

Система взаимодействия игрока с компаньоном с применением нейронных сетей для решения NLP задач

Зав. кафедрой _____ С.Д. Махортов д.ф.- м.н., доцент
Обучающийся _____ П.Н. Парамонов, 2 курс (маг), д/о
Руководитель _____ В.С. Тарасов, ст. преподаватель

Воронеж 2023

Введение

- Игры стремительно совершенствовались
- Техническое развитие игр опережает развитие геймдизайна
- Недостаточно технической составляющей для высоких оценок
- Развитие геймдизайна по многим направлениям

Введение

Проблема:

- Недостаточное вовлечение игрока во взаимодействие с NPC
 - Отсутствие интереса к NPC
 - Беснадобность NPC
 - Минимизация контакта, только если это необходимо
- Ограниченный и минимальный набор функций NPC
 - Список диалоговых фраз заранее известен
 - Если расширять возможности, список будет также увеличиваться, что приведет к неудобству

Постановка задачи

- Разработать механику взаимодействия с компаньоном:
 - Игрок вводит любую фразу в диалоговое окно с NPC
 - NPC должным образом реагирует на это, выполнив необходимую игровую функцию
 - Учитывается тон фразы
- Разработать наиболее подходящие модели нейронной сети:
 - Минимально возможное время обучения моделей
 - Максимально возможная точность прогнозирования моделей

Игровые функции NPC

- Интерактивные способности:
 - Рассказать о мире
 - Напомнить суть текущего задания
 - Дать совет по тактике боя
 - Дать совет по прохождению текущего задания
- Боевые способности:
 - Тренировочный бой(меч, булава, топор, рукопашный бой)
 - Вылечить игрока
 - Перейти в ближний/дальний бой
- Различные способности:
 - Обмениваться ресурсами
 - Взломать замок
 - Исследовать территорию

Средства реализации

- Python 3.10.9
- Tensorflow 2.10.0
- Keras 2.10.0
- KerasTuner 1.3.5
- Google Colaboratory
- DataSpell

CPU: AMD Ryzen 5 3500U 2.1GHz

RAM: 5.71 GB

Обучающие данные тональности

- Источник Kaggle
- Содержит набор сообщений сопоставленный с соответствующими классами
- Количество данных:
 - 160000 для обучения
 - 64000 для тестов

comment_text	toxic	severe_toxic	obscene	threat	insult	identit
I agree with the above. I look for some references...	0	0	0	0	0	
bigjkim why do you like Jews so much ? why are you...	0	0	0	0	0	
"REPLY: There is no such thing as Texas Commerce ...	0	0	0	0	0	
Nietzsche's Last Man Was Nietzsche's concept of t...	0	0	0	0	0	
Ha! Riiiiiight....because you and the other two moro...	1	0	0	0	0	
"*I have decided to take the next step and have you...	0	0	0	0	0	

Обучающие данные определения действия

- Источник ChatGPT
- Набор сообщений с соответствием с номером класса действия
- Количество данных:
 - 7000 для обучения
 - 1400 для тренировки

	C1 ÷	C2	÷
1	1	I'd like to know more about your interpersonal skil...	
2	9	We should incorporate sparring into our fist fight ...	
3	3	What is the end goal of our current mission?	
4	10	Can you please use your healing herbs?	
5	6	We should spend some time training with swords.	
6	10	Is there a way to fight effectively with little hea...	
7	1	Share your story of overcoming self-doubt and stepp...	
8	13	Do you want to exchange equipment with me?	
9	6	Let's work on your sword fighting skills.	
10	1	Describe the moments that have taught you the value...	
11	3	What is the task that we need to complete before an...	

Множество гиперпараметров

- Количество слоев в скрытом слое
- Количество нейронов на каждом слое
- Размерность вектора в слое Embedding
- Функция активации на каждом слое
- Метод оптимизации
- Коэффициенты регуляризации
- Количество ядер свертки
- Длина окна свертки

Реализация тональности

Тип сети	Время подбора (GPU)	Время обучения (CPU)	Inference	AUC	Accuracy
CNN	23 минуты	39 секунд	3ms/step	0,96269	0,9975
GRU	2 часа 18 минут	13 минут	25ms/step	0,95986	0,9961
LSTM	2 часа 1 минута	12 минут	19ms/step	0,95914	0,9852
RNN	1 час 18 минут	23 минуты	14ms/step	0,95076	0,9976
FNN	54 минуты	43 минуты	81ms/step	0,95034	0,3222

Архитектура сети

- Слой 1: Embedding(output_dim=36, regularizer=1e-4)
- Слой 2: Conv1D (filters=32, kernel_size=5, FA=elu, regularizer=1e-4)
- Слой 3: MaxPooling1D(pool_size=2)
- Слой 4: Flatten()
- Слой 5: Dense(units=108, FA=elu, regularizer=1e-4)
- Слой 6: Dense(units =6, FA=sigmoid)
- Optimizer=rmsprop
- Loss=binary_crossentropy

Процесс обучения сети

```
Epoch 1/5
125/125 [=====] - 10s 63ms/step - loss: 0.1412 - auc: 0.8930 - accuracy: 0.9793 - val_loss:
0.0890 - val_auc: 0.9589 - val_accuracy: 0.9940
Epoch 2/5
125/125 [=====] - 7s 56ms/step - loss: 0.0812 - auc: 0.9640 - accuracy: 0.9783 - val_loss: 0.0758
- val_auc: 0.9683 - val_accuracy: 0.9909
Epoch 3/5
125/125 [=====] - 7s 56ms/step - loss: 0.0726 - auc: 0.9695 - accuracy: 0.9464 - val_loss: 0.0734
- val_auc: 0.9667 - val_accuracy: 0.9431
Epoch 4/5
125/125 [=====] - 7s 59ms/step - loss: 0.0690 - auc: 0.9718 - accuracy: 0.9840 - val_loss: 0.0747
- val_auc: 0.9627 - val_accuracy: 0.9026
Epoch 5/5
125/125 [=====] - 8s 61ms/step - loss: 0.0669 - auc: 0.9727 - accuracy: 0.9908 - val_loss: 0.0686
- val_auc: 0.9663 - val_accuracy: 0.9941
```

```
model.evaluate(x_test, y_test)
```

```
Executed in 7s, 18 Apr at 01:24:44
```

```
2000/2000 [=====] - 7s 3ms/step - loss: 0.0840 - auc: 0.9627 - accuracy: 0.9975
```

```
[0.08404318988323212, 0.9626981019973755, 0.9975460171699524]
```

Реализация выбора действия

Тип сети	Время подбора (GPU)	Время обучения (CPU)	Inference	Accuracy
CNN	6 минут	20 секунд	8ms/step	0,9821
FNN	9 минут	11 секунд	38ms/step	0,9686
LSTM	53 минуты	15 секунд	8ms/step	0,9086
RNN	41 минута	8 секунд	8ms/step	0,8879
GRU	54 минуты	35 секунд	39ms/step	0,85550

Архитектура сети

- Слой 1: Embedding(output_dim=116, regularizer=1e-4)
- Слой 2: Conv1D (filters=172, kernel_size=3, FA=selu, regularizer=1e-4)
- Слой 3: MaxPooling1D(pool_size=2)
- Слой 4: Flatten()
- Слой 5: Dense(units=140, FA=selu, regularizer=1e-4)
- Слой 6: Dense(units =15, FA=softmax)
- Optimizer=rmsprop
- Loss=categorical_crossentropy

Процесс обучения сети

```
Epoch 1/5
45/45 [=====] - 6s 80ms/step - loss: 2.0205 - accuracy: 0.5488 - val_loss: 1.0437 -
val_accuracy: 0.7597
Epoch 2/5
45/45 [=====] - 3s 68ms/step - loss: 0.4966 - accuracy: 0.9511 - val_loss: 0.6785 -
val_accuracy: 0.8391
Epoch 3/5
45/45 [=====] - 4s 79ms/step - loss: 0.2297 - accuracy: 0.9872 - val_loss: 0.2613 -
val_accuracy: 0.9639
Epoch 4/5
45/45 [=====] - 3s 78ms/step - loss: 0.1372 - accuracy: 0.9968 - val_loss: 0.1682 -
val_accuracy: 0.9802
Epoch 5/5
45/45 [=====] - 4s 79ms/step - loss: 0.0972 - accuracy: 0.9986 - val_loss: 0.1456 -
val_accuracy: 0.9837

model_action.evaluate(x_test, y_test)
Executed in 482ms, 22 Apr at 23:53:13

44/44 [=====] - 0s 8ms/step - loss: 0.1455 - accuracy: 0.9821

[0.1454996019601822, 0.9821428656578064]
```

Вывод

- Разработана механика взаимодействия с компаньоном, благодаря которой:
 - Игрок может ввести необходимую ему фразу в диалоговое окно с NPC
 - После чего классифицируется данная фраза по функциям NPC и по тональности
 - На основе этого NPC выполнить требуемую функцию
- Разработаны наиболее подходящие модели нейронной сети:
 - Для классификации функций NPC наиболее подходящей является – CNN, с временем обучения в 39 секунд, Inference = 3ms/step, AUC = 0,96269 и Accuracy = 0,9975
 - Для классификации тональности фраз наиболее подходящей является – CNN, с временем обучения в 20 секунд, Inference = 8ms/step и Accuracy = 0,9821

Спасибо за внимание