

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 6
по дисциплине «Проектирование интеллектуальных систем»

Тема: «Использование рекуррентных нейронных сетей для предсказания
временных рядов»

ИСПОЛНИТЕЛЬ:

группа ИУ5-24М

Шапиев М.М.

ФИО

подпись

"__" _____ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Терехов В.И.

ФИО

подпись

"__" _____ 2020 г.

Москва - 2020

Задание

На основе приведенного кода нужно выполнить 3 упражнения. Для успешного запуска нужно структурировать код следующим образом:

1. Генерация данных
2. Определение параметров нейросети
3. Описание графа вычислений
4. Описание функции потерь и оптимизатора
5. Обучение и валидация нейронной сети
6. Визуализация результатов

Упражнение 1. X - обучающий набор данных, Y - тестовый набор данных. Возвращаемые массивы имеют форму (seq_length, batch_size, output_dim). Для данного упражнения параметр isTrain можно игнорировать, т.к. тестирование происходит на тех же данных, что и обучение. Всего генерируется 2 ряда для обучающей и контрольной выборки, это можно интерпретировать как двумерные данные.

Упражнение 2.

Функция генерации по сути аналогична функции генерации для 1-го упражнения, но теперь генерируется сигнал, являющийся суперпозицией двух частот, выбранных случайно. Длина последовательности составляет 15 элементов (т.е. всего получается 30 элементов, 15 для прошлого и 15 для будущего).

Упражнение 3. Здесь функция генерации похожа на функцию генерации для упражнения 2, но в значения X введены шумы. Длина последовательности составляет 30 элементов.

Реализация

1) Упражнение 1

Параметры нейросети:

```
1  #Набор обучающих и тестовых данных
2  sample_x, sample_y = generate_x_y_data_v2(isTrain = True, batch_size = 3)
3
4  #Длина последовательности
5  seq_length = sample_x.shape[0]
6  #Размер пакета количество, по которому усредняется градиент
7  batch_size = 5
8
9  #Размерность выходных данных
10 output_dim = input_dim = sample_x.shape[-1]
11 #Количество скрытых нейронов в каждой ячейке
12 hidden_dim = 12
13 #Количество ячеек
14 layers_stacked_count = 2
15
16 #Параметры оптимизатора
17 #Скорость обучения
18 learning_rate = 0.007
19 #Количество итераций по обучающей выборке
20 nb_iters = 150
21 #Дополнительные параметры алгоритма оптимизации
22 lr_decay = 0.92
23 momentum = 0.5
24 # Коэффициент L2 регуляризации
25 lambda_l2_reg = 0.003
```

Значения функции потерь:

Step0/150,train loss:16.2618465423584,	TEST loss:8.940359115600586
Step10/150,train loss:1.7378686666488647,	TEST loss:2.9567458629608154
Step20/150,train loss:1.2143529653549194,	TEST loss:0.7605609893798828
Step30/150,train loss:0.8891664743423462,	TEST loss:0.7118458151817322
Step40/150,train loss:0.611620306968689,	TEST loss:0.605688750743866
Step50/150,train loss:0.727796733379364,	TEST loss:0.5906921625137329
Step60/150,train loss:0.5477001667022705,	TEST loss:0.6039848923683167
Step70/150,train loss:0.644220232963562,	TEST loss:0.4937879741191864
Step80/150,train loss:0.9800869226455688,	TEST loss:0.8530012369155884
Step90/150,train loss:1.4052412509918213,	TEST loss:1.2518302202224731
Step100/150,train loss:0.551878809928894,	TEST loss:0.5354585647583008
Step110/150,train loss:0.4329961836338043,	TEST loss:0.41733497381210327
Step120/150,train loss:0.43849197030067444,	TEST loss:0.4546097218990326
Step130/150,train loss:0.4534367322921753,	TEST loss:0.6511718034744263
Step140/150,train loss:0.5010781288146973,	TEST loss:0.34124743938446045
Step150/150,train loss:0.3854043781757355,	TEST loss:0.41447481513023376
Fin.train loss:0.3854043781757355,	TEST loss:0.41447481513023376

График изменения ошибки:

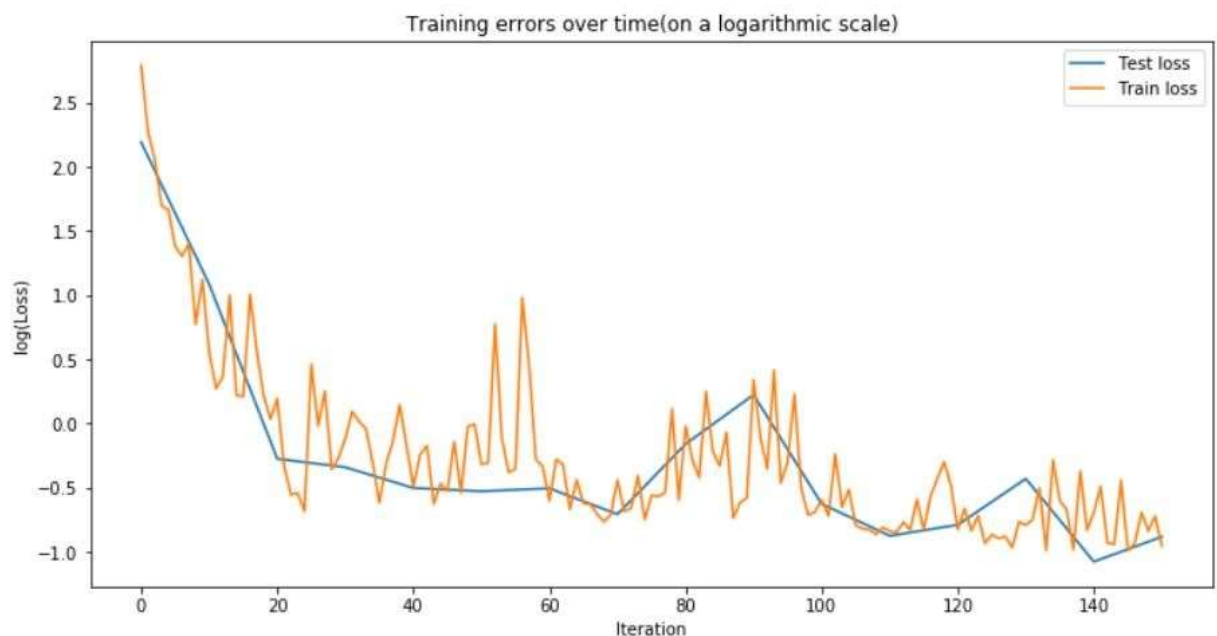
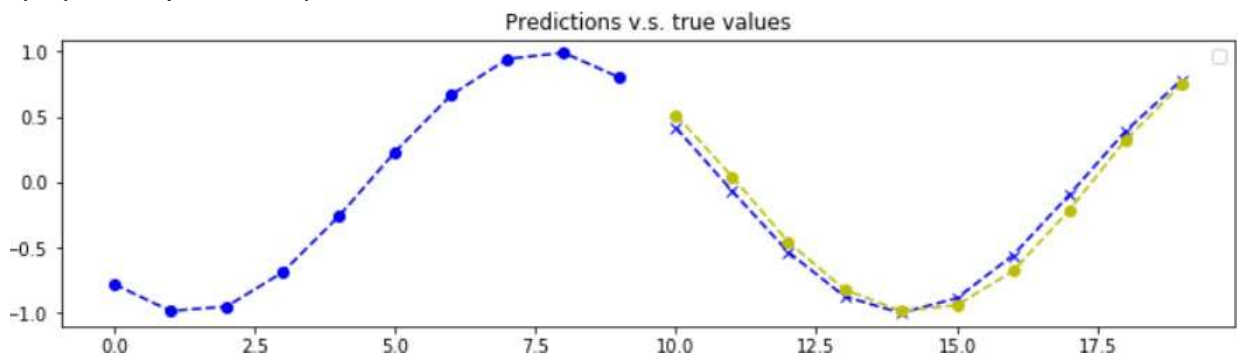


График полученных предсказаний:



2) Упражнение 2

Параметры нейросети:

```
1 # Данные имеют размерность (seq_length , batch_size , output_dim)
2 sample_x , sample_y = generate_x_y_data_v2(isTrain=True , batch_size=3)
3
4 # Длина последовательности (в данных примерах одинаковая для обучающих и тестовых данных)
5 seq_length = sample_x.shape[0]
6
7 # Размер пакета количество(тестовых примеров), по которому усредняется градиент
8 batch_size = 120
9 # Размерность выходных данных
10 output_dim = input_dim = sample_x.shape[-1]
11 # Количество скрытых нейронов в каждой ячейке
12 hidden_dim = 70
13 # Количество ячеек рекуррентной сети (в глубину)
14 layers_stacked_count = 2
15
16 # Параметры оптимизатора
17 # Скорость обучения маленькая (скорость обучения позволяет алгоритму не расходиться во время обучения)
18 learning_rate = 0.007
19 # Количество итераций по обучающей выборке
20 nb_iters = 2800
21 # Дополнительные параметры алгоритма оптимизации
22 lr_decay = 0.8
23 momentum = 0.3
24 # Коэффициент L2 регуляризации
25 lambda_l2_reg = 1e-12
```

Значения функции потерь:

Step2630/2800,train loss:58.934688568115234,	TEST loss:81.41954803466797
Step2640/2800,train loss:56.0072135925293,	TEST loss:62.38401412963867
Step2650/2800,train loss:70.33123016357422,	TEST loss:95.32917022705078
Step2660/2800,train loss:71.82191467285156,	TEST loss:68.03392028808594
Step2670/2800,train loss:72.50523376464844,	TEST loss:67.37962341308594
Step2680/2800,train loss:53.35747528076172,	TEST loss:77.05794525146484
Step2690/2800,train loss:39.97764587402344,	TEST loss:83.2818374633789
Step2700/2800,train loss:68.40777587890625,	TEST loss:99.17269134521484
Step2710/2800,train loss:70.8269271850586,	TEST loss:92.32915496826172
Step2720/2800,train loss:71.65814971923828,	TEST loss:82.37958526611328
Step2730/2800,train loss:44.4577522277832,	TEST loss:50.11127853393555
Step2740/2800,train loss:50.22694778442383,	TEST loss:52.58091354370117
Step2750/2800,train loss:46.955421447753906,	TEST loss:68.56304931640625
Step2760/2800,train loss:75.70114135742188,	TEST loss:53.27568435668945
Step2770/2800,train loss:39.30817413330078,	TEST loss:47.11541748046875
Step2780/2800,train loss:67.09012603759766,	TEST loss:69.53251647949219
Step2790/2800,train loss:75.06305694580078,	TEST loss:64.72932434082031
Step2800/2800,train loss:72.59973907470703,	TEST loss:72.10797882080078
Fin.train loss:72.59973907470703,	TEST loss:72.10797882080078

График изменения ошибки:

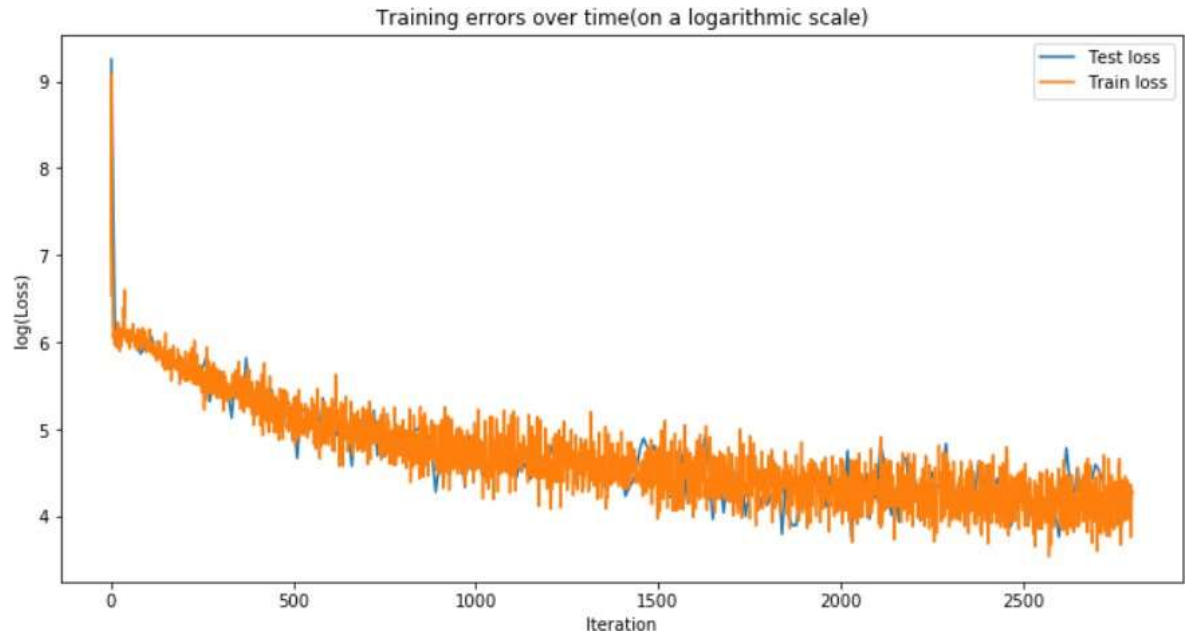
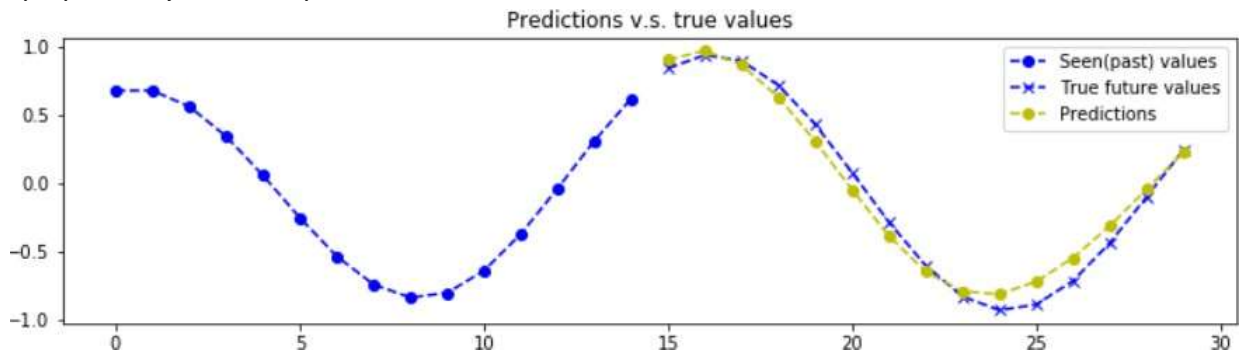


График полученных предсказаний:



3) Упражнение 3

Параметры нейросети:

```
1 # Данные имеют размерность (seq_length, batch_size, output_dim)
2 sample_x, sample_y = generate_x_y_data_v3(isTrain=True, batch_size=3)
3
4 # Длина последовательности (в данных примерах одинаковая для обучающих и тестовых данных)
5 seq_length = sample_x.shape[0]
6
7 # Размер пакета количество(тестовых примеров), по которому усредняется градиент
8 batch_size = 130
9 # Размерность выходных данных
10 output_dim = input_dim = sample_x.shape[-1]
11 # Количество скрытых нейронов в каждой ячейке
12 hidden_dim = 60
13 # Количество ячеек рекуррентной сети (в глубину)
14 layers_stacked_count = 2
15
16 # Параметры оптимизатора
17 # Скорость обучения маленькая (скорость обучения позволяет алгоритму не расходиться во время обучения)
18 learning_rate = 0.007
19 # Количество итераций по обучающей выборке
20 nb_iters = 3000
21 # Дополнительные параметры алгоритма оптимизации
22 lr_decay = 0.8
23 momentum = 0.3
24 # Коэффициент L2 регуляризации
25 lambda_l2_reg = 1e-12
```


Значения функции потерь:

Step2830/3000,train	loss:105.11546325683594,	TEST	loss:121.11521911621094
Step2840/3000,train	loss:103.68354034423828,	TEST	loss:94.68636322021484
Step2850/3000,train	loss:112.65629577636719,	TEST	loss:99.17383575439453
Step2860/3000,train	loss:84.44066619873047,	TEST	loss:100.52761840820312
Step2870/3000,train	loss:107.45015716552734,	TEST	loss:116.56285858154297
Step2880/3000,train	loss:92.41710662841797,	TEST	loss:70.2393798828125
Step2890/3000,train	loss:97.6463851928711,	TEST	loss:103.5279541015625
Step2900/3000,train	loss:94.96153259277344,	TEST	loss:98.58307647705078
Step2910/3000,train	loss:84.53778839111328,	TEST	loss:106.72590637207031
Step2920/3000,train	loss:78.39085388183594,	TEST	loss:112.3362045288086
Step2930/3000,train	loss:95.72964477539062,	TEST	loss:88.39141082763672
Step2940/3000,train	loss:86.88092803955078,	TEST	loss:132.61337280273438
Step2950/3000,train	loss:92.83494567871094,	TEST	loss:144.47769165039062
Step2960/3000,train	loss:89.96983337402344,	TEST	loss:115.16890716552734
Step2970/3000,train	loss:94.01921844482422,	TEST	loss:88.25679779052734
Step2980/3000,train	loss:108.96233367919922,	TEST	loss:86.05457305908203
Step2990/3000,train	loss:85.21257781982422,	TEST	loss:92.31494140625
Step3000/3000,train	loss:103.99127197265625,	TEST	loss:116.44924926757812
Fin.train	loss:103.99127197265625,	TEST	loss:116.44924926757812

График изменения ошибки:

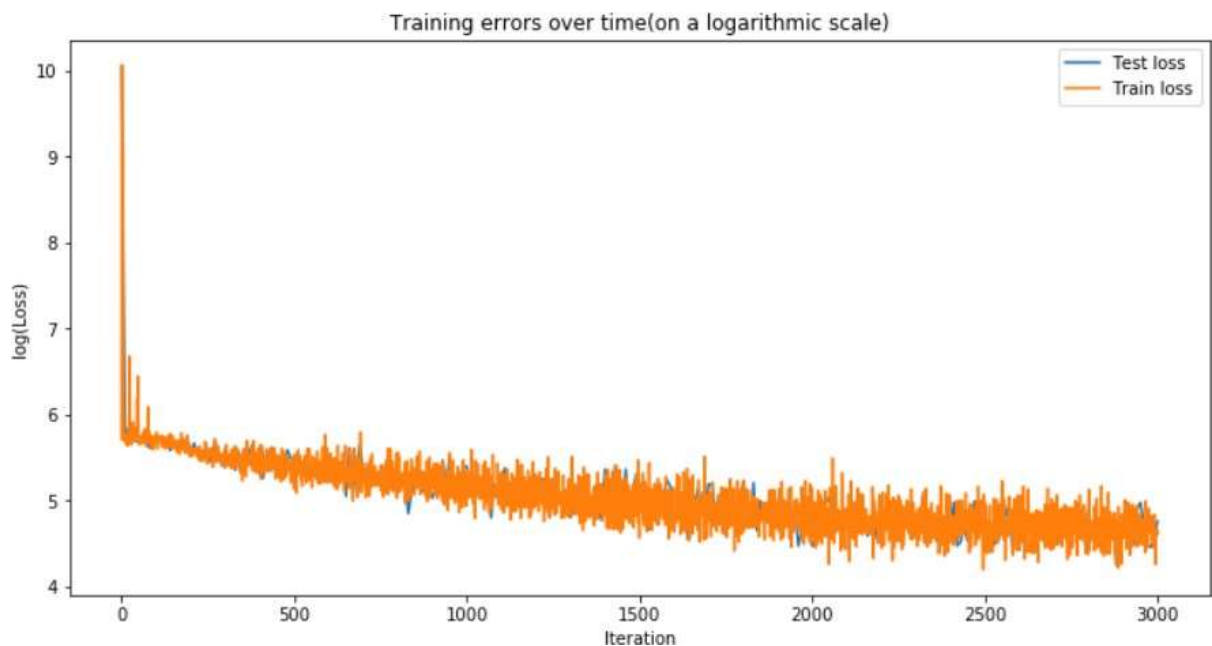
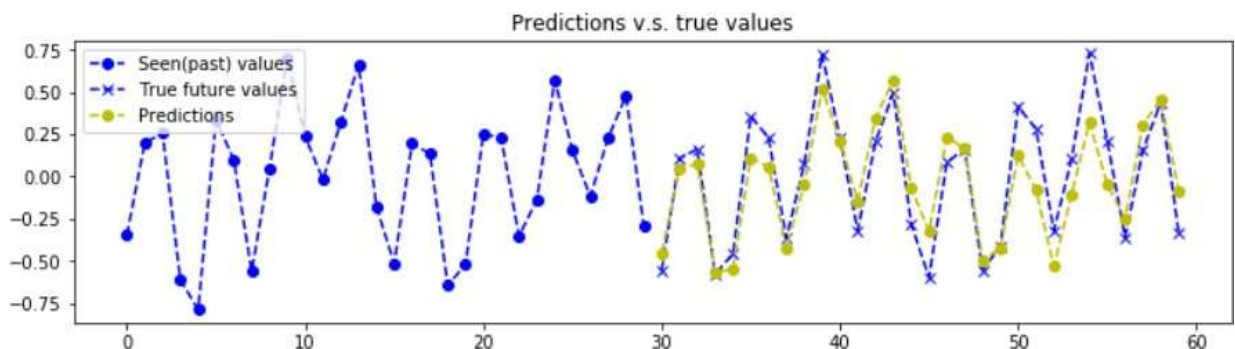


График полученных предсказаний:



Контрольные вопросы

1. В чём преимущество рекуррентных нейронных сетей по сравнению с обычными персептронами?

Рекуррентные сети выполняют одну и ту же задачу для каждого элемента последовательности, выход зависит от результатов предыдущих вычислений.

2. Что такое регуляризации и зачем она нужна?

Метод добавления некоторых дополнительных ограничений к условию с целью решить некорректно поставленную задачу или предотвратить переобучение.

3. Что такое пакетный, мини-пакетный и онлайнный градиентный спуск?

Пакетный градиентный спуск: параметры обновляются после вычисления градиента ошибки по всему обучающему набору.

Мини-пакетный градиентный спуск: параметры обновляются после вычисления градиента ошибки относительно подмножества обучающего набора.

Стохастический(онлайнный) градиентный спуск: параметры обновляются после вычисления градиента ошибки по отношению к одному обучающему примеру.

Литература

[1] Документация по tensorflow. <https://www.tensorflow.org/>.

[2] J. Brownlee. Encoder-decoder recurrent neural network models for neural machine translation. <https://machinelearningmastery.com/encoder-decoder-recurrent-neural-network-models-neural-machine-translation/>.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.

[4] Архангельская Е. Николенко С., Кадурын А. Глубокое обучение. Погружение в мир нейронных сетей. СПб.: Питер, 2018.