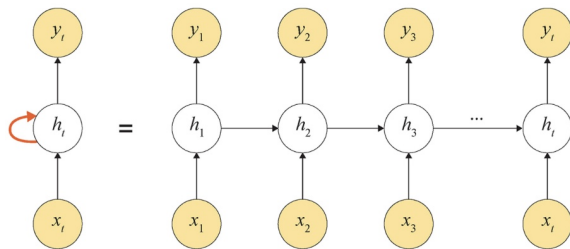




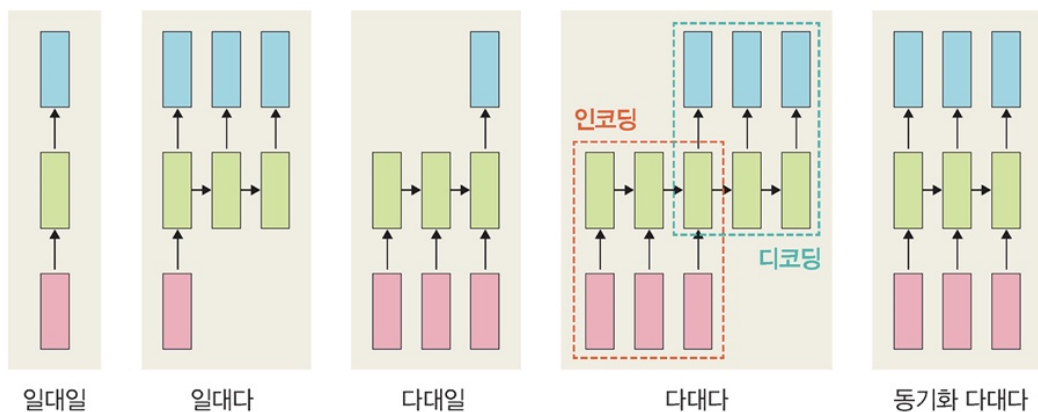
3. 순환 신경망(RNN)

3-0. RNN(Recurrent Neural Network)



- 시간적으로 연속성이 있는 데이터를 처리하기 위한 인공 신경망
 - 이전 은닉층이 현재 은닉층의 입력이 되면서 반복되는 순환 구조를 가짐
 - 기억(= 현재까지의 입력 데이터를 요약한 정보)을 가짐
 - 첫 번째 입력(x_1)이 들어오면 첫 번째 기억(h_1)이 만들어지고, 두 번째 입력(x_2)이 들어오면 기존 기억(h_1)과 새로운 입력을 참고해 새 기억(h_2)을 만듦
- ⇒ 외부 입력과 자신의 이전 상태를 입력받아 현재 상태를 갱신

RNN의 유형



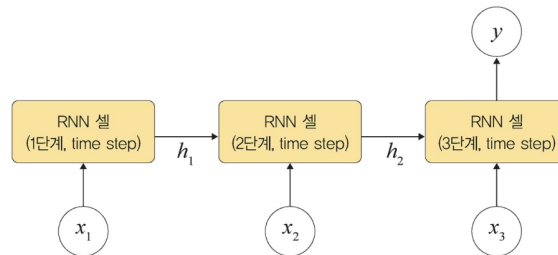
일대일

- 순환이 $x \rightarrow$ RNN이라고 말하기 어려움
- 순방향 네트워크가 대표적임

일대다

- 입력이 하나이고 출력이 다수인 구조
- 이미지 캡션(= 이미지를 입력해 이미지에 대한 설명을 문장으로 출력)이 대표적

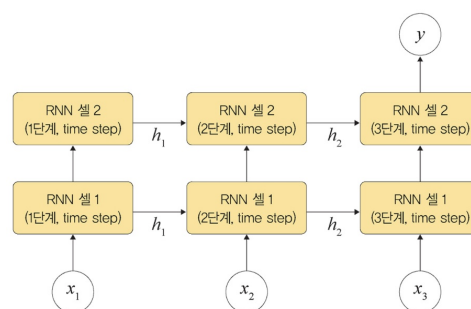
다대일



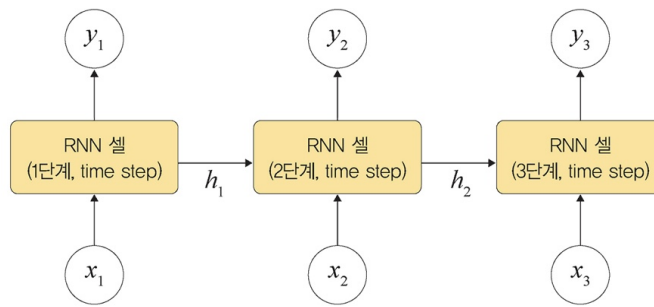
- 입력이 다수이고 출력이 하나인 구조
- 감성 분석(= 문장을 입력해서 긍정/부정을 출력)이 대표적
- 구현

```
self.em = nn.Embedding(len(TEXT.vocab.stoi), embedding_dim)
self.rnn = nn.RNNCell(input_dim, hidden_size) # RNN 적용
self.fc1 = nn.Linear(hidden_size, 256) # 완전연결층
self.fc2 = nn.Linear(256, 3)
```

- 층을 쌓아 올려 적층된 구조로도 구현 가능



다대다



- 입력과 출력이 다수인 구조
- 자동 번역기 등이 대표적
- 파이토치에서는 `seq 2 seq` 방식을 주로 활용

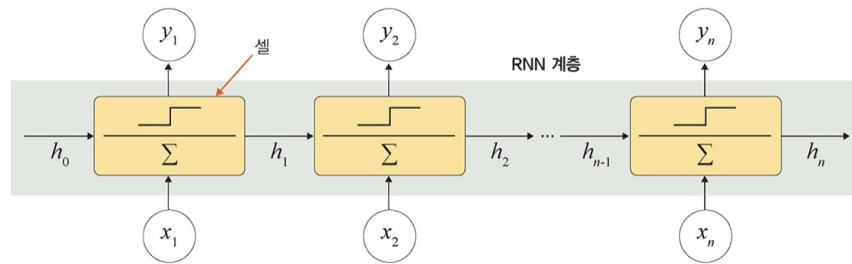
```

Seq2Seq(
    (encoder) : Encoder(
        (embedding) : Embedding(7855, 256)
        (rnn) : LSTM(256, 512, num_layers = 2, dropout =
        (dropout) : Dropout(p = 0.5, inplace = False)
    )
    (decoder) : Decoder(
        (embedding) : Embedding(5893, 256)
        (rnn) : LSTM(256, 512, num_layers = 2, dropout =
        (fc_out) : Linear(in_features = 512, out_feature
        (dropout) : Dropout(p = 0.5, inplace = False)
    )
)
  
```

- 동기화 다대다
 - 4와 같이 입력과 출력이 다수인 구조
 - 언어 모델(= 문장에서 다음에 나올 단어를 예측)이 대표적

3-1. RNN 계층과 셀

- RNN은 내장된 계층뿐만 아니라 셀 레벨의 API도 지원
- RNN 셀은 오직 '하나'의 단계만 처리
 - RNN 셀은 RNN 계층의 for loop 구문을 갖는 구조
 - 실제 계산에 사용되는 RNN 계층의 구성 요소
 - ⇒ 단일 입력과 과거 상태를 가져와 출력과 새로운 상태를 생성



- RNN 계층은 셀을 래핑하여 동일한 셀을 여러 단계에 적용
- 셀 유형
 - nn.RNNCell: SimpleRNN 계층에 대응되는 RNN 셀
 - nn.GRUCell: GRU 계층에 대응되는 GRU 셀
 - nn.LSTMCell: LSTM 계층에 대응되는 LSTM 셀