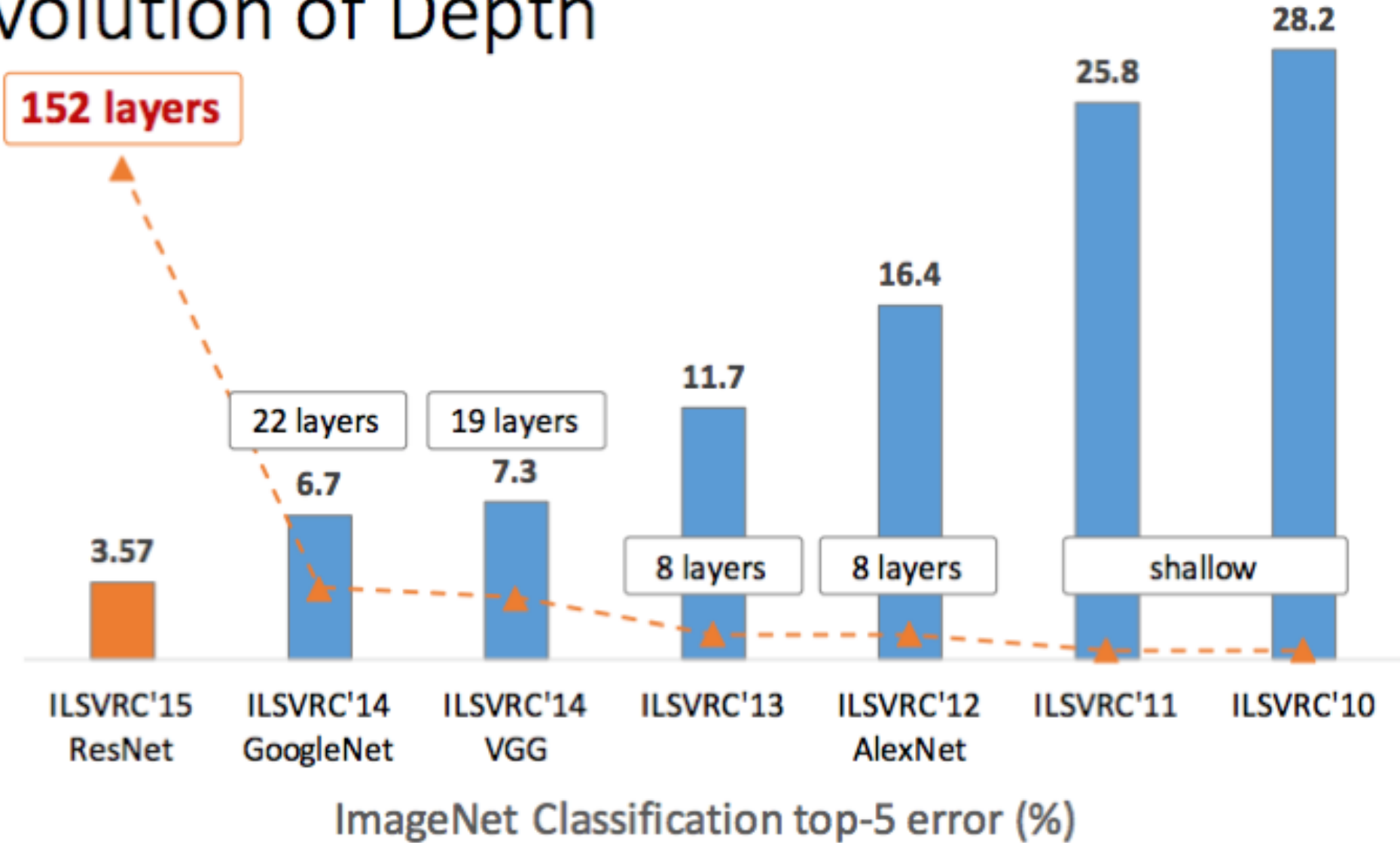
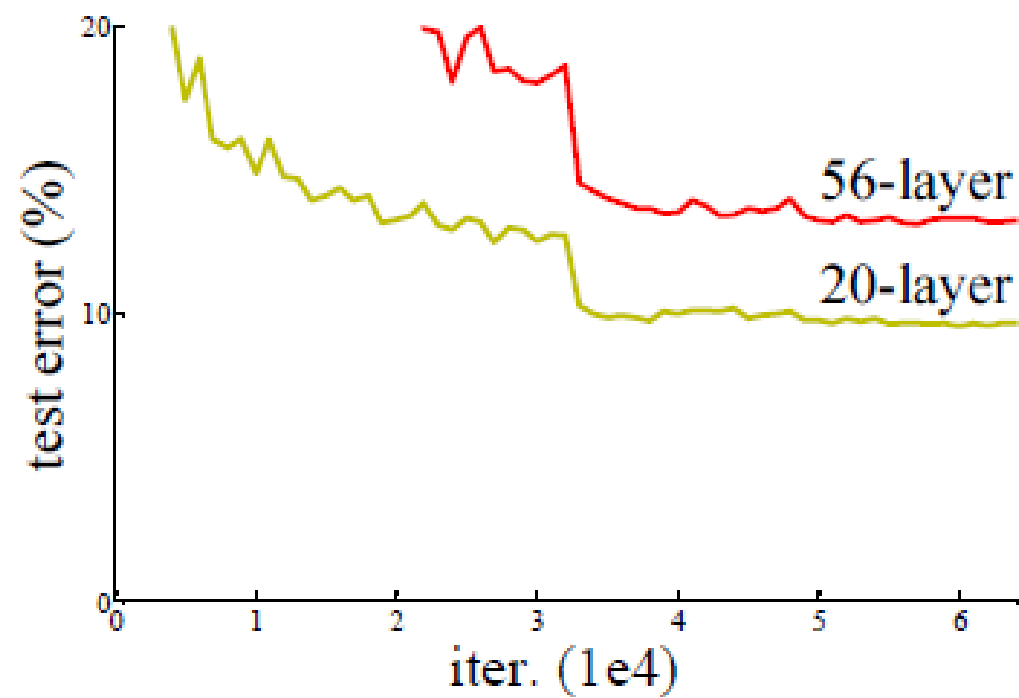
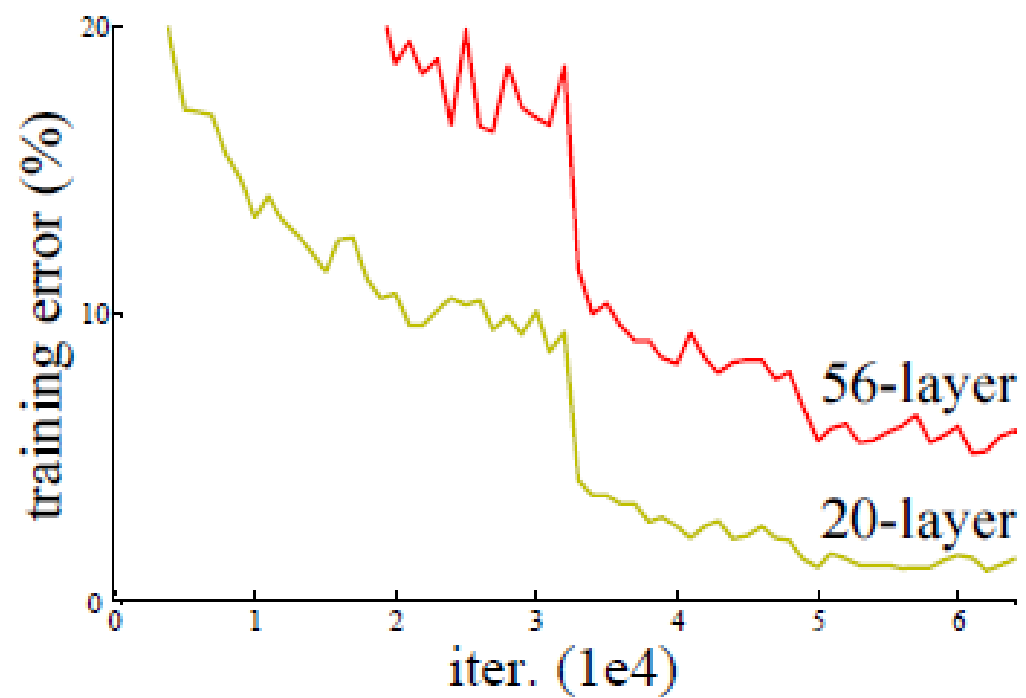
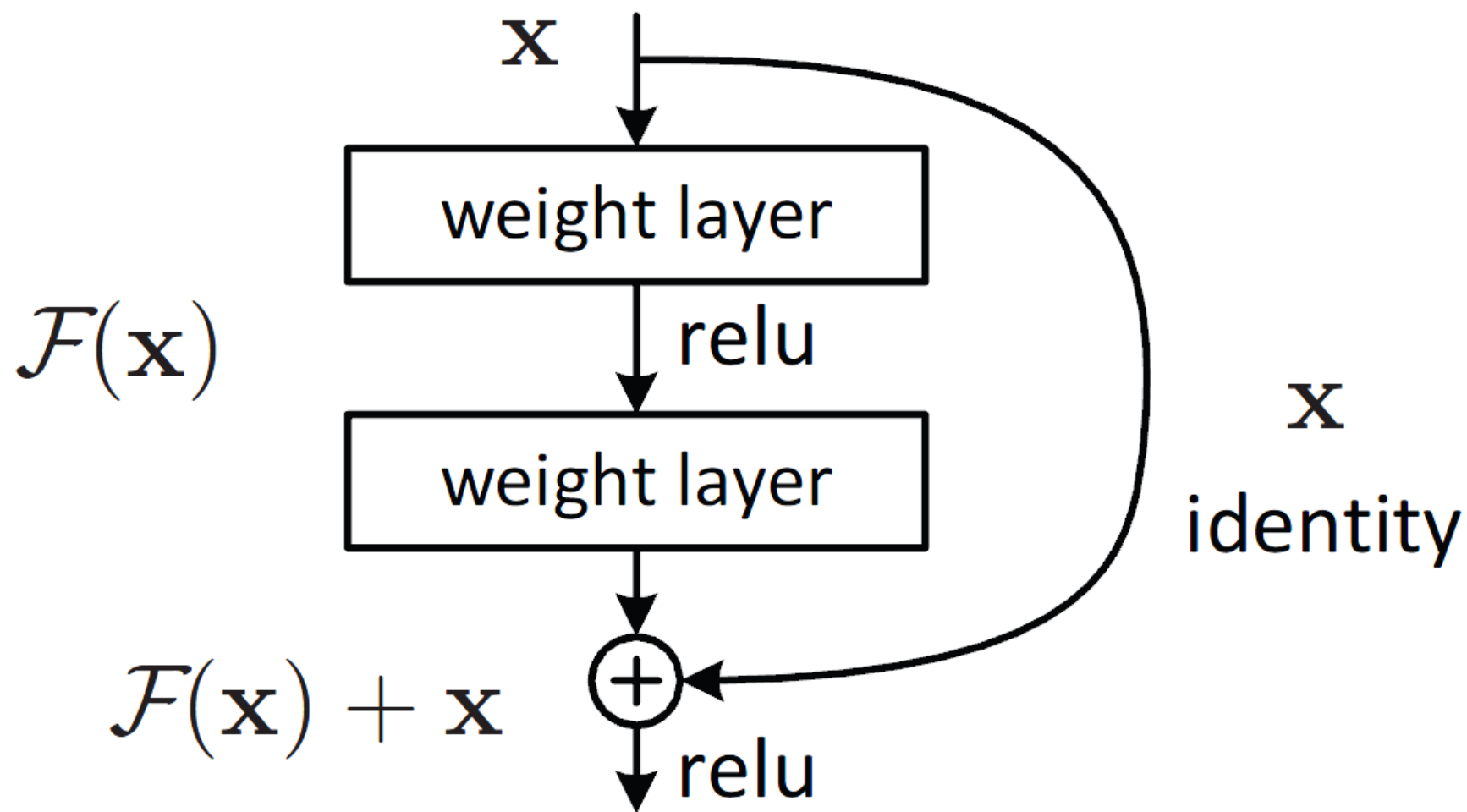


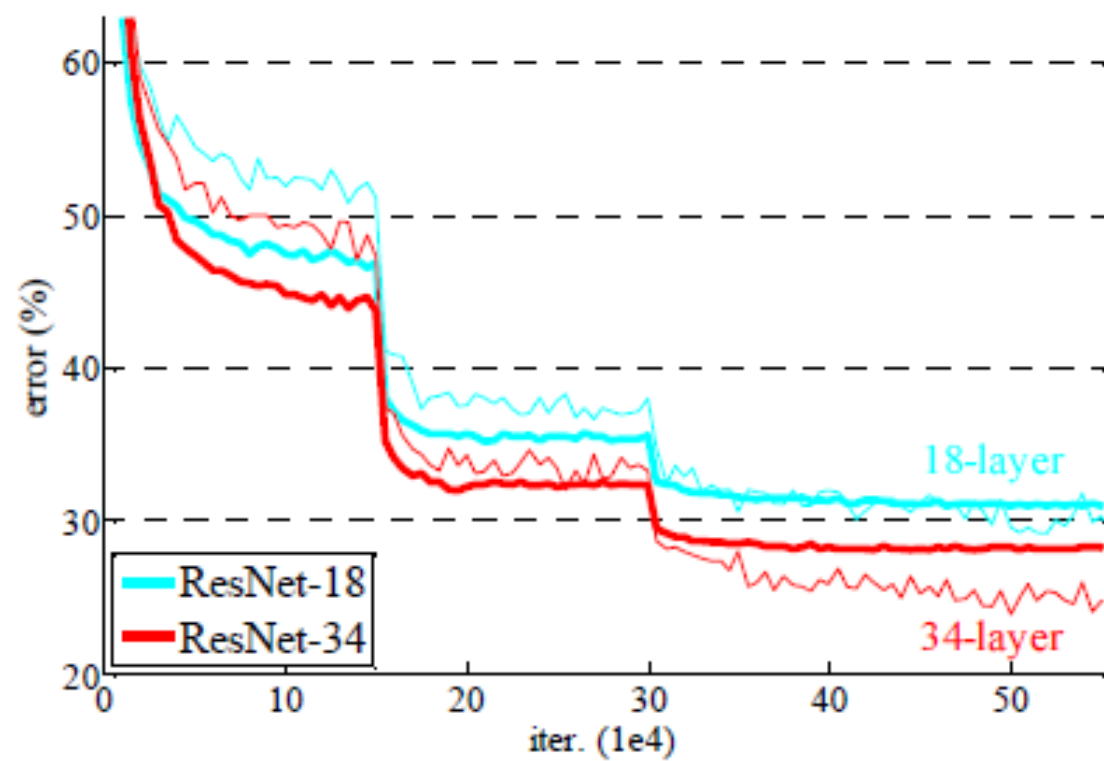
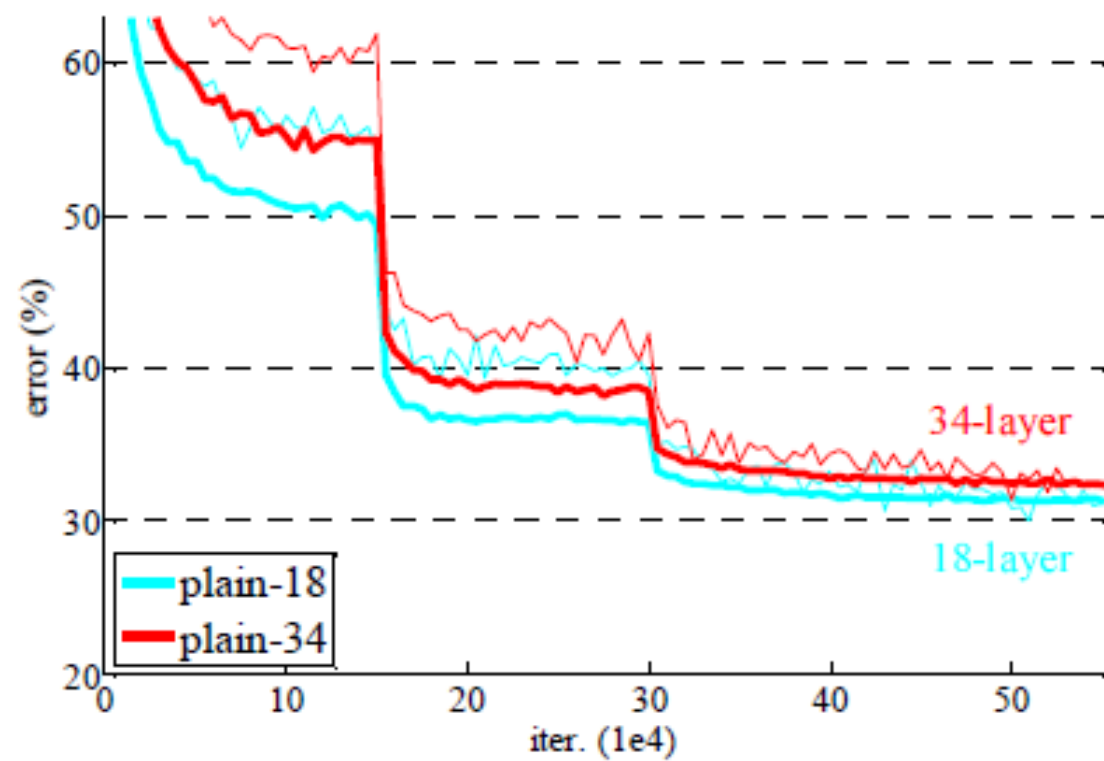
Revolution of Depth

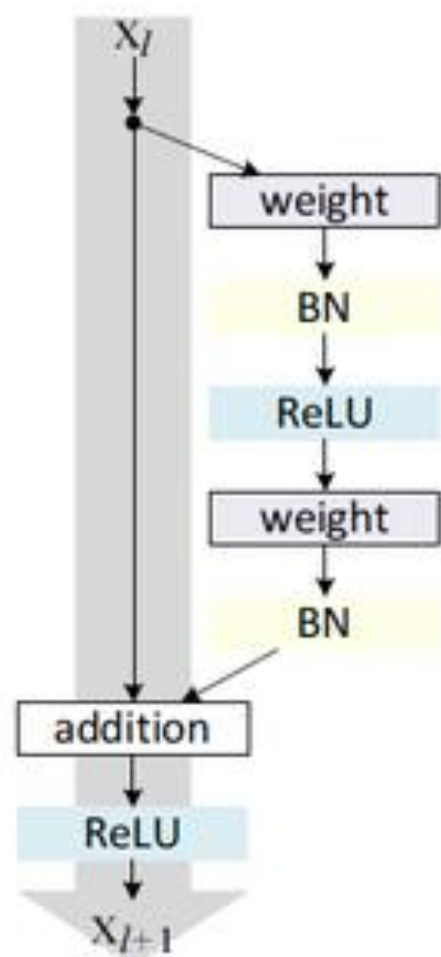


Degradation problem

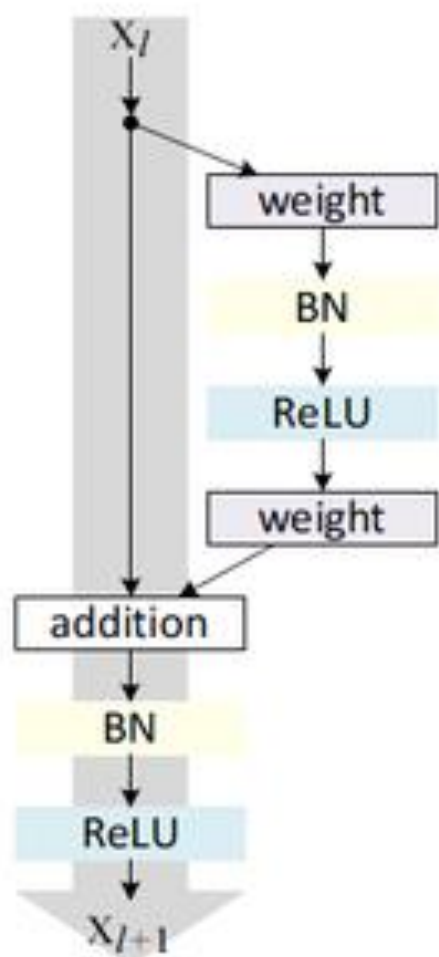




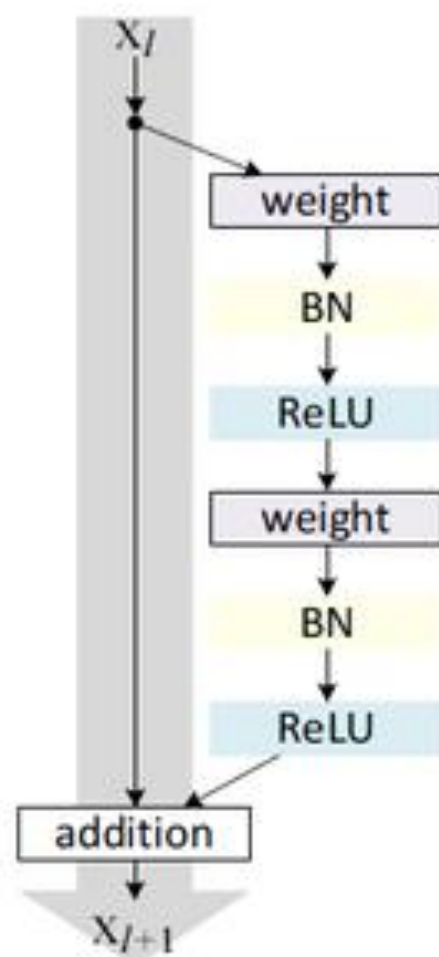




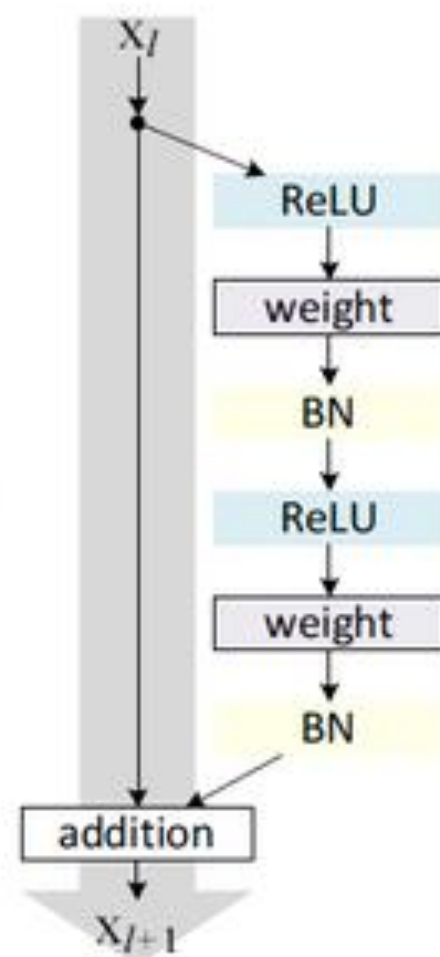
(a) original



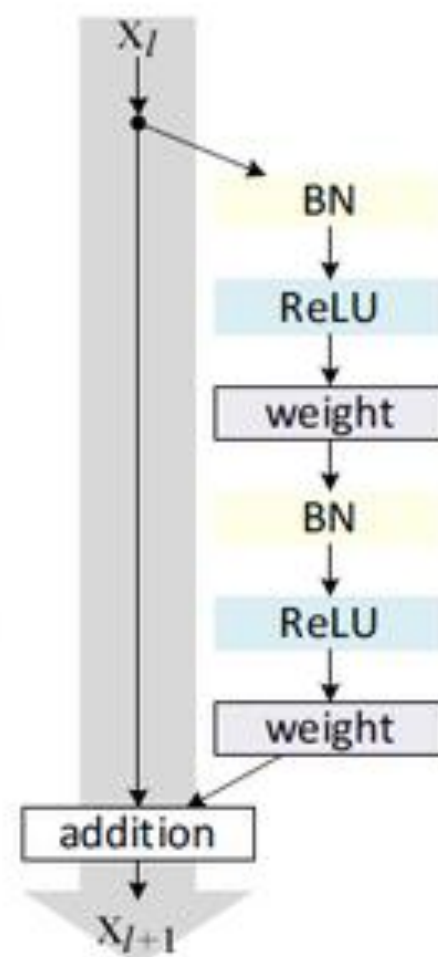
(b) BN after
addition



(c) ReLU before
addition



(d) ReLU-only
pre-activation



(e) full pre-activation

력을 받아 레이블(10개)마다 예측값을 내게 됩니다.

⑥ self._make_layer() 함수를 더 자세히 들여다보겠습니다.

```
def _make_layer(self, planes, num_blocks, stride):
    strides = [stride] + [1]*(num_blocks-1)
    layers = []
    for stride in strides:
        layers.append(BasicBlock(self.in_planes, planes, stride))
        self.in_planes = planes
    return nn.Sequential(*layers)
```

이 함수는 멤버 변수인 self.in_planes 채널 개수로부터 직접 입력받은 인수인 planes 채널 개수만큼을 출력하는 BasicBlock을 생성합니다.

- **layer1** : 16채널에서 16채널을 내보내는 BasicBlock 2개
- **layer2** : 16채널을 받아 32채널을 출력하는 BasicBlock 1개와 32채널에서 32채널을 내보내는 BasicBlock 1개
- **layer3** : 32채널을 받아 64채널을 출력하는 BasicBlock 1개와 64채널에서 64채널을 출력하는 BasicBlock 1개

여기서 주목해야 할 점은 16채널을 받아 32채널을 출력하는 BasicBlock과 32채널을 받아 64채널을 출력하는 BasicBlock입니다. 이렇게 증폭하는 역할을 하는 모듈들은 shortcut 모듈을 따로 갖게 됩니다. shortcut 모듈은 이전 입력을 중간층에 더해 주어 **이미지**의 맥락이 보존될 수 있도록 하는 역할을 합니다.

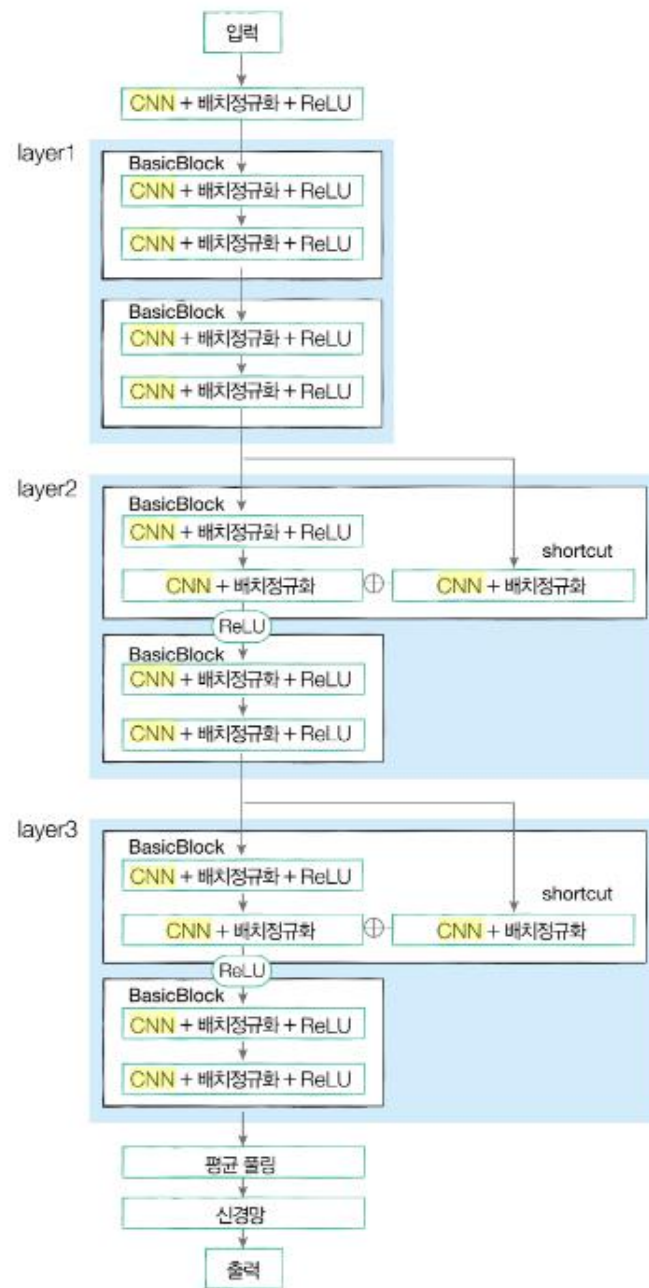


그림 5-16 이번 예제에 쓰이는 ResNet 모델