ДОДАТОК А

Графічний матеріал атестаційної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

ФАКУЛЬТЕТ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА УПРАВЛІННЯ
КАФЕДРА КІТС

Модель згорткової мережі для мультиагентної кооперації

Магістрант гр. КІТм-19-1 — Тельний М.А.
Науковий керівник — проф. Аксак Н. Г.

Харків 2020

Мета роботи

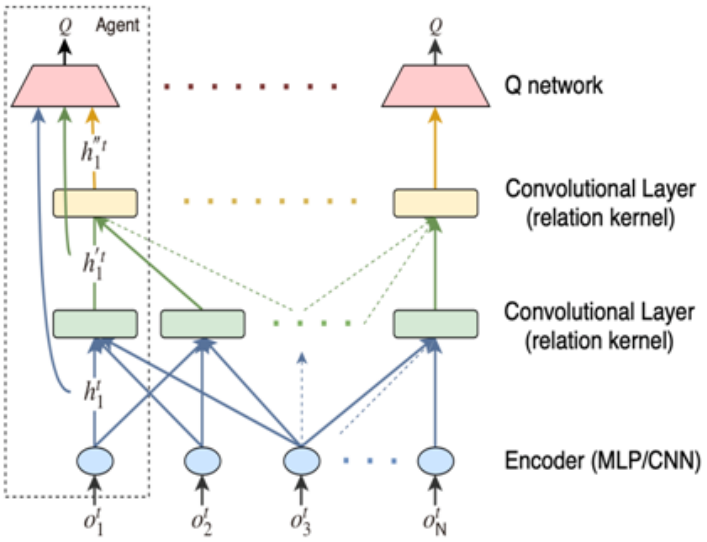
Метою атестаційної роботи є розробка моделі згорткової мережі для мультиагентної кооперації.

Для досягнення даної мети необхідно вирішити наступні задачі:

- проаналізувати та розглянути сучасні роботи та публікації для визначення актуальності роботи;
- визначити сфери використання, переваги, нелодіки та перспективи розвитку нейронних мереж та мультиагентних систем;
- розглянути сучасні методи розробки нейронних мереж;
- провести аналіз моделей мультиагентної кооперації;
- реалізувати модель згоркової мережі для мультиагентної кооперації.

2

Структура DGN



o_i^t - локальне спостереження

a_i^t - дія

r_i^t - винагорода

h_i^t - прихований вектор ознак

Q - мережа

3

Згортка графів

На кожному кроці кортеж

$$(O, A, O', R, C)$$

зберігається у буфері відтворення $\boldsymbol{B}$,
де $O = \{o_1, \cdots, o_N\}$, $A = \{a_1, \cdots, a_N\}$, $O' = \{o'_1, \cdots, o'_N\}$, $R = \{r_1, \cdots, r_N\}$, $C = \{C_1, \cdots, C_N\}$.

Потім випадково відбирається міні-партію зразків S з буфера $\boldsymbol{B}$ і мінімізуються втрати

$$L(\theta) = \frac{1}{S}\sum_{S}\frac{1}{N}\sum_{i=1}^{N}(y_i - Q(O_i, a_i; \theta))^2 \quad (1)$$

де $y_i = r_i + \gamma max_{a'} Q(O'_i, a'_i, \theta')$, $O_i \subseteq O$ позначає сукупність спостережень усіх агентів у рецептивних полях i, γ - коефіцієнт знижки, модель параметризована по θ.

Відношення між i та $j \in E_i$ обчислюється як

$$a_{ij}^m = \frac{\exp(\tau \cdot W_q^m h_i \cdot (W_k^m h_j)^T}{\sum_{e \in \varepsilon_i} \exp(\tau \cdot W_q^m h_i \cdot (W_k^m h_e)^T} \quad (2)$$

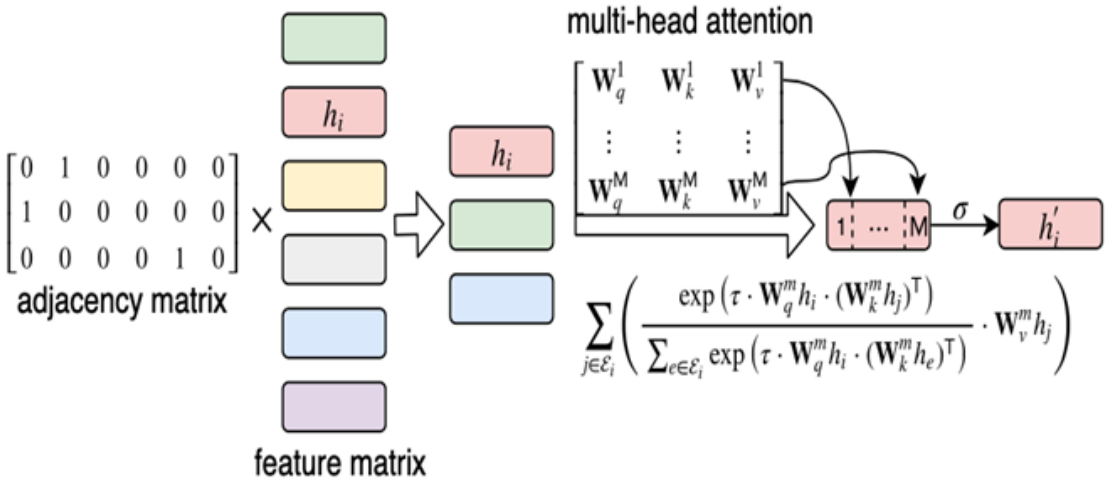
де τ - коефіцієнт масштабування.

Вихід згорткового шару

$$h'_i = \sigma(Concat[\sum_{j \in \varepsilon_i} a_{ij}^m W_v^m h_j, \forall m \in M] \quad (3)$$

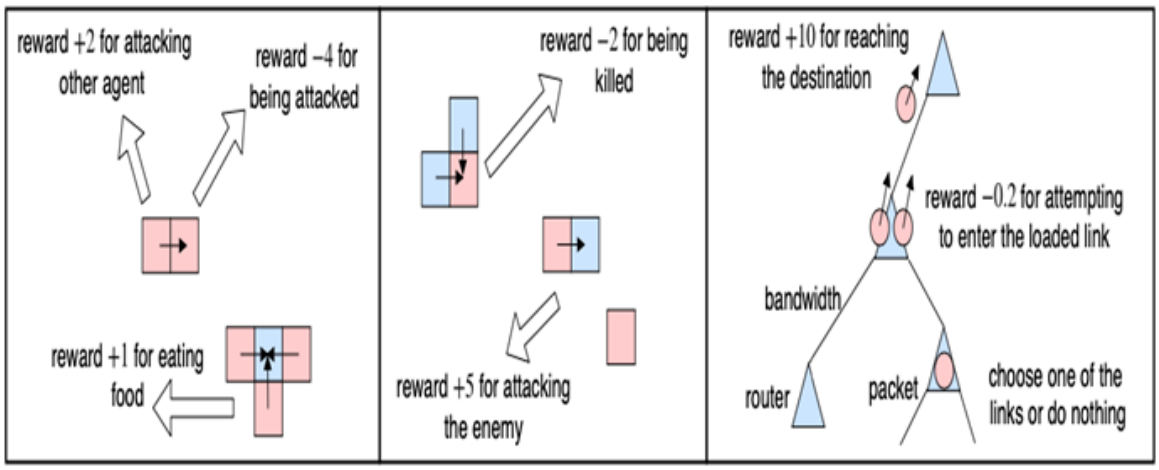
4

Обчислення згорткового шару



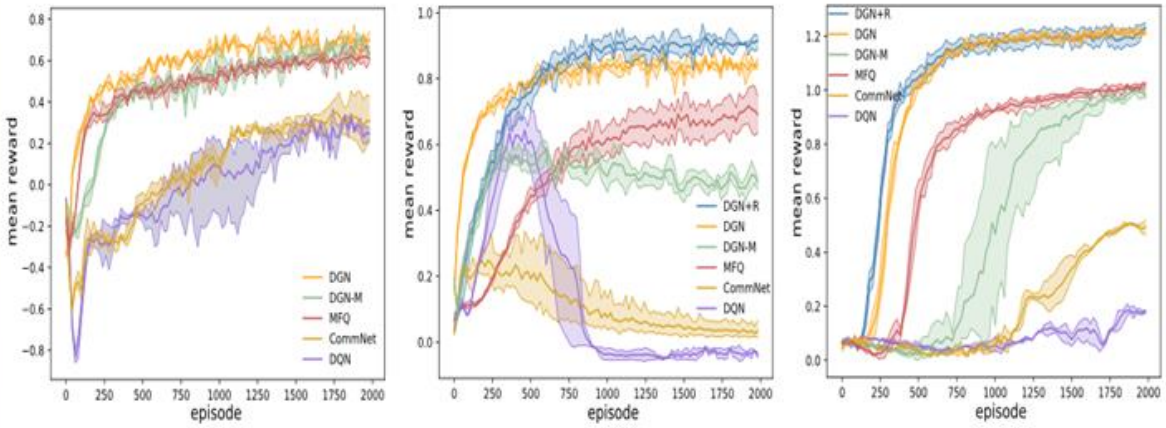
5

Ілюстрація експериментальних сценаріїв



6

Навчання моделі з точки зору середньої винагороди



а) джунглі б) битва в) маршрутизація

7

Результати експериментів

Таблиця.1 – Ілюстрація середньої винагороди

(N,L)		DGN	DGN-M	MFQ	CommNet	DQN
(20,12)	mean reward	0.70	0.66	0.62	0.30	0.24
	#attacks	0.91	1.89	2.74	5.44	7.35
(50,12)	mean reward	0.67	0.63	0.57	0.27	0.20
	#attacks	0.91	1.88	3.13	6.35	9.02

Таблиця2 – Середня винагорода, вбивства, смерті та коефіцієнт вбивства-смерті

	DGN+R	DGN	DGN-M	MFQ	CommNet	DQN
mean reward	0.91	0.84	0.50	0.70	0.03	-0.03
#kills	220	208	121	193	7	2
#deaths	97	101	84	92	27	74
Kill-death ratio	2.27	2.06	1.44	2.09	0.26	0.03

Таблиця3 – Середня винагорода, середня затримка пакетів даних та пропускна здатність

(N,L)		Floyd	Floyd with BL	DGN+R
(20,20)	mean reward			1.23
	delay	6.3	8.7	8.0
	throughput	3.17	2.30	2.50
(40,20)	mean reward			0.86
	delay	6.3	13.7	9.8
	throughput	6.34	2.91	4.08
(40,20) retrained	mean reward			0.94
	delay	6.3	13.7	10.2
	throughput	6.34	2.91	3.92

8

Результати експериментів

Таблиця.4 – Середня винагорода, середня затримка пакетів даних та пропускна здатність

(N,L)		DGN	DGN-M	MFQ	CommNet	DQN
(20,20)	mean reward delay throughput	1.21 8.1 2.47	0.99 9.8 2.04	1.02 9.4 2.13	0.49 18.6 1.08	0.18 46.7 0.43
(40,20)	mean reward delay throughput	0.83 10.0 4.00	0.70 12.7 3.15	0.78 11.8 3.39	0.39 23.5 1.70	0.12 83.6 0.49
(40,20) retrained	mean reward delay throughput	0.90 10.5 3.81	0.78 12.2 3.27	0.76 12.8 3.12	0.35 21.2 1.86	0.05 112.2 0.35

9

Висновки

- В ході виконання магістерської роботи був зроблений обзор та аналіз сучасних робот та публікацій за тематикою згоркових нейронних мереж в мультиагентній кооперації та в результаті проведенних досліджень була розроблена модель згорткової мережі для мультиагентної кооперації.
- Також були розглянуті сучасні методи розробки нейронних мереж, проведений аналіз моделей мультиагентної кооперації.
- Розроблена модель на мові Python за допомогою інструментів TensorFlow та Keras. З їх допомогою довелося створити модель, яка розпізнає зображення з точністю 83%.

13

ДОДАТОК Б

Реалізація моделі

```
import numpy
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization,
Activation
from keras.constraints import maxnorm
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.datasets import cifar10

# fix random seed for reproducibility
seed = 21
numpy.random.seed(seed)

# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()

# normalize inputs from 0-255 to 0.0-1.0
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train = X_train / 255.0
X_test = X_test / 255.0

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# Create the model
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=X_train.shape[1:], padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Conv2D(128, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dropout(0.2))
```

```
model.add(Dense(256, kernel_constraint=maxnorm(3)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(128, kernel_constraint=maxnorm(3)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(num_classes))
model.add(Activation('softmax'))

epochs = 25
optimizer = 'Adam'

model.compile(loss='categorical_crossentropy', optimizer=optimizer,
metrics=['accuracy'])

print(model.summary())

model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs,
batch_size=64)

# Final evaluation of the model

scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

ДОДАТОК В

Результат друку

```
Results:
Layer (type) Output Shape Param #
=================================================
conv2d_1 (Conv2D) (None, 32, 32, 32) 896
_________________________________________________
activation_1 (Activation) (None, 32, 32, 32) 0
_________________________________________________
dropout_1 (Dropout) (None, 32, 32, 32) 0
_________________________________________________
batch_normalization_1 (Batch (None, 32, 32, 32) 128
_________________________________________________
conv2d_2 (Conv2D) (None, 32, 32, 64) 18496
_________________________________________________
activation_2 (Activation) (None, 32, 32, 64) 0
_________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64) 0
_________________________________________________
dropout_2 (Dropout) (None, 16, 16, 64) 0
_________________________________________________
batch_normalization_2 (Batch (None, 16, 16, 64) 256
_________________________________________________
conv2d_3 (Conv2D) (None, 16, 16, 64) 36928
_________________________________________________
activation_3 (Activation) (None, 16, 16, 64) 0
_________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64) 0
_________________________________________________
dropout_3 (Dropout) (None, 8, 8, 64) 0
_________________________________________________
batch_normalization_3 (Batch (None, 8, 8, 64) 256
_________________________________________________
conv2d_4 (Conv2D) (None, 8, 8, 128) 73856
_________________________________________________
activation_4 (Activation) (None, 8, 8, 128) 0
_________________________________________________
dropout_4 (Dropout) (None, 8, 8, 128) 0
_________________________________________________
batch_normalization_4 (Batch (None, 8, 8, 128) 512
_________________________________________________
```

```
flatten_1 (Flatten) (None, 8192) 0
_________________________________________________________________
dropout_5 (Dropout) (None, 8192) 0
_________________________________________________________________
dense_1 (Dense) (None, 256) 2097408
_________________________________________________________________
activation_5 (Activation) (None, 256) 0
_________________________________________________________________
dropout_6 (Dropout) (None, 256) 0
_________________________________________________________________
batch_normalization_5 (Batch (None, 256) 1024
_________________________________________________________________
dense_2 (Dense) (None, 128) 32896
_________________________________________________________________
activation_6 (Activation) (None, 128) 0
_________________________________________________________________
dropout_7 (Dropout) (None, 128) 0
_________________________________________________________________
batch_normalization_6 (Batch (None, 128) 512
_________________________________________________________________
dense_3 (Dense) (None, 10) 1290
_________________________________________________________________
activation_7 (Activation) (None, 10) 0
=================================================
Total params: 2,264,458
Trainable params: 2,263,114
Non-trainable params: 1,344
```

№	Позначення	Найменування	Дод. відомості
		Текстові документи	
1	ГЮІК.50ХХХХ.011 ПЗ	Пояснювальна записка	80 с.
		Графічні матеріали	
2		Слайд-презентація	10 слайдів

Змін.	Арк.	Номер докум.	Підп.	Дата	ГЮІК.50ХХХХ.011 ВД			
Розроб.		Тельний М.А.			Модель згорткової мережі для мультиагентної кооперації	Літ	Аркуш	Аркушів
Перевір.		Аксак Н.Г.				У	1	1
						ХНУРЕ Кафедра КІТС		
Н.контр.					Відомість атестаційної роботи			
Затв.		Руденко О.Г.						