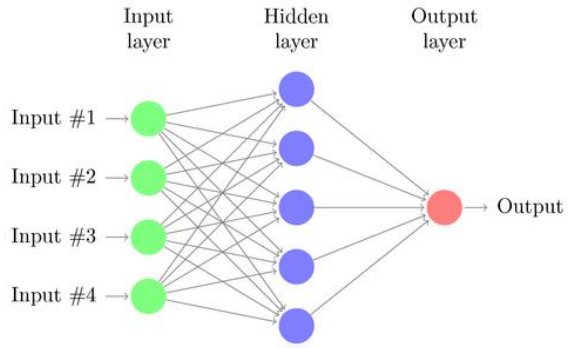
A gentle introduction to CNNs

David Wallis

16th June 2020

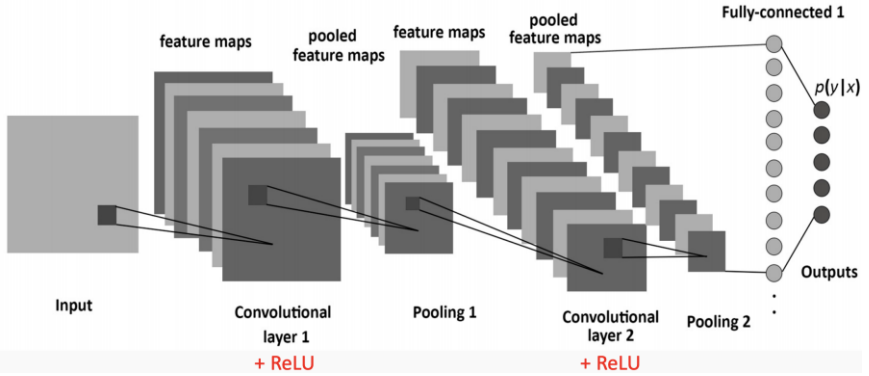
Why use a CNN?

- If using a traditional neural network to classify images, with each input representing a pixel, the number of weights quickly explodes
- Spatial relationships between pixels not taken into account



Standard multi-layer perceptron (traditional neural network)

Want to use spatial relationships

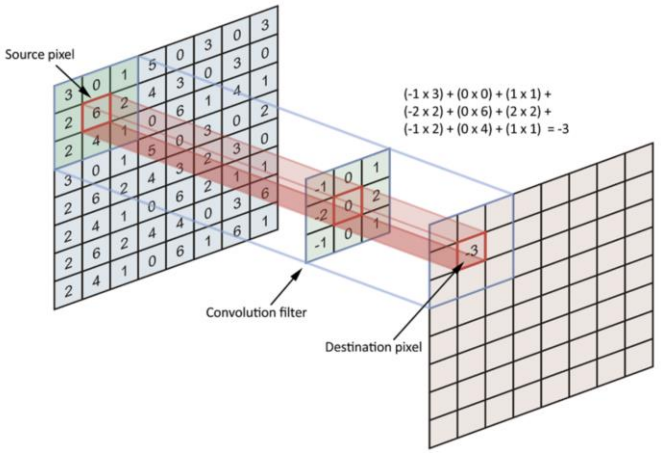


Example convolutional neural network with two convolutional layers

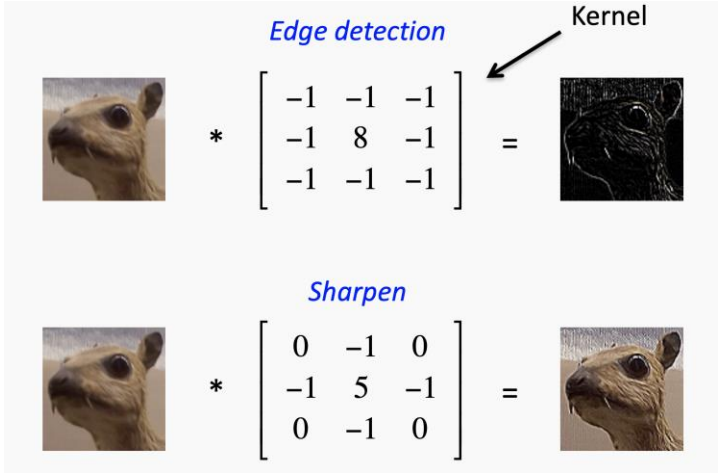
- Instead use convolutional neural network, utilising successive feature maps
- Used by Alex Krizhevsky to win 2012 ImageNet image classification competition

Architecture of a CNN – Convolutional Layer

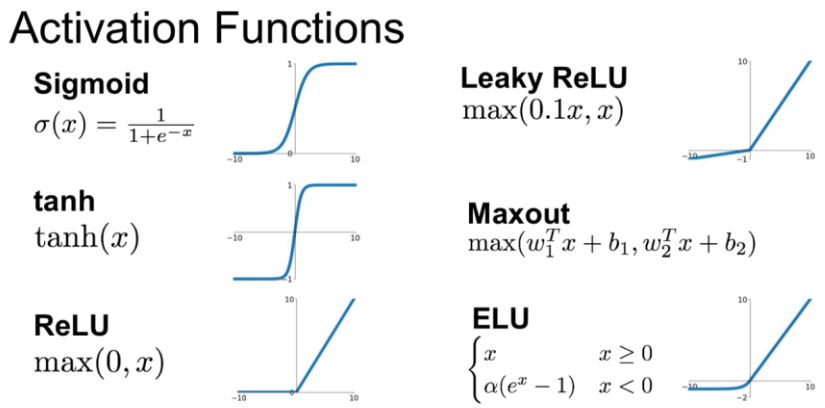
- Input image is convolved with a filter
- Edge detection and sharpening are examples of convolutions
- In a CNN the filters are learnt through training rather than pre-defined
- An activation function is then applied to introduce non-linearity (as in a traditional neural network)



3x3 filter acting on image. Filters can be different sizes depending on the task



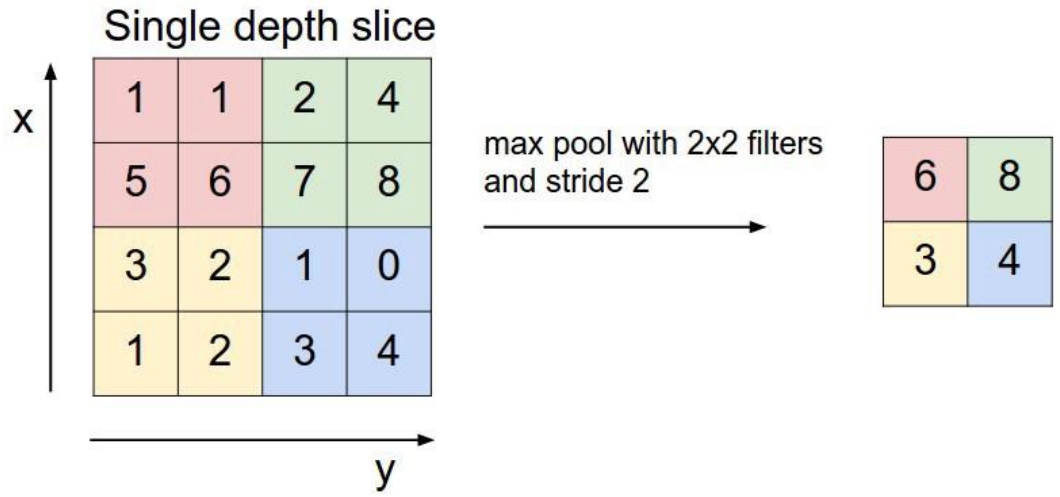
Edge detection and sharpen, examples of pre-defined filters



Example activation functions

Architecture of a CNN – Pooling Layer

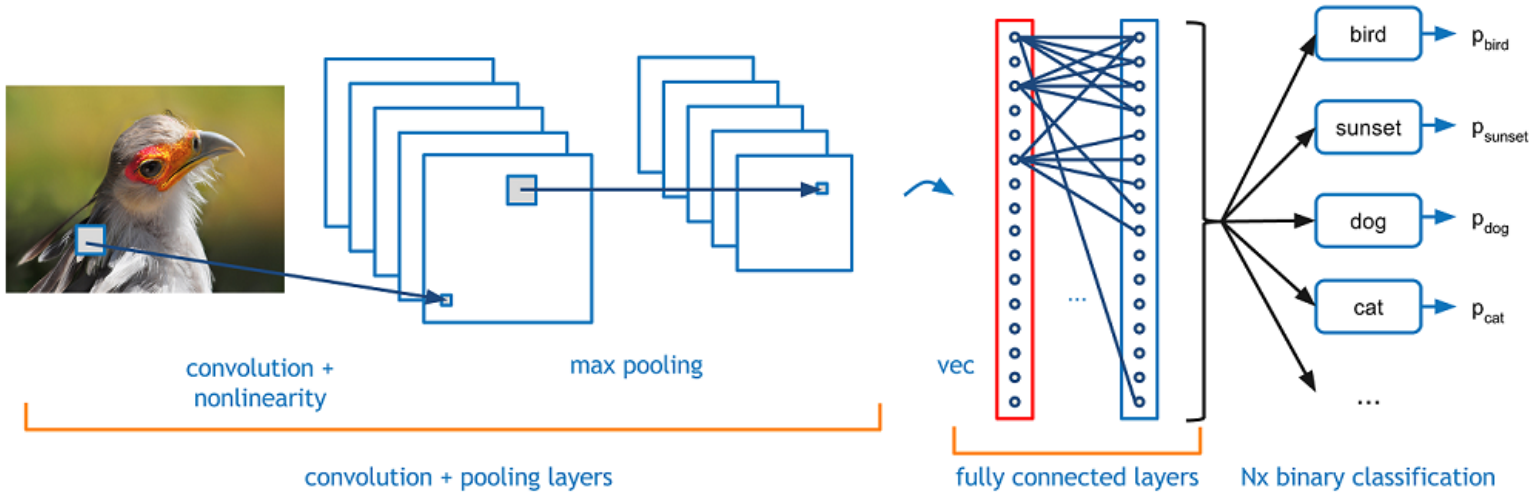
- Pooling layers are commonly used to reduce the number of parameters and control overfitting
- The size and stride of the pooling can be changed
- Two common pooling methods: Max Pooling and Average Pooling



Max Pooling example

Architecture of a CNN – Fully Connected Layer

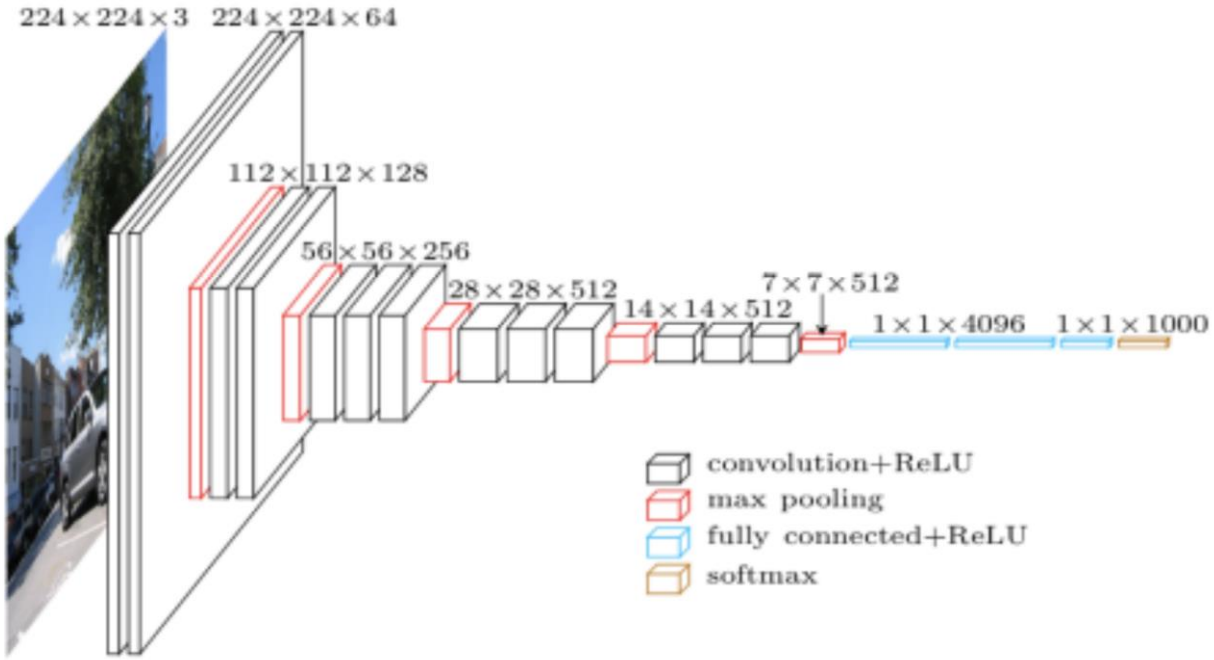
- The output from a series of convolutional layers is a set of high-level features, hopefully well trained to be discriminative for the task
- A traditional multi-layer perceptron can then be used to combine these features in a non-linear manner to classify the image
- ‘Fully Connected’ because every node from one layer is connected to every node in the next



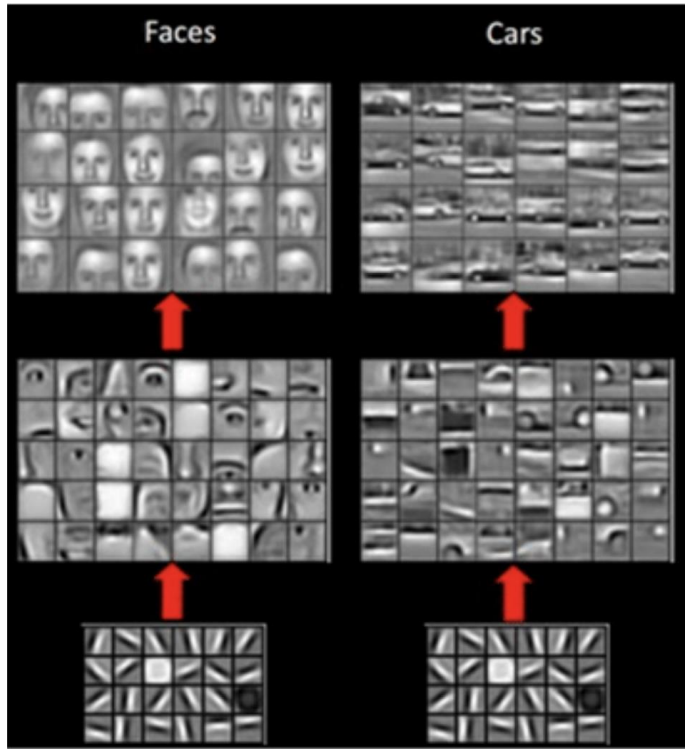
Example of network showing fully connected layers before classification

Putting it together – Example Network

- Network is trained as with a traditional neural network, using a gradient descent
- Successive layers learn increasingly complex features of the images

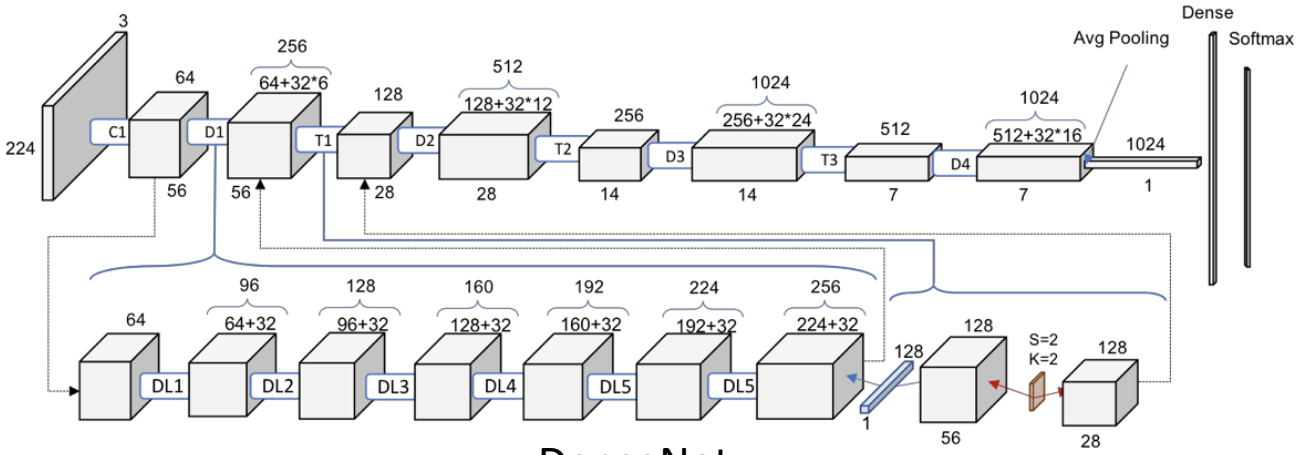


Example network showing convolutional, max pooling, and fully connected layers

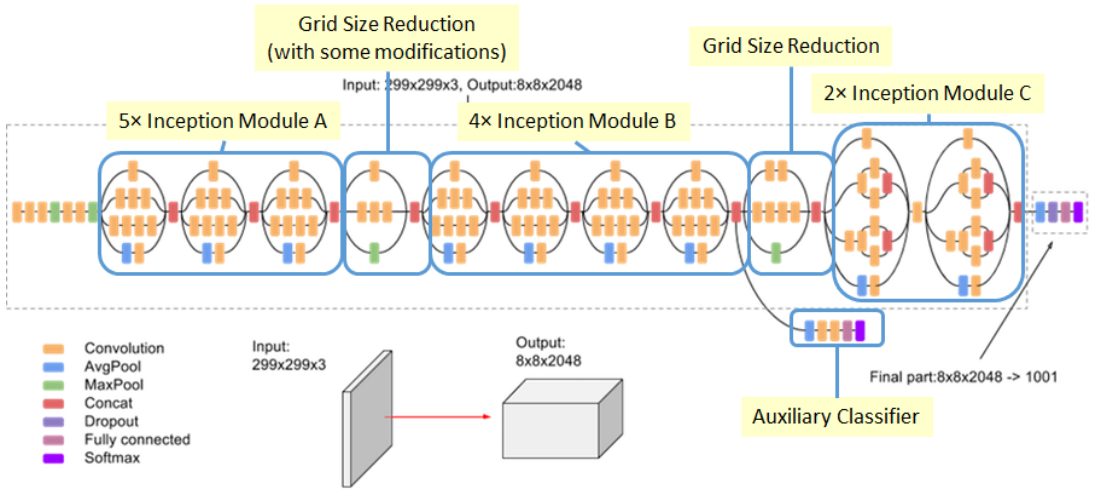


Features learnt in successive layers of CNN, going from basic geometric features to full objects

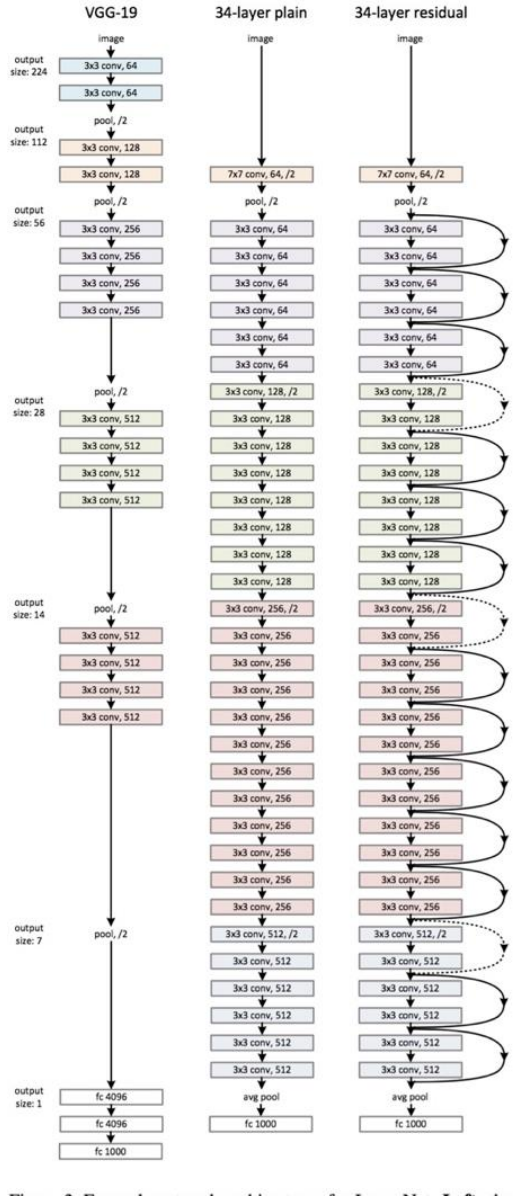
More Complicated Networks



DenseNet



Inception v3



ResNet 34

Further Reading/References

Basic Introduction:

- https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
- https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17
- https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac
- https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/

Stanford Course on Computer Vision:

- http://cs231n.stanford.edu/slides/2017/

Cool CNN Visualisation:

- https://www.cs.ryerson.ca/~aharley/vis/conv/

Network Architectures:

- Inception v3: https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c
- ResNet: https://arxiv.org/pdf/1512.03385.pdf
- DesnetNet: https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a