統計書報討論(一)

Deep learning

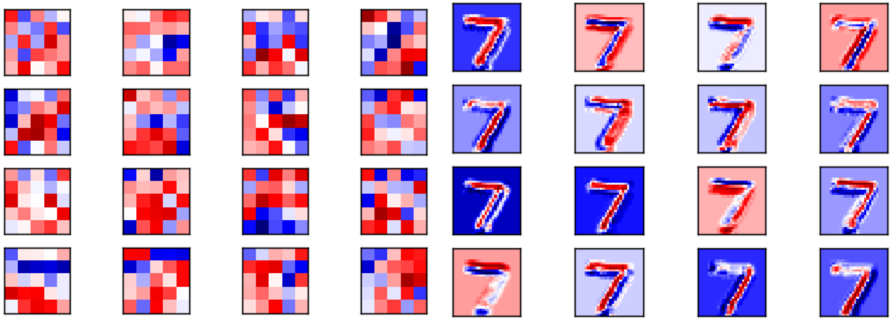
Reporter: Bo-Zen Xiao

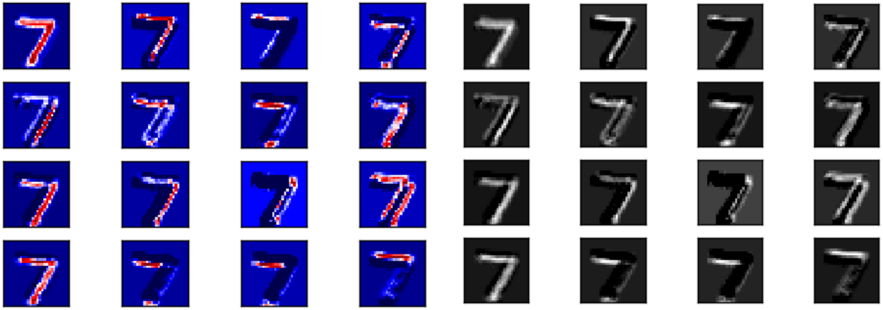
2017-05-19

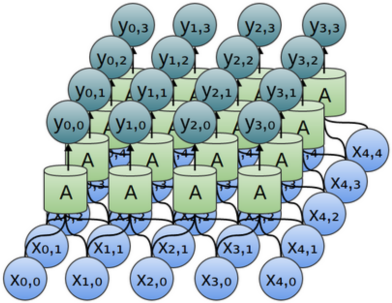
Outline

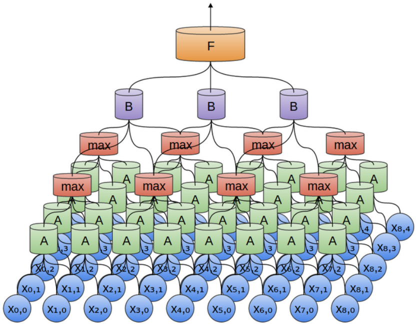
Structured Outputs

- Convolutional networks can be used to output a high-dimensional, structured object, rather than just predicting a class label for a classification task or a real value for a regression task.
- For example, the model might emit a tensor S, where $S_{i,j,k}$ is the probability that pixel (j, k) of the input to the network belongs to class i.
  This allows the model to label every pixel in an image and draw precise masks that follow the outlines of individual objects.






y0,3
y1,3
y2,3
y3,3
y0,2
y1,2
y2,2
y3,2
y0,1
y1,1
y2,1
y3,1
y0,0
y1,0
y2,0
y3,0
A
X4,4
X4,3
X4,2
X0,1
X1,1
X2,1
X3,1
X4,1
X0,0
X1,0
X2,0
X3,0
X4,0


F
B
max
A
X8,4
X8,3
X8,2
X0,1
X1,1
X2,1
X3,1
X4,1
X5,1
X6,1
X7,1
X8,1
X0,0
X1,0
X2,0
X3,0
X4,0
X5,0
X6,0
X7,0
X8,0

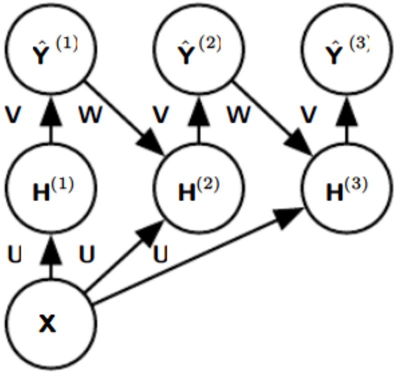
The Issue Of Outputs

- One issue that often comes up is that the output plane can be smaller than the input plane.
- The greatest reduction in the spatial dimensions of the network comes from using pooling layers with large stride.
- In order to produce an output map of similar size as the input, one can avoid pooling altogether. In principle, one could use a pooling operator with unit stride.

The Concept Of Recurrent Convolutional Network

- One strategy for pixel-wise labeling of images is to produce an initial guess of the image labels, then refine this initial guess using the interactions between neighboring pixels.
- Repeating this refinement step several times corresponds to using the same convolutions at each stage.

The Figure Of Recurrent Convolutional Network



Data Types

- The data used with a convolutional network usually consists of several channels, each channel being the observation of a different quantity at some point in space or time.

Different formats of data

	Single channel	Multi-channel
1-D	Audio waveform	Skeleton animation data
2-D	Black image data	Color image data
3-D	CT scans	Color video data

Varying Spatial Extents Data

- One advantage to convolutional networks is that they can also process inputs with varying spatial extents.
- For example, consider a collection of images, where each image has a different width and height. the kernel is simply applied a different number of times depending on the size of the input, and the output of the convolution operation scales accordingly.

Efficient Convolution Algorithms

- Modern convolutional network applications often involve networks containing more than one million units. Powerful implementations exploiting parallel computation resources are essential.
  However, in many cases it is also possible to speed up convolution by selecting an appropriate convolution algorithm.

Fourier Transform

- Convolution is equivalent to converting both the input and the kernel to the frequency domain using a Fourier transform, performing point-wise multiplication of the two signals, and converting back to the time domain using an inverse Fourier transform.
  For some problem sizes, this can be faster than the naive implementation of discrete convolution.

Separable Convolution

- When a d-dimensional kernel can be expressed as the outer product of d vectors, one vector per dimension, the kernel is called separable.
- When the kernel is separable, naive convolution is inefficient. It is equivalent to compose d onedimensional convolutions with each of these vectors.
- If the kernel is w elements wide in each dimension, then naive multidimensional convolution requires $O(w^d)$ runtime and parameter storage space, while separable convolution requires $O(w \times d)$ runtime and parameter storage space.
- Of course, not every convolution can be represented in this way.

Conclusion of Efficient Convolution Algorithms

- Designing faster ways of performing convolution or approximate convolution without harming the accuracy of the model is an active area of research.
- Even techniques that improve the efficiency of only forward propagation are useful because in the commercial setting, it is typical to devote more resources to deployment of a network than to its training.

How To Reduce The Cost of Convolutional Network

- Typically, the most expensive part of convolutional network training is learning the features.
- The output layer is usually relatively inexpensive due to the small number of features provided as input to this layer after passing through several layers of pooling.
- When performing supervised training with gradient descent, every gradient step requires a complete run of forward propagation and backward propagation through the entire network.
- One way to reduce the cost of convolutional network training is to use features that are not trained in a supervised fashion.

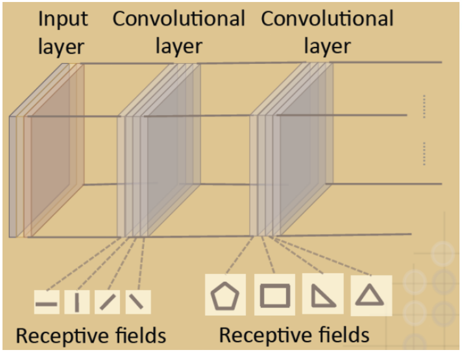
Unsupervised Training

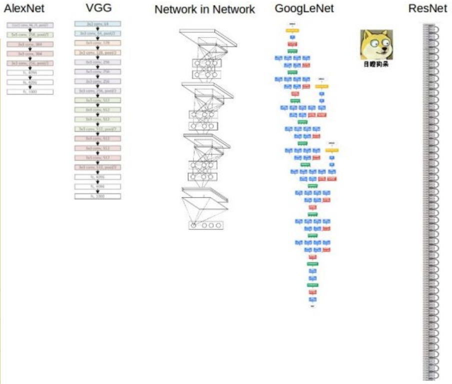
- For example by setting each kernel to detect edges at a certain orientation or scale. Finally, one can learn the kernels with an unsupervised criterion. For example, apply k-means clustering to small image patches, then use each learned centroid as a convolution kernel.

Random Filters

- Layers consisting of convolution following by pooling naturally become frequency selective and translation invariant when assigned random weights.
- They argue that this provides an inexpensive way to choose the architecture of a convolutional network: first evaluate the performance of several convolutional network architectures by training only the last layer, then take the best of these architectures and train the entire architecture using a more expensive approach.

Conclusion of CNN





Reference

- Goodfellow, I., Bengio, Y.,and Courville, A.(2016). *Deep Learning*. Retrieved from http://www.deeplearningbook.org
- http://colah.github.io/posts/2014-07-Conv-Nets-Modular/
- http://ithelp.ithome.com.tw/articles/10187521

Thank You
for your
Attention