

1. 제목: LSTM 알고리즘을 활용한 동행복권 번호 예측 프로그램

2. 요약

2-1) 데이터 수집

동행복권의 1회부터 1,064회차까지의 당첨 번호 데이터를 수집함.

(출처: 「동행복권」 <https://dhlottery.co.kr/>)

2-2) 데이터 전처리

수집한 데이터를 전처리하여 LSTM 알고리즘이 분석할 수 있는 형태로 가공.

이 과정에서, 당첨 번호를 One-Hot Encoding 방식으로 변환함.

데이터를 train: (0, 800), val: (801, 900), test: (901, 1063)로 나눔.

2-3) LSTM 모델 학습

전처리 된 데이터를 바탕으로 LSTM 모델을 학습.

2-4) 모델 검증

학습이 완료된 모델이 테스트케이스에서 얼마의 상금을 획득할 수 있는지 평가.

2-5) 미래 복권번호 예측 및 구매하여 확인

모든 복권 데이터를 학습하여 미래의 복권번호를 예측하고, 실제 구매해 봄으로써 확인.

3. 사용 데이터 요약

3-1) 동행복권의 1회부터 1,064회차까지의 당첨 번호, 등수별 상금 데이터.

(출처: 「동행복권」 <https://dhlottery.co.kr/>)

4. 상세 수행 내용

전제: 지금까지의 로또 번호를 딥러닝에서 학습 시키면 딥러닝이 로또 번호의 원리를 깨달아서 그 원리대로 그다음 회 로또 번호를 예측할 수 있다.

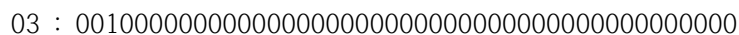
4-1) 데이터 수집

동행복권 홈페이지에서 1,064회까지의 당첨 번호, 당첨금액을 엑셀 파일로 수집합니다.

데이터 목록:

회차

5등 당첨금액



[illegible]

이전 회차의 당첨 번호를 넣어서 다음 회차의 당첨 번호를 예측하는 것입니다.

1회 당첨 번호를 모델에 입력하면, 모델에서 2회 당첨 번호를 알려준다고 예를 들겠습니다.

[illegible]

4-4) 샘플의 구성

(1회차 당첨번호, 2회차 당첨번호)
(2회차 당첨번호, 3회차 당첨번호)
(3회차 당첨번호, 4회차 당첨번호) ...

```
ohbins
X[0]: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
Y[0]: [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.
 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
```

4-5) 데이터 셋 구성

훈련셋: (1회 입력, 2회 출력)부터 (800회 입력, 801회)까지 700개 샘플

검증셋: (801회, 802회)부터 (900회, 901회)까지 100개 샘플

시험셋: (901회, 902회)부터 (1063회, 1064회)까지 163개 샘플

4-6) 모델 구성

모델을 딥러닝 라이브러리 중 하나인 텐서플로의 케라스를 이용합니다.

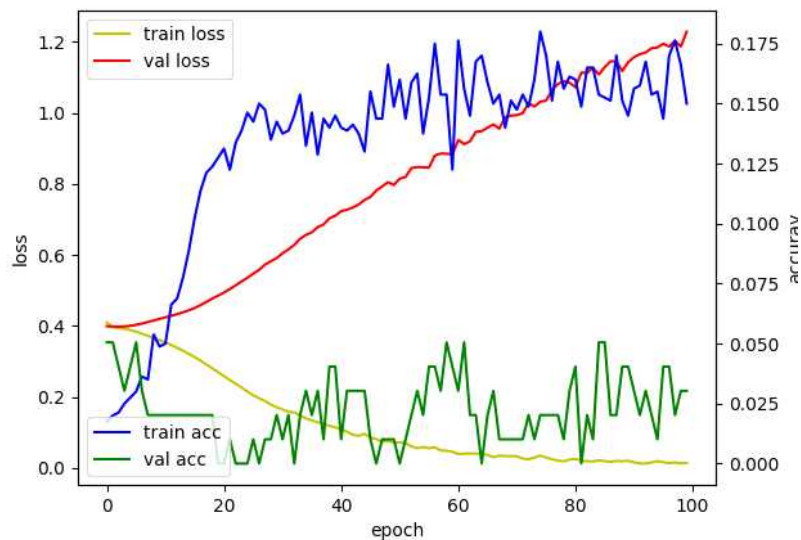
45개의 벡터로 출력합니다.

각각의 벡터는 0.0과 1.0 사이의 실숫값으로 나옵니다.

멀티레이블 문제라 출력층의 활성화함수를 sigmoid로 설정하였습니다.

4-7) 모델 학습

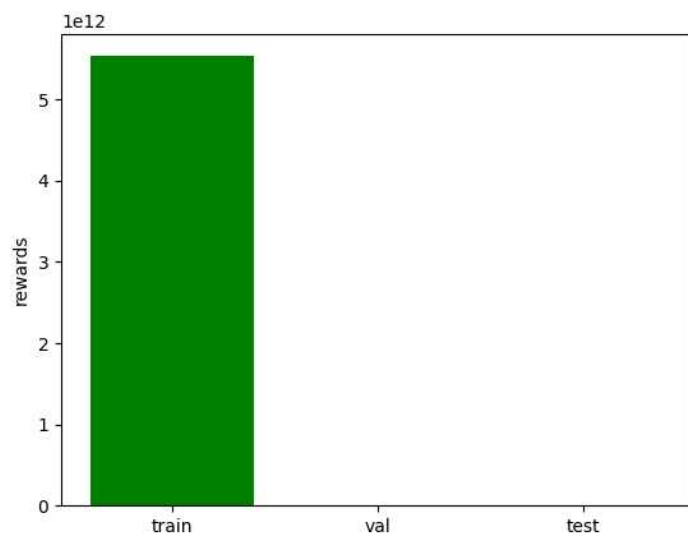
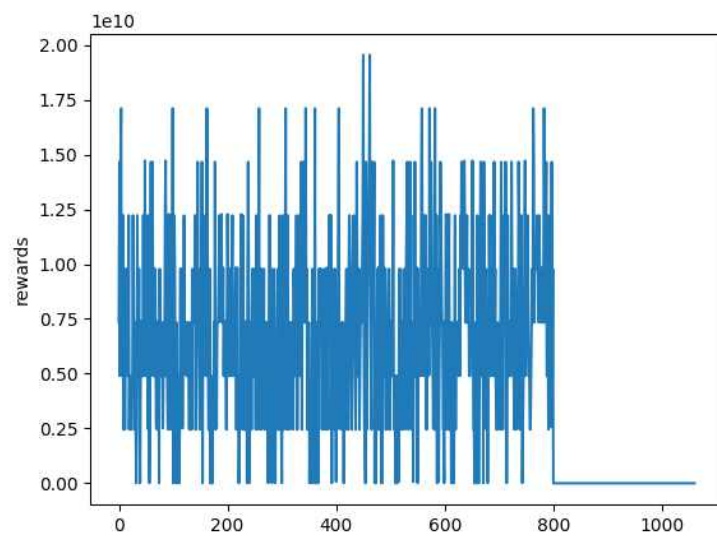
100 에포크 학습 시행. 매 에포크 저장된 훈련셋의 손실 값 및 정확도와 검증셋의 손실값 및 정확도에 대해서 기록한 것을 그래프로 표현.



4-8) 모델 검증

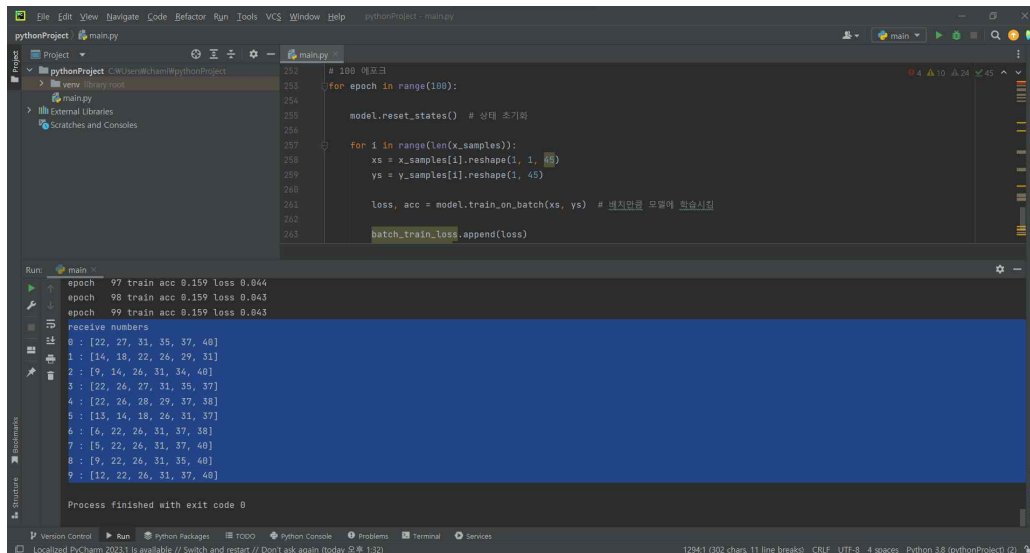
모델이 10개의 복권번호를 출력하고 훈련셋, 검증셋, 시험셋에서 얼마나 많은 상금을 획득하였는지 검증합니다.

1. (a)회차 당첨번호를 모델에 입력한 후 모델 출력으로부터 10판의 번호를 생성한 뒤 (a+1)회차와 비교하여 등수와 상금 계산
2. (a+1)회차 당첨번호를 모델에 입력한 후 모델 출력으로부터 10판의 번호를 생성한 뒤 (a+2)회차와 비교하여 등수와 상금 계산
- .
- .
- .



4-8) 미래 복권번호 예측

1,064개의 훈련셋을 학습시켜 미래의 복권번호를 예측합니다.



```
252 # 100 에포크
253 for epoch in range(100):
254     model.reset_states() # 상태 초기화
255     for i in range(len(x_samples)):
256         xs = x_samples[i].reshape(1, 1, 45)
257         ys = y_samples[i].reshape(1, 45)
258         loss, acc = model.train_on_batch(xs, ys) # 배치만큼 모델에 학습시킴
259         batch_train_loss.append(loss)
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

epoch 97 train acc 0.159 loss 0.844
epoch 98 train acc 0.159 loss 0.843
epoch 99 train acc 0.159 loss 0.843
Receive numbers
0 : [22, 27, 31, 35, 37, 40]
1 : [14, 18, 22, 26, 29, 31]
2 : [9, 14, 26, 31, 34, 40]
3 : [22, 26, 27, 31, 35, 37]
4 : [22, 26, 28, 29, 37, 38]
5 : [13, 14, 18, 26, 31, 37]
6 : [6, 22, 26, 31, 37, 38]
7 : [5, 22, 26, 31, 37, 40]
8 : [9, 22, 26, 31, 35, 40]
9 : [12, 22, 26, 31, 37, 40]
Process finished with exit code 0

receive numbers

- 0 : [22, 27, 31, 35, 37, 40]
- 1 : [14, 18, 22, 26, 29, 31]
- 2 : [9, 14, 26, 31, 34, 40]
- 3 : [22, 26, 27, 31, 35, 37]
- 4 : [22, 26, 28, 29, 37, 38]
- 5 : [13, 14, 18, 26, 31, 37]
- 6 : [6, 22, 26, 31, 37, 38]
- 7 : [5, 22, 26, 31, 37, 40]
- 8 : [9, 22, 26, 31, 35, 40]
- 9 : [12, 22, 26, 31, 37, 40]

4-8) 실제로 복권을 구매하여 확인

로또6/45

1069회차 (2023.05.27.)

당첨번호	당첨금액	당첨판매점
1 10 18 22 28 31 + 44	1등 당첨금 1,863,217,554원 (당첨 복권수 14개)	



결과: 5등 1개

5. 구글 코랩 파일

<https://colab.research.google.com/drive/15TUETccTHHr7Oq4g35TrX9XaRMAYBYOV?usp=sharing>

6. 분석 및 고찰

모델 학습 결과를 분석해보면 모델 학습에서는 일반적인 학습 결과와는 다르게 val loss가 처음부터 끝까지 지속해서 증가하는 모습을 볼 수 있습니다. 이는 모델이 훈련 데이터에 대해서는 좋은 성능을 보이지만 새로운 데이터에 대해서는 일반화하지 못한다는 뜻입니다. 이는 과적합(overfitting)의 징후로 추정됩니다. 과적합은 모델이 훈련 데이터에 너무 맞추어져서 새로운 데이터에 대한 일반화 성능이 떨어지는 현상입니다. 결국 본 모델은 복권번호를 효과적으로 예측하지 못함을 알 수 있습니다. 이는 모델 검증 결과에서 더 확실히 드러납니다. train set인 800회까지는 많은 상금을 획득한 것이 보이지만, 그 이후로는 급격히 획득 상금의 액수가 줄어든 것을 보입니다. 1,064회의 train set을 학습하여 예측한 미래 예측 복권번호를 실제로 사용한 결과도 일반적인 무작위 선택과 큰 차이가 없음을 알 수 있습니다. 결국 인터넷상에서 광고하는 'AI 복권번호 추천', 'AI 로또 예상 번호'는 아무런 효력이 없는 상품이라는 것이 확인되었습니다.

N | ai 로또 예상번호

VIEW

인물 미술쟁 2023.02.19.

이번주 로또 예상번호 AI가 알려줍니다!

인공지능 AI 로또 예상번호 최근에는 인공지능 AI를 통해 로또 예상번호를 예측하고 있습니다. 또한 그 적중률 또한 굉장히 좋아서 화제가 되고 있습니다. 지금 바...

[로또 예상번호, 인공지능 AI 추천](#)
[로또 번호 추천 인공지능 AI를 통한 예상번호 받기](#)

이야기 보따리 7일 전

1069회 이번주 로또 예상번호 빅데이터 조합

오늘의 분석은 저의 로또 분석방향을 AI인 Chat GPT에 적용하여 예상번호를 조합하도록 하겠습니다. 최근 여러 AI들이 생기고 있는데, 다음에 기회가 되면 다양한 AI...

[1068회 이번주 로또 예상번호 빅데이터 조합](#)
[1067회 이번주 로또 예상번호 빅데이터 조합](#)

달달한 인생 어제

로또 1069회 예상번호 조합 공유

여러분은 원하는 성과를 이루고 계시는지 궁금한데 올해는 로또 1등 당첨을 기원하며 AI 로또 데이터로 나온 분석 자료 공유합니다. 2023년 5월 27일 토요일 추첨 예...

[역대로로당첨번호 분석으로 당첨되기!](#)

jochanyang0님의블로그 2021.10.14.

로또986회예상번호 AI분석 숫자 무료로 받기

1회차의 번호를 학습을해서 다음회차를 예측해서 어떤게 맞았는지 학습을하고 그 령지 않은 번호들은... 알려드리겠습니다 로또986회예상번호 AI분석 숫자 무료로 ...

