Cs231n lecture 12 summary: Visualizing and Understanding

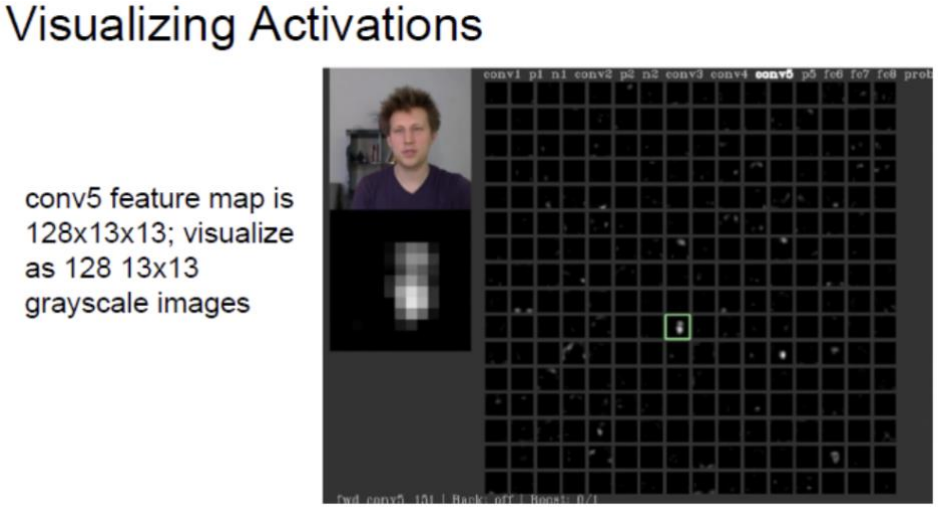
1. What's going on inside ConvNets?
2. Gradient Ascent
3. Feature Inversion
4. Neural Style Transfer

1. What's going on inside ConvNets?

ConvNet 안에서 일어나는 일들을 Visualizing하는 것이 중요한 이유는 딥러닝이 잘 작동하는 이유를 시각화해서 설명하고 납득시키기 위해서이다.

딥러닝 내부의 첫 번째 layer는 입력 이미지와 필터의 Weight 값을 내적하여 만든 layer이다. 따라서 우리의 눈으로 보았을 때 edge 성분이 많이 검출되는 것을 확인할 수 있다.

이제 이 필터들이 layer가 깊어지면서 합성곱 연산이 이루어지고 더 복잡해진다. 따라서 깊은 layer의 필터는 우리가 직관적으로 이해하기는 어려운 필터들이 된다. Last layer에서는 한 이미지에 4096 차원의 특징 벡터들이 존재한다. 따라서 각 이미지마다 특징 벡터들을 Nerest Neighbor 알고리즘을 돌리게 되면 유사한 이미지가 검출되는 것을 확인할 수 있다. 이 4096차원의 특징 벡터를 PCA 알고리즘 등으로 2차원으로 차원 축소를 하게 되면 군집화 된 모습을 볼 수 있다. 아까 중간의 깊은 layer들은 우리가 직관적으로 이해하기 어렵다고 했지만 모든 layer가 그런 것은 아니다. Activation map을 통과시킨 layer의 모습을 시각화하면 이처럼 사람의 얼굴 부분이 활성화된 layer가 있다는 것을 확인할 수 있다.



지금까지는 고정적인 1개의 이미지가 들어왔을 때 각 layer에서 어떤 반응을 보이는지를 관찰하였다. 그렇다면 일반적인(General)한 이미지가 들어왔을 때 어떤 이미지가 들어와야 각 뉴런들의 활성화가 최대치가 되는지 알아볼 수 있지 않을까?

이러한 방법으로 Visualization한 방법을 **Maximally Activation Patches**라고 한다.
또 다른 방법으로는 **Occlusion Experiments**라는 방법이 있는데 이는 입력의 어떤 부분이 Classification을 결정했는지 알아보기 위해 사용한 방법이다. 입력 이미지의 일부분을 가린 후 분류를 잘 하는지 확인하는 것인데, 만약 이미지를 가렸는데 네트워크 score에 변화가 있다면 네트워크는 그 부분을 분류하는 데 크게 영향을 미치는 요소로 판단했다고 볼 수 있다.
다른 방법으로 **Saliency Maps**라는 방법도 있는데 이는 어떤 픽셀을 보고 이미지를 분류했는지 알아내는 방법이다. 이 방법을 가지고 segmentation을 진행할 수는 있지만 그렇게 성능이 좋지는 않다.

2. Gradient Ascent

우리는 Gradient를 구할 때 Backpropagation 방법으로 구하는 경우가 많았다. 이것의 이는 입력 이미지가 들어왔을 때 Weight 값을 업데이트시키기 위해 사용했던 방법인데 이것의 반대 개념이 Gradient Ascent이다. 네트워크의 weight값은 고정시키고 해당 뉴런을 활성화시키는 General한 입력 이미지를 찾아내는 방법이다.

3. Feature Inversion

이 개념 또한 네트워크의 다양한 layer에서 어떤 요소들을 포착하고 있는지 알아볼 수 있는 방법이다. 이미지의 특정 layer에서 activation map을 추출한 다음 이 activation map을 가지고 이미지를 재구성하는 방법이다. 여기서 gradient ascent를 이용하는데, 스코어를 최대화하지 않고 특징 벡터의 거리가 최소화되는 방향으로 update를 진행한다.