#26 - Basic of RNN

주제 : Basic of RNN (Recurrent Neural Network)

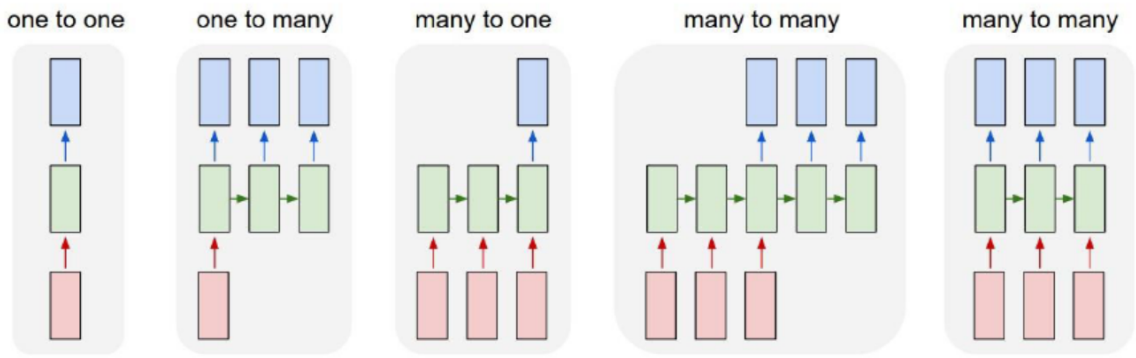
링크 :

[#26.Lec] (강추) Basic of RNN (Recurrent Neural Network) - 딥러닝 홀로서기
강의 자료 링크 : https://github.com/heartcored98/Standalone-DeepLearning/blob/master/Lec7/Lec7-A.pdf자료 저장소 링크 : https://github.com/heartcored98/Standalone-DeepLea...
https://youtu.be/bPRfnlG6dtU

1. Dealing with Sequential Data

- 만약 상사가 다음 4가지 문제를 내주었을때,
  ① 이미지를 주고 캡션을 자동적으로 매기는 것
  ② 기업이 파산할지 안할지 예측
  ③ 문장을 해석해서 다른 언어로 번역
  ④ 문장을 주고 각 토큰의 품사를 알아 맞추는 것 (Multi-label Classification)
  → 이 모든 문제의 특징은, Sequential Data를 다루는 문제라는 공통점을 가지고 있다
  ❀ Sequential Data (순차 데이터) : 말, 텍스트, 이미지 등 순서(Order)가 있는 데이터

- Type of Task Dealing with Sequential Data
  - RNN으로 풀 수 있는 문제는 다음의 4가지이며, 문제를 풀기 전 어떤 유형인지 파악
  - One to One
    - X_1, X_2를 주고 Y를 예측하는 것, 이미지를 주고 무슨 Class인지 예측하는 것
  - One to Many
    ① 이미지를 주고 캡션을 생성하는 문제
    → 이미지를 CNN모델을 통해서 Feature Vector를 생성하여 이미지로 부터 자연어를 매핑
  - Many to One
    ② Bankrupcy될지 안될지를 예측하는 문제
    → Discrete한 Time Step을 보고 Sequence에 대한 Feature Vector를 생성하여 Binary Classification
  - Many to Many
    ③ 문장을 해석하여 (단어1, 단어2, 단어3) 다른 언어로 번역하는 문제
    → 문장의 각 단어에 대한 Feature Vector를 생성하여 다른 언어 스페이스에서 맵핑
  - Many to Many
    ④ 단어$1(W_1)$, 단어$2(W_2)$, 단어$3(W_3)$의 품사를 알맞추는 문제
    → 입력된 Input : 각 단어 (W_1, W_2, W_3)에 대해서 Time Step별로 Task 수행



2. Classical Approach for Time Series Analysis

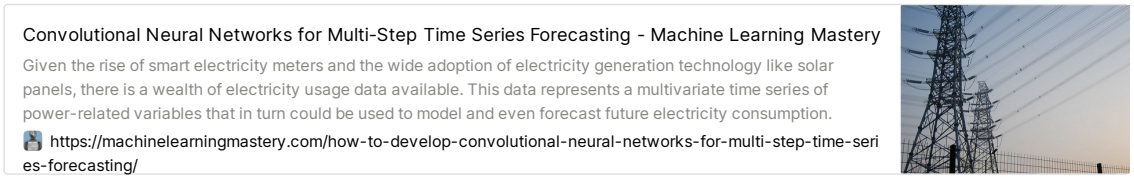
- Time Series Analysis를 푸는 기존의 방식
  - Time domain analysis
    - 파형을 보고, 폭, 움직임, 파형의 높이를 보고 분석
  - Frequency domain analysis
    - Fourier transformation (주파수 대역(스펙트럼) 변형)을 하거나, Wavelets을 하는것
  - Nearest neighbors analysis
    - Dynamic time warping (DTW) 알고리즘, Sequential한 데이터끼리 어떻게 Similarity를 측정할 것인가에 대한 방식
  - Probabilistic Model
    - 주어진 시퀀스에 대해서 그 다음에 나올 값이 무엇인지 확률적으로 예측하는 모델
  - (S)AR(I)MA(X) models
    - 통계적 방식으로 시계열 내에 Auto Correlation이 있는지 확인하는 것
  - Decomposition
    - 전체적인 트렌트 + 주기적으로 변동하는 항목 + 오류의 합으로 Time Series가 만들어져 있다고 가정하여 각각의 항목을 Time Series로부터 분해하는 방식
  - Nonlinear Dynamics
    - 미분 방정식을 활용하여 변화하는 시스템을 편미분하는 모델
  - Machine Learning
    - 위에서 나온 기법들을 통해서 Feature들을 머신러닝 기법으로 Regression, Classification하는 방식

  - Deep Learning은 위와 같은 기존의 Time Series Analysis 방식처럼 Menual로 각각의 Feature를 구하여 분석하는 방식이 아닌, 알아서 Feature를 구하여 Regression, Classification하기 위하여 사용함

3. Deep Learning Dealing with Sequential Data

- 지금까지 배운 Deep Learning 방식으로 Sequential Data 문제를 풀수 있을까?
  - MLP
    - Fully Connected Layers를 켜켜이 쌓아놓는 방식임
    - w matrix가 고정되어 있기에, 따라서 임의의 길이의 sequential를 다루기가 어려움
    - w 차원에 맞추어 resize하던지, 차원을 축소, 확장해야 하기 때문에 정보의 손실이 발생
    - 다양한 파형이 있을 수 있는데, 그 파형을 파악하기 위한 파라메터가 굉장히 많이 필요
  - CNN
    - Convolution + Pooling + Fully Connected Layers를 쌓아놓는 방식

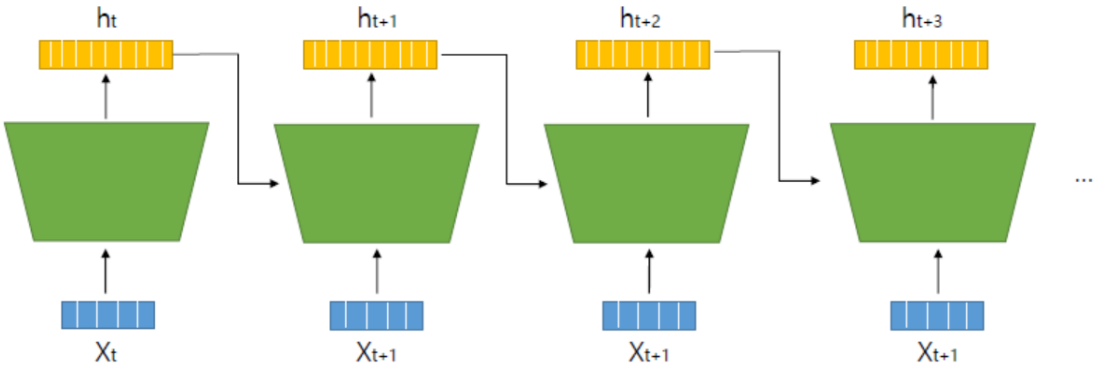
- Time Series Analysis가 수월하게 동작한다 (하단 링크에 설명)



- CNN이 나오기 전, 이런 Sequential Data처리를 위해 RNN을 만들어냈음

4. Recurrent Neural Network

- 개념
  - Sequence Data를 분석할 때 Time Step T에서의 입력된 Input을 처리하고 Output을 내보낼 때, 전후 관계는 고려되지 않는 것 문제가 발생됨
  - **따라서 이전 Step에서 나온 Output과 현재 상태에서의 Input을 함께 고려하여 현재 Step에서 Output을 새롭게 만들자는 것이 핵심**
  - 예시) $InputX_t$를 모델에 넣어 h_t를 Output으로 도출하여 이를 다시 X_{t+1}과 함께 모델의 Input값으로 입력하여 최종적으로 $h_{t+1}, h_{t+2}, h_{t+3}$까지 도출



- 위의 Sequence(전후관계) Featurizing 을 구체적으로 설명하자면,

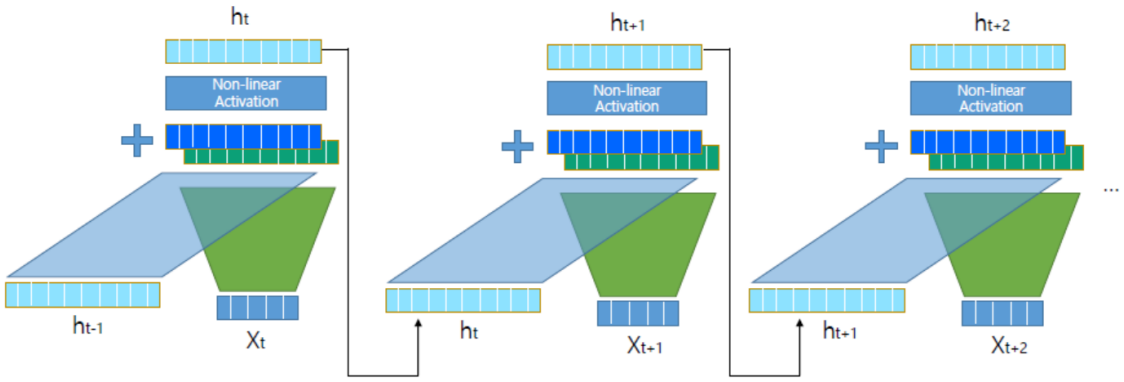
  ① 짙은 초록색 박스 : $InputX_t$를 모델에 통과시켜 벡터를 생성(Linear Transformation하면(=wx)차원이 Projection되는 것과 같은 원리)

  ② 짙은 파란색 박스 : h_{t-1}을 0 또는 랜덤으로 초기화하여 동일한 Dimension(h_{t-1}의 요소가 100개라면 100 x 100)의 Matrix를(=w) 곱해서 해당하는 Hidden Vector를 생성

  ③ 초록 박스 + 파란 박스 : Elementwise(요소끼리) 더하여 이를 Non-Linear Activation Function을(Sigmoid, hipertangent, ReLu 등) 취해준다

  ④ 이 결과값을 Time Step T에서의 h_t로 활용함

  ⑤ ④의 결과값을 활용하여 ①②③④ 순서대로 다시 수행하여 $h_{t+1}, h_{t+2}, h_{t+3}$까지 도출



Q1. h를 바로 사용하지 않고 행렬을 곱해서 사용하는 이유는?

- H값은 이전 Step의 결과물이고, 이를 Non-Linear Activation에 활용할 형태로 변형해서 사용해야 하기 때문에 정방행렬을 곱하여 Non-Linear한 형태로 변환하는 것임

Q2. 입력되는 데이터 길이가 다르다면 어떻게 처리해야 하나요?

- 입력 데이터의 길이가 다르면, 처리가 안될 것이다. 따라서 데이터를 정비하여(Interpolation) 데이터의 형태를 정비한 후 수행해야 한다

Q3. 데이터가 문장이면 어떻게 처리해야 하나요?

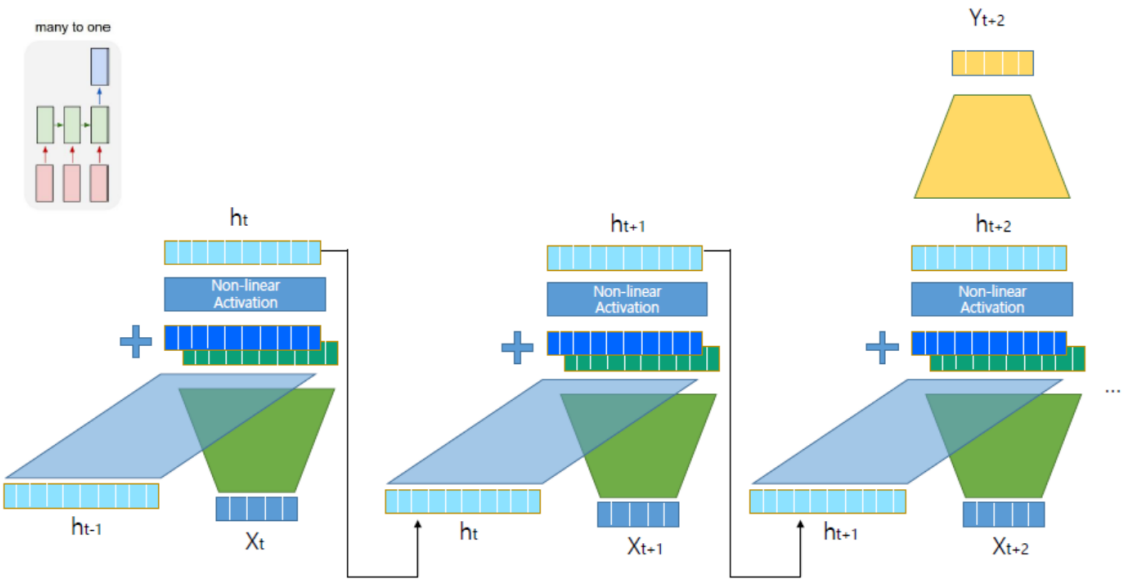
- 전체 문장에 대한 Vocabulary set을 만들어서 각 토큰을 One-Hot Encoding으로 표현

  ex) I Have Apple → I, Have, Apple → Vocabulary Set : 3 → $X_t : 100$ → $X_{t+1} : 010$ → $X_{t+2} : 001$으로 표현

Q4. 문맥상 같은 단어의 의미차이는 어떻게 고려하나요?

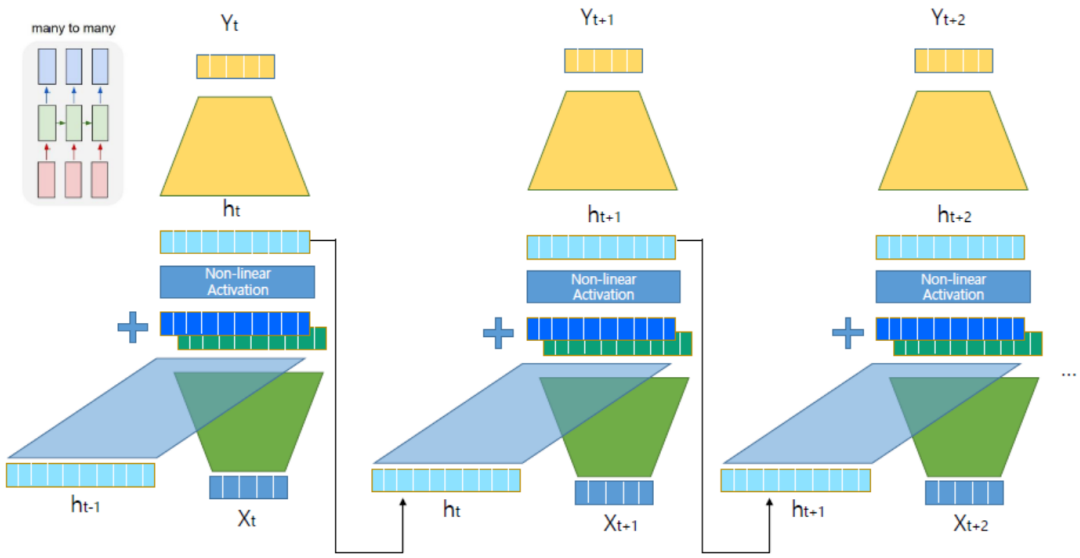
- Contexturizing Embed 방법으로 주위 단어의 맥락을 보고 해당 단어를 Vector를 어떻게 잘 만들것인가 → '엘모어, 트랜스포머, 벌트'의 키워드로 검색해서 공부해보아~

- Many to One
  - Sequence Length가 3인 Sequence라 가정하면, 위의 모델에서 마지막 h에 원하는 모델 하나만 새로 붙이면 됨



- Many to Many
  - 각각의 h_t에 파라메터가 동일한 모델을 씌워서 계산하여 Loss를 바탕으로 업데이트를 한 후, Y_t, Y_{t+1}, Y_{t+2}를 도출

※ 업데이트 : Binary Classification 문제라면 각 모델에서 도출된 Loss를 합산하여 이를 바탕으로 Backpropagation으로 Gradient를 계산하여 Optimizing하면 파라메터가 업데이트됨



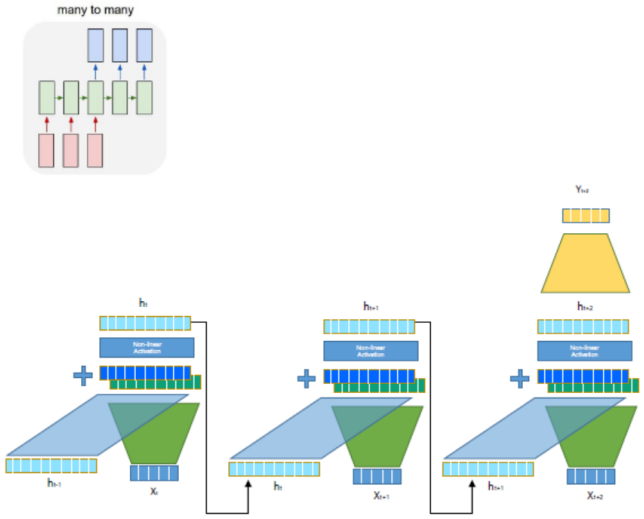
- Many to Many
  - 위의 예에서 다른 언어로 번역하는 과정을 Many to Many로 보자면,
    - "I Have Apple"을 한글로 번역하는 과정으로 예를 들자

      ① Encoding Process

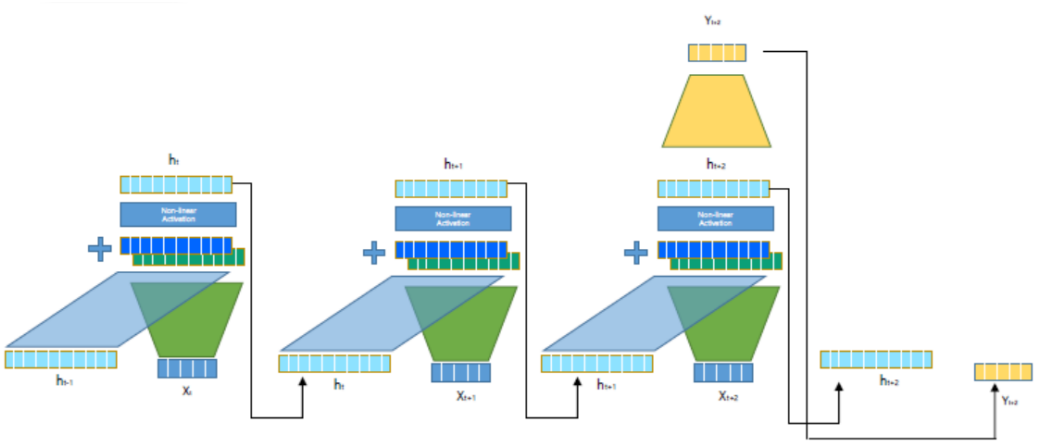
      - "I Have Apple"을 각각 Vocabulary Set으로 분류하여 Feature Vector를 생성하는 과정

      ② Decoding Process

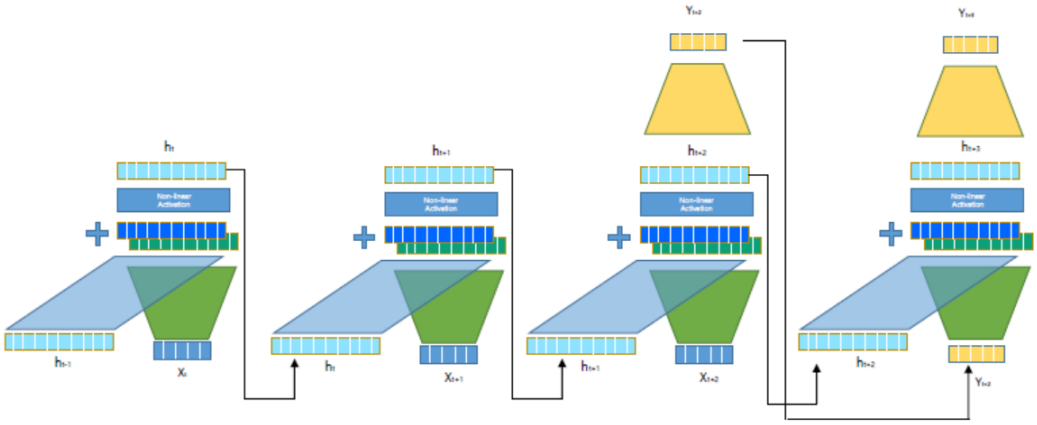
      - 마지막 모델을 통하여 Y_{T+2} = "나는"이라는 단어를 예측



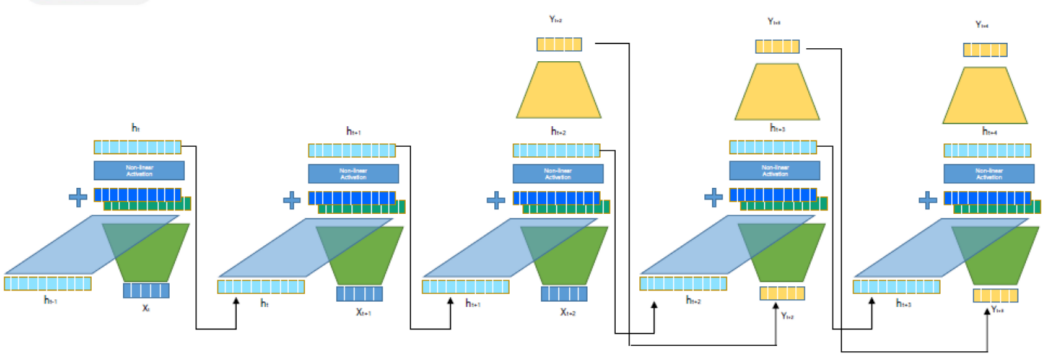
      - 예측한 Y_{T+2}와 이전 단계의 h_{T+2}를 Input 값으로 활용



- 모델을 통하여 다시금 Y_{T+3} = "가졌다"라는 단어를 예측



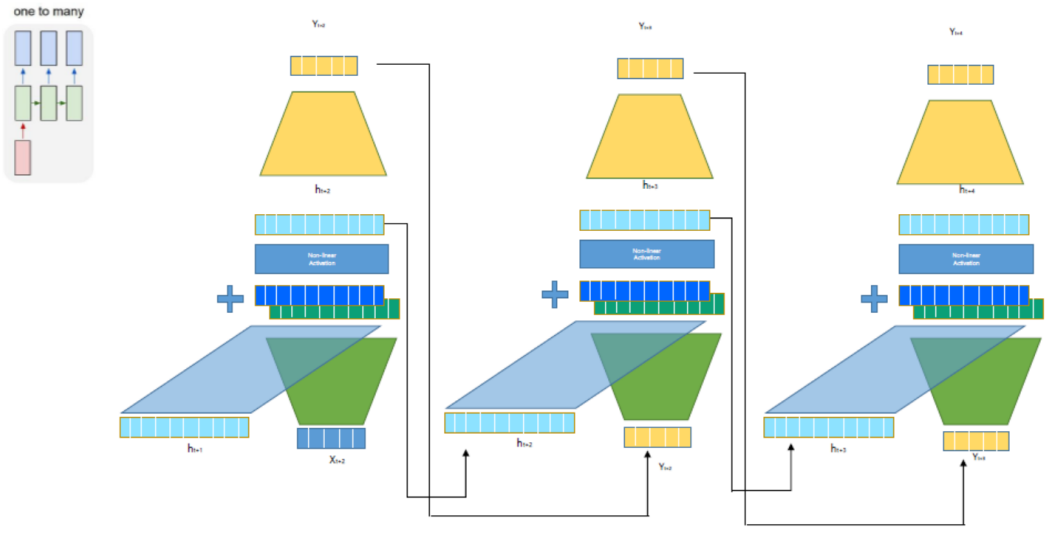
- 마지막 모델을 통하여 Y_{T+4} = "사과"라는 단어를 예측



- 최종적으로 Cross Entropy를 활용하여 True Y와의 Loss를 계산하여 모델을 평가

- 주의점
  - Encoding Process와 Decoding Process에서 사용하는 파라메터가 다르다
  - 영어로부터 Feature Vector를 추출하는 모델과, 영어로부터 한글을 생성시키는 모델이기 때문
  - 이는 곧, h_{t+2}로부터 의미는 공유하되 각각의 Language Space는 각각 다르다는 것이다

- One to Many
  - 이미지가 어떤 클래스인지 알아내는 과정
  - (Encoding) CNN을 통과한 이미지 벡터를 X_{t+2}에입력하여 Decoding과정을 거치면 클래스를 판별할 수 있음



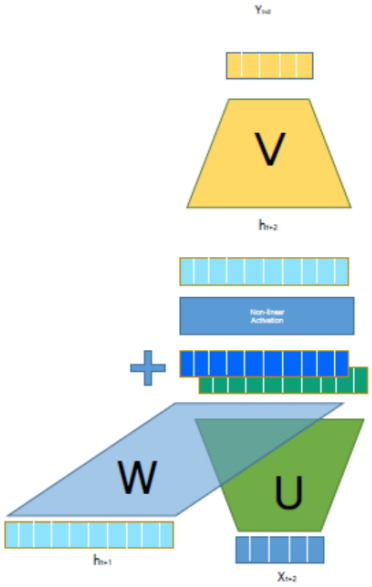
Q. Sequence길이가 다르면 어떻게 처리 하나요?

1) Max Sequence Length를 정해놓고 Sequence를 작성 (Max보다 길면 끊어서)

2) 동시에 Batch 갯수의 Sequence를 처리할 수 있으니 Forward를하다가 먼저 끝나는 순서대로 문장을 리스트에 입력하는 방식 (문장의 앞과 뒤에 (BOS : begin of sentence, EOS : end of sentence 토큰을 넣어줌)

5. Recurrent Neural Network with Math

- 수학적 해석 과정
  - $h_{t+2} = f(Ux_t + Wh_{t-1})$
    - $InputX_{t+2}$을 U로 Projection 한 후, h_{t-1}을 W로 Projection 한 후 Element wise하게 합산
    - Non-Linear Activation Function f를 적용
  - $f(x) = tanh(x)$
    - 위에서 사용한 f 함수는 RNN에서 보통 tanh (x)를 주로 사용함 (-1 ~ 1사이의 값)
  - $y_t = f(Vh_t)$
    - Y는 V라는 파라메터를 통해서 Projection시킨 후 f를 적용
  - $f(x) = x$
    - Y_{t+2}에서 사용한 f 함수는 문제, 기호에 따라서 원하는 모델을 사용 (Linear Regression, Logistic Regression, MLP 등등)



f(x) = tanh(x)

$$h_t = f(Ux_t + Wh_{t-1})$$

$$y_t = f(Vh_t)$$

f(x) = x

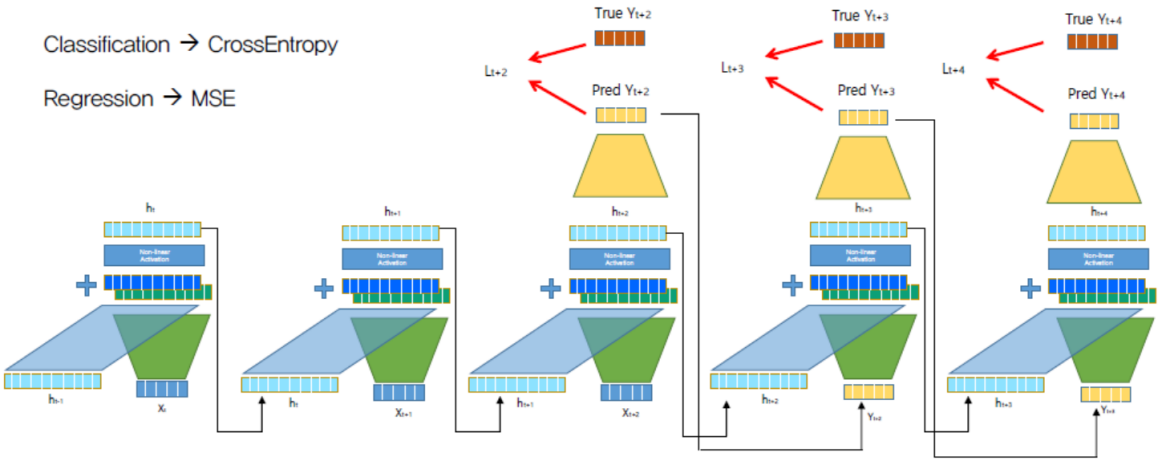
6. How can we evaluate it?

- 평가 방식
  - True Y와 Pred Y의 Loss를 구하여 평균을 구함
  - 각 시퀀스 별로 계산하여 마지막 Loss 값을 통하여 Backward 하면 미분값을 통하여 업데이트
  - 마지막 Task가 Classification이라면 Loss 함수를 CrossEntropy로 사용
  - 마지막 Task가 Regression이라면 Loss 함수를 MSE로 사용

$$Loss(\theta) = \sum_{t} loss(y_{true,t}, y_{pred,t})$$

Classification → CrossEntropy

Regression → MSE



Q. Gradient Vanishing 문제는 어떻게 해결하는가?

- 이를 해결한 모델이 LSTM이다!

7. Summary

- 다양한 Task에서는 Sequential data를 사용해야 한다

- RNN은 이를 쉽게 해결한다
- 기본적으로 RNN은 새로운 Input과 이전 Step의 Output을 활용하는 것이 핵심
- RNN을 활용하여 다양한 Task (one to one, many to one, many to many)를 해결가능