

MATLAB Problem of the Sixth Practice Day

Convolution in Neural Network

이번 시간에는 Convolutional neural network 에서 사용되는 간단한 형태의 Convolutional layer 를 구현하도록 한다.

Convolutional layer 는 image 의 일부 영역인 patch 별로 feature vector 를 계산하는 layer 이다.

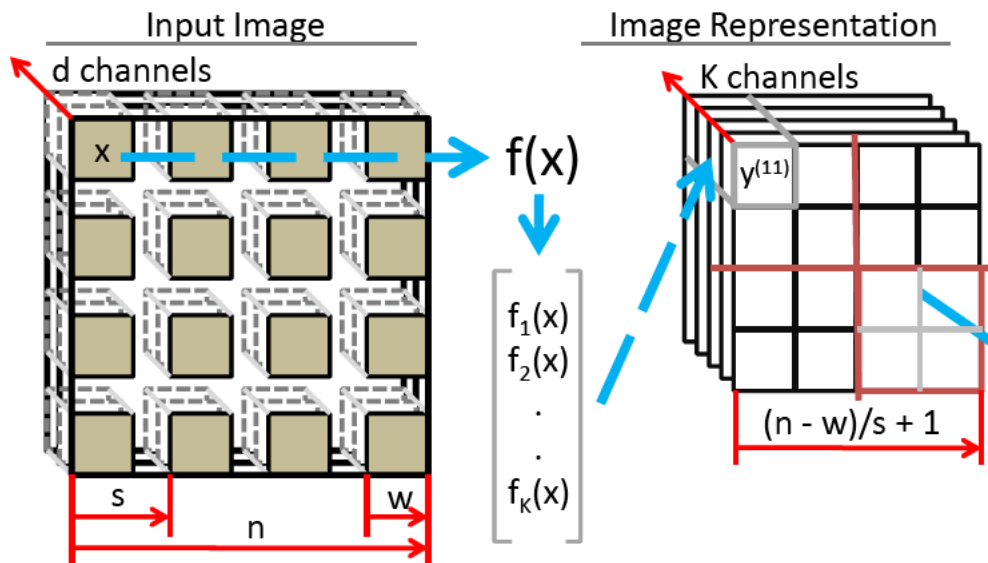


Figure 1 Convolution obtains the new image representation given an input image

흑백 image 는 image 에 대한 정보가 1 개 채널의 gray-level 로 주어지는 반면, Red-Green-Blue (RGB) 형태의 color image 의 경우, 빨강, 녹색, 청색에 대한 $d=3$ 개 채널의 정보로 주어진다. 이러한 image 에서 receptive field size (가로, 세로의 길이)가 w 인 patch 들을 임의로 골라낼(sample) 수 있다. 이때, 각 patch 들은 가로, 세로의 길이가 w 이고, channel 이 d 개인 $w \times w \times d$ array 가 된다. 이러한 patch 들을 같은 크기의 s 만큼 움직이며 골라낼 수 있는데, 이때 s 를 stride 라고 한다. 이와 같이 골라내면 하나의 image 로부터 총 n_p^2 개의 patch 들을 얻게 된다 ($n_p = \lfloor \frac{n-w}{s} \rfloor + 1$). 이때, 각 patch 로부터 K 길이의 feature vector 를 계산하여, patch 의 위치대로 나열하면 $n_p \times n_p \times K$ array 가 되며, 이러한 계산을 하는 layer 를 convolutional layer 라고 한다. 각 patch 로부터 feature vector 를 계산하는 방법은 다양하다.

이번 실습 시간에는 convolution 을 하기 위해 K-means clustering 을 활용하도록 한다. MNIST dataset 은 $d=1$ 개의 채널이 있고, image 의 가로, 세로의 길이는 $n=28$ 이다. 이를 $w = 14$ 크기의 patch 들을 $s = 14$ 만큼 움직이며 고르면, 한 image 당 $n_p^2 = 4$ 개의 patch 를 구할 수 있다. 즉, Training data 1,000 개가 주어질 경우, 4,000 개의 patch 가 나오게 된다. 이 patch 들을 vector 형태로 변환하면, K-means clustering 을 수행할 수 있다. 이때 한 patch 마다 각 cluster centroid 로부터의 거리를 계산하면, K 개의 거리가 나오고, 이를 K 길이의 feature vector 로 사용할 수 있다. 각 image 당 이러한 feature vector 의 개수는 patch 의 개수인 4 가 된다.

이번 시간에는 위의 설명대로 구현한 convolutional Layer로부터 얻어진 feature vector 를 input 으로 활용하여, 각자 원하는 형태의 classification algorithm (linear classifier, Gaussian classifier, radial basis function network, multi-layer perceptron 등)을 이용하여 training data 1,000 개에 대해 학습한 후 test data 100 개에 대한 running time, accuracy 를 계산해 보도록 한다. 이를 convolutional layer 를 사용하지 않았을 경우의 비교하여, 그 결과를 메일 본문에 기재하여 source code 와 함께 chhan@bi.snu.ac.kr 로 제출하도록 한다.

[참고]

수학적으로 discrete convolution 은 다음과 같이 정의된다.

Definition: The discrete convolution of discrete-time functions f and g is given by

$$f * g[n] = \sum_m f[n - m]g[m].$$

어떤 시점 n 에서의 convolution 은 함수 f 를 n 을 기준으로 시간 축을 반대로 뒤집은 후 각 m 에서 g 와 곱하여 더한 것이라고 볼 수 있다. Continuous convolution 에 대해 아래의 그림 2 를 참조할 수 있다. Continuous convolution 은 덧셈 대신 적분을 적용하면 되고, 그림 2 에서 shaded area 에 해당한다.

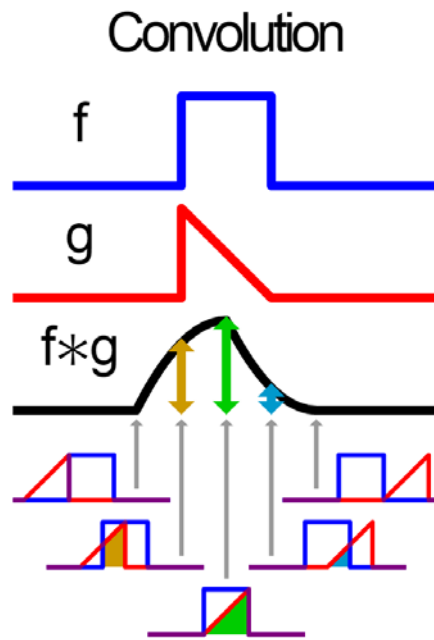


Figure 2 (Continuous) Convolution

Convolutional layer 에서 stride 가 1 이고, feature vector 를 계산할 때 k-means clustering 대신 matrix 곱(혹은 k 개의 vector 와의 내적)을 사용하면, convolutional layer 에서 수행하는 연산은 k 개의 2-dimensional discrete convolution 에 해당된다.

References

Adam Coates, Honglak Lee, Andrew Y. Ng, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," in AISTATS 14, 2011.

Wikipedia, <https://en.wikipedia.org/wiki/Convolution>, Retrieved Nov. 18, 2015.