Лабораторная работа № 2 по Нейроинформатике

Линейная нейронная сеть. Правило обучения Уидроу-Хоффа

Выполнила: Тимофеева Наталья М8О-408Б-19

Вариант № 16

Часть 1

Аппроксимация

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras import layers
from matplotlib import pyplot as plt
from timeit import default_timer as timer
```

Задаём сигнал

```
def signal(t):
    return np.sin(-np.sin(t) * t * t + t)
```

Создание модели

```
epochs = 15
discrete = 5
t = np.arange(0.5, 4, 0.01)
y = signal(t).tolist()

model = keras.models.Sequential()
model.add(keras.layers.Dense(1, input_dim = discrete, 
activation='linear',
                             kernel_initializer = 
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0),
                             bias_initializer = 
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0)))

optimazer = keras.optimizers.SGD(learning_rate = 0.01)
model.compile(loss = 'mse', optimizer = optimazer, metrics = ['mae'])
```

Подготовка данных

```
windows = [y[i:i + discrete] for i in range(0, len(y) - discrete)]
expectations = [y[i] for i in range(discrete, len(y))]
# Проверка соответствия размеров
assert len(windows) == len(expectations)
```

Обучение модели

```
time_start = timer()
hist = model.fit(windows, expectations, batch_size = 1, epochs = 
epochs, shuffle = True)
time_end = timer()
```

```
Epoch 1/15
345/345 [==============================] - 0s 634us/step - loss:
0.0439 - mae: 0.1408
Epoch 2/15
345/345 [==============================] - 0s 656us/step - loss:
0.0242 - mae: 0.1012
Epoch 3/15
345/345 [==============================] - 0s 669us/step - loss:
0.0156 - mae: 0.0808
Epoch 4/15
345/345 [==============================] - 0s 689us/step - loss:
0.0101 - mae: 0.0654
Epoch 5/15
345/345 [==============================] - 0s 641us/step - loss:
0.0066 - mae: 0.0534
Epoch 6/15
345/345 [==============================] - 0s 755us/step - loss:
0.0043 - mae: 0.0432
Epoch 7/15
345/345 [==============================] - 0s 660us/step - loss:
0.0028 - mae: 0.0359
Epoch 8/15
345/345 [==============================] - 0s 672us/step - loss:
0.0019 - mae: 0.0299
Epoch 9/15
345/345 [==============================] - 0s 686us/step - loss:
0.0013 - mae: 0.0250
Epoch 10/15
345/345 [==============================] - 0s 632us/step - loss:
8.9070e-04 - mae: 0.0215
Epoch 11/15
345/345 [==============================] - 0s 686us/step - loss:
6.4744e-04 - mae: 0.0183
Epoch 12/15
345/345 [==============================] - 0s 647us/step - loss:
4.8768e-04 - mae: 0.0166
Epoch 13/15
345/345 [==============================] - 0s 626us/step - loss:
3.8476e-04 - mae: 0.0150
Epoch 14/15
345/345 [==============================] - 0s 648us/step - loss:
3.1272e-04 - mae: 0.0136
Epoch 15/15
345/345 [==============================] - 0s 680us/step - loss:
2.7072e-04 - mae: 0.0127
```

```
print('Эпох: {0}'.format(epochs))
print('Время обучения: {0} секунд'.format(int(time_end - time_start)))
print('Функция потерь MSE: {0}'.format(min(hist.history['loss'])))
print('Метрика качества MAE: {0}'.format(min(hist.history['mae'])))
```

```
Эпох: 15
Время обучения: 3 секунд
Функция потерь MSE: 0.00027072490775026382
Метрика качества MAE: 0.012732269242405891
```

Предсказание

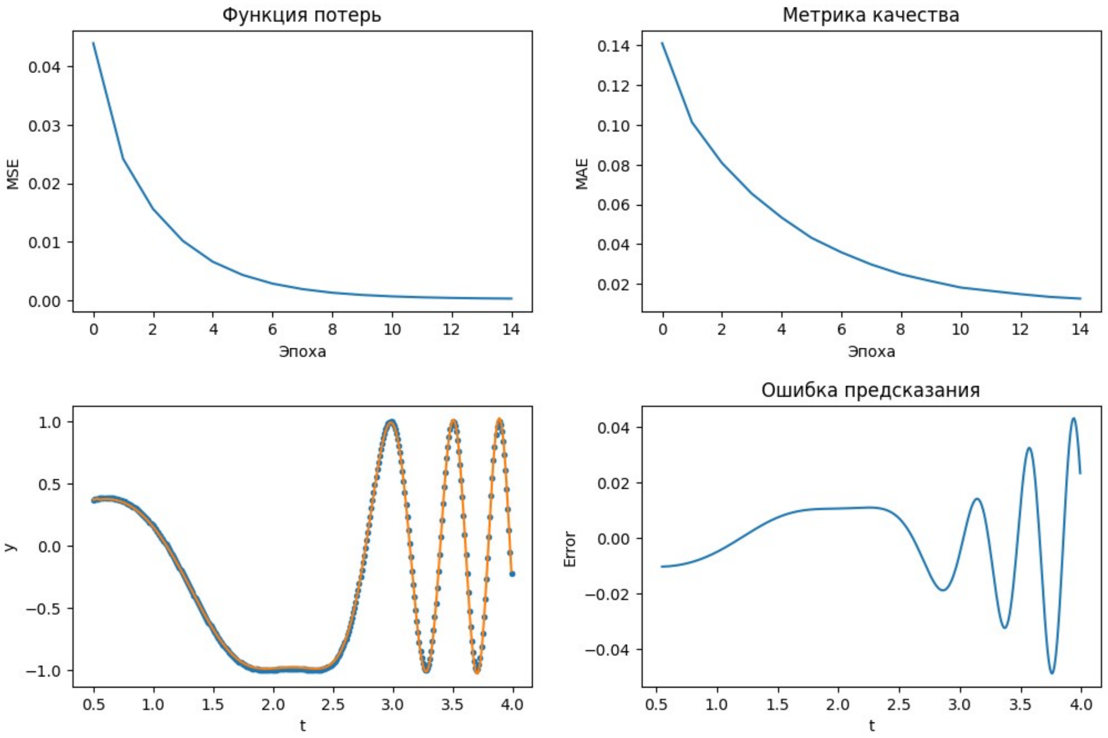
```
y_predicted = y[:discrete]
errors = []
for i in range(0, len(expectations)):
    predicted = model.predict([windows[i]])
    y_predicted += [predicted]
    errors += [predicted.item() - expectations[i]]
```

```
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 28ms/step
```

```
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 41ms/step
```

```
fig, axes = plt.subplots(2, 2, figsize=(10, 6.5))
fig.tight_layout(h_pad = 4, w_pad = 4)

axes[0, 0].set_title('Функция потерь')
axes[0, 0].set_xlabel('Эпоха')
axes[0, 0].set_ylabel('MSE')
axes[0, 0].plot(hist.history['loss'])

axes[0, 1].set_title('Метрика качества')
axes[0, 1].set_xlabel('Эпоха')
axes[0, 1].set_ylabel('MAE')
axes[0, 1].plot(hist.history['mae'])

axes[1, 0].set_xlabel('t')
axes[1, 0].set_ylabel('y')
axes[1, 0].plot(t, y, '.')
axes[1, 0].plot(t, y_predicted)

axes[1, 1].set_title('Ошибка предсказания')
axes[1, 1].set_xlabel('t')
axes[1, 1].set_ylabel('Error')
axes[1, 1].plot(t[discrete:], errors)
```

```
[<matplotlib.lines.Line2D at 0x236628a1480>]
```


Функция потерь
MSE
0.04
0.03
0.02
0.01
0.00
0
2
4
6
8
10
12
14
Эпоха
Метрика качества
MAE
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0
2
4
6
8
10
12
14
Эпоха
y
1.0
0.5
0.0
−0.5
−1.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
t
Ошибка предсказания
Error
0.04
0.02
0.00
−0.02
−0.04
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
t

Лабораторная работа № 2 по Нейроинформатике

Линейная нейронная сеть. Правило обучения Уидроу-Хоффа

Выполнила: Тимофеева Наталья М8О-408Б-19

Вариант № 16

Часть 2

Фильтрация

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras import layers
from matplotlib import pyplot as plt
from timeit import default_timer as timer
```

Задаём сигналы

```
def noised_signal(t):
    return (1 / 8) * np.cos(-5 * t ** 2 + 10 * t - 5)
def true_signal(t):
    return np.cos(-5 * t ** 2 + 10 * t - 5)
```

Создание модели

```
epochs = 30
discrete = 5
t = np.arange(0, 2.5, 0.01)
y_noised = noised_signal(t).tolist()
y_true = true_signal(t).tolist()

model = keras.models.Sequential()
model.add(keras.layers.Dense(1, input_dim = discrete,
activation='linear',
                             kernel_initializer =
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0),
                             bias_initializer =
keras.initializers.RandomNormal(stddev = 0.5, mean = 0.0)))

optimazer = keras.optimizers.SGD(learning_rate = 0.01)
model.compile(loss = 'mse', optimizer = optimazer, metrics = ['mae'])
```

Подготовка данных

```
windows = [y_noised[i:i + discrete] for i in range(0, len(y_noised) -
discrete)]
expectations = [y_true[i] for i in range(discrete, len(y_true))]
# Проверка соответствия размеров
assert len(windows) == len(expectations)
```

Обучение модели

```
time_start = timer()
hist = model.fit(windows, expectations, batch_size = 1, epochs =
epochs, shuffle = True)
time_end = timer()
```

```
Epoch 1/30
245/245 [==============================] - 0s 834us/step - loss:
0.3514 - mae: 0.5334
Epoch 2/30
245/245 [==============================] - 0s 762us/step - loss:
0.2320 - mae: 0.4367
Epoch 3/30
245/245 [==============================] - 0s 750us/step - loss:
0.1611 - mae: 0.3661
Epoch 4/30
245/245 [==============================] - 0s 766us/step - loss:
0.1122 - mae: 0.3028
Epoch 5/30
245/245 [==============================] - 0s 740us/step - loss:
0.0810 - mae: 0.2573
Epoch 6/30
245/245 [==============================] - 0s 746us/step - loss:
0.0604 - mae: 0.2220
Epoch 7/30
245/245 [==============================] - 0s 766us/step - loss:
0.0468 - mae: 0.1936
Epoch 8/30
245/245 [==============================] - 0s 734us/step - loss:
0.0379 - mae: 0.1719
Epoch 9/30
245/245 [==============================] - 0s 751us/step - loss:
0.0320 - mae: 0.1575
Epoch 10/30
245/245 [==============================] - 0s 737us/step - loss:
0.0280 - mae: 0.1456
Epoch 11/30
245/245 [==============================] - 0s 732us/step - loss:
0.0255 - mae: 0.1355
Epoch 12/30
245/245 [==============================] - 0s 728us/step - loss:
0.0236 - mae: 0.1299
Epoch 13/30
245/245 [==============================] - 0s 709us/step - loss:
0.0225 - mae: 0.1234
Epoch 14/30
245/245 [==============================] - 0s 853us/step - loss:
0.0218 - mae: 0.1218
Epoch 15/30
245/245 [==============================] - 0s 775us/step - loss:
```

```
0.0211 - mae: 0.1184
Epoch 16/30
245/245 [==============================] - 0s 753us/step - loss:
0.0208 - mae: 0.1148
Epoch 17/30
245/245 [==============================] - 0s 734us/step - loss:
0.0206 - mae: 0.1162
Epoch 18/30
245/245 [==============================] - 0s 705us/step - loss:
0.0203 - mae: 0.1144
Epoch 19/30
245/245 [==============================] - 0s 683us/step - loss:
0.0202 - mae: 0.1113
Epoch 20/30
245/245 [==============================] - 0s 735us/step - loss:
0.0201 - mae: 0.1112
Epoch 21/30
245/245 [==============================] - 0s 762us/step - loss:
0.0200 - mae: 0.1107
Epoch 22/30
245/245 [==============================] - 0s 700us/step - loss:
0.0198 - mae: 0.1106
Epoch 23/30
245/245 [==============================] - 0s 755us/step - loss:
0.0197 - mae: 0.1105
Epoch 24/30
245/245 [==============================] - 0s 732us/step - loss:
0.0194 - mae: 0.1087
Epoch 25/30
245/245 [==============================] - 0s 752us/step - loss:
0.0193 - mae: 0.1091
Epoch 26/30
245/245 [==============================] - 0s 747us/step - loss:
0.0195 - mae: 0.1080
Epoch 27/30
245/245 [==============================] - 0s 745us/step - loss:
0.0193 - mae: 0.1092
Epoch 28/30
245/245 [==============================] - 0s 770us/step - loss:
0.0193 - mae: 0.1076
Epoch 29/30
245/245 [==============================] - 0s 751us/step - loss:
0.0192 - mae: 0.1081
Epoch 30/30
245/245 [==============================] - 0s 764us/step - loss:
0.0192 - mae: 0.1073
```

```python
print('Эпох: {0}'.format(epochs))
print('Время обучения: {0} секунд'.format(int(time_end - time_start)))
print('Функция потерь MSE: {0}'.format(min(hist.history['loss'])))
print('Метрика качества MAE: {0}'.format(min(hist.history['mae'])))
```

```
Эпох: 30
Время обучения: 6 секунд
Функция потерь MSE: 0.01922464370727539
Метрика качества MAE: 0.10733126103878021
```

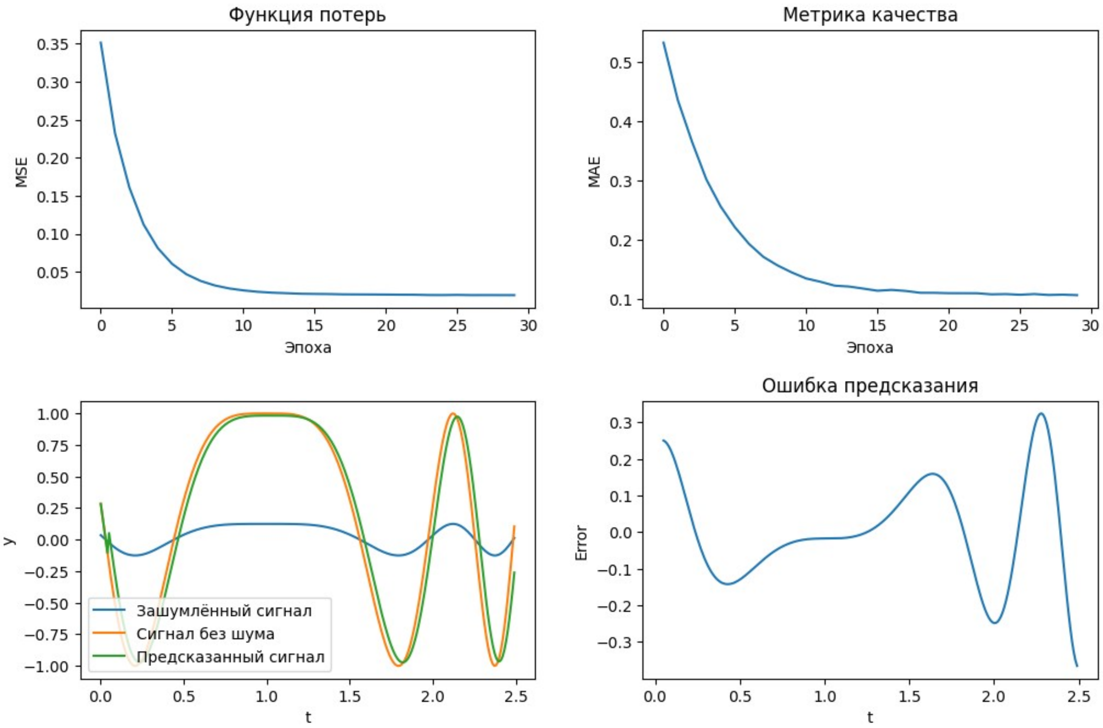
Предсказание

```
predicted = model.predict(windows)
y_predicted = y_true[:discrete] + predicted.flatten().tolist()
errors = [i - j for i, j in zip(predicted, expectations)]
```

```
8/8 [==============================] - 0s 863us/step
```

```
fig, axes = plt.subplots(2, 2, figsize=(10, 6.5))
fig.tight_layout(h_pad = 4, w_pad = 4)

axes[0, 0].set_title('Функция потерь')
axes[0, 0].set_xlabel('Эпоха')
axes[0, 0].set_ylabel('MSE')
axes[0, 0].plot(hist.history['loss'])

axes[0, 1].set_title('Метрика качества')
axes[0, 1].set_xlabel('Эпоха')
axes[0, 1].set_ylabel('MAE')
axes[0, 1].plot(hist.history['mae'])

axes[1, 0].set_xlabel('t')
axes[1, 0].set_ylabel('y')
axes[1, 0].plot(t, y_noised, label = 'Зашумлённый сигнал')
axes[1, 0].plot(t, y_true, label = 'Сигнал без шума')
axes[1, 0].plot(t, y_predicted, label = 'Предсказанный сигнал')
axes[1, 0].legend()

axes[1, 1].set_title('Ошибка предсказания')
axes[1, 1].set_xlabel('t')
axes[1, 1].set_ylabel('Error')
axes[1, 1].plot(t[discrete:], errors)
```

```
[<matplotlib.lines.Line2D at 0x14cb494a680>]
```


Функция потерь
MSE
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
5
10
15
20
25
30
Эпоха
Метрика качества
MAE
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
Эпоха
y
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Зашумлённый сигнал
Сигнал без шума
Предсказанный сигнал
0.0
0.5
1.0
1.5
2.0
2.5
t
Ошибка предсказания
Error
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.5
1.0
1.5
2.0
2.5
t