THE UNIVERSITY OF MELBOURNE

Neural networks II: Deep learning

Semester 1, 2021

Kris Ehinger

Outline

- Introduction to deep learning
- Network architecture
- What is it doing?

What is "deep learning"?

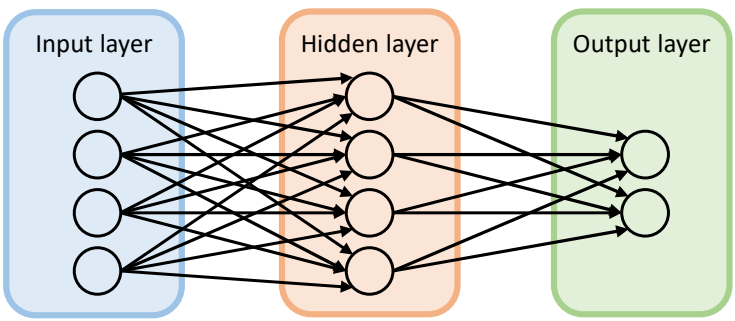
representation learning (feature learning)

allow a system to automatically discover the representation needed for feature detection or classification

- Deep learning refers to neural networks with a large number of hidden layers
  - Network **depth** = number of layers
  - Network **width** = number of neurons per layer
- Focus on **representation learning**, i.e., transformation of raw inputs into a latent intermediate representation that is more amenable to learning

What is “deep learning”?

- One hidden layer:



- 33 hidden layers:

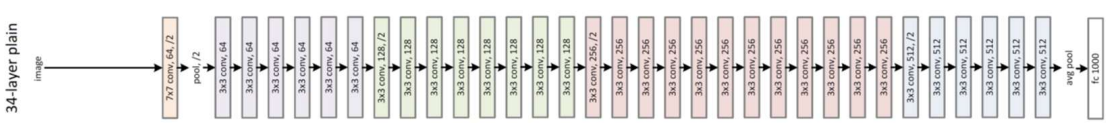
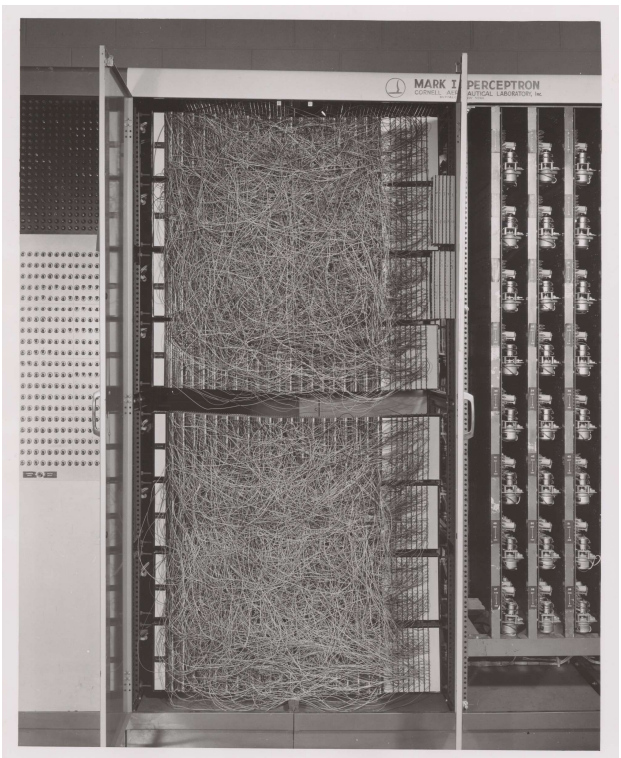


Image: Hu, Zhang, Ren, & Sun (2016)

Deep learning “revolution”



- Starting around 2010, neural networks became the state-of-the-art for many ML problems
- But perceptron and backpropagation are from the 1950s-60s
- What changed?

Cornell University News Service records, #4-3-15. Division of Rare and Manuscript Collections, Cornell University Library. https://digital.library.cornell.edu/catalog/ss:550351

What changed?

- Graphics processing units (GPUs) for fast parallel computation
  - CUDA (Nvidia, 2007) – platform for parallel processing on GPU
- Algorithm improvements:
  - Unsupervised pretraining, ReLU activation function, regularisation (e.g., dropout)
- The internet
  - Massive amounts of text, photos, etc. *with human-annotated labels*

Large-scale image datasets

IMAGENET

http://www.image-net.org/

14 million images
1000 object classes
Collected by image-searching for class names

Open images dataset

https://g.co/dataset/open-images
9 million images
19,700 classes

YouTube-8M

https://research.google.com/youtube8m/
6 million videos
3800 object classes

Architecture of a deep neural network

Convolutional neural network

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)

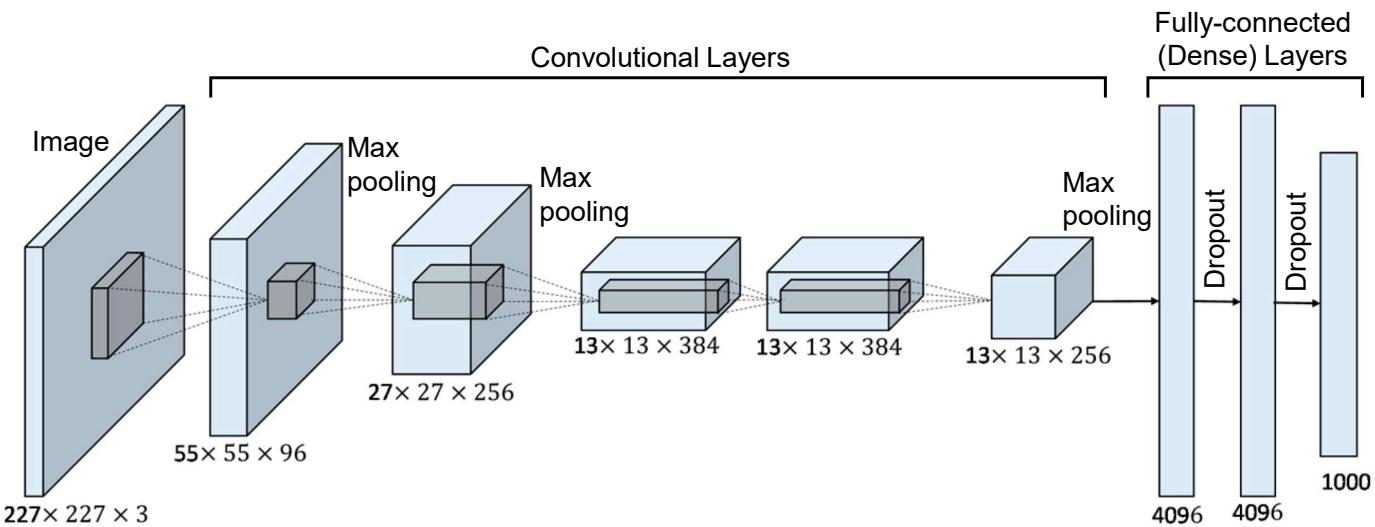


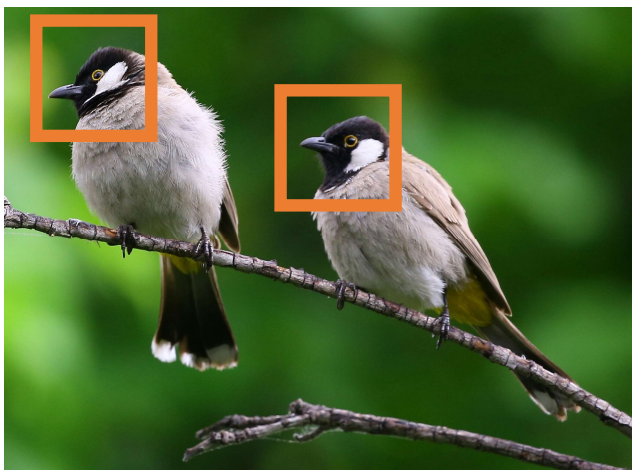
Image: Han, Zhong, Cao, & Zhang (2017)

Convolution

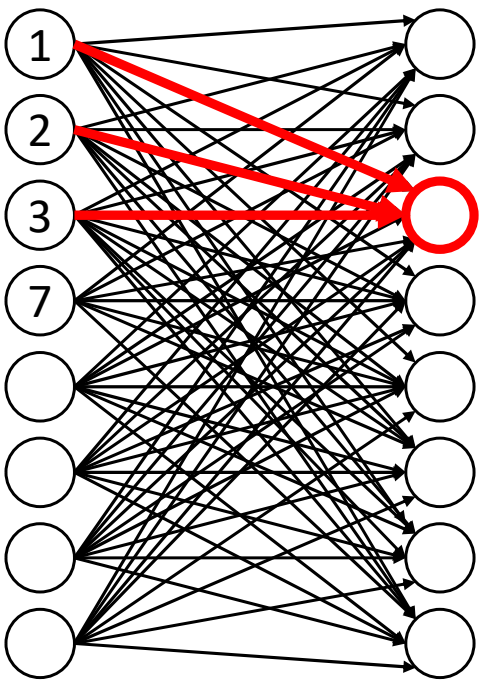
This is how i felt while watching this film. I loved it. It was hilarious.

I loved it, it was fun and moved quickly, no boring drawn out scenes.

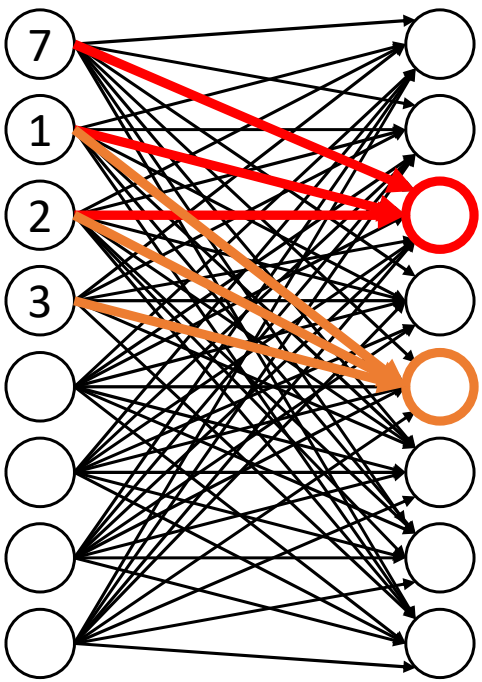
I just got it and it is a great movie!! I loved it!

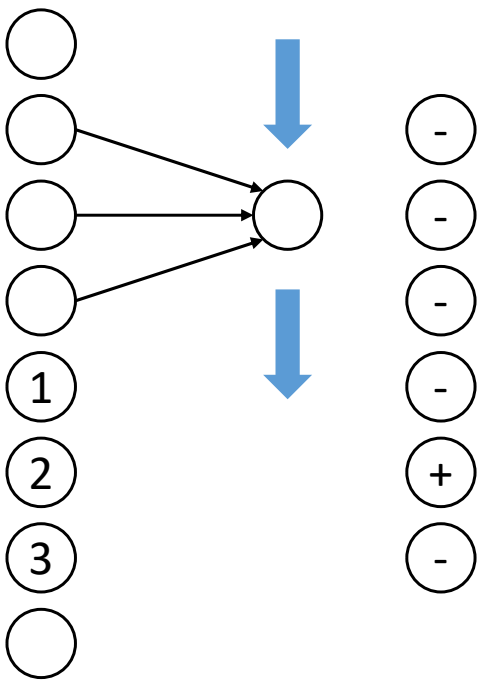


Convolution



Convolution





Convolution

- Convolutions are defined by
  - A **kernel**, which is a matrix overlaid on the input and computes an element-wise product with the input
  - A **stride** which defines how many positions in the input to advance the kernel on each iteration (stride = 1 means the kernel will operate on every input)

Convolutional neural network

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)

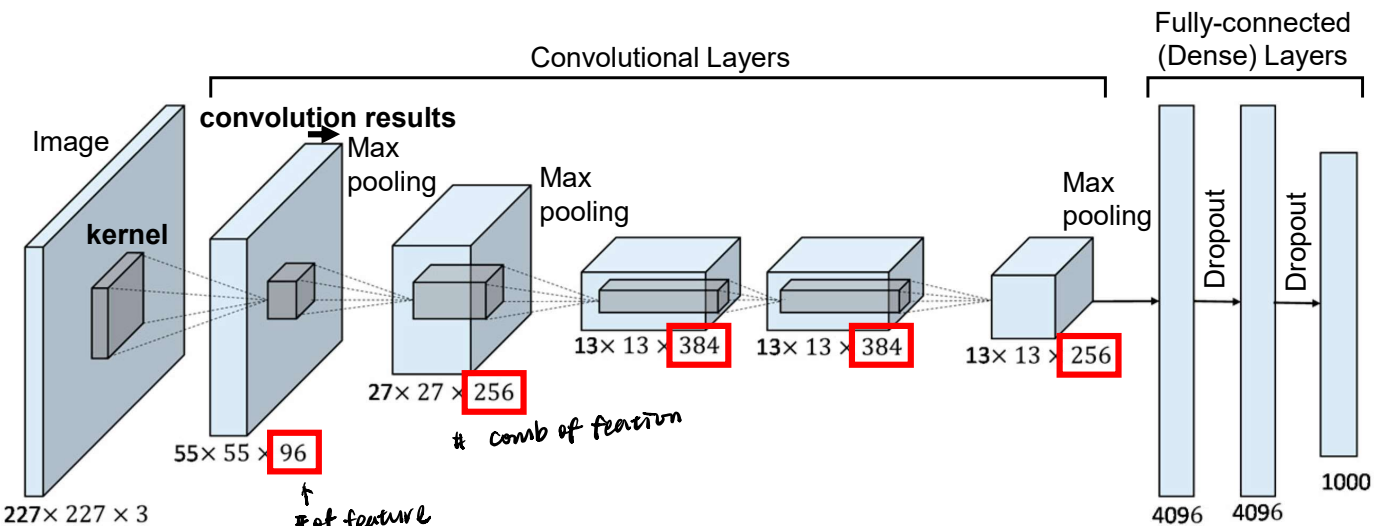
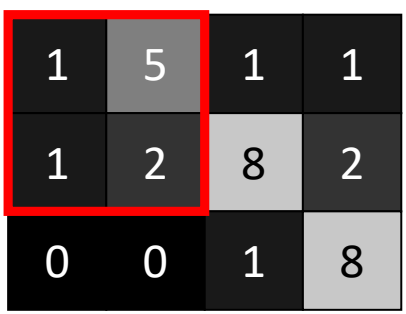


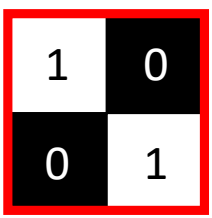
Image: Han, Zhong, Cao, & Zhang (2017)

Convolution

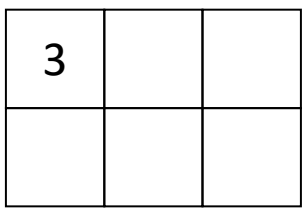
- Consider a 3x4 image and 2x2 kernel, stride = 1



Image



Kernel



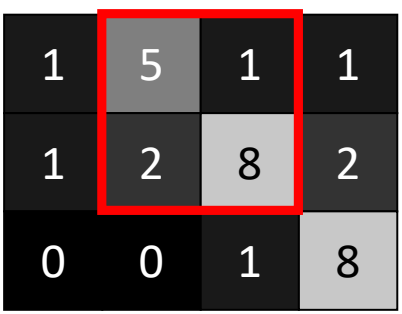
Convolution output

Convolution 1:

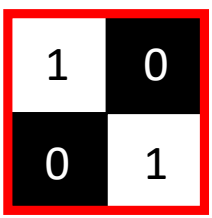
1x1	5x0
1x0	2x1

Convolution

- Consider a 3x4 image and 2x2 kernel, stride = 1



Image



Kernel

3	13	

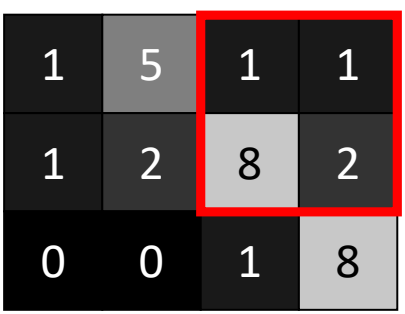
Convolution output

Convolution 2:

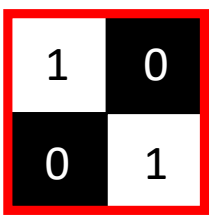
5x1	1x0
2x0	8x1

Convolution

- Consider a 3x4 image and 2x2 kernel, stride = 1



Image



Kernel

3	13	3

Convolution output

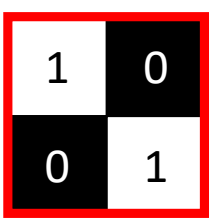
Convolution 3:

1x1	1x0
8x0	2x1

Convolution

- Consider a 3x4 image and 2x2 kernel, stride = 1

Image



Kernel

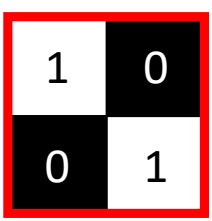
3	13	3
1		

Convolution output

Convolution

- Consider a 3x4 image and 2x2 kernel, stride = 1

Image



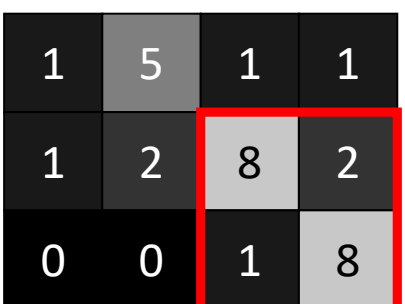
Kernel

3	13	3
1	3	

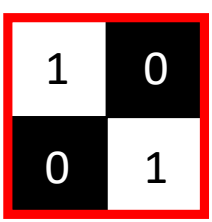
Convolution output

Convolution

- Consider a 3x4 image and 2x2 kernel, stride = 1



Image



Kernel

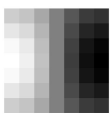
3	13	3
1	3	16

Convolution output

Convolutional layer 1

Input = image

Neuron 1 kernel:



Output:

Neuron 2 kernel:

Output:

...

...

Convolutional layer 2

Neuron 1 kernel:

?

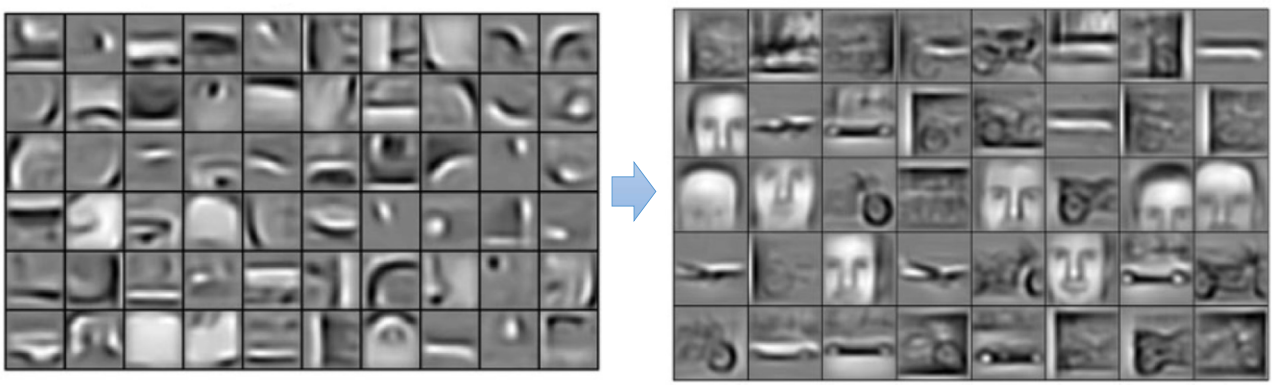
Output:

?

Input = stack of layer 1 outputs

What does each layer learn?

- Layers learn a hierarchy of image features



Images: Lee et al. (2011)

Fully-connected vs. Convolutional

Fully-connected layer

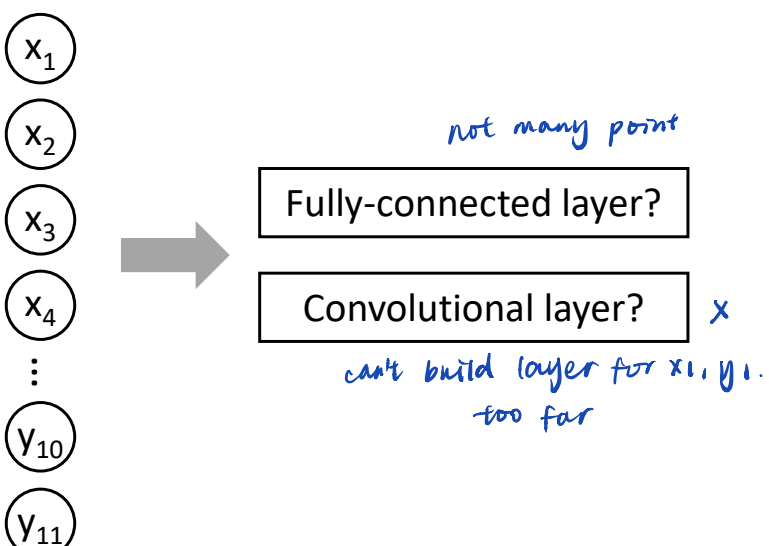
- Each neuron is connected to every neuron in the last layer
- The neuron learns some combination of the last layer's responses
- The output to the next layer is the neuron's response

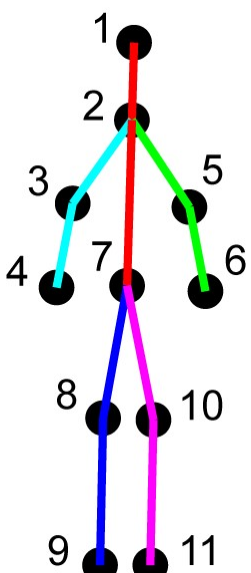
Convolutional layer

- Each neuron is connected to a small patch of all of the last layer's outputs
- The neuron learns a convolutional kernel
- The output to the next layer is the input convolved with the neuron's kernel

Example: Layer types

- What type of layer would you use for the keypoint data in the Asst. 1 pose classification task?





Convolutional layers

- Advantages of convolutional layers
  - Efficient – just one neuron learns to recognize a feature like “vertical edge” everywhere in the input
  - Preserves spatial relations (output represents “where” features are present, not just “what” features are present)
- Disadvantages of convolutional layers
  - Limited kernel size means model is limited to learning local features

Convolutional neural network

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)

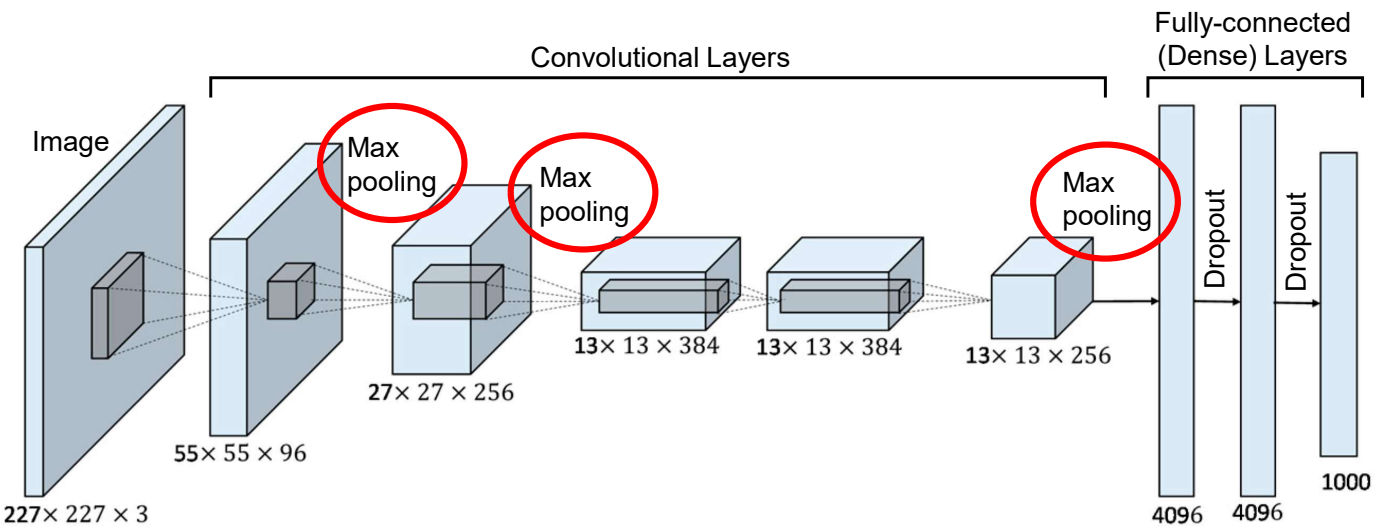
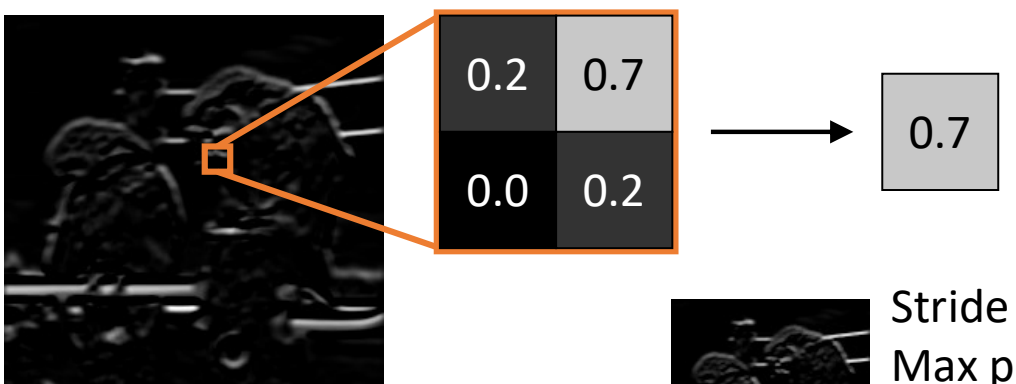


Image: Han, Zhong, Cao, & Zhang (2017)

Max pooling

- Within a small window (e.g., 2x2 pixels) in the kernel's output map, take the highest output and discard the rest



Stride = 2
Max pool output is ½ height and ½ width of input

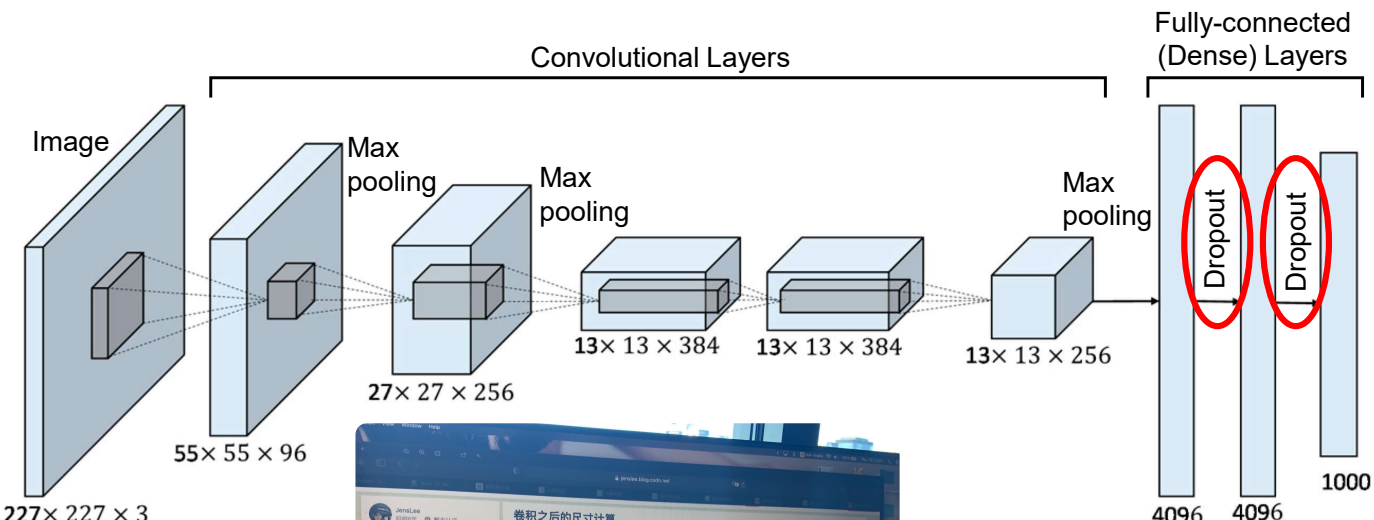
Max pooling

- Why do max pooling?
- Adds **translation invariance** – network response is the same even if features are in a slightly different position in the image
- Reduces the amount of data going to the next layer, which reduces computation time
- Tradeoff – it discards some information

dimension reduction

Convolutional neural network

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)



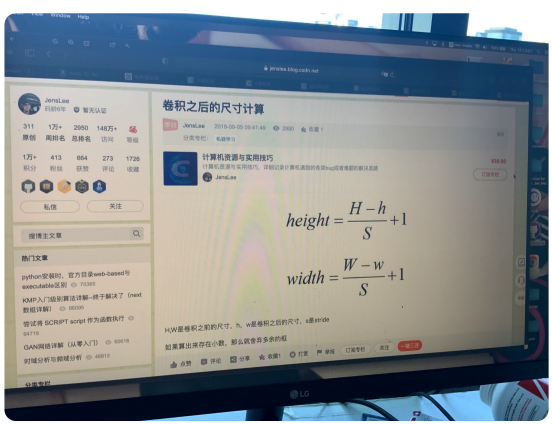


Image: Han, Zhong, Cao, & Zhang (2017)

Dropout

- Randomly discard some neurons (set output = 0)
- Form of regularization, forces neurons to find usefully features independently of each other
- Effectively, trains multiple architectures in parallel

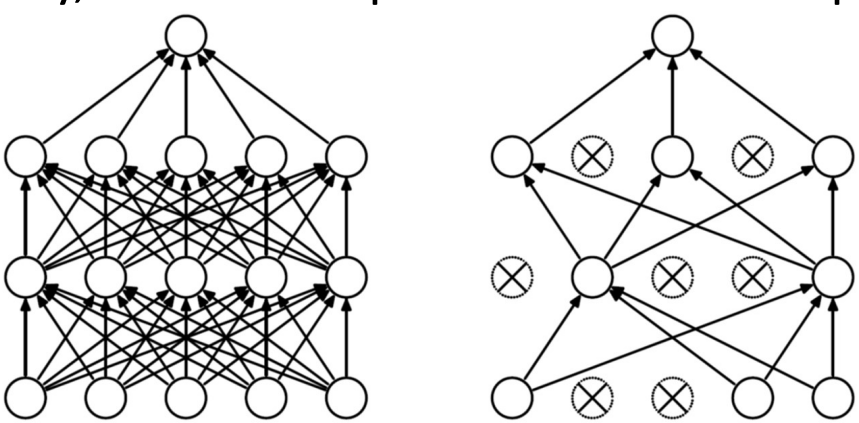


Image: Srivastava, Hinton, Krizhevsky, Sutskever, Salakhutdinov (2014)

What is it doing?

Human-like performance?

ImageNet classification

Russakovsky et al. (2014):

"With a sufficient amount of training, a human annotator is still able to outperform the GoogLeNet result… by approximately 1.7%."

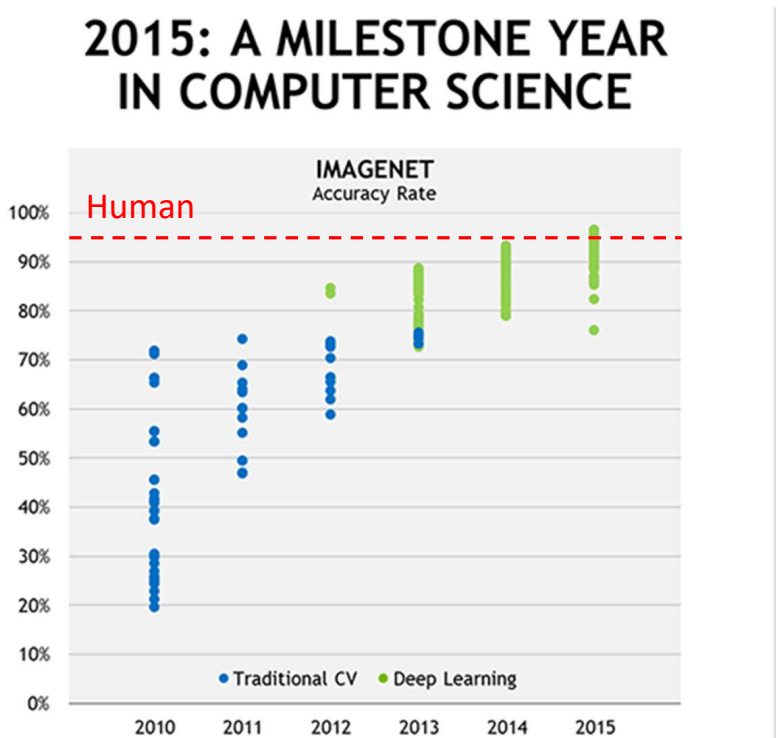


Image: https://blogs.nvidia.com/wp-content/uploads/2016/01/2-milestone-web1.gif

Human-like performance?

ImageNet Classification Failures (GoogLeNet 2014)



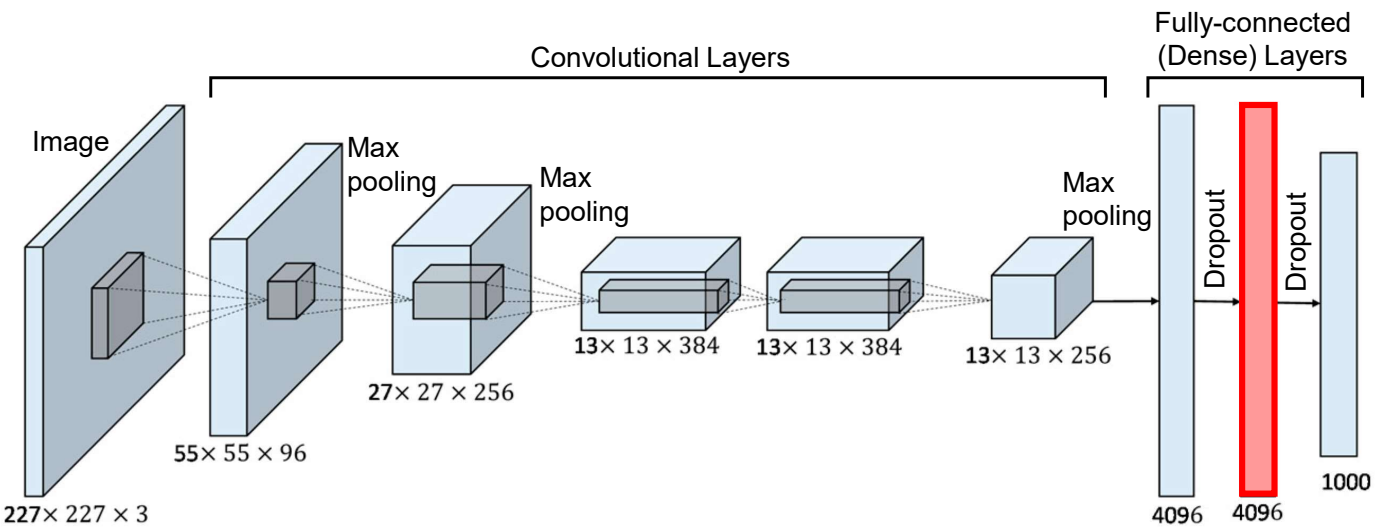


ruler	**king crab**	**sidewinder**	**saltshaker**	**reel**	**hatchet**
pencil box	pizza	maze	pill bottle	stethoscope	vase
rubber eraser	strawberry	gar	water bottle	whistle	pitcher
ballpoint pen	orange	valley	lotion	ice lolly	coffeepot

Russakovsky et al. (2014)

Generalisation

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)

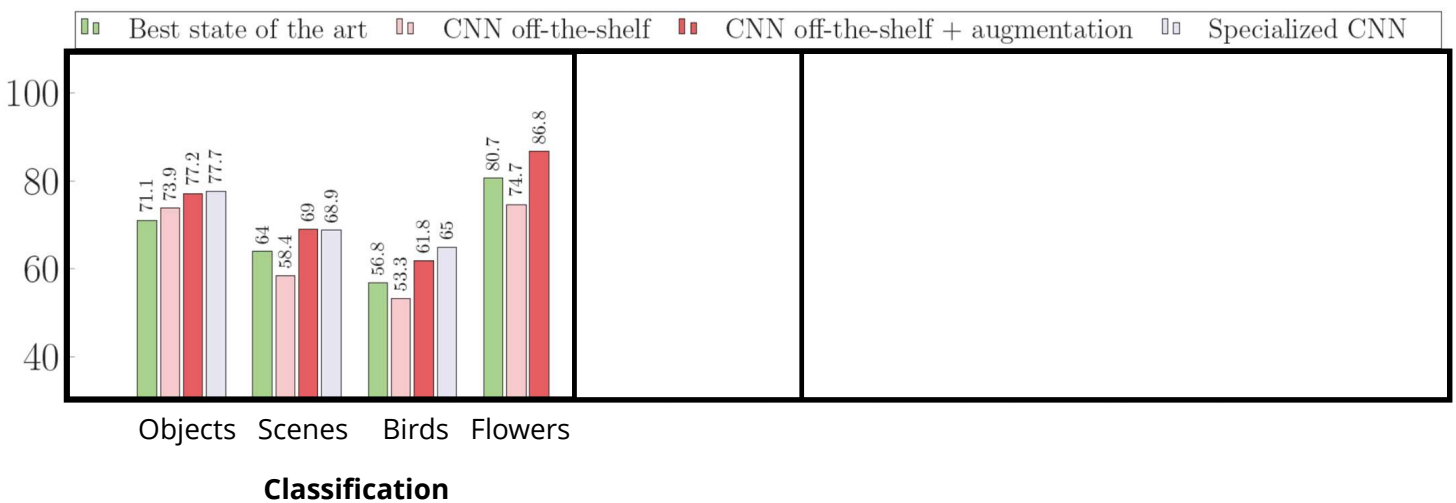


Embedding of an input = the network's response to the input at some layer

Image: Han, Zhong, Cao, & Zhang (2017)

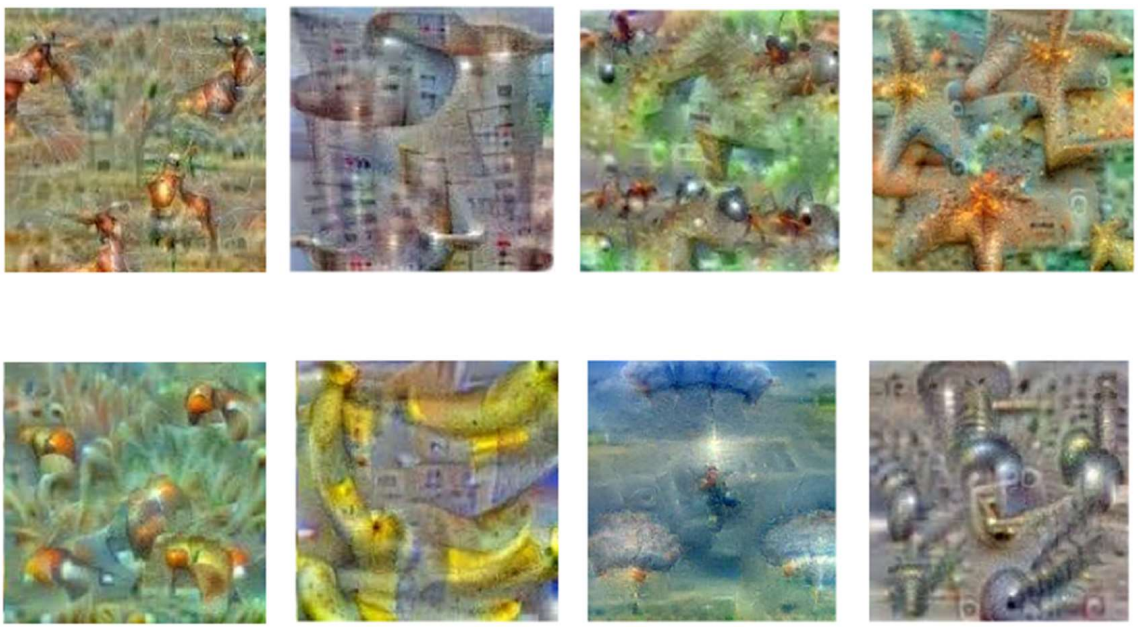
Generalisation

- **Embeddings** from neural networks are good representations for a range of tasks



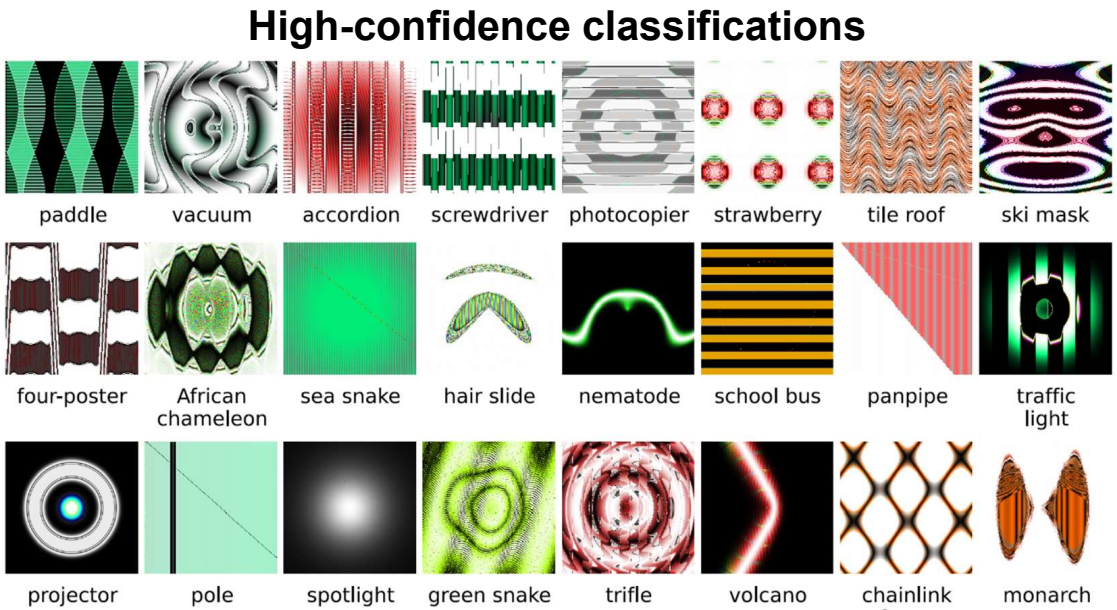
Razavian et al. (2014)

Visualising classes



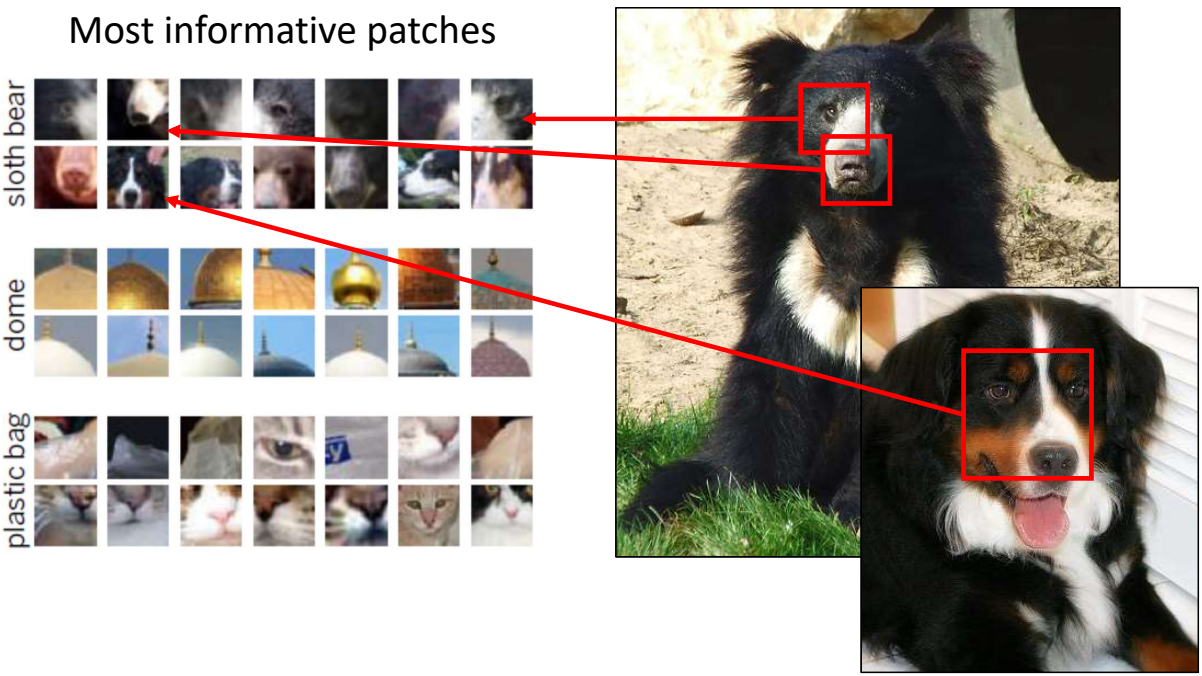
Google AI Blog: Mordvintsev, Olah, & Tyka (2015)
https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

Visualising classes



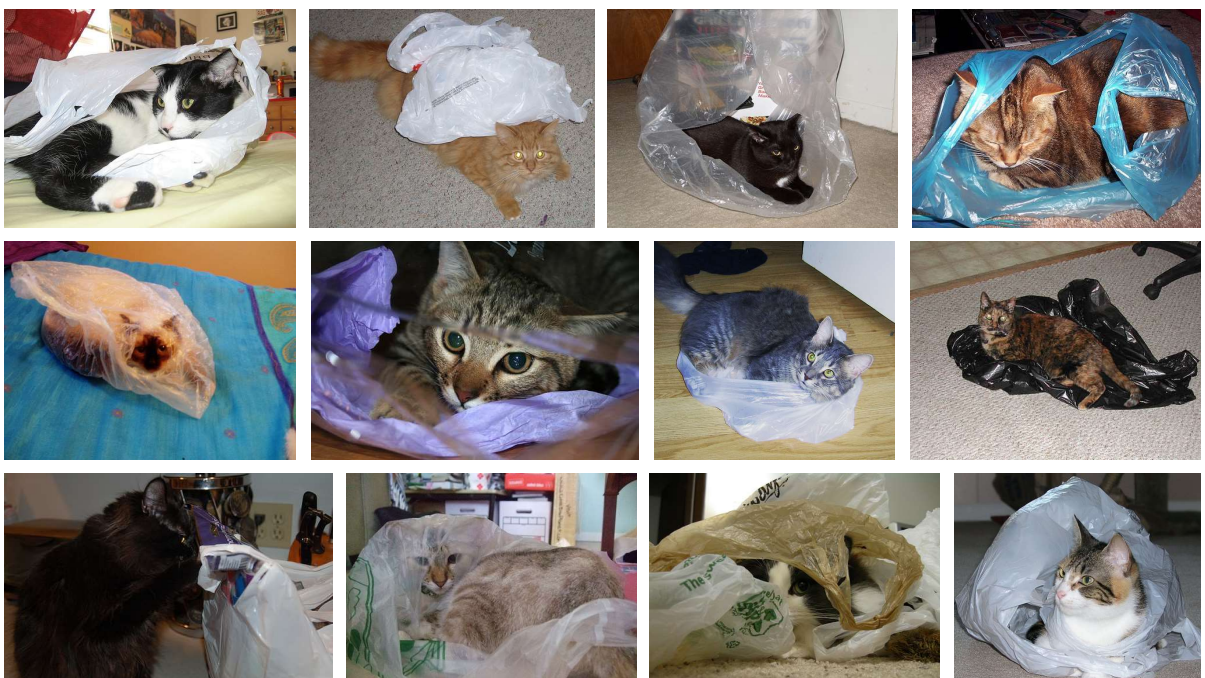
Nguyen, Yosinski, & Clune (2015)

Visualising classes



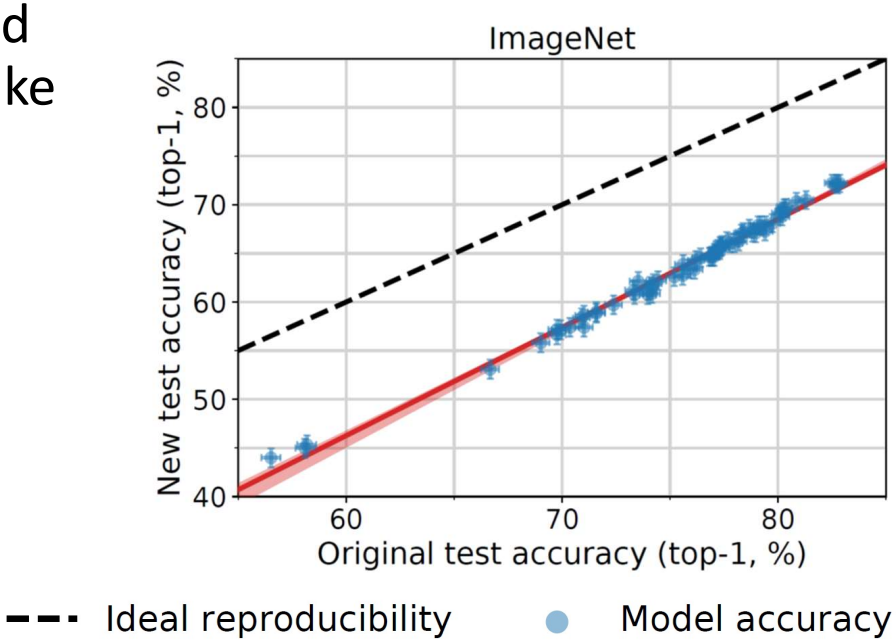
Brendel & Bethge (2019)

ImageNet “plastic bag” class



Generalisation, continued

- Models trained on one task (like ImageNet classification) may not generalize to very similar tasks (like ImageNet v2)



Recht, Roelofs, Schmidt, & Shankar (2019)

Generalisation, continued

- Models are very sensitive to some types of noise

Train on regular images: 

Can recognize: 

Can't recognize:

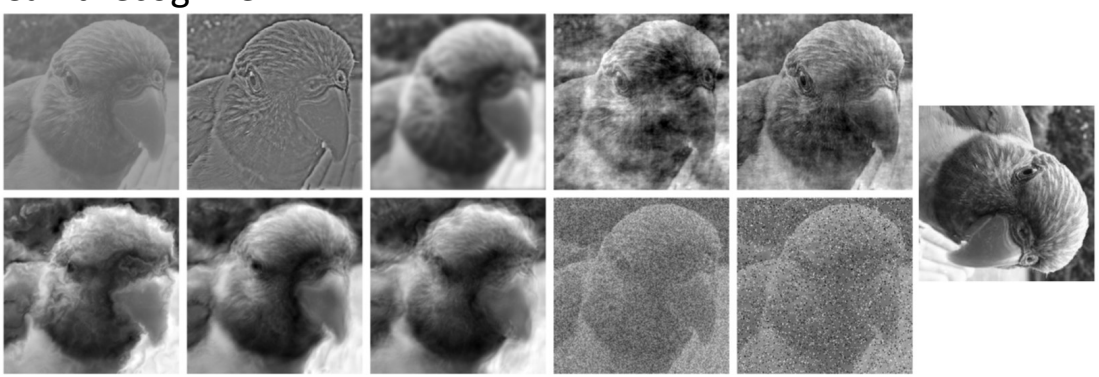
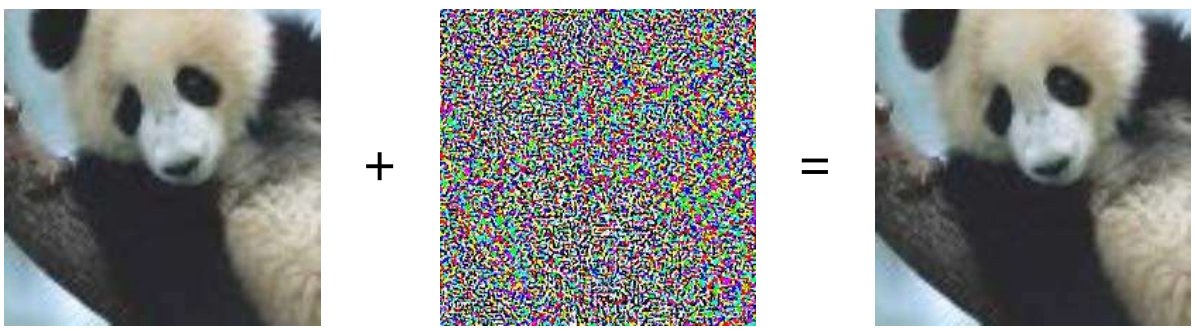


Image: Geirhos, Temme, Rauber, Schütt, Bethge, & Wichmann (2018)

Adversarial images

- Adding small amounts of noise to an image can completely change the model's perception

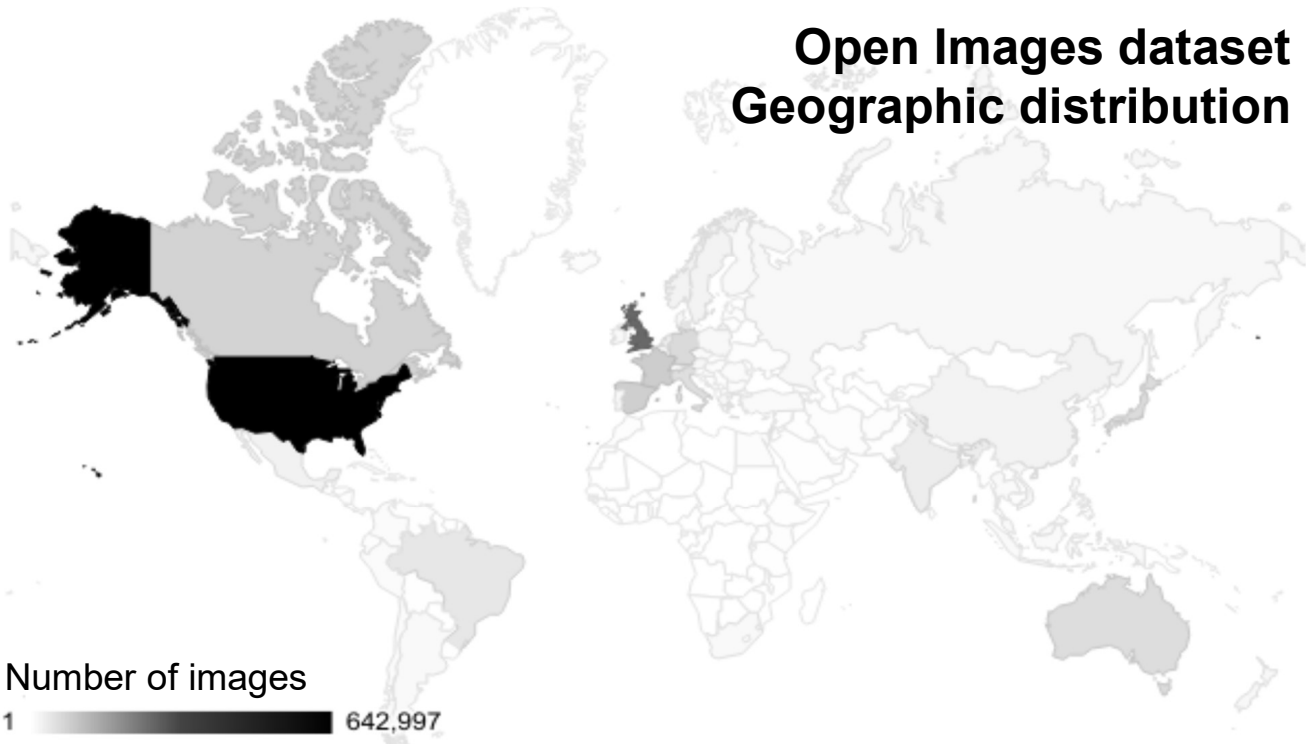


Original image: "panda"
(57.7% confidence)

New image: "gibbon"
(99.3% confidence)

Image: Goodfellow, Shlens, & Szegedy (2015)

A note about dataset construction



Shankar, Halpern, Breck, Atwood, Wilson, & Sculley (2017)

Object detection: Geographic bias

Ground truth: Soap **Nepal, 288 $/month**

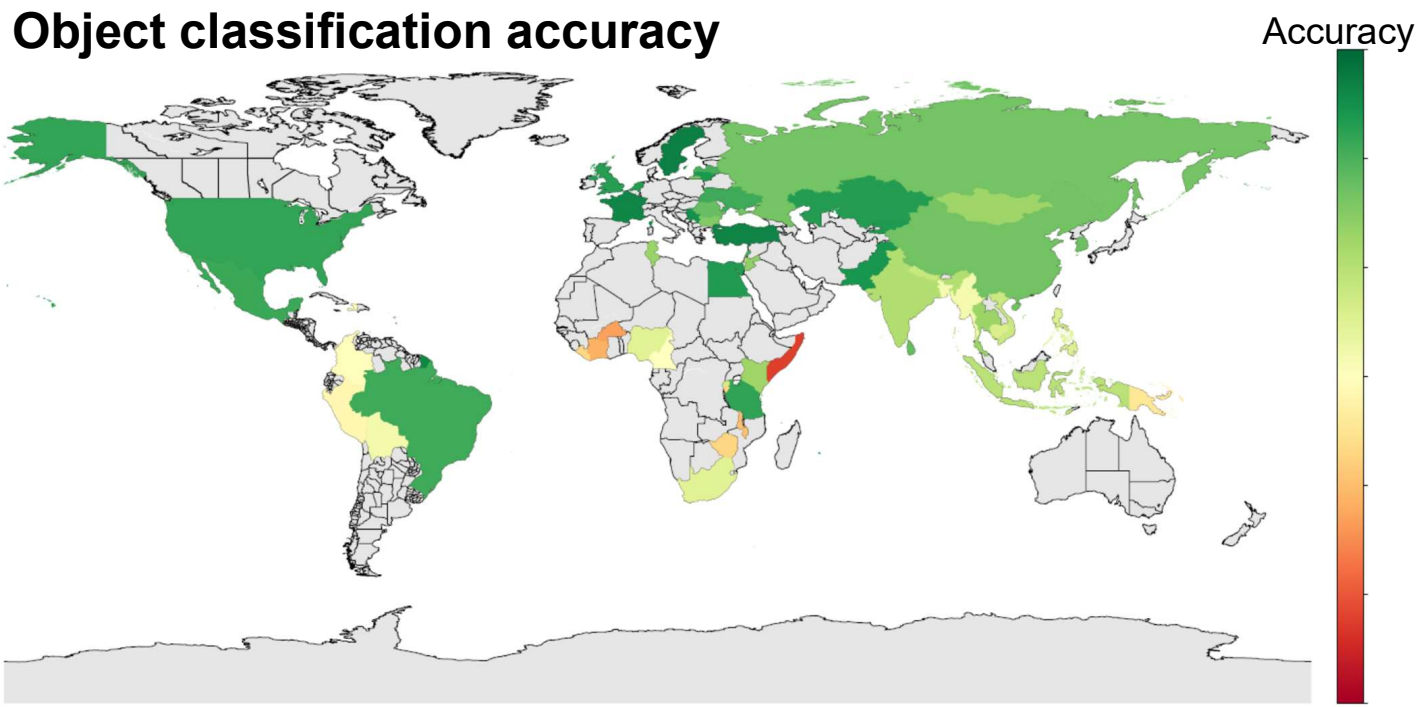
Azure: food, cheese, bread, cake, sandwich
Clarifai: food, wood, cooking, delicious, healthy
Google: food, dish, cuisine, comfort food, spam
Amazon: food, confectionary, sweets, burger
Watson: food, food product, turmeric, seasoning
Tencent: food, dish, matter, fast food, nutriment

Ground truth: Soap **UK, 1890 $/month**

Azure: toilet, design, art, sink
Clarifai: people, faucet, healthcare, lavatory, wash closet
Google: product, liquid, water, fluid, bathroom accessory
Amazon: sink, indoors, bottle, sink faucet
Watson: gas tank, storage tank, toiletry, dispenser, soap dispenser
Tencent: lotion, toiletry, soap dispenser, dispenser, after shave

DeVries, Misra, Wang, & van der Maalen (2019)

Object detection: Geographic bias

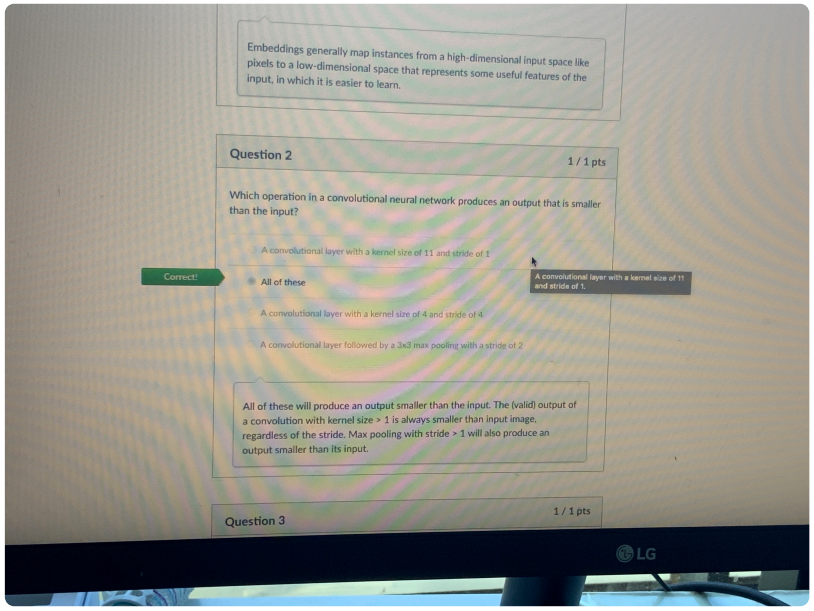


DeVries, Misra, Wang, & van der Maalen (2019)

Summary

- In theory, neural networks make few assumptions… but in practice architecture decisions will constrain what the model can (or is likely to) learn
- Advantages
  - State-of-the-art performance
  - Embeddings may be useful for a range of tasks
- Disadvantages
  - Difficult to interpret
  - Requires large training datasets
  - Sensitive to noise, dataset bias

making decision of the architecture


Embeddings generally map instances from a high-dimensional input space like pixels to a low-dimensional space that represents some useful features of the input, in which it is easier to learn.
Question 2
1 / 1 pts
Which operation in a convolutional neural network produces an output that is smaller than the input?
A convolutional layer with a kernel size of 11 and stride of 1
Correct!
All of these
A convolutional layer with a kernel size of 11 and stride of 1.
A convolutional layer with a kernel size of 4 and stride of 4.
A convolutional layer followed by a 3x3 max pooling with a stride of 2
All of these will produce an output smaller than the input. The (valid) output of a convolution with kernel size > 1 is always smaller than input image, regardless of the stride. Max pooling with stride > 1 will also produce an output smaller than its input.
Question 3
1 / 1 pts
LG