# LSTM 기반의 강남역 탑승인원 예측

## 영문제목


**요 약**

한국정보과학회는 정보과학에 관한 기술을 발전, 보급시키고 회원상호간의 친목을 도모하기 위하여 1973 년 3 월 3 일에 설립되었으며, 정보통신부에 '사단법인 한국정보과학회'로 등록되었다. 학회의 주요 활동은 1) 컴퓨터 기술 및 이론에 관한 새로운 연구결과를 발표하는 기회를 제공하고, 2) 국내의 컴퓨터 관련 기술 개발에 참여 하며, 3) 국제적 학술 교류 및 협력 증진을 도모 하고, 4) 회원 상호간의 친목을 증진시키는 것이다.


### 1. 서 론

출퇴근 시간대 혹은 특정 호선들의 차량은 수용률이 150% 이상을 넘어가며, 이는 시간이 지날수록 증가하는 추세이다. 이로인해 현대인들이 지하철 이용에 불편함을 겪고 있다. 광역 버스에는 남은 좌석수가 표시가 되지만 지하철에는 탑승 인원 표시를 지원하지 않는다. 따라서 본 논문에서는 서울 특별시에서 제공받은 지하철 호선별 역별 승·하차 인원 정보 데이터인 시계열 데이터를 예측할 수 있는 방법이 필요하고 시계열 예측에 적합한 LSTM(Long Short-Term Memory)[각주]를 사용하여 강남역의 탑승 인원을 최대한 정확하게 예측하기위해 실험을 하였고, 높은 정확도를 보였다. 또한 실험을 통해 추후 연구 계획을 세웠다.

### 2. 관련 기술

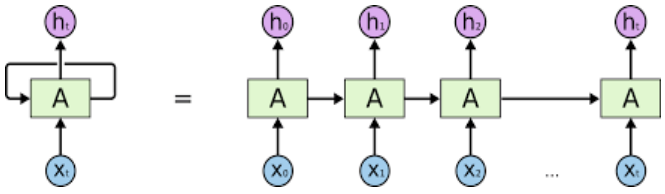



Fig. 1 RNN 모델 [각주]

RNN(Recurrent Neural Networks)[각주]은 Fig 1 과 같이 입력 데이터와 이전 데이터가 함께 다음 입력으로 들어가도록 연결 되어 있다. 따라서 과거의 데이터를 이용해 다음 데이터에 영향을 주는지를 학습할 수 있고, 이러한 특성으로 시계열 데이터의 형태를 지닌 데이터의 학습이 용이하다.

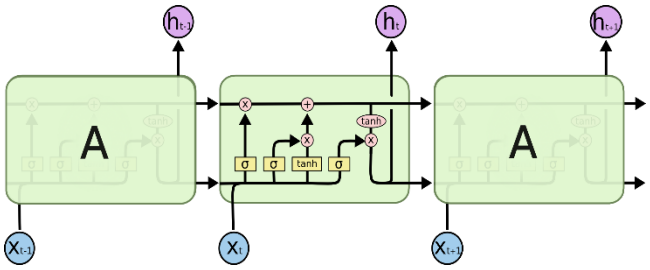



Fig. 2 LSTM 구조 [각주]

LSTM 은 RNN 에서 파생된 모델로써 여러 개의 게이트(gate)가 붙어있는 셀(cell)로 이루어져 있으며 이 셀들이 하는 역할은 정보를 버리고, 저장하고, 업데이트하고, 내보내는 기능들로 이루어져 있다. 각 셀은 셀에 연결된 게이트의 값을 보고 무엇을 저장할지, 언제 정보를 내보낼지, 언제 쓰고 언제 지울지를 결정한다. LSTM 은 이러한 셀 구조를 가짐으로써 RNN 의 대표적인 문제인 기울기 소실 및 발산 문제와 장기 의존성 문제를 해결할 수 있다.

### 3. 데이터셋

서울시에서 운영하는 홈페이지인 정보소통광장 에서 제공하는 서울시 지하철호선별 역별 승하차 인원 정보[각주]를 데이터를 이용하였다. 사용한 데이터의 기간은 2008 년 1 월 1 일부터 2017 년 09 월 30 일 까지이며, 지하철 운행 시간대 별로 구간을 나누어 승차 및 하차 인원을 역별로 나누어 엑셀형태로 저장하였다. 본 연구에서 원하는 예측값은 각 시간대별로 탑승인원을 구하는 것이므로, 승차인원에서 하차인원을 뺀 값에서 시간대 별로 나누어진 데이터를 하나의 일일 데이터로 만들었다.

## 4. 실험환경 및 모델 설정

본 실험에서는 python3.6 과 Keras 를 사용한다. 데이터셋은 탑승인원(승차인원 - 탑승인원)의 값을 입력으로 받는다. 데이터는 출퇴근 시간대에 가장 큰 값을 가지고, 05 시, 11 시 이후에는 급격히 감소하는 것을 알수 있었다. 입력의 범위의 격차가 크다면 Gradient Descent 를 적용하기 까다로워지지만, 데이터를 정규화 하면 쉽고 빠르게 최적화 지점을 찾을수 있다. 따라서 본 논문에서는 원본(raw)데이터를 0 과 1 사이의 값으로 정규화를 실시하였다. 이 중에서 2018 년기준 가장 많은 승·하차 인원을 가진 강남역을 기준으로 삼아 실험을 진행한다. 강남역의 데이터의 총 개수는 71,220 개이며, training 에 80%, test 에 20%의 데이터를 할당하였다. 원본데이터가 1 일을 20 개의 시간으로 나누어져 있기 때문에 lookback 을 20 으로 설정하였다.

데이터의 학습 모델은 LSTM 으로 구성한다. 모델의 성능 평가 척도는 평균 제곱 오차 (MeanSquareError, MSE)를 사용한다. MSE 는 실제 데이터 값과 예측 값의 차이의 제곱을 의미한다. 차이가 크면 클수록 MSE 는 더욱 커지게 된다. 따라서 이 값의 차이를 최대한 줄임으로써 실제 값과 예측 값의 차이를 줄이는 것이 이번 실험의 목표이다.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

위 수식과 같이 입력값 $y_i$에서 예측값 $\tilde{y}_i$를 뺀값을 제곱한뒤 모두 합하여 평균을 낸다.

학습의 loss 는 MeanSquerError, optimizer 는 rmsprop, epochs 는 10, batchsize 는 16 으로 설정하였다. 모델의 마지막에는 하나의 출력값을 내보내기 위해 Dense(1)을 추가하였다

따라서 본 예측 모델의 네트워크를 다음과 같이 구성한다.

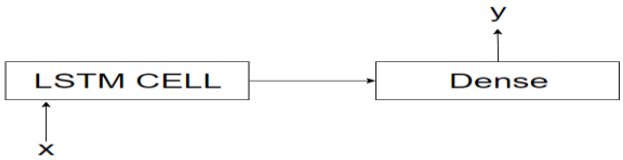



Fig. 3.1 (a) LSTM + Dense(1)

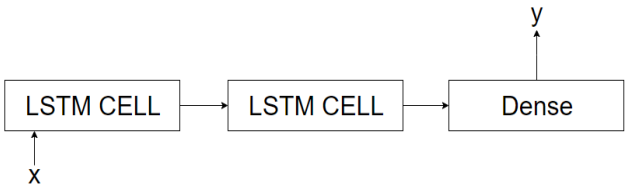



Fig. 3.2 (b) LSTM(512) + LSTM + Dense(1)

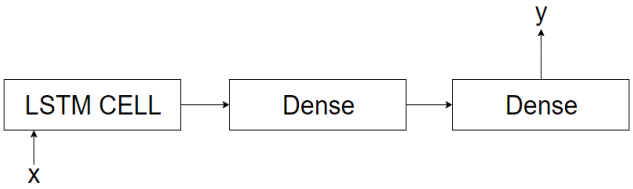



Fig. 3.3 (c) LSTM(512) + Dense + Dense(1)

x 는 탑승인원인 입력 데이터이고, y 는 학습된 모델에서의 탑승인원 예측값이다.

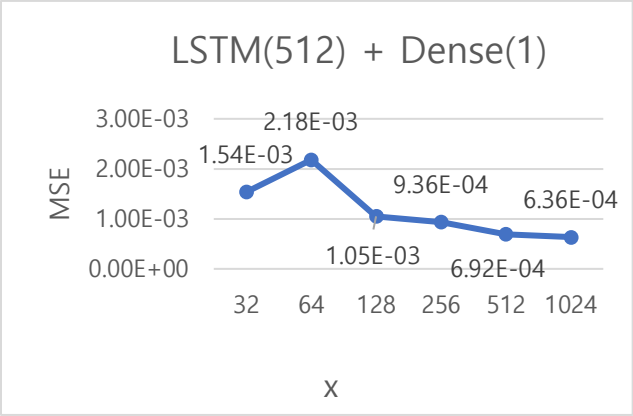
## 5. 실험 및 분석

본 실험에서는 LSTM Layer 를 최소값의 MSE 를 얻을 수 있는 구체적인 구현 모델을 실험하였다.

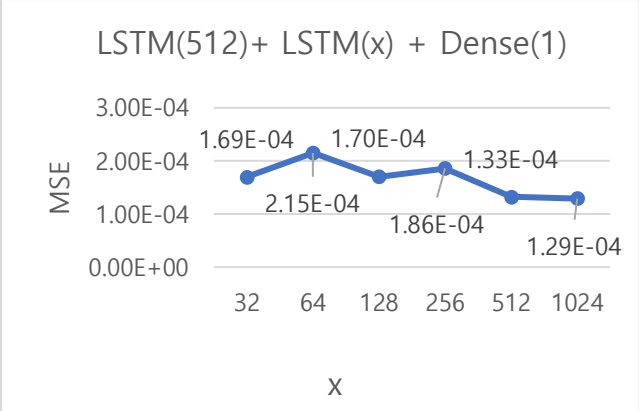
| Model | MSE |
|---|---|
| (a) LSTM + Dense(1) | |
| LSTM(32) | 1.5367935e-3 |
| LSTM(64) | 2.1808513e-3 |
| LSTM(128) | 1.0510251e-3 |
| LSTM(256) | 9.3585515e-4 |
| LSTM(512) | 6.916265e-4 |
| LSTM(1024) | 6.3569e-4 |
| (b) LSTM + LSTM + Dense(1) | |
| LSTM(512)+LSTM(32) | 1.693891e-4 |
| LSTM(512)+LSTM(64) | 2.1497853e-4 |
| LSTM(512)+LSTM(128) | 1.7026807e-4 |
| LSTM(512)+LSTM(256) | 1.861279e-4 |
| LSTM(512)+LSTM(512) | 1.3251218e-4 |
| LSTM(512)+LSTM(1024) | 1.2900409e-4 |
| (c) LSTM + Dense + Dense(1) | |
| LSTM(512)+Dense(32) | 7.3264446e-4 |
| LSTM(512)+ Dense(64) | 6.963257e-4 |
| LSTM(512)+ Dense(128) | 6.466417e-4 |
| LSTM(512)+ Dense(256) | 5.3646386e-4 |
| LSTM(512)+ Dense(512) | 8.9407247e-4 |
| LSTM(512)+ Dense(1024) | 8.351417e-4 |

table. 1 실험 결과표

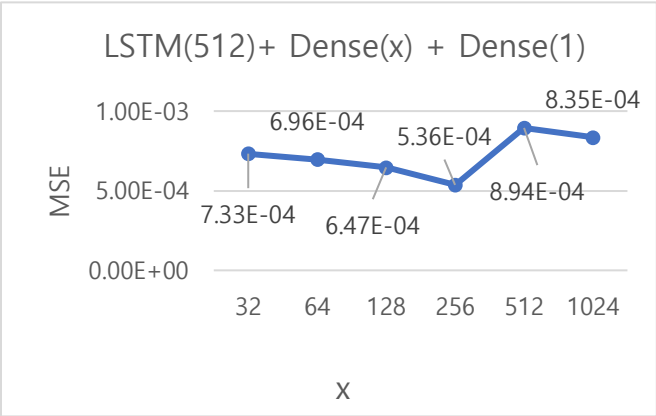
위 결과표를 시각화 하여 각 네트워크를 설명하겠다.




(a)모델의 경우 단순 LSTM layer 만으로도 괄목할만한 성과를 보인다. 하지만 더욱 낮은 오차를 구하기 위하여 Hidden Layer 를 추가하여 실험한다.




(b)모델의 경우 Unit 의 수가 높아질수록 오차가 많이 줄어드는것을 알수 있다. 하지만 Unit 의 개수가 많아질 수록 오차가 크게 떨어지지 않는 것을 확인할 수 있다. 이는 입력 데이터의 크기에 비해 모델이 너무 복잡하여 과도한 파라미터를 가지므로 오히려 학습 능력이 줄어든다는것을 알 수 있다.




(c)모델의 경우 (b) 모델 보다 오차가 큰 이유는 Dense 의 완전 연결(Fully-Connected)은 시계열 데이터의 예측 정확성이 떨어진다는 것을 의미한다. 그 이유는 과거의 데이터를 고려하지 않았기 때문이다.

## 6. 결론 및 향후 연구

본 논문에서는 강남역의 시간별 탑승인원에 대해 Keras 의 LSTM layer 와 Dense layer 를 이용하여 MSE 를 줄이는 방향으로 연구를 진행하였다.
이번 실험에서는 강남역 하나에 대한 탑승 인원 예측을 진행 했다. 이를 확장시켜 모든 역에 대한 탑승 인원을 예측하면 지하철 수용인원 대비 탑승인원을 계산하여 전 역의 지하철 포화도를 계산할 수 있을 것이다. 이를 통해 이용객들에게 출퇴근 시간대 특정 시점에서의 지하철 포화도를 제공할 수 있으며, 비교적 적은 포화도에서 편안한 지하철 이용을 가능하게 하는 서비스를 만들 수 있다. 많은 사람들이 이러한 포화도를 보고 지하철을 이용한다면 분산효과를 기대할수 있으며 이는 지하철 이용의 불편 감소로 이어지는 긍정적인 영향을 가져올수 있다. 또한 정부기관에서는 미래의 탑승 인원을 예상하여 지하철 시간표를 조정할 수 있고, 예산할당의 지표로 삼을 수 있다. 현재 데이터셋은 시간별로 구간이 나누어진 데이터를 합쳐서 예측한 모델이지만, 학습된 시계열 모델을 선형 회귀모델로 바꾸어 분 단위의 탑승인원 예측을 가능하게 할 수 있다. 또한 LSTM 을 사용했던 모델의 개선을 통해 모델의 정확도를 더욱 높일 수 있다.

### 참 고 문 헌

fig1 https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5

fig2 http://colah.github.io/posts/2015-08-Understanding-LSTMs/