**Convolutional Neural Networks**

**Citations:** Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2014). Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

Mahendran, A., & Vedaldi, A. (2014). Understanding deep image representations by inverting them. *arXiv preprint arXiv:1412.0035*.

Background:

In the last few years deep convolutional neural networks have been seeing an explosion in literature and on the internet. They differ from traditional networks by making the explicit assumption that the input data is an image. This allows convolutional or deep networks as they are more commonly called draw another inspiration from nature – receptive fields of vision. The deep network can focus on specific parts of the image using a convolution of the image to focus on specific features of the image.

  
Figure X: Receptive field of deep network vs receptive field in biological neuron

Convolutional networks consist of convolutional layers – which act as receptive fields, followed by pooling layers – which decrease the amount of features and pixels the next convolutional and pooling layers can focus on. These convolutional + pooling layers are stacked many times until finally connected to some sort of classical neural network with some hidden layers and then finally the output layer as the classifier.



Figure X: ConvNets arrange the neurons in 3 Dimensions (width, height, depth)

The theory is that recognizable features will be close to each other – this allows us to create these receptive fields by convolutional layers. The other option is to create a conventional neural network in which the first neuron layer is connected to every single pixel of the image. This will potentially result in a higher descriptive ability of the network at the cost of greatly increased complexity. Also the neural network will not exploit the fact that pixels very far from each other are not likely to be part of the same feature.

Successful Applications of Deep Networks:

Deep networks have been very successful in classifying many different kinds of objects. Some of the best networks are able to classify as many as 22000 different categories learned from a set of 15 million images [ImageNet]. Despite highly optimized code and 3 high performance video cards – the ImageNet network takes about 5 days to train according to the authors.

Another very successful network is outlined in a paper from Google – the GoogLeNet [Google]. It employs many classical computer vision techniques along with the raw computational power of CNN (convolutional neural networks). Some of the novel techniques utilized in the architecture from Google are using very stacks of very small convolutional layers in order to abstract features away from each other and using parallel streams of pooling data which causes some connections to be sparser.

Problems and possible future improvements of CNNs:

Move to improvements section

The main way of increasing model complexity and computational potential is by adding more layers to the topology of the network. This can be problematic because signals have to be extremely strong in order to make it through all the layers. This can be solved by having some neurons skip several layers forward into the network – creating a sparse topology. This is in fact more akin to how neurons fire in biological systems.

It is still not well understood in literature on what exactly creates a better topology for CNNs. A possible, yet very computationally costly solution is use genetic algorithms to rearrange the topology in an automatic way. The exciting potential of this strategy is that a good neural network can “evolve” to solve difficult problems. An unfortunate side effect of the above is the fact that power consumption and computation cost continues to increase with more complicated models and more hyperparameters. A way of speeding up computation and of controlling wasted electricity is using special built hardware such as FPGAs (field programmable logic arrays) for the task of searching through the evolutionary search space. This could potentially result in ground-breaking work only being accessible to large and established tech companies, because no one else can afford expensive FPGAs.