

# 딥러닝을 이용한 실시간 게임 승률 예측

이선훈, 오흥선  
한국기술교육대학교 컴퓨터공학부  
e-mail : seonhoon1002@gmail.com, ohhs@koreatech.ac.kr

## Predicting Winning Rates of LOL in Real-time using Deep Learning

Seon Hoon Lee, Heung-Seon Oh  
School of Compute Science and Engineering, KOREATECH

### 요 약

일반적으로 멀티 온라인 배틀 게임은 게임의 참가자들이 팀을 이루어 전략을 짜고 협력하여 주어진 목적을 성취하면 승리한다. 게임에서는 승리를 판가름 할 수 있는 다양한 요소(e.g. 골드, 아이템, 캐릭터의 레벨 등)들이 있다. 본 논문에서는 게임 플레이 중에 다양한 요소를 분석하여 실시간으로 승률을 예측할 수 있는 딥러닝 기반의 모델을 제안하고 이를 리그오브레전드 게임에 적용하여 그 결과를 분석하였다.

### 1. 서론

리그오브레전드는 두 팀으로 나뉘어서 상대방의 최종 건물(넥서스)을 깨면 이기는 것을 목표로 하는 멀티 온라인 배틀 게임이다. 팀은 블루 팀과 레드 팀으로 나뉘고 각 팀당 인원수는 5 명으로, 팀원은 서로 다른 포지션을 맡아 게임에 임한다.

이 게임은 올림픽 정식 종목으로 논의될 만큼 게임 대회가 활성화 되어있다. 그러나 대회를 시청하다 보면 누가 이기고 있는 것인지 게임에 익숙하지 않은 초보자는 알기가 어렵다. 만약에 실시간으로 어느 팀이 유리한지 정확한 수치를 알 수 있다면, 해설자의 입장에선 경기를 설명하기 쉬울 것이고, 시청자들의 입장에선 경기가 어떻게 흘러가고 있는 것인지 알 수 있어서 게임을 더욱 즐길 수 있을 것이다.

본 연구에서는 실시간 승률을 예측하기 위하여 게임에서 승패와 관련도가 높은 특징(feature)를 추출하고 이를 이용하여 시계열 데이터를 처리하는데 적합한 recurrent neural network(RNN) 방법 중 gated recurrent unit(GRU) [1] 방법에 입력하여 승률을 예측하였다. 여기서 말하는 실시간 승률이란 분(minute)당 승률을 의미한다.

본 논문의 구성은 다음과 같다. 2 장에서는 연구에서 사용된 모델을 소개한다. 3 장에서는 사용한 데이터와 학습방법에 대해 소개하고, 4 장에서는 결과를 분석한다. 5 장에서 연구의 결과와 추후 어떤 분야에 응용되면 좋을지 소개하면서 결론을 맺도록 한다.

### 2. RNN 모델

$p_t$ : t 분일 때 블루팀 승률,  $q_t = 1 - p_t$ : t 분일 때 레드팀 승률,  $x_t$ : t 분 일 때 게임에 대한 정보라고 하면, GRU[1]의 t 시간의 현재의 히든 상태(hidden state)는

다음과 같이 정의된다.

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \quad (\text{식 1})$$

(식 1)에서 이전의 히든 상태인  $h_{t-1}$ 와 현재의 히든 상태  $\hat{h}_t$  조합하여 현재의 히든 상태를 생성한다.

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (\text{식 2})$$

(식 2)는 업데이트 게이트이며 과거의 정보를 얼마나 볼지 결정한다. 여기서  $W^{(z)}$ ,  $U^{(z)}$  는 각각  $x_t$  와  $h_{t-1}$  를 선형 결합하는 행렬이다.

$$\hat{h}_t = \tanh(Wx_t + U(r_t \odot h_{t-1})) \quad (\text{식 3})$$

(식 3)에서  $\odot$ 는  $r_t$ 와  $h_{t-1}$ 사이의 요소 곱을 의미한다. 여기서 쓰인  $W$ ,  $U$  는 각각 리셋 게이트, 업데이트 게이트의 가중치 행렬을 나타낸다.

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (\text{식 4})$$

(식 4)는 리셋 게이트로 과거 정보를 얼마나 볼 것인지 결정한다. 여기서  $W^{(r)}$ ,  $U^{(r)}$  는 각각  $x_t$  와  $h_{t-1}$  를 선형결합하는 행렬이다.

$$y_t = \text{softmax}(W_{yh}h_t) \quad (\text{식 5})$$

(식 5)는 GRU 에서 나온 결과 값을 softmax 함수에 넣는 과정이다.  $W_{yh}$ 는  $h_t$  벡터를 선형 결합하는 행렬이고 softmax 의 클래스는  $C = \{\text{블루팀 승리}, \text{레드팀 승리}\}$ 로  $|C|=2$  이다.

### 3. 실험

#### 3.1 데이터

<표 1>은 학습에 사용한 데이터의 출처, 개수, 그리고 특징을 나타낸다.

<표 1> 학습에 사용된 데이터

데이터 출처	리그오브레전드 API <sup>1</sup>
전체 데이터 수 (게임)	51,084
학습 데이터 수	40,867
테스트 데이터 수	10,217
게임 시간	50 분
입력 특징 벡터( $x_t$ )	[골드 차이, 경험치 차이, 타워개수차이, 에픽 몬스터 사냥 수 차이, 바론 차이, 협곡의 전령 차이, 장로드래곤차이, 챔피언 킬수 차이]
정답 벡터( $y_t$ )	[블루 팀 승리, 레드 팀 승리]

<표 1>에서처럼 데이터는 롤 API 를 통해 추출한 51,084 개의 ‘개인/2 인 랭크 게임’ 을 사용하였다. 여기서 말하는 ‘개인/2 인 랭크 게임’ 이란, 자신의 점수를 올리고 내릴 수 있는 게임으로 40,867 랭크 게임을 학습데이터에, 10,217 랭크 게임을 테스트 데이터로 사용했다.

게임 시간은 50 분을 기준으로 했으며, 학습데이터는

$x_t \in \mathbb{R}^8$  인 벡터이다. 각 속성은 게임에서 중요하다고 생각하는 속성들을 뽑은 것이다. 정답지는  $y_t \in \mathbb{R}^2$  이다. [1,0]이면 블루 팀이 이긴 경우고, [0,1]이면 레드 팀이 이긴 경우다.

#### 3.2 학습

GRU 의 학습 데이터는 50 분의 게임데이터이다. 50 분 미만의 게임데이터는 zero-padding 을 수행하였다. 학습률은 0.001 로 하였고, epoch 은 20 회를 설정하였으며, GRU 의 히든 상태의 크기인  $h_t$  는 12 로 설정하였다. Optimizer 는 Adam[2] 을 사용하며 교차 엔트로피(cross entropy)를 이용한 손실함수(loss function)를 정의하였다. <표 2>는 모델의 학습에 필요한 설정을 보여준다.

<표 2> 학습에 적용한 매개변수 및 최적화 기법

전체 시퀀스 길이(시간)	50(분)
학습률	0.001
배치 사이즈	50
epoch	20
GRU $h_t$ 사이즈	12
Optimizer	Adam
Loss	cross entropy

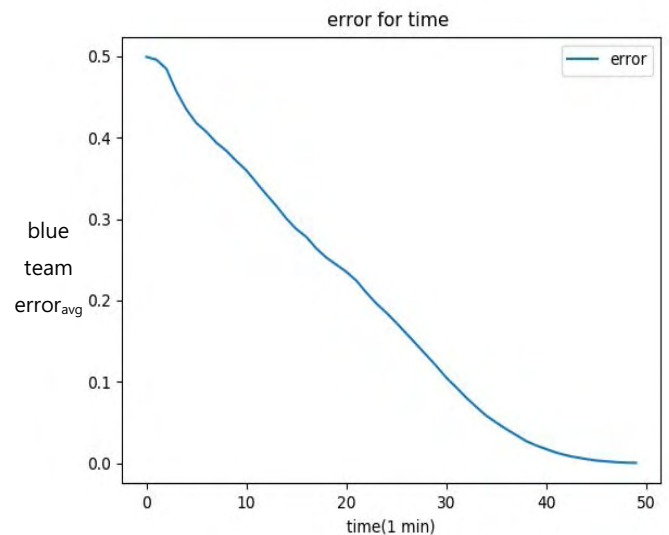
### 4. 결과 분석

각 분당 승률 오차는 다음과 같은 수식을 통해 측정하였다. 여기서 승률 오차는 블루 팀에 대한 오차이며,  $error_t^{game}$  는 한 게임에 대한 해당 시간의 오차를 의미한다. 예를 들어  $error_{20}^1$  은 1 번 경기에서 20 분 일 때, 승률 오차를 의미한다.

$error_{avg}$  (average error per time)는 모든 데이터에 대한  $error_t^{game}$  의 평균값이다.

$$error_t^{game} = \begin{cases} 1 - p_t^{game}, & \text{if } 0.5 > p_t^{game} \\ 0 & \text{else} \end{cases} \quad (\text{식 } 6)$$

$$error_{avg} = \frac{\sum_{g \in \text{game}} error_t^{game}}{(\text{total number of games})} \quad (\text{식 } 7)$$



(그림 1) 시간에 따른 블루팀의  $error_{avg}$

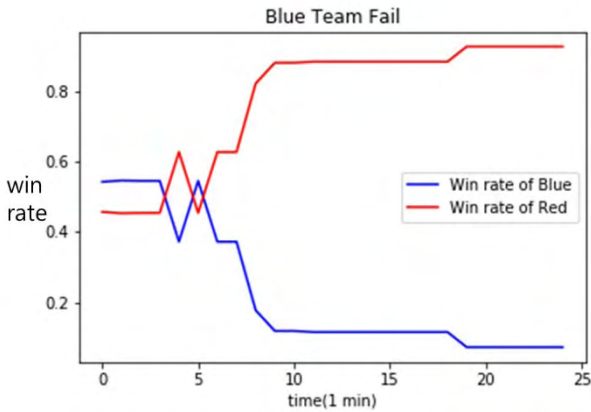
(그림 1)은 테스트 데이터 10,217 게임을 바탕으로 계산한 오차이다. 세로축은 분당 승률 오차를 의미하고 가로축은 1 분 단위의 시간을 의미한다. 그래프를 보면 시간이 지날수록 오차가 줄어드는 것을 확인할 수 있다. 경기시간이 0 분일 때는 모델의 예측 값이 맞을 확률이 50%이지만, 시간이 흐를수록 모델의 예측이

<sup>1</sup> <https://developer.riotgames.com/>

정확해지는 것을 볼 수 있다.

참고문헌

- [1] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014.
- [2] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015, pp. 1-13.



(그림 2) 레드팀이 이긴 승률 그래프

(그림 2)는 임의의 한 게임에 대해서 승률을 예측한 그래프이다. (그림 2)에서 5~6 분대에 벌어진 격차가 경기가 끝날 때까지 줄어들지 못한 것을 알 수 있다. 이를 통해서 이 게임의 비전문가도 블루 팀이 언제부터 경기의 승패가 언제 기울어졌는지, 또 얼마나 게임이 불리한 것인지 한눈에 파악할 수가 있다.

5. 결론

본 논문에서는 GRU 모델을 이용하여 실시간 게임 승률을 예측하고 그 결과를 분석하였다. 게임 진행 중에 명시적으로 승률을 보여준다면 초보자들도 경기를 쉽게 이해할 수 있고, 더 나아가서 프로게이머들 또한 이 결과를 바탕으로 피드백을 받을 수 있을 것으로 기대한다. 추후에는 리그오브레전드 뿐만 아니라 스타크래프트, 오버워치 등 다른 게임에서도 실시간 승률을 알 수 있다면, 게임 경기를 좀더 재밌게 관람하는 동시에 선수들에게는 명시적인 피드백을 줄 수 있을 것이라고 기대된다.