Deep Learning Notes 9.6 - 9.9

Key Words : Structured Outputs; Data Types; Efficient Convolution Algorithm; Random or Unsupervised Features

1. Structured Outputs

1. 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.

2. Typically this object is just a **tensor**, emitted by a standard convolutional layer. For example, if $\mathbf{S}$ denotes the output tensor, then $\mathbf{S}_{i,j,k}$ is the probability that pixel (j,k) of the input image to the network belongs to class i.

3. The biggest problem is that **the output plane can be smaller than the input plane** since the greatest reduction in the spatial dimensions of the network comes from **using pooling layers with large stride**. The solutions are 1) avoid pooling altogether, 2) emit a lower-resolution grid of labels, 3) use a pooling operator with unit stride.

2. Data Types

4. 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. 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.

Figure.1 shows the 1D case

	Single channel	Multi-channel
1-D	Audio waveform: The axis we convolve over corresponds to time. We discretize time and measure the amplitude of the waveform once per time step.	Skeleton animation data: Animations of 3-D computer-rendered characters are generated by altering the pose of a "skeleton" over time. At each point in time, the pose of the character is described by a specification of the angles of each of the joints in the character's skeleton. Each channel in the data we feed to the convolutional model represents the angle about one axis of one joint.

Figure 1: The 1D case

Figure.2 shows the 2D case

2-D	Audio data that has been preprocessed with a Fourier transform: We can transform the audio waveform into a 2D tensor with different rows corresponding to different frequencies and different columns corresponding to different points in time. Using convolution in the time makes the model equivariant to shifts in time. Using convolution across the frequency axis makes the model equivariant to frequency, so that the same melody played in a different octave produces the same representation but at a different height in the network's output.	Color image data: One channel contains the red pixels, one the green pixels, and one the blue pixels. The convolution kernel moves over both the horizontal and vertical axes of the image, conferring translation equivariance in both directions.

Figure 2: The 2D case

Figure.3 shows the 3D case

3-D	Volumetric data: A common source of this kind of data is medical imaging technology, such as CT scans.	Color video data: One axis corresponds to time, one to the height of the video frame, and one to the width of the video frame.

Figure 3: The 3D case

5. Another advantage to convolutional networks is that they can also process inputs with varying spatial extents.

3. Efficient Convolution Algorithm

6. 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.

4. Random or Unsupervised Features

7. 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 as input to this layer after passing through several layers of pooling.

8. When performing supervised training with gradient descent, every gradient step requires a complete 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 supervis

9. There are three basic strategies for obtaining convolution kernels without supervised training. One is to simply initialize them randomly. Another is to design them by hand, for example by setting each kernel to detect edges at a certain orientation or scale. Finally, one can learn kernels with an unsupervised criterion.