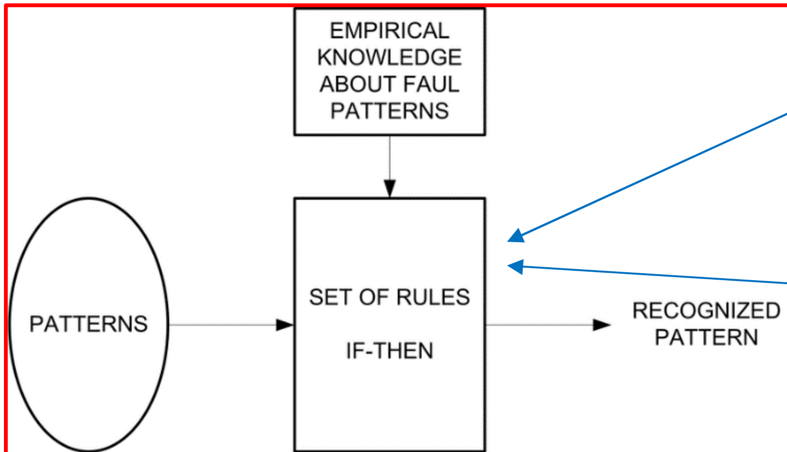
1. Motivation
2. Feature Learning
3. CNNs for Image Processing
4. Conclusion

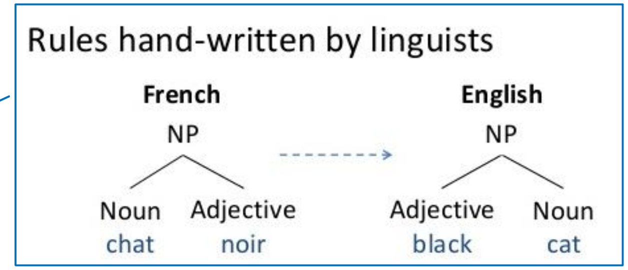
Why now?

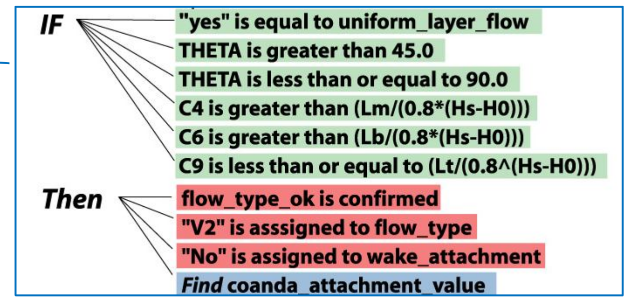
1. Basic science
   - Backpropagation did not work / overfitting…
   - *now*: developed method for training, better activation functions, better architectures….
   - Need for lots training data…
   - *now*: we have massive amounts + unsupervised pre-training
2. GPU computing
   - Neural networks are very very long to train… (days, weeks)
   - *now:* use of GPU's which are optimized for very fast matrix multiplication
3. Open Access to resources
   - *now* : Access to DL methods, code and frameworks
   - *now* : Fast turnaround from idea to implementation

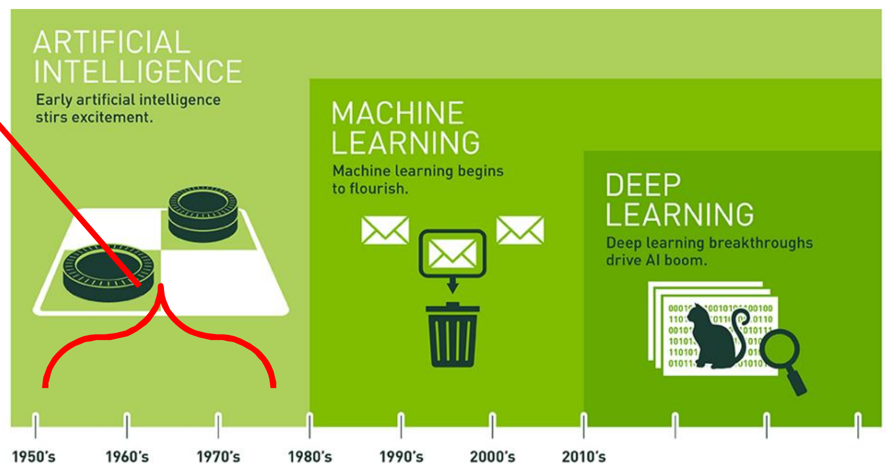
History of AI

Rules written by experts
(eg. linguistics, medical doctors,...)





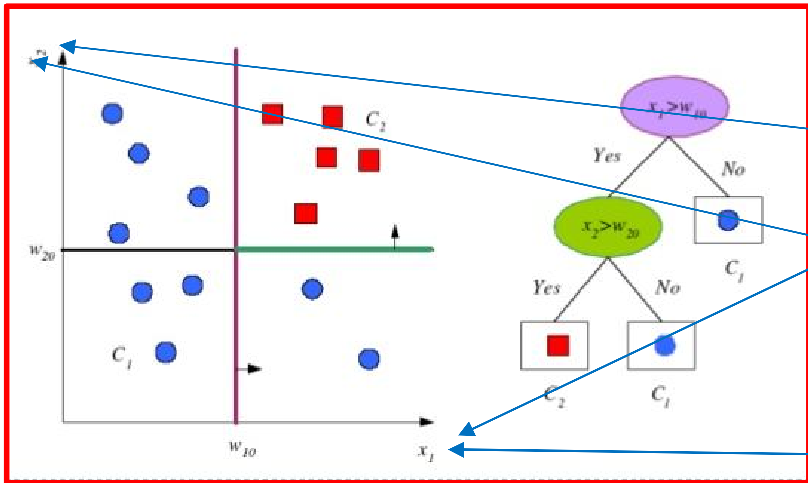


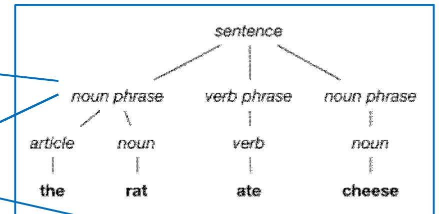


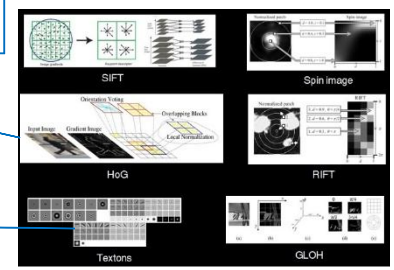
https://www.researchgate.net/profile/Dubravko_Miljkovic/publication/268239364/figure/fig30/AS:394719407427587@147111
http://www.cormix.info/images/RuleTreeExample.jpg

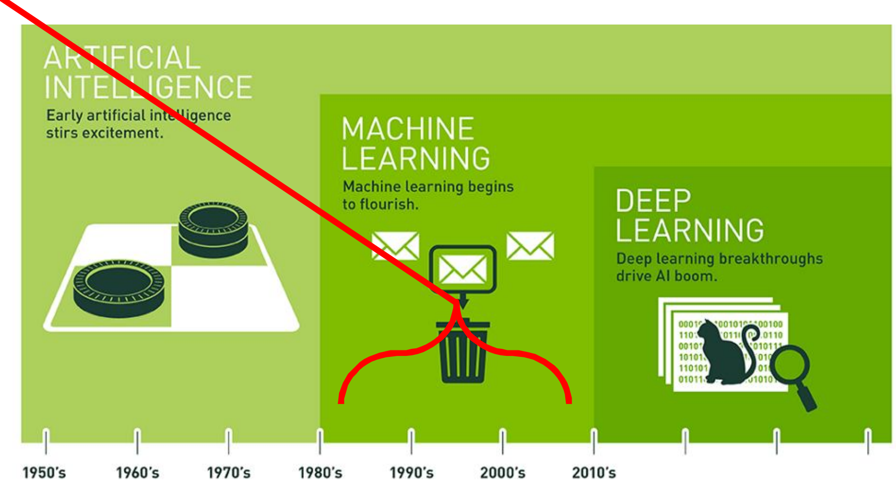
History of A I

Rules learns via the data ;-)
But: features identified by the experts
(eg. linguistics, medical doctors,...)



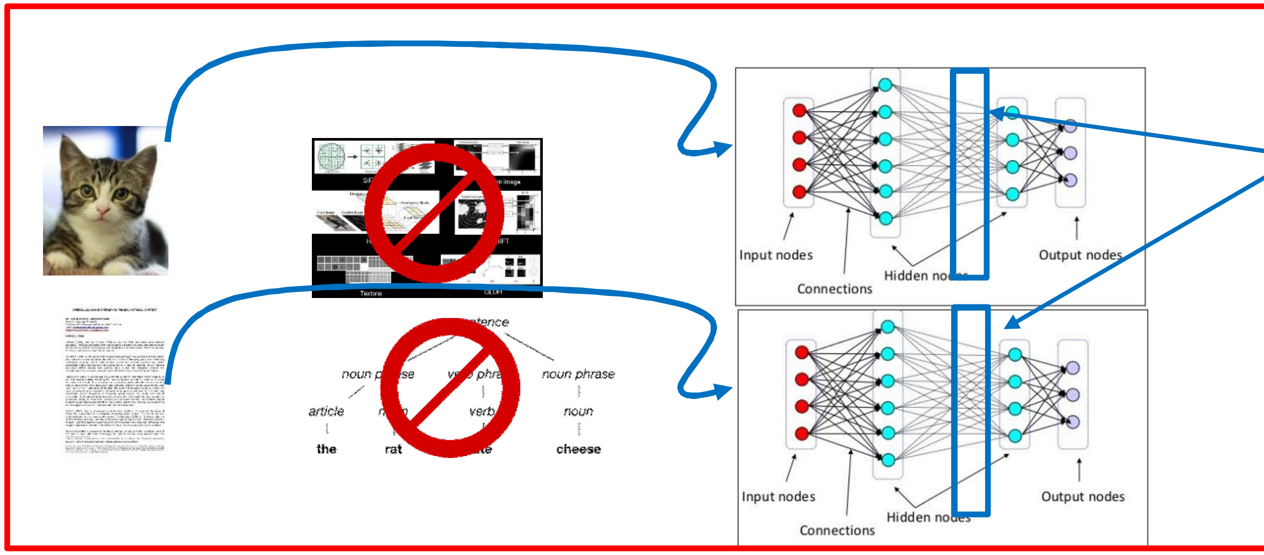




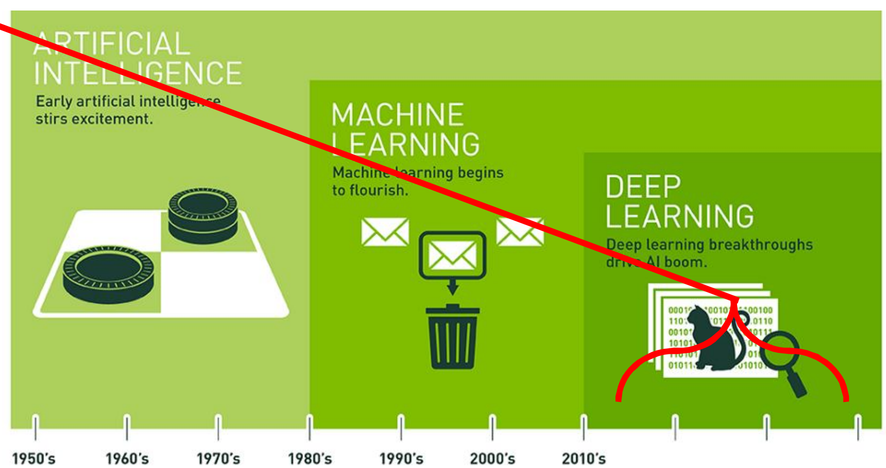


https://www.linkedin.com/pulse/goedels-incompleteness-theorem-emergence-ai-eberhard-schoeneburg/

History of A I



Rules AND features learned from the data



https://www.linkedin.com/pulse/goedels-incompleteness-theorem-emergence-ai-eberhard-schoeneburg/

Conclusion

- Deep Learning is thriving !
  - *vision*
  - *image processing*
  - *speech recognition*
  - *natural language processing*
  - *...*
- Canada is a world leader in Deep Learning
  1. Montreal: (Bengio et al.) [MILA]
  2. Toronto: (Hinton et al.) [Vector Institute]
  3. Edmonton: (Sutton et al.) [AMII]