DenseNet 리뷰

Reference

- Densely Connected Convolutional Networks, Gao Huang et al.

1. Introduction

CNN 모델들이 매우 깊어지면서 입력의 특징이 마지막 layer로 갈수록 흐려져 좋은 결과를 보이지 못하는 경우가 다수 발생하였다. ResNet은 이를 해결하기 위한 첫 시도로 identity mapping을 통해서 이전 layer의 input을 residual function의 결과물과 함께 다음 layer의 input으로 전달한다. DenseNet은 ResNet의 업그레이드 버전으로 바로 이전 layer의 input만 받는 ResNet과는 달리 이전 layer의 모든 input을 받고 이를 summation이 아니라 concatenate한다. 구체적인 아키텍쳐를 살펴보자.

2.DenseNet

하나의 이미지 x_0가 L개의 layers로 구성된 신경망을 통ㅇ과한다고 생각하자. 각 layers는 non linear transformation인 $H_l(\cdot)$을 수행하고 이는 BN, ReLU, Pooling, Conv으로 이루어져 있다고 생각하자. l번째 layer의 output을 x_l이라고 한다.

DenseNet은 ResNet과 비교 분석하면 이해가 더 빠르므로 ResNet을 우선 살펴보자.

Traditional convolutional networks

기존의 CNN은 l번째 layer의 output을 $l+1$번째 layer의 input으로 그대로 사용했다. 즉,

$$x_l = H_l(x_{l-1}) \qquad (1)$$

ResNets

ResNets은 기존의 CNN에서 identity mapping을 추가하였다.

$$x_l = H_l(x_{l-1}) + x_{l-1} \qquad (2)$$

identity mapping을 추가한 ResNets은 forward, backward propagation이 identity function을 통해서 직접적으로 흐른다는 장점이 있다. 즉, gradient vanishing 현상이 덜하다는 것이다. 하지만 identity function과 residual function의 output이 summation으로 합쳐지기 때문에, 신경망에서 정보의 흐름을 늦출 수 있다는 단점이 있다.

Dense connectivity

ResNets의 l번째 layer는 $l-1$번째의 output 추가로 받았지만 DenseNet은 $l-1$번째 까지의 모든 output을 받는다. 즉,

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]) \qquad (3)$$

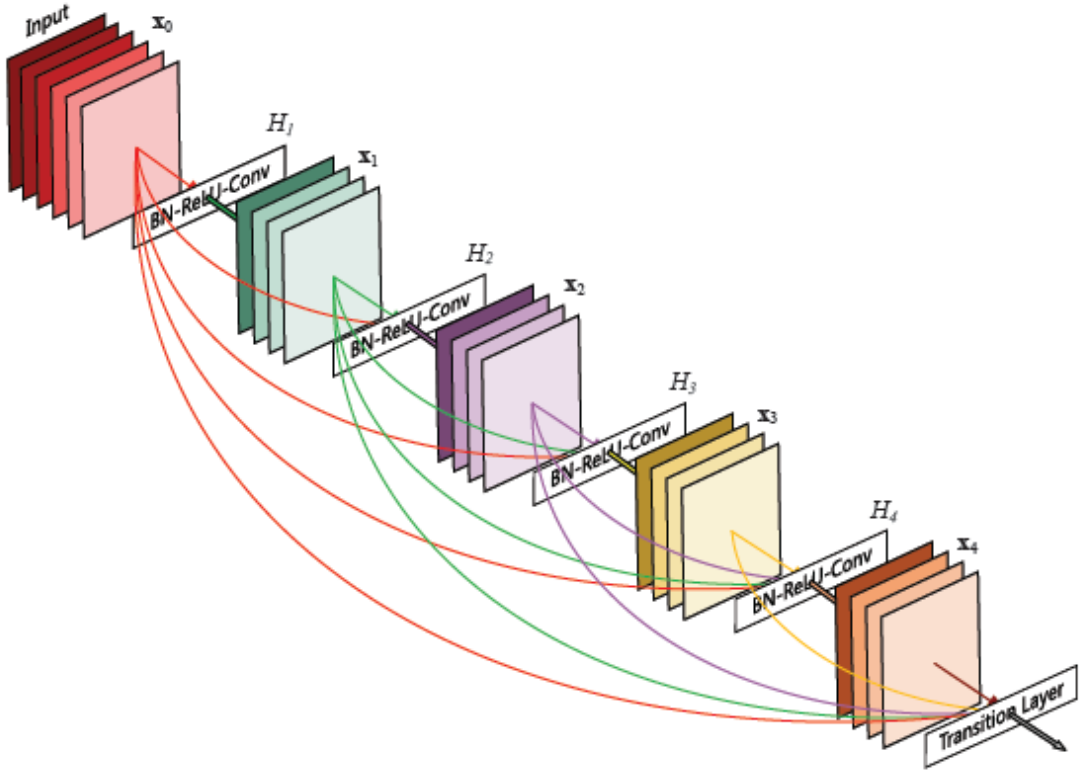

Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

위 그림은 DenseNet의 아키텍쳐를 보여준다. 가장 처음 layer인 빨간색 layer을 보자. 빨간색 layer의 output feature map이 이후의 모든 layer이 input으로 들어감을 빨간색 선을 통해서 보여주고 있다.

(3)의 Dense Connectivity가 (2) Resnet과 또 다른 점은, 이전의 output을 $H_l(\cdot)$과 더하는 것이 아니라 concatenate한다는 점이다. 따라서 l번째 layer에서 학습된 feature map은 그 다음의 모든 layer에서 접근할 수 있다. 이를 통해 신경망에서 feature reuse를 할 수 있는 것이다.

Composite function

$H_l(\cdot)$을 BN, ReLU, 3×3 Conv으로 구성한다.

Pooling layers

(3)의 conctenation은 feature maps의 크기가 변한다면 불가능하다. 하지만 down sampling을 통해서 feature maps의 크기를 조정할 수 있다. 좀 더 원활한 down sampling을 위해 신경망을 아래와 같이 dense blocks으로 나눈다. dense blocks 사이는 transition layers라고 부른다.

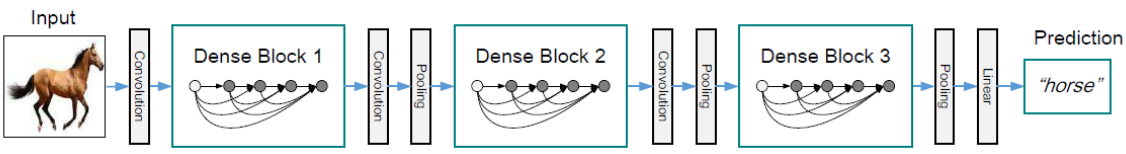

Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

Growth rate

만약 각 layer의 H_l이 k개의 feature maps을 만든다면 l번째 layer는 $k_0 + k \times (l-1)$개의 input feature maps을 가진다. 여기서 k_0는 l번째 layer의 channel의 수이다. k를 growth rate hyperparameter라고 하자. 논문에서는 작은 growth rate이 적절하다고 말한다.

Bottleneck layers

각 layer가 k개의 feature maps을 만든다할지라도 densenet 특성상 이전의 feature maps을 모두 받기 대문에 더 많은 inputs을 가진다. ResNet에서도 쓰인 bottleneck layer은 1×1 cov layer와 3×3 conv layer를 순차적으로 통과시켜 feature maps의 갯수를 줄이고 computational efficiency를 향상시킨다. densenet에서도 bottleneck layer을 사용한다. BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3) 순이다.

Compression

모델의 compactness를 향상시키기 위해 transition layers에서 feature maps을 줄인다. $0 < \theta \leq 1$의 범위를 가지는 compression factor을 통해 dense block에서 오는 m개의 feature maps를 $[\theta m]$개의 feature maps으로 줄인다. 만약 $\theta = 1$이라면 feature maps의 수는 변하지 않는다.