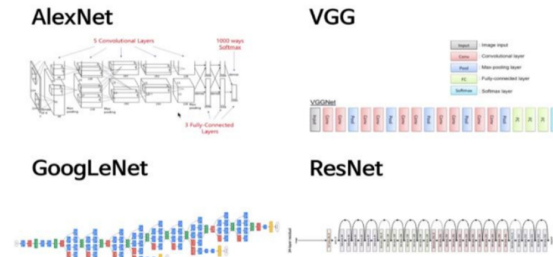
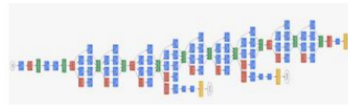


STEP15. 복잡한 계산 그래프 (이론 편)

■ CNN Architectures



■ 복잡하게 연결된 계산 그래프의 미분 계산



[딥러닝 GoogLeNet]

- 지금까지는 일직선 계산 그래프의 미분 계산만 다루었음.

그림 15-1 일직선 계산 그래프



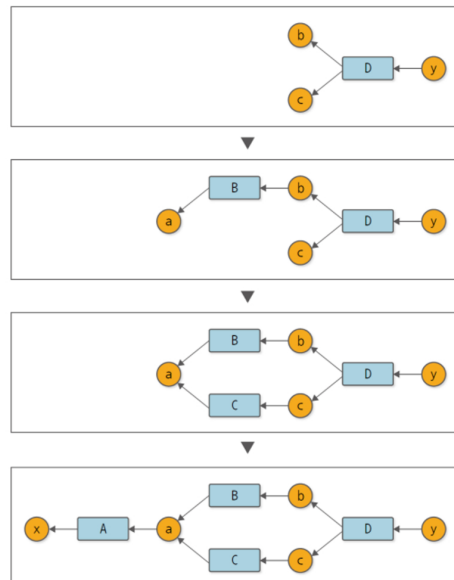
위상 (topology) == 그래프의 연결된 형태
이번 단계의 목표는 다양한 위상의 계산 그래프에 대응하는 것이다.
어떤 모양으로 연결된 계산 그래프라도 제대로 미분할 수 있도록
새로운 아이디어 도입

역전파의 올바른 순서

복잡하게 연결된 그래프의 올바른 순서

- 변수를 반복해서 사용하면 역전파 때는 출력 쪽에서 전파하는 미분값을 더해야 함.

- 변수 a에 2개의 미분값이 모두 전파되어야 함.
- 반드시 지켜야 할 규칙은 함수 B와 C의 역전파를 모두 끝내고 나서 함수 A를 역전파 해야 함.

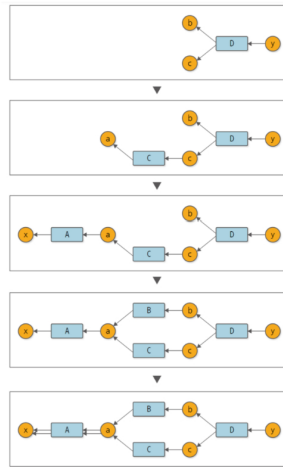


DeZero가 14단계까지 구현 사항 파악

- funcs 리스트 구현 부분 파악 필요
- 역전파의 흐름 파악
 - while 블록에서 처리할 함수의 후보를 funcs 리스트의 끝에 추가함.
- 다음에 처리할 함수를 그 리스트의 끝에서 꺼냄

복잡한 그래프의 미분 계산시의 현 DeZero의 문제점

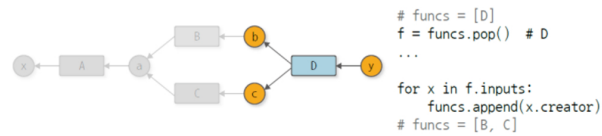
- 함수 순서 처리가 D, C, A, B, A가 됨. C 다음에 A로 바로 이어짐
- 함수 A의 역전파가 두번 일어나는 것도 문제



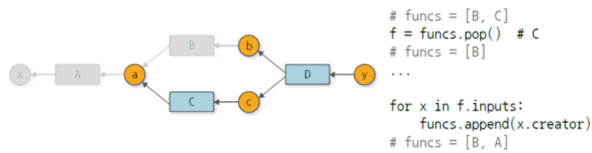
문제 발생 원인

- 한줄로 나열된 계산 그래프를 다룸
- 리스트에서 함수를 꺼내 처리하는 순서를 고려하지 않음

그림 15-6 함수 D의 역전파(오른쪽은 funcs 리스트를 처리하는 코드)



- Funcs 리스트 [B, A]에서 마지막 원소 A가 꺼내짐



함수 우선순위

함수 우선 순위 필요성

- funcs 리스트에서 지금까지 '마지막 원소'만 꺼냄
- 예로 funcs 리스트 [B, A] 에서 출력 쪽에 더 가까운 B를 꺼낼 수 있어야 함

함수 우선 순위 설정

- 순전파 때 '함수' 가 '변수'를 만들어냄. 이는 '부모-자식 관계' 일
- 이 관계를 기준으로 함수와 변수의 '세대(generation)'을 기록함
- '세대' 가 우선순위에 해당함
- 역전파 시 세대수가 큰 쪽부터 처리하면 '부모' 보다 '자식' 이 먼저 처리됨을 보장

그림 15-8 순전파 때의 함수와 변수 '세대'

