

AI504 17강 정리

- Handling Variable-Length Sequences
 - Image to Label -> 입력 size 고정
 - Sentence to Label -> 입력 size 샘플에 따라 다름
- Bag of Words
 - 문장에 단어가 몇 번 나왔는가 셈
 - 벡터로 나타내려면 모든 단어에 대해 하나의 차원을 필요로 함 -> 차원 매우 큼
 - ◆ 그러나 그 중 실제로 사용되는 차원은 매우 적음
 - 순서를 고려하지 않음
- Recurrent Neural Network (RNN)
 - $h_t = f(Wh_t + Ux_{t-1} + b)$
 - 이전 단계와 현재 입력을 받아서 다음 단계로 출력함
 - 매번 같은 weight 사용
 - 각 timestep 마다 새로운 정보가 추가되는 Feedforward NN이라 볼 수 있음
- RNN의 적용
 - Sequence-level Classification/Regression
 - ◆ 참/거짓 판별, 주제 판별
 - Classification/Regression at each step
 - ◆ Language Modeling: 주어진 문장에서 다음에 오기 적절한 단어 찾기
 - ◆ Part-of-speech tagging: 단어를 보고 품사 맞추기
 - Sequence-to-sequence
 - ◆ 번역, 질문에 답하기
- Sequence-level Classification
 - 참/거짓 판별: "This movie is as impressive as a preschool Christmas play"
 - ◆ 첫 timestep에 x_1 으로 this 입력 - h_1 출력, 두 번째는 x_2 로 movie 입력 - h_2 출력 -> play까지 반복

- ◆ 마지막으로 나온 h_{10} 이 첫 문장의 representation
- Classification/Regression at each step
 - Language Modeling: "This movie is as impressive as a preschool Christmas play"
 - $P(\text{"This movie is as impressive as a preschool Christmas play"})$ <- 매우 작음
 - $\Rightarrow P(\text{This}) \cdot P(\text{movie}|\text{This}) \cdot P(\text{is}|\text{This, movie}) \cdot \dots \cdot P(\text{play}|\text{This, movie, ..., Christmas})$
 - $P(w_t | w_1, w_2, \dots, w_{t-1})$ 처리할 수 있는 모델 필요
 - ◆ \Rightarrow RNN
 - This가 들어오면 movie를, movie가 들어오면 is를 예측하도록 학습
 - x_1 을 첫 입력으로 받음 $\rightarrow \text{Softmax}(W_w h_1 + b_w) \rightarrow y_1' \rightarrow \text{Loss}(y_1, y_1')$ ($y_1 = x_2$)
 - ◆ Bidirectional RNN
 - 위의 RNN의 학습 과정에 더해 x_{10} 을 첫 입력으로 받아 문장을 역방향으로 진행하여 학습을 함
- RNN의 한계
 - Vanishing gradient: long sequence = long backpropagation
 - 먼 과거의 입력을 잊어버림
 - Exploding gradient도 존재
- Gated Recurrent Unit (GRU)
 - Recurrence에 더 복잡한 hidden unit computation 사용
 - Main Idea
 - ◆ Keep around memories to capture long distance dependencies
 - ◆ Allow error messages to flow at different strengths depending on the inputs
 - 기존의 RNN에 각각 다른 weight를 가지는 update gate와 reset gate 추가됨
 - ◆ \Rightarrow 3세트의 weight (RNN은 1세트)
- Long Short Term Memory (LSTM)
 - 더 복잡해진 unit
 - 4세트의 weight(Input, Forget, Output, New Memory Cell)

- h_t 라는 하나의 출력만 있던 RNN이나 GRU와는 달리 두 가지를 출력함

- ◆ Memory Cell, Hidden State(h_t)

- Sequence-to-Sequence

- Variable-length sequence를 입력 받아 Variable-length sequence를 predict(generate)

- 두 개의 RNN 필요

- ◆ 입력과 출력이 모두 Variable-length sequence이기 때문

- Encoder의 출력 = 하나의 h_t

- Decoder는 각 timestep마다 한 번씩 입력을 받아야 함

- => Decoder의 initial state를 h_t 로 하고 각 timestep의 입력은 이전 timestep의 출력으로 함 (일반적인 RNN의 initial state는 0)

- Attention

- 모든 timestep의 hidden layer를 사용해 prediction을 하기

- => Context Vector c

- ◆ $c = a_1h_1 + a_2h_2 + \dots + a_{10}h_{10}$

- ◆ $a_1 + a_2 + \dots + a_{10} = 1$

- ◆ a_i 는 MLP를 사용해 생성됨

- 장점: a_i 는 각 단어의 usefulness를 나타냄

- Seq2Seq에서의 사용

- ◆ 각 timestep에서의 prediction y_i 은 hidden layer s_i 에 의해 결정됨

- ◆ s_i 는 이전 단계의 s_{i-1} , y_{i-1} 그리고 c_i 에 의해 결정됨

- ◆ c_i 는 s_{i-1} 과 $h_{1:T}$ 에 의해 결정됨