

Звіт до комп'ютерного практикуму №3.

Рекурентні нейронні мережі

ПІБ: Грисюк Михайло Олександрович

Група: ІК-21мп

Мета роботи: ознайомитись з принципами побудови, навчання та використання мереж зі зворотніми зв'язками, дослідити вплив параметрів моделі, алгоритму навчання та даних на результати роботи.

Завдання: для задачі прогнозування (або іншої) на основі типового датасету створити рекурентну мережу. Навчити її, перевірити результат на тестовій вибірці, оцінити результати. Провести дослідження впливу параметрів на результати роботи мережі.

Номер варіанту: 5

Завдання для варіанту: Для задачі прогнозування на основі типового датасету **Boston Housing** створити рекурентну мережу. Навчити її, перевірити результат на тестовій вибірці, оцінити результати. Провести дослідження **впливу параметрів рекурентного шару** на результати роботи мережі. **Базова РНН.**

Засоби виконання практикуму: Дану комп'ютерну практику було виконано в середовищі VSCode зі спеціально встановленими розширеннями Jupiter та іншими. Використовувалась мова програмування Python та фреймворком TensorFlow, це є самі популярні інструменти для створення та навчання нейронних мереж.

Набір даних (датасет): Датасет *boston_housing* складаються із 505 прикладів. Це є вектор який складається з 13 елементів і результатом якого є число (MEDV). Ця інформація, зібрана Службою перепису населення США щодо житла в районі Бостона, Массачусетс. Нижче наведено опис стовпців набору даних:

- **CRIM** - рівень злочинності на душу населення по містах
- **ZN** - частка земель житлової забудови, виділених на ділянки площею понад 25 000 кв.
- **INDUS** - частка акрів нероздрібного бізнесу на місто.
- **CHAS** – фіктивна змінна річки Чарльз (1, якщо тракт обмежує річку; 0 інакше)
- **NOX** - концентрація оксидів азоту (частини на 10 мільйонів)
- **RM** - середня кількість кімнат на житло
- **AGE** - частка квартир, які займають власники, побудованих до 1940 року
- **DIS** - зважені відстані до п'яти бостонських центрів зайнятості
- **RAD** - індекс доступності радіальних магістралей
- **TAX** - ставка податку на повну вартість майна за 10 000 доларів США
- **PTRATIO** - співвідношення учень/вчитель за містом
- **B** - $1000(B_k - 0,63)^2$, де B_k - частка темношкірих у містах
- **LSTAT** - % нижчий статус населення
- **MEDV** – середня вартість будинків, зайнятих власниками, у 1000 доларів США

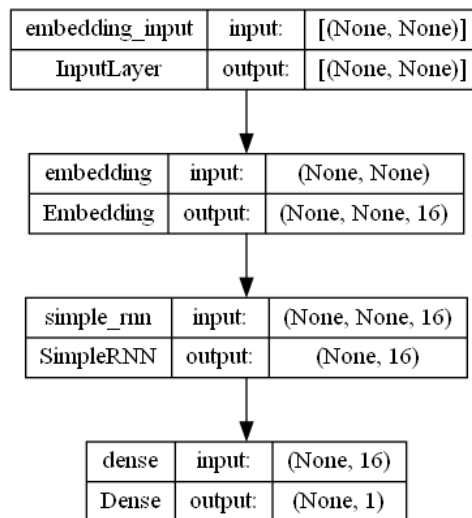
Boston Housing Dataset

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
14.4208	0.0	18.1	0.0	0.74	6.461	93.3	2.0026	24.0	666.0	20.2	27.49	18.05	9.6
0.51183	0.0	6.2	0.0	0.507	7.358	71.6	4.148	8.0	307.0	17.4	390.07	4.73	31.5
0.04297	52.5	5.32	0.0	0.405	6.565	22.9	7.3172	6.0	293.0	16.6	371.72	9.51	24.8
2.3139	0.0	19.58	0.0	0.605	5.88	97.3	2.3887	5.0	403.0	14.7	348.13	12.03	19.1
0.06899	0.0	25.65	0.0	0.581	5.87	69.7	2.2577	2.0	188.0	19.1	389.15	14.37	22.0
8.49213	0.0	18.1	0.0	0.584	6.348	86.1	2.0527	24.0	666.0	20.2	83.45	17.64	14.5
7.36711	0.0	18.1	0.0	0.679	6.193	78.1	1.9356	24.0	666.0	20.2	96.73	21.52	11.0
0.09604	40.0	6.41	0.0	0.447	6.854	42.8	4.2673	4.0	254.0	17.6	396.9	2.98	32.0
0.06664	0.0	4.05	0.0	0.51	6.546	33.1	3.1323	5.0	296.0	16.6	390.96	5.33	29.4
0.07165	0.0	25.65	0.0	0.581	6.004	84.1	2.1974	2.0	188.0	19.1	377.67	14.27	20.3
0.1146	20.0	6.96	0.0	0.464	6.538	58.7	3.9175	3.0	223.0	18.6	394.96	7.73	24.4
10.233	0.0	18.1	0.0	0.614	6.185	96.7	2.1705	24.0	666.0	20.2	379.7	18.03	14.6
0.1712	0.0	8.56	0.0	0.52	5.836	91.9	2.211	5.0	384.0	20.9	395.67	18.66	19.5
4.75237	0.0	18.1	0.0	0.713	6.525	86.5	2.4358	24.0	666.0	20.2	50.92	18.13	14.1
5.58107	0.0	18.1	0.0	0.713	6.436	87.9	2.3158	24.0	666.0	20.2	100.19	16.22	14.3
0.97617	0.0	21.89	0.0	0.624	5.757	98.4	2.346	4.0	437.0	21.2	262.76	17.31	15.6
22.0511	0.0	18.1	0.0	0.74	5.818	92.4	1.8662	24.0	666.0	20.2	391.45	22.11	10.5
9.91655	0.0	18.1	0.0	0.693	5.852	77.8	1.5004	24.0	666.0	20.2	338.16	29.97	6.3
0.37578	0.0	10.59	1.0	0.489	5.404	88.6	3.665	4.0	277.0	18.6	395.24	23.98	19.3
0.17142	0.0	6.91	0.0	0.448	5.682	33.8	5.1004	3.0	233.0	17.9	396.9	10.21	19.3

Попередня обробка даних: немає

Модель машинного навчання:

На вході у нас буде вектор 13x1 з числовими значеннями, далі буде рекурентний шар зі SimpleRNN і на виході буде шар з 1 нейроном . Виходить така конструкція мережі:



Код моделі навчання:

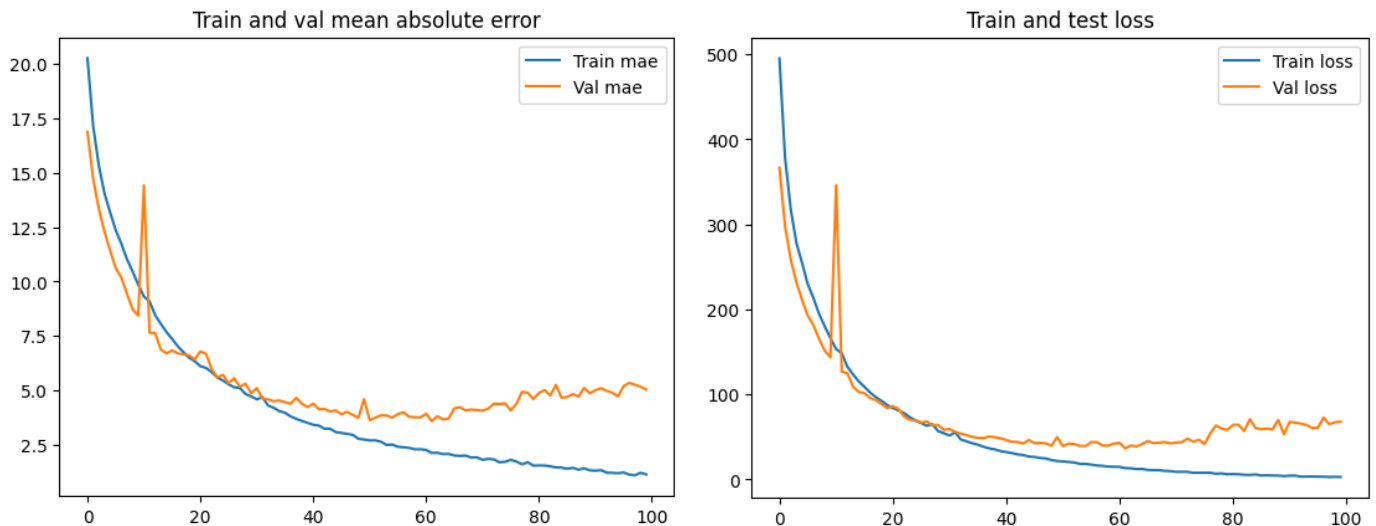
```
def simpleRNN_regression_model():
    model = tf.keras.Sequential()
    model.add(tf.keras.layers.Embedding(
        input_dim=max_value,
        output_dim=16,
        mask_zero=True))
    model.add(tf.keras.layers.SimpleRNN(16))
    model.add(tf.keras.layers.Dense(1))
    return model
```

Константи:

```
BATCH_SIZE = 20  
EPOCHS = 100
```

Навчання моделі: Тут ми використовували модель рекурентної мережі. Ми використовували алгоритм навчання лінійної регресії з визначенням максимально точного результату, а для функції втрат тут можна використати **RMSprop**.

Результати навчання:

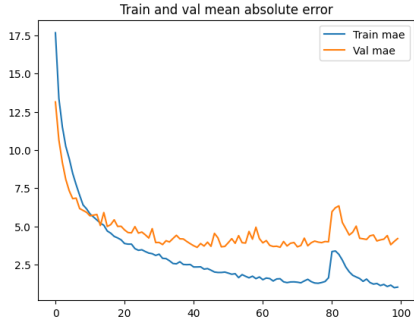
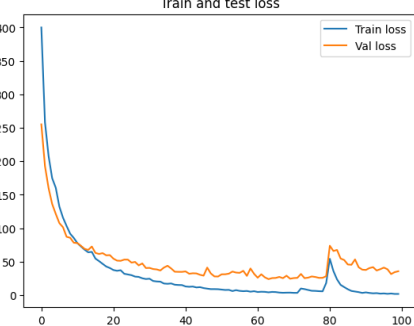


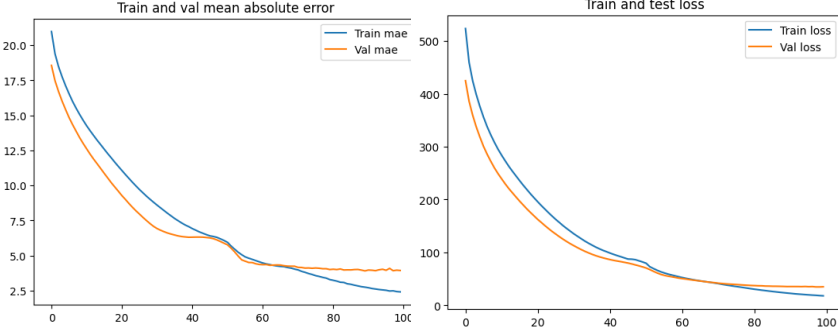
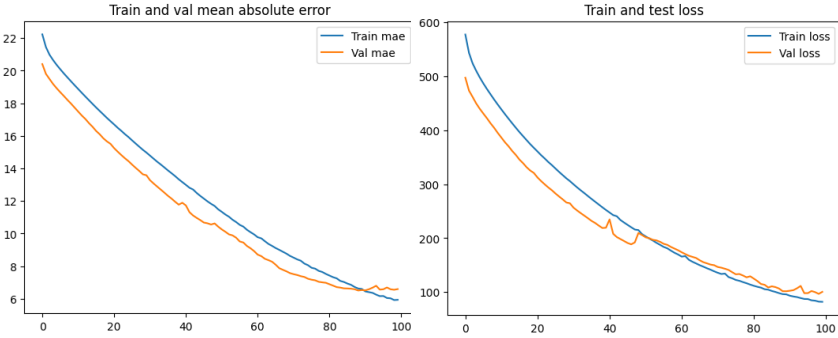
```
2/2 [=====] - 0s 5ms/step - loss: 36.5123 - mae: 4.5314  
test_loss : 36.51232147216797  
test_mae : 4.531397342681885
```

Оцінка результатів навчання: Як ми бачимо наша модель навчається, але в нас відбувається перенавчання приблизно 40-50 епох. Щоб покращити результат нам можна змінити параметр рекурентного шару або зменшити кількість епох.

Задача дослідження: Вплив параметрів рекурентного шару.

Результати експериментального дослідження:

Варіант покращення результату	Середня максимальна різниця	Втрати	Оцінка результату
SimpleRNN (64)			<pre>train_loss: 1.6812 train_mae: 1.0194 val_loss: 35.5835 val_mae: 4.1982 test_loss: 34.69 test_mae : 4.44</pre> <p>Погіршення результатів. Стрибки у графіку та інше вказують на перенавчання. Можна зменшити кількість епох.</p>

SimpleRNN (16)		<p> train_loss: 17.8964 train_mae: 2.4067 val_loss: 35.0120 val_mae: 3.9321 test_loss: 40.25 test_mae : 3.87 </p> <p>Найкраще покращення результату.</p>
SimpleRNN (8)		<p> train_loss: 81.9407 train_mae: 5.9377 val_loss: 100.4283 val_mae: 6.5952 test_loss: 132.99 test_mae : 8.315 </p> <p>Погіршення результатів. Відбувається недовчання, можна збільшити кількість епох.</p>

Висновки за результатами дослідження: Згідно нашими результатами можна зробити висновок, що при зміні параметрів рекурентного шару результати середньої абсолютної різниці та втрат змінюються. При збільшені значення параметру наша модель починає перенавчатись, а коли будемо зменшувати параметр – недонавчатись. Згідно наших результатів краще використовувати **SimpleRNN(16)** при заданих константах.