

# 거대심층구조 신경망

부산대학교 정보컴퓨터공학부

15 학번 이연걸

## 1. 개요

합성곱 신경망은 딥러닝 연구의 핵심적인 주제이다. 이번에는 합성곱 신경망 모델 가운데 '거대 구조'의 대표적 사례인 인셉션 모델과 '깊은 구조'의 대표적 사례인 레스넷 모델을 사용하고 비교한다. 특히 레스넷 모델에서는 보틀넥 모듈의 특징 및 장단점을 살펴본다. 데이터셋은 꽃 이미지를 활용한다.

## 2. 구현 과정

### 2.1 인셉션 모델


인셉션 모델은 인셉션 모듈이라고 하는 병렬 처리 합성곱 신경망 구조를 반복적으로 활용하여 신경망의 규모를 크게 늘린 모델이다. 인셉션 모듈은 하나의 입력을 몇몇 분기에서 병렬 처리한 후 그 결과들을 한데 모아 다음 단계로 전달한다.

이번 실험에서는 특히 인셉션-v3 모델을 사용하면서 'actions'의 옵션값을 'LA'로 지정하여 선형 연산 후에 활성화 함수를 적용하게 된다.

```
In [15]: model_flower_LA.exec_all(report=2)
```

Model model\_flower\_LA train started:  
Epoch 2: cost=1.601, accuracy=0.240/0.230 (1710/1710 secs)  
Epoch 4: cost=1.601, accuracy=0.244/0.220 (1645/3355 secs)  
Epoch 6: cost=1.601, accuracy=0.244/0.230 (1727/5082 secs)  
Epoch 8: cost=1.601, accuracy=0.241/0.220 (1364/6446 secs)  
Epoch 10: cost=1.601, accuracy=0.252/0.230 (1686/8132 secs)  
Model model\_flower\_LA train ended in 8132 secs:  
Model model\_flower\_LA test report: accuracy = 0.242, (79 secs)

Model model\_flower\_LA Visualization



추정확률분포 [18,25,18,17,21] => 추정 dandelion : 정답 dandelion => 0  
추정확률분포 [18,25,18,17,22] => 추정 dandelion : 정답 rose => X  
추정확률분포 [19,24,18,18,22] => 추정 dandelion : 정답 tulip => X

그림 1 - 인셉션-v3 모델을 적용한 경우

## 2.2 레스넷 모델

레스넷 모델은 여러 딥러닝 모델 가운데에서도 특히 깊은 구조의 신경망이다. 레스넷 모델은 매우 많은 수의 레지듀얼 블록을 차례로 연결해 만든다. 여기에서 레지듀얼 블록은 합성곱 계층의 처리 결과에 원래의 입력을 더하는 방법을 이용해 각 합성곱 계층의 학습이 미분 수준의 작은 변화에 집중되도록 만들어주는 간단하면서도 독특한 병렬 처리 구조다.

깊은 구조의 신경망은 학습시키기가 몹시 어려웠고 드롭아웃이나 배치 정규화 등 여러 가지 정규화 기법을 처방해도 어느 한도 이상의 깊이 증가는 오히려 정확도를 떨어뜨리는 결과를 초래하곤 했다. 여러 노력 끝에 레스넷 모델은 이를 극복한 대표적인 사례에 해당한다.

이번 실험에서는 레스넷 모델 중에서 레지듀얼 모델과 이를 변형시킨 보틀넥 모델을 사용하고 결과를 비교해본다.

```
Total parameter count: 173637
Model residual_flower train started:
  Epoch 2: cost=1.227, accuracy=0.490/0.220 (482/482 secs)
  Epoch 4: cost=1.137, accuracy=0.552/0.330 (422/904 secs)
  Epoch 6: cost=1.034, accuracy=0.599/0.460 (437/1341 secs)
  Epoch 8: cost=0.972, accuracy=0.610/0.430 (383/1724 secs)
  Epoch 10: cost=0.947, accuracy=0.631/0.560 (388/2112 secs)
Model residual_flower train ended in 2112 secs:
Model residual_flower test report: accuracy = 0.543, (22 secs)

Model residual_flower Visualization
```



```
추정확률분포 [16,69,10, 0, 4] => 추정 dandelion : 정답 rose => X
추정확률분포 [38,44,10, 4, 4] => 추정 dandelion : 정답 tulip => X
추정확률분포 [ 0, 0, 9, 0,91] => 추정 tulip : 정답 tulip => 0
```


그림 2 – 레지듀얼-34 모델을 적용한 경우

다음은 보틀넥 모듈을 활용한 보틀넥-152 모델의 실험이다. 보틀넥 모듈은 내부 채널 수와 외부채널 수를 달리 하는 방법을 이용해 레지듀얼 모듈과 비슷한 양의 연산을 수행하면서도 외형적인 채널 수를 4 배로 늘려주는 변형된 형태의 레스넷 모듈이다. 깊이가 깊어진 데다가 출력 계층에 제공되는 채널 수가 4 배로 늘어난 탓에 파라미터 수가 크게 늘었다.

하지만 VGG-19 모델과 비교했을 때 거의 1/3 수준이며 플레인-34 모델과 레지듀얼-34 모델보다 높은 정확도를 가지는 장점이 있다.

```
Total parameter count: 189925
Model bottleneck_flower train started:
  Epoch 2: cost=1.191, accuracy=0.513/0.250 (1032/1032 secs)
  Epoch 4: cost=1.064, accuracy=0.581/0.390 (558/1590 secs)
  Epoch 6: cost=0.995, accuracy=0.608/0.370 (480/2070 secs)
  Epoch 8: cost=0.894, accuracy=0.648/0.440 (443/2513 secs)
  Epoch 10: cost=0.836, accuracy=0.677/0.440 (805/3318 secs)
Model bottleneck_flower train ended in 3318 secs:
Model bottleneck_flower test report: accuracy = 0.460, (46 secs)

Model bottleneck_flower Visualization
```



```
추정확률분포 [41,30,28, 0, 0] => 추정 daisy : 정답 daisy => 0
추정확률분포 [46,49, 5, 0, 0] => 추정 dandelion : 정답 daisy => X
추정확률분포 [ 6,33,40, 2,18] => 추정 rose : 정답 tulip => X
```

그림 3 – 보틀넥-152 모델을 적용한 경우

### 3. 총평

	인셉션 모델	레스넷 모델	
	인셉션 - v3	레지듀얼-34	보틀넥-152
정확도	24.2%	54.3%	46.0%

인셉션 모델보다 레스넷 모델이 정확도에서 2 배정도 높은 결과가 나왔다. 인셉션 모델의 경우, 2 시간이 넘도록 학습을 진행했지만 정확도가 24.2%로 상당히 낮게 나왔다. 시각화에서 민들레(dandelion)를 답으로 추정하면서 추정확률 분포가 거의 같았다. 이를 통해 아마도 모든 데이터에 대해 민들레(dandelion)으로 추정하고 있는 상태이고 평가 데이터 셋 안의 민들레 비율이 24.2%일 것임을 예측할 수 있다. 이 결과로부터 학습이 아직 크게 부족한 상태임을 확인할 수 있지만 중간 로그 상태를 보면 cost 가 1.601로 나아지지 않는 상황이다. 따라서 학습 횟수를 늘리더라도 더 나은 결과는 얻어지지 않을 것이다.

레스넷 모델을 살펴본다. 보틀넥 모델의 경우 레지듀얼 모델보다 정확도는 낮지만 cost 가 줄어드는 양상을 보면 오히려 나을 정도로 그에 못지 않는 성능을 보여주었다. 따라서 두 모델은 학습 횟수를 늘리면 더 좋은 품질을 얻을 가능성이 크다. 물론 규모가 커서 학습 시간은 오래 걸릴 것이다.