딥 러닝을 이용한 COVID-19 확진자 수 예측

Prediction of Number of COVID-19 Confirmed Cases Using Deep Learning

김홍식, Hanyang University, Department of Computer Software

초 록

본 논문에서는 딥 러닝을 이용하여 코로나바이러스감염증-19(COVID-19)의 확진자 수를 예측하는 모델을 제안한다. 이 모델은 특정 국가 또는 지역의 특정 일자 기준 전후 일정 기간 동안의 COVID-19 확진자 수에 대한 데이터를 입력으로, 해당 일자로부터 일정 기간이 지난 후의 날짜의 COVID-19 확진자 수를 출력으로 하여, 과거 및 현재의 확진자 수에 해당하는 입력 데이터가 주어졌을 때 미래의 확진자 수에 해당하는 출력 데이터를 예측하는 인공신경망(Artificial Neural Network, ANN) 모델이다. 이 모델이 미래의 COVID-19 확진자 수 예측에 유효하다는 것을 보인다.

키워드: Artificial Neural Network (ANN), Activation Function (활성화 함수)

1. 서론

2020년 초 전 세계를 강타한 이슈 중 가장 큰 것은 아마도 코로나바이러스감염증-19(COVID-19)일 것이다. 이로 인해 사회의 각종 일정이 연기되거나 취소되고, 각종 경제 지표가 추락하는 등 사회 전반적으로 악영향이 생기고 있다. 전 세계적으로 확진자 수가 증가하는 추세에 있는 가운데, 그 추세를 예측하는 것은 COVID-19의 종식을 예상하는 데 있어서, 더 나아가 전 세계의 사회 및 경제가 언제 정상화될지를 예측하는 데 있어서 매우 중요한 것으로 여겨지고 있다. 확진자 수의 증가 추세는 전염병이 특정 지역에 퍼지기 시작하는 초기 단계의 경우 느리게 증가하면서 증가 속도가 점점 빨라지는 추세를 보인다. 그러다가 전파 속도가 빠른 중간 단계에서 증가 추세가 가장 빠르며, 시간을 x축, 확진자 수를 y축으로 하는 함수의 변곡점, 즉 도함수의 극대점은 이 지점을 전후하여 존재한다. 방역 체계가 구축된 후반에 이르러서는 전파 속도가 느려진다. 이러한 추세적 특징은 특정 단체에 의한 대규모 전염, 해당 국가 또는 지역의 COVID-19 확진 여부 판정 기준 변경 등 돌발적인 변수가 생기거나 데이터가 조작되지 않는 한 유효하므로, 이 특징을 학습하면 미래의 확진자 수를 예측할 수 있다.

본 논문에서는 과거 및 현재의 COVID-19 확진자 수를 입력으로 하고, 미래의 확진자 수를 출력으로 하는 인공신경망을 이용한다. 특정 지역의 일정 기간의 COVID-19 확진자 수 데이터에 대해, 그 데이터의 각 값을 해당 기간의 중간 시점의 일정 날짜의 확진자 수로 나눈 값을 선형적으로 가공한 데이터를 입력 데이터로, 입력값 데이터가 나타내는 기간 중 마지막 날로부터 일정 기간이 지난 날짜의 확진자 수를 입력값 데이터가 나타내는 기간의 마지막 날짜의 확진자 수로 나눈 값을 선형적으로 가공한 데이터를 출력 데이터로 하여 인공신경망을 이용하여 학습한 후, 입력 데이터와 같은 형태의 데이터가 주어졌을 때 미래의 확진자 수에 해당하는 출력 데이터를 예측한다. 이때 입력 데이터는 N일간의 확진자 수 데이터가 입력으로 주어진다고 할 때 N의 크기를 가진 1차원 배열이고, 출력 데이터는 1개의 실수 값이다.

2. 관련 연구

성균관대학교 물리학과 교수팀은 2020년 2월 12일경 로지스틱 모형(Logistic model)을 이용하여 중국의 확진자 수 데이터를 기반으로 그 추이를 예측하였다. 로지스틱 모형이란 추이 데이터가 특정 미분방정식을 따른다고 가정하고 미래의 추이를 예측하는 모형이다. [1]

COVID-19 Novel Coronavirus EDA & Forecasting Cases [2] 에서는 Prophet이라는 모델을 이용하여 전 세계 확진자 수 증가 추이를 예측했는데, 이것은 연도별, 주별, 날짜별, 계절별, 그리고 휴일에 따른 효과를 이용하여 데이터 추이를 예측하는 모델이다.

COVID-19:Analysis+ EDA+ Forecasting [3] 에서는 다음 날의 증감 폭을 전 날의 증감 폭과 동일하게 예측하는 Naïve approach, 최근 30일간의 이동평균을 이용하는 방법인 Moving average, 데이터의 고수준 트렌드를 이용하여 최적의 linear equation을 찾는 Holt linear, 그리고 average smoothing 방법에 따라 서로 다른 smoothing 방법을 사용하는 Exponential smoothing, 마지막으로 ARIMA (Autoregressive integrated moving average)라는 방법을 사용하여 확진자 수와 사망자 수의 증감을 예측하였다.

COVID-19 Visualizations, Predictions, Forecasting [4] 에서는 Linear Regression, SVM(Support Vector Machine) Regressor, Holt's Linear Model, Holt's Winter Model, AR(Auto Regressive) Model,

MA(Moving Average) Model, ARIMA Model, 그리고 Facebook's Prophet Model을 이용하여 COVID-19 확진자 수의 증가 추이를 예측하였는데, Root Mean Squared Error의 값은 ARIMA 모델이 가장 작았으며 Linear Regression이 가장 컸다.

3. 딥 러닝을 활용한 COVID-19 확진자 수 예측 모델

3.1. Overview

본 논문에 사용된 raw data는 Novel Corona Virus 2019 Dataset [5] (Version 42) 의 time_series_covid_19_confirmed.csv 파일에 있는 데이터이다. 이 데이터에서 일정 기간 동안의 확진자 수가 일정 값 이상인 경우의 해당 기간 동안의 데이터를 모두 추출하여 입력 데이터 및 출력 데이터로 지정하였다. 이들 중 비교적 과거에 해당하는 N일간의 일정 기간 데이터를 가공한 데이터를 입력 데이터로, 가장 나중 날짜의 데이터 값을 가공한 데이터를 출력 데이터로 지정하였다. 인공신경망은 크기가 N인 입력 데이터를 입력, 크기가 1인 출력 데이터를 출력으로 한다.

3.2. Data preprocessing

입력 데이터로 주어진, raw data에서의 연속된 N일간의 확진자 수 데이터를 $\{C_0, C_1, \dots, C_{N-1}\}$, 이들 중 기준이 되는 데이터를 K번째 데이터로 하여 그 데이터를 C_K $(0 \le K \le N-1)$, raw data에서 확진자 수 데이터가 나타내는 날짜의 마지막 날짜부터 P일이 경과한 날짜의 확진자 수 데이터를 출력 데이터로 하여 그 데이터를 C_{after_P}라고 하면 인공신경망의 입력 데이터와 출력 데이터는 다음과 같다.

입력 데이터:

$$\left\{\frac{C_0}{C_K}-1, \frac{C_1}{C_K}-1, \dots, \frac{C_{N-1}}{C_K}-1\right\}$$

출력 데이터:

$$Sigmoid\left(Scaling\left(\frac{C_{after\_P}}{C_{N-1}}-1\right)\right) \quad \dots (1)$$

$$\text{where } Sigmoid(x) = \frac{1}{1+e^{-x}}$$

이때 출력 데이터에 Sigmoid 함수만 적용한 채로 학습하면 학습 효율이 저하되는 문제가 발생하기 때문에, 딥 러닝에서 activation function을 지정하는 것처럼 출력 데이터에 대한 Scaling을 해야 하는데, 이를 위한 Scaling 함수는 다음과 같다.

$$Scaling(x) = x \cdot scalingRate - 1 \quad \dots (2)$$

where $scalingRate$ is an appropriate value of weight to scale output values of model

Raw data에서 신경망의 입력 및 출력으로 지정할 각 데이터는 <Algorithm 1>의 알고리즘을 이용하여 추출한다.

```
R := numRegion // 전체 데이터의 지역 개수
D := numDate // 전체 데이터의 날짜 개수
N := numTrainDate // 학습할 날짜 개수
C := minimumC // 최소 확진자 수

confirmData := data[R][D] // 확진자 수 데이터
inputData := data[][N] // 입력 데이터
outputData := data[][1] // 출력 데이터

For r in R:
  For d in D:
    validDs := Find All d that all values in
    confirmData[R][d:d+N-1+P] is larger
    than or equal to C

    For validD in validDs:
      Append confirmData[R][validD:validD+
      N-1] to inputData // 입력 데이터 지정
      Append confirmData[R][validD+N+P-
      1] to outputData // 출력 데이터 지정
```

<Algorithm 1> raw data에서 입력 및 출력 데이터로 활용할 데이터를 추출하는 알고리즘

3.3. Training

인공신경망은 크기가 N인 각 입력 데이터를 입력으로, 이에 대응되는 크기가 1인 각 출력 데이터를 출력으로 하여 학습하며, 중간에 4개의 은닉층이 있다. 각 은닉층은 모두 크기가 32이며, 활성화 함수로 ReLU 함수 [6]를 사용하는데, 이것은 수식 (3)으로 나타내어진다.

$$ReLU(x) = \max(0, x) \quad \dots (3)$$

출력층은 활성화 함수로 sigmoid 함수를 사용한다. 또한, 신경망은 Adam optimizer [7]를 사용하며, 이때 learning rate = 0.0005이다. 이러한 신경망의 구조를 표로 나타내면 <Table 1>과 같다.

Layer	Neuron #	Activation Function	Parameter #
Input	N		
Hidden 0	32	ReLU	(N+1)*32
Hidden 1	32	ReLU	1056
Hidden 2	32	ReLU	1056
Hidden 3	32	ReLU	1056
Output	1	Sigmoid	33

<Table 1> COVID-19 확진자 수 예측을 위한 인공신경망의 구조

4. 실험 결과

4.1. Parameters

N	15
K	7
P	1
C	50
scalingRate	All cases in $\{2^k, 1.25*2^k, 1.5*2^k, 1.75*2^k\}$ $where$ $0 \le k \le 4$
epoch	0, 1, 2, 25, 50, 100

<Table 2> 실험을 위한 Parameter Setting

실험을 위해서 3.2.에서 언급한 parameter 값을 <Table 2>와 같이 정한다. 기준 시점의 확진자 수에 대한 그 1일 후의 확진자 수의 증가율은 P=1일 때 수식 (4)로 나타내어지는데, 대부분의 지역에서 그 값은 0이거나 0에 가까운 작은 양수일 것이다.

$$\frac{C_{after\_P}}{C_{N-1}} - 1 \qquad \dots (4)$$

따라서 수식 (1), (2)를 참고하면, scalingRate의 값이 1.0과 같이 매우 작으면 Scaling 함수의 결과값이 대부분의 지역에서 음수가 될 것이며, 매우 크면 대부분의 지역에서 양수가 될 것임을 알 수 있다. 이 2가지 경우 모두에서 Sigmoid 함수의 결과값의 범위가 매우 제한적이며, 이 경우 학습을 제대로 하지 못한다. 각 지역에서의 수식 (4)의 값의 평균을 IncRateAvg라 할 때, 수식 (5)를 만족시키면 Scaling 함수의 결과값의 평균값이 0이 되므로 양수인 경우와 음수인 경우가 모두 존재하여 비교적 원활하게 학습이 이루어질 것이다.

$$scalingRate = \frac{1}{IncRateAvg} \qquad \dots (5)$$

또, 여기서 epoch는 인공신경망이 dataset 전체를 1회 학습하는 과정을 말한다. 즉 epoch 수는 신경망의 전체 데이터에 대한 학습 횟수를 의미한다. 여기서 0, 1, 2를 지정한 이유는 학습이 거의 되지 않았을 때의 정확도를 구하기 위한 것이다.

4.2. Test Method

매 회 테스트 시마다 입력 데이터와 출력 데이터의 쌍 각각에 대하여 80%의 확률로 학습 데이터로 지정하고 20%의 확률로 테스트 데이터로 지정하여 학습 및 테스트 데이터를 생성한다. 이것은 학습 데이터와 테스트 데이터의 비율이 대략적으로 80:20 이라는 것을 의미하며, 확률적으로 지정되는 것이기 때문에 정확히 80:20 비율로 지정된다는 것을 의미하지는 않는다.

테스트용 입력 데이터를 인공신경망에 입력시켜서 나오는 출력값을 Y라고 할 때, 이 값을 해당 입력 데이터에 대한 수식 (4)의 값 Y'로 환원시키기 위하여 수식 (6)을 이용한다.

$$Y' = \frac{invSigmoid(Y) + 1}{scalingRate} \qquad \dots (6)$$

$where\ invSigmoid\ is\ inverse\ function\ of\ Sigmoid,$

$$invSigmoid(x) = \ln\frac{x}{1-x}$$

4.3. Metrics

모델의 정확도를 평가하기 위해 사용하는 Metric에는 difAvg, 0difAvg, accuracyRate가 있고, 이들 metric의 값을 구하기 위한 수식에 사용되는 값들로 result, testValOutput이 있다. 본 논문에서는 metric 중 현재 시점의 확진자 수에 대한 P일 후의 확진자 수의 증가율의 절대 오차의 제곱의 평균값인 difAvg의 값을 error 값으로 사용한다.

Result: 각 테스트용 입력 데이터를 신경망에 넣은 결과로 출력되는 출력값을 Y라고 할 때, 이것을 (6)을 이용하여 (4)의 값으로 환원시킨 Y'의 값

testValOutput: 각 테스트용 입력 데이터에 대응되는, raw data에 있는 출력 데이터 C_{after_P}와 그 입력 데이터의 가장 나중 날짜의 확진자 수인 C_{N-1}을 이용할 때의 수식 (4)의 결과값

difAvg: 테스트 데이터의 전체집합 S, S에 속한 각 테스트 데이터 T_i에 대해 result, testValOutput의 값을 각각 R_i, V_i라 할 때 다음 수식의 결과값으로, 오차제곱합의 평균을 의미한다.

$$\frac{1}{|S|}\sum_{T_i \in S}(R_i - V_i)^2 \qquad \dots (7)$$

0difAvg: 수식 (7)에서 모든 테스트 데이터 T_i에 대해 $R_i = 0$을 대입했을 때의 값으로, 모든 테스트 데이터에 대해 result의 값이 0일 때의 오차제곱합의 평균을 나타낸다. 이것은 아무런 학습 없이 다음 날의 확진자 증가율을 0으로 예측했을 때, 즉 현재 날짜와 같다고 예측했을 때의 오차를 의미한다. 전체 dataset의 1318개의 데이터 중 절반 이상인 752개의 데이터에서 수식 (4)의 값이 0이고, 그 평균이 0.0204로 매우 작으므로 학습이 전혀 안 되었을 때 확진자 증가율을 0으로 예측하는 것은 유의미하다.

accuracyRate: 0difAvg와 difAvg의 비율로 수식 (8)로 계산한다. 이 값이 1이면 result의 값을 모두 0으로 예측했을 때와 정확도가 동일하다는 것을 의미하며 사실상 학습이 전혀 되지 않았다는 것을 의미하고, 이 값이 1 초과의 양수이며 전체적으로 학습이 많이 될수록 증가하는 추세이면 유의미하게 학습이 되었다는 것을 의미한다.

$$accuracyRate = \frac{0difAvg}{difAvg} \qquad \dots (8)$$

4.4. Environment for Test

테스트에 사용된 CPU 프로세서는 Intel® Core™ i5-1035G7 CPU @ 1.20GHz 1.50GHz의 x64 기반 프로세서이고 16.0GB RAM을 사용하며, 64-bit OS를 사용하였다. 사용한 운영 체제는 Windows 10 Pro edition이며, Python 3.7.4 버전 및 Tensorflow와 Keras 라이브러리를 이용하여 테스트하였다.

4.5. Test Result

<Figure 1>, <Figure 2>, <Figure 3>에서 각 칸은 scalingRate를 그 칸이 속한 행의 가장 왼쪽 칸의 값으로, epoch를 그 칸이 속한 열의 가장 위쪽 칸의 값으로 지정하여 실험했을 때의 difAvg, 0difAvg, accuracyRate 값을 각각 나타낸다. 단, 여기서 difAvg, 0difAvg 값은 실제 값을 100배 한 값이다. AVG25+ 열은 학습이 거의 되지 않아서 실질적으로 무의미한 epoch 0, 1, 2일 때의 결과를 제외한, epoch 25, 50, 100일 때의 결과값의 평균을 의미한다.

<Figure 1>에 나타난 difAvg 값의 경우, AVG25+ 열의 값은 scalingRate 1.00 ~ 28.00 구간에서 노이즈를 제외하고 큰 변화를 보이지 않으며, 그 값이 최소에 가까운 scalingRate의 값은 4.00 ~ 8.00 내외이다. Epoch 0, 1, 2일 때에 비해 epoch 25, 50, 100일 때의 값이 현저히 작으므로 유의미하게 학습이 되었다고 할 수 있다.

<Figure 2>에 나타난 0difAvg 값은 신경망의 학습과는 무관하고 validation output data의 값에 따라서만 결정되므로 epoch 및 scalingRate의 영향을 받지 않는다. 노이즈는 학습 및 테스트 데이터로 선정된 데이터에 영향을 받아 나타났다. 전체적으로 볼 때 그 평균값은 0.003 ~ 0.004 정도이며, 최솟값은 0.001863, 최댓값은 0.005855로 나타났다.

<Figure 3>에 나타난 accuracyRate 값의 경우 AVG25+ 열의 값이 scalingRate 2.00 ~ 8.00 구간에서 가장 높게 나타나는 것을 알 수 있다. Epoch 0, 1, 2일 때의 값보다 Epoch 25, 50, 100일 때의 값이 현저히 높게 나타났으며, 또한 대부분 2.0 이상의 값을 기록하였고 평균적으로 3.0~4.0 정도를 기록한 것으로 보아 학습이 유의미하게 진행되었다고 볼 수 있다.

이 결과들을 종합해 볼 때, scalingRate 1.00 ~ 28.00의 전체 구간에서의 정확도의 평균은 노이즈를 제외하고 큰 차이가 없다는 것을 알 수 있으며, 그 중에서도 scalingRate 2.00 ~ 8.00 구간에서 가장 높다는 것을 알 수 있다. 또한 accuracyRate의 값이 0~2 epoch를 포함하여 전체적으로 epoch에 따른 증가 추세를 보이며, 25, 50, 100 epoch에서 그 값이 대부분 2 이상의 양수로 나타난 것을 볼 때 학습이 유의미하게 진행되었다고 할 수 있다.

difAvg	0	1	2	25	50	100	AVG25+
1.00	98.4362	48.7491	31.5436	0.0819	0.1936	0.1379	**0.1378**
1.25	58.2498	33.6191	9.8655	0.0998	0.0675	0.1409	**0.1027**
1.50	41.0859	28.2530	7.4801	0.1389	0.1237	0.1043	**0.1223**
1.75	32.0651	18.5720	3.3532	0.0371	0.2040	0.1204	**0.1205**
2.00	22.4599	12.7437	7.1417	0.2191	0.0676	0.0939	**0.1269**
2.50	15.3897	6.7654	1.1043	0.1715	0.1113	0.1350	**0.1393**
3.00	10.6543	6.5892	2.7638	0.0789	0.1226	0.1121	**0.1045**
3.50	7.3458	5.2990	1.2328	0.1007	0.1425	0.1535	**0.1322**
4.00	5.3089	3.1952	0.8468	0.0978	0.0846	0.1900	**0.1241**
5.00	3.4177	1.7129	0.9523	0.0359	0.0666	0.1501	**0.0842**
6.00	2.4991	1.7021	0.5548	0.1220	0.1828	0.2012	**0.1687**
7.00	1.7525	1.1788	0.3326	0.0579	0.0862	0.1816	**0.1086**
8.00	1.5252	1.1075	0.5231	0.0795	0.0909	0.0893	**0.0866**
10.00	0.8705	0.7540	0.2695	0.0920	0.0662	0.1127	**0.0903**
12.00	0.6957	0.5213	0.3630	0.1615	0.1091	0.0837	**0.1181**
14.00	0.5160	0.3951	0.2712	0.0677	0.1630	0.1994	**0.1434**
16.00	0.4961	0.3332	0.2272	0.0986	0.0955	0.1142	**0.1028**
20.00	0.4158	0.3870	0.2671	0.1141	0.1702	0.1238	**0.1360**
24.00	0.2739	0.2231	0.1841	0.0932	0.0841	0.1796	**0.1190**
28.00	0.3989	0.3142	0.3115	0.1369	0.2092	0.0667	**0.1376**
MIN	**0.2739**	**0.2231**	**0.1841**	**0.0359**	**0.0662**	**0.0667**	**0.1417**
MAX	**98.4362**	**48.7491**	**31.5436**	**0.2191**	**0.2092**	**0.2012**	**29.8931**

<Figure 1> difAvg 값에 대한 측정 결과

0difAvg	0	1	2	25	50	100	AVG25+
1.00	0.2722	0.2227	0.4002	0.1863	0.3122	0.3056	**0.2680**
1.25	0.2668	0.2497	0.2326	0.3495	0.2383	0.3403	**0.3094**
1.50	0.2940	0.3525	0.3459	0.3802	0.3872	0.3277	**0.3650**
1.75	0.3866	0.3843	0.4823	0.1883	0.3226	0.3281	**0.2797**
2.00	0.4132	0.4553	0.4011	0.4259	0.3111	0.2860	**0.3410**
2.50	0.4176	0.3314	0.3942	0.4636	0.3961	0.4434	**0.4344**
3.00	0.3485	0.3840	0.2406	0.4503	0.3647	0.3899	**0.4016**
3.50	0.4910	0.4816	0.5291	0.2125	0.4166	0.4104	**0.3465**
4.00	0.4072	0.4044	0.5625	0.3103	0.2439	0.5855	**0.3799**
5.00	0.4495	0.3655	0.3961	0.2019	0.2014	0.4607	**0.2880**
6.00	0.3100	0.2486	0.4906	0.4219	0.4214	0.4252	**0.4228**
7.00	0.2163	0.2690	0.3069	0.3492	0.3389	0.3642	**0.3508**
8.00	0.4631	0.4996	0.4965	0.2004	0.3128	0.2783	**0.2638**
10.00	0.2666	0.5362	0.4036	0.3124	0.2931	0.3743	**0.3266**
12.00	0.3793	0.3051	0.4078	0.4846	0.3090	0.3489	**0.3808**
14.00	0.2821	0.3165	0.3523	0.3380	0.4054	0.4073	**0.3836**
16.00	0.3469	0.3750	0.3323	0.3775	0.2581	0.3829	**0.3395**
20.00	0.4335	0.4807	0.3830	0.4579	0.4488	0.3927	**0.4331**
24.00	0.2594	0.4206	0.3040	0.3138	0.2312	0.4618	**0.3356**
28.00	0.4121	0.3440	0.5218	0.3677	0.4293	0.2607	**0.3526**
MIN	**0.2163**	**0.2227**	**0.2326**	**0.1863**	**0.2014**	**0.2607**	**0.2200**
MAX	**0.4910**	**0.5362**	**0.5625**	**0.4846**	**0.4488**	**0.5855**	**0.5181**

<Figure 2> 0difAvg 값에 대한 측정 결과

accuracy	0	1	2	25	50	100	AVG25+
1.00	0.0028	0.0046	0.0127	2.2748	1.6125	2.2158	**2.0344**
1.25	0.0046	0.0074	0.0236	3.5029	3.5304	2.4148	**3.1494**
1.50	0.0072	0.0125	0.0462	2.7360	3.1309	3.1406	**3.0025**
1.75	0.0121	0.0207	0.1438	5.0725	1.5816	2.7244	**3.1262**
2.00	0.0184	0.0357	0.0562	1.9439	4.6023	3.0465	**3.1976**
2.50	0.0271	0.0490	0.3570	2.7027	3.5582	3.2848	**3.1819**
3.00	0.0327	0.0583	0.0871	5.7065	2.9749	3.4766	**4.0527**
3.50	0.0668	0.0909	0.4292	2.1104	2.9237	2.6731	**2.5691**
4.00	0.0767	0.1266	0.6643	3.1741	2.8825	3.0814	**3.0460**
5.00	0.1315	0.2134	0.4159	5.6255	3.0215	3.0684	**3.9051**
6.00	0.1240	0.1460	0.8843	3.4592	2.3052	2.1138	**2.6261**
7.00	0.1234	0.2282	0.9226	6.0323	3.9333	2.0058	**3.9905**
8.00	0.3036	0.4511	0.9491	2.5190	3.4421	3.1169	**3.0260**
10.00	0.3063	0.7112	1.4976	3.3961	4.4260	3.3198	**3.7140**
12.00	0.5451	0.5852	1.1234	3.0013	2.8305	4.1683	**3.3334**
14.00	0.5468	0.8009	1.2992	4.9902	2.4876	2.0424	**3.1734**
16.00	0.6993	1.1254	1.4624	3.8294	2.7023	3.3540	**3.2952**
20.00	1.0424	1.2422	1.4340	4.0139	2.6369	3.1731	**3.2746**
24.00	0.9469	1.8854	1.6508	3.3660	2.7508	2.5719	**2.8962**
28.00	1.0333	1.0948	1.6748	2.6864	2.0519	3.9085	**2.8823**
MIN	**0.0028**	**0.0046**	**0.0127**	**1.9439**	**1.5816**	**2.0058**	**0.9252**
MAX	**1.0424**	**1.8854**	**1.6748**	**6.0323**	**4.6023**	**4.1683**	**3.2343**

<Figure 3> acuracyRate 값에 대한 측정 결과

5. 결론

본 논문에서는 과거 및 현재의 COVID-19 확진자 수 데이터를 이용하여 미래의 확진자 수를 예측하는 인공신경망 모델을 제안하였고, accuracyRate라는 지표가 학습이 진행됨에 따라 유의미하게 증가하는 것을 확인하였으며 이에 따라 학습이 유의미하게 이루어진다는 것을 보였다. COVID-19 확진자 추이는 대부분의 지역에서 초기에 서서히 증가하다가 중기에 급증하고, 말기에 이르러 증가 추세가 둔화되는 경향을 보이기 때문에 이러한 경향적 추세를 딥 러닝을 통해 학습하면 미래의 확진자 수를 쉽게 예측할 수 있을 것이라고 보았으며 실제로 어느 정도 학습이 되었다. 그러나 다음과 같은 한계점 때문에 학습에 일부 지장이 있었다.

- **데이터 양의 부족에 따른 학습의 부진:** 본 논문에서 설정한 parameter의 값에 따라 raw data로부터 추출한 학습용 입출력 데이터는 모두 1318개인데, 이 정도는 딥 러닝을 통한 정교한 학습을 진행하기에 부족한 양이다. 이는 COVID-19 이슈가 2020년 1월에 본격적으로 발생하여, 논문 작성 시점을 기준으로 발생한 지 겨우 2개월밖에 되지 않았기 때문이다.

- **각종 돌발 변수:** 앞서 언급한 일반적인 확진자 수 증가 추이를 따르다가 돌발 변수 때문에 추이가 급격히 변동하는 경우가 있다. 예를 들어 대한민국의 경우 첫 확진자가 발생한 2020년 1월 20일부터 그해 2월 18일까지의 추이로 보아서는 그해 2월 18일이 증가 추세가 둔화되는 시점에 해당하였다고 볼 수 있으나, 그해 2월 19일경 모 사이비 종교의 COVID-19 집단 감염 사태로 인해 급격한 상승 추이로 전환되었다.

본 논문에서는 1일 후의 확진자 수를 예측하는 학습에 대한 실험만을 진행했지만, 시간이 지나서 데이터가 축적되어 위와 같은 한계점이 일부 극복되면 parameter의 값을 적절히 조정하여 2일, 3일, 더 나아가 7일 이상 이후의 확진자 수를 예측하고, 그 정확도도 높아질 것으로 기대된다. 딥 러닝의 경우 데이터의 개수가 많을수록 일반적인 수학적 모델 또는 머신러닝 모델과의 정확도 차이가 더 유의미하게 벌어지기 때문에 데이터가 축적될수록 본 논문에서 제안한 모델의 가치는 높아질 것으로 예상한다. 충분한 횟수의 돌발 변수 발생과 함께 데이터가 충분히 많이 쌓이면 돌발 변수의 발생 가능성을 예측할 수도 있을 것이다. 그러나 이를 위해서는 지역별 COVID-19 확산 초기 및 중기 데이터도 수집해야 하므로, 아직 확진자가 발생하지 않은 많은 지역에서 유의미한 학습 데이터가 나올 수 있도록 확진자가 충분히 많이 발생해야 한다. 그런데 그것은 곧 COVID-19의 광범위한 확산을 의미하므로, 그것보다는 모델에 이와 같은 한계점은 있지만 COVID-19 바이러스가 빨리 극복 및 종식되어 세계 사회 및 경제가 다시 정상화되고, 각종 사회 일정들이 다시 정상적으로 진행되기를 바란다.

6. 참고 문헌

[1] 중앙일보, https://news.joins.com/article/23704860, 2020

[2] Wei Hao Khoong, COVID-19 Novel Coronavirus EDA & Forecasting Cases, at https://www.kaggle.com/khoongweihao/covid-19-novel-coronavirus-eda-forecasting-cases, 2020

[3] Deepak Deepu, COVID-19:Analysis+ EDA+ Forecasting, at https://www.kaggle.com/deepakdeepu8978/covid-19-analysis-eda-forecasting, 2020

[4] Neel, COVID-19 Visualizations, Predictions, Forecasting, at https://www.kaggle.com/neelkudu28/covid-19-visualizations-predictions-forecasting, 2020

[5] SRK, Novel Corona Virus 2019 Dataset, Version 42, at https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset/version/42, 2020

[6] Vinod Nair and Geoffrey E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, 2010

[7] Diederik P. Kingma and Jimmy Lei Ba, ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, 2014