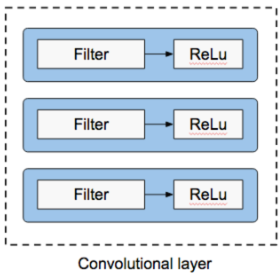
**Deformable Convolutional Network – Jifeng Die et al.**

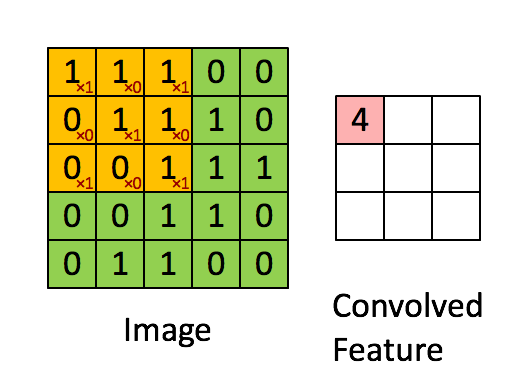
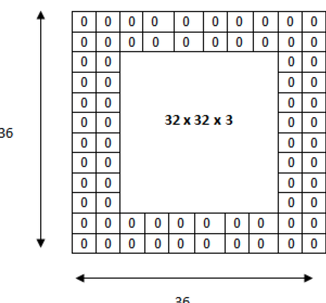
**1. 배경(CNN 개념)**

- Convolutional Layer는 Data의 특성을 추출하는 Filter와 Filter값을 비선형 값으로 바꾸어주는 Activation 함수로 구성.



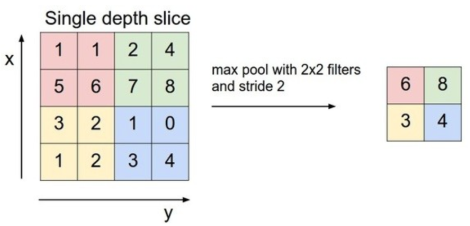
- Filter : data에 특징이 있는지 없는지 검출해주는 함수

[3x3 Filter 적용] [Padding] – Filter후 작아지는 것을 개선

- Activation Function : filter에서 추출된 값은 정량적이므로 비선형 값으로 바꿔주는 것임.

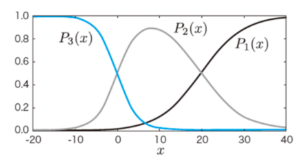
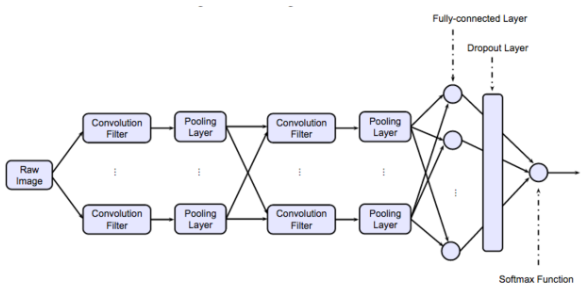
- Pooling(sub sampling) : 특징이 잘 추출된 Map은 크든 작든 판단을 할 수 있다. 따라서 filter에서 추출된 Activation map을 임의로 줄여 Computing 속도를 높이도록 하는 작업. Max pooling은 Activation map을 MxN으로 잘라낸 후 가장 큰 값을 뽑아내는 방법.



- Fully Connected layer : Convolution에서 추출된 data를 Neural Network에 넣는 것.

- Softmax 함수 : x가 P1, P2, P3 각각 어떤 확률을 가지는지 판단.

[Full Connected layer] [Soft-max 함수]



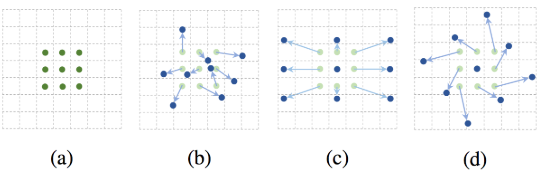
**2. Abstract**

- 기존 CNN은 기하학적으로 일정한 패턴을 가정하고 있어 복잡한 것은 유연하기 힘들고 Object detection을 위한 사람의 작업이 필요

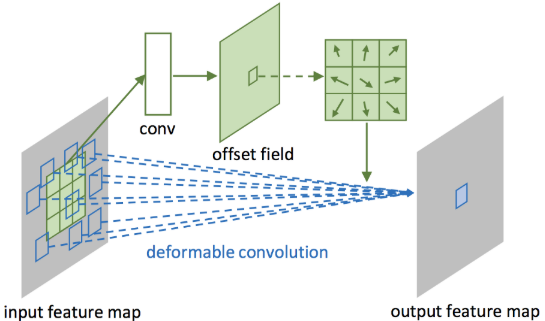
**3. Key Point**

- Deformable Convolution

Sampling Grid에 2D Offset을 더한 것. CNN의 고정된 Gird 대비(a)대비 (b),(c),(d)와 같이 패턴을 변형 시킬 수 있음.



Deformable convolution은 기존 convolution대비 하나의 convolution이 더 있음(빨간 원).



Regular grid R은 다음과 같음.



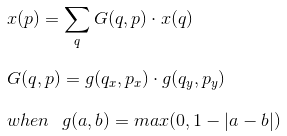
임의의 픽셀 [https://camo.githubusercontent.com/39cc33ff86e1dc7bba25119a120deb9911bb7059/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f705f30](https://camo.githubusercontent.com/39cc33ff86e1dc7bba25119a120deb9911bb7059/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f705f30)를 중심으로 하는 3 x 3 Convolution 연산을 다음과 같이 정의.



각 픽셀을 일정 Offset만큼 이동. 얼마나 이동할지는 [https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e](https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e)이 다음과 같이 결정



단순하게 [https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e](https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e)만큼 해당 Offset 방향으로 더 이동(여기서 [https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e](https://camo.githubusercontent.com/abbe3d96a8f7726c18c75f56dfc013cb62b18d3d/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f25354344656c7461253230705f6e)은 실수). 픽셀 위치에 실수는 존재할 수 없으므로, Offset의 적용을 쌍선형 보간법(Bilinear Interpolation)으로 수행하였습니다.

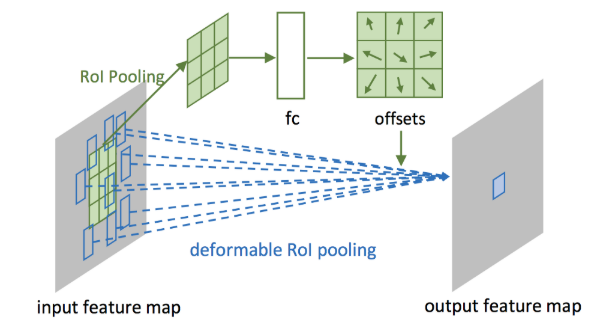
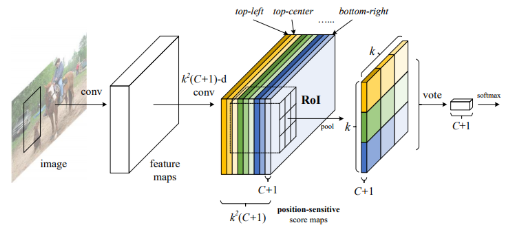


여기서 [https://camo.githubusercontent.com/dccf3897a7546f11b217deec4b2ed7170a094f7e/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f71](https://camo.githubusercontent.com/dccf3897a7546f11b217deec4b2ed7170a094f7e/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f71)는 Feature Map 상의 모든 공간을 순회.

**Deformable RoI Pooling**

- deformable ROI pooling

ROI(region of interest) pooling은 크기가 변하는 사각형 입력 region을 고정된 크기의 feature로 변환하는 과정임. ROI pooling layer와 offset을 학습하기 위한 layer로 구성됨. Deformable convolution과 다른 점은 offset을 학습하는 부분에 convolution이 아니라 fully connected layer가 있는 것임

**4. 결론**

data 특징을 w를 통해서 찾는 것이 아니라 입력 data x에서 직접 찾으려는 시도를 함