Eric He

eh1885

Homework 2, Question 1

a. Convolutional networks make two assumptions about the structure of the input data. Say (1) what these assumptions are, (2) what advantages we gain by using a convolution when they are true. Give an example of a data set where these assumptions would not hold.

Convolutional networks assume:

1. Locality

The information of a feature is given by a function of the feature's neighbors. Information can be found by looking at local groups of features, rather than by looking at each feature itself. In the example of pictures, a pixel is uninformative on its own, but the local group of pixels is informative. Particularly informative local groups form "motifs" which can be captured by the cross-correlation operator.

2. Stationarity

CNNs assume that these motifs can be found anywhere in the data, so the cross-correlation operator is swept across the entire image. In the case of images, informative motifs like an edge or a face can be found at any position in the image.

Example

Standard business data, e.g. the Iris flower dataset, would make no sense to have a convolution. Every feature in the Iris flower dataset makes sense on its own, and the features are inherently orderless and without concept of a "neighborhood" in the sense that a pixel has a clearly defined neighborhood. It would not make sense to cross-correlate features, and it would especially not make sense to sweep a cross-correlation operator across the features.

Derivative of cross-correlation operator

The Jacobian coefficients can be calculated in a straightforward manner.

$$\frac{\delta z[k]}{\delta x[i]} = y[(i+k)\text{mod} n]$$

Here we take advantage of the fact that $x[i]$ only shows up in one term of the summation.

$$\frac{\delta z[k]}{\delta y[i]} = x[(i-k)\text{mod} n$$

$x[k]$ is multiplied by the index of y shifted k units to the right (under modulo n).

This means that $y[k]$ is multiplied by the x which is shifted k units to the left of $y[k]$: $x[(i-k)\text{mod} n$.

Dimensionality of FC net vs ConvNet

For f:

- The dimensionality of the output space is (1000, 1) since it is obtained through a (1000,100) x (100, 1) operation. The elementwise operation does not change the shape of the output.
- The number of trainable parameters is given by the shape of the weight matrix, (1000, 100) -> 1000 * 100 = 1e+5.
- The matrix multiplication requires 1000 * 100 = 1e+5 operations, the elementwise nonlinearity requires 1000 operations, giving us O(|W|).

For g:

- Each convolution filter returns a vector of size 100, and there are 10 such filters. Then the dimension of the output space is given by (|data|, |filter count|, |x|) = (100, 10, 1).

- The number of trainable parameters is given by the sum of the weights of all weight matrices. Each convolution filter has 3 weights, and there are 10 convolution filters, so there are 30 weights to train.
- One convolution operation takes 3 operations, corresponding to the size of the filter. The operation occurs 100 times for one pass over the filter. There are 10 filters. This gives us a computational cost of O(|filter count| |*filter sweep count*| |filter size|) = O(10 *100* 3) = O(3e+4).