Лабораторная работа № **4** по Нейроинформатике

Сети с радиальными базисными элементами

Выполнила**:** Тимофеева Наталья М**8**О**-408**Б**-19**

Вариант № **16**

Часть **1**

Классификация

In [237]:

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

In [238]:

```
class RBFLayer(keras.layers.Layer):
    def __init__(self, output_dim, **kwargs):
        self.output_dim = output_dim
        super(RBFLayer, self).__init__(**kwargs)

    def build(self, input_shape):
        self.mu = self.add_weight(name = "mu",
                                  shape = (input_shape[1], self.output_dim),
                                  initializer = tf.keras.initializers.RandomUniform(minv
al = -1, maxval = 1),
                                  trainable = True)
        self.sigma = self.add_weight(name = "sigma",
                                     shape = (self.output_dim,),
                                     initializer = "random_normal",
                                     trainable = True)
        super(RBFLayer, self).build(input_shape)

    def call(self, inputs):
        diff = backend.expand_dims(inputs) - self.mu
        output = backend.exp(backend.sum(diff ** 2, axis = 1) * self.sigma)
        return output
```

In [239]:

```
def ellipse(t, a, b, x0, y0):
    x = x0 + a * np.cos(t)
    y = y0 + b * np.sin(t)
    return x, y
```

In [240]:

```
def parabola(t, p, x0, y0):
    x = x0 + t ** 2 / (2. * p)
    y = y0 + t
    return x, y
```

In [241]:

```
def rotate(x, y, alpha):
    xr = x * np.cos(alpha) - y * np.sin(alpha)
    yr = x * np.sin(alpha) - y * np.cos(alpha)
```

```
    return xr, yr
```

In [242]:

```
t = np.arange(0, 2 * np.pi, 0.001)
```

Создаём эллипсы и параболу

In [243]:

```
x1, y1 = ellipse(t, 0.4, 0.15, 0.1, -0.15)
x1, y1 = rotate(x1, y1, np.pi / 6)

x2, y2 = ellipse(t, 0.7, 0.5, 0, 0)
x2, y2 = rotate(x2, y2, np.pi / 3)

x3, y3 = parabola(t, 1, 0, -0.8)
x3, y3 = rotate(x3, y3, np.pi / 2)
```

Преобразуем данные

In [244]:

```
d1 = [[[x, y], [1., 0., 0.]] for x, y in zip(x1, y1)]
d2 = [[[x, y], [0., 1., 0.]] for x, y in zip(x2, y2)]
d3 = [[[x, y], [0., 0., 1.]] for x, y in zip(x3, y3)]
```

Объединяем и перемешиваем

In [245]:

```
dataset = d1 + d2 + d3
np.random.shuffle(dataset)
```

Разбиваем на обучающую и тестовую выборки

In [246]:

```
separ = int(len(dataset) * 0.8)
train_data = dataset[:separ]
test_data = dataset[separ:]

train_input = [i[0] for i in train_data]
train_output = [i[1] for i in train_data]
```

Создание модели

In [247]:

```
model = keras.models.Sequential()
model.add(RBFLayer(3, input_dim = 2))
model.add(keras.layers.Dense(3, activation = 'sigmoid'))
model.compile(loss = 'mse', optimizer = 'adam', metrics = ['mae'])
```

Обучение модели

In [248]:

```
len(train_data)
```

Out[248]:

```
15081
```

In [249]:

```
epochs = 1000
```

```
batch_size = int(len(train_data) * 0.01)

time_start = timer()
hist = model.fit(train_input, train_output, batch_size = batch_size, epochs = epochs)
time_end = timer()
```

```
Epoch 1/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.2787 - mae: 0.4703
Epoch 2/1000
101/101 [==============================] - 0s 918us/step - loss: 0.2339 - mae: 0.4480
Epoch 3/1000
101/101 [==============================] - 0s 898us/step - loss: 0.2183 - mae: 0.4362
Epoch 4/1000
101/101 [==============================] - 0s 898us/step - loss: 0.2068 - mae: 0.4264
Epoch 5/1000
101/101 [==============================] - 0s 888us/step - loss: 0.1981 - mae: 0.4174
Epoch 6/1000
101/101 [==============================] - 0s 967us/step - loss: 0.1914 - mae: 0.4097
Epoch 7/1000
101/101 [==============================] - 0s 898us/step - loss: 0.1859 - mae: 0.4022
Epoch 8/1000
101/101 [==============================] - 0s 918us/step - loss: 0.1813 - mae: 0.3959
Epoch 9/1000
101/101 [==============================] - 0s 937us/step - loss: 0.1772 - mae: 0.3900
Epoch 10/1000
101/101 [==============================] - 0s 993us/step - loss: 0.1736 - mae: 0.3846
Epoch 11/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1703 - mae: 0.3797
Epoch 12/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1674 - mae: 0.3749
Epoch 13/1000
101/101 [==============================] - 0s 897us/step - loss: 0.1645 - mae: 0.3702
Epoch 14/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1618 - mae: 0.3659
Epoch 15/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1592 - mae: 0.3613
Epoch 16/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1566 - mae: 0.3571
Epoch 17/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1541 - mae: 0.3528
Epoch 18/1000
101/101 [==============================] - 0s 908us/step - loss: 0.1517 - mae: 0.3486
Epoch 19/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1494 - mae: 0.3445
Epoch 20/1000
101/101 [==============================] - 0s 965us/step - loss: 0.1472 - mae: 0.3404
Epoch 21/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1451 - mae: 0.3366
Epoch 22/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1430 - mae: 0.3325
Epoch 23/1000
101/101 [==============================] - 0s 938us/step - loss: 0.1412 - mae: 0.3290
Epoch 24/1000
101/101 [==============================] - 0s 957us/step - loss: 0.1394 - mae: 0.3253
Epoch 25/1000
101/101 [==============================] - 0s 937us/step - loss: 0.1377 - mae: 0.3218
Epoch 26/1000
101/101 [==============================] - 0s 898us/step - loss: 0.1361 - mae: 0.3183
Epoch 27/1000
101/101 [==============================] - 0s 977us/step - loss: 0.1346 - mae: 0.3151
Epoch 28/1000
101/101 [==============================] - 0s 928us/step - loss: 0.1332 - mae: 0.3118
Epoch 29/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1319 - mae: 0.3086
Epoch 30/1000
101/101 [==============================] - 0s 967us/step - loss: 0.1307 - mae: 0.3057
Epoch 31/1000
101/101 [==============================] - 0s 937us/step - loss: 0.1295 - mae: 0.3028
Epoch 32/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1284 - mae: 0.2998
Epoch 33/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.1274 - mae: 0.2973
```

```
Epoch 970/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1326
Epoch 971/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1330
Epoch 972/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.0623 - mae: 0.1326
Epoch 973/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1327
Epoch 974/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1327
Epoch 975/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.0622 - mae: 0.1327
Epoch 976/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1328
Epoch 977/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1327
Epoch 978/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1324
Epoch 979/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1325
Epoch 980/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0623 - mae: 0.1327
Epoch 981/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1329
Epoch 982/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1325
Epoch 983/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1327
Epoch 984/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
Epoch 985/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1326
Epoch 986/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1325
Epoch 987/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
Epoch 988/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1328
Epoch 989/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1323
Epoch 990/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
Epoch 991/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1326
Epoch 992/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1325
Epoch 993/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1326
Epoch 994/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.0622 - mae: 0.1321
Epoch 995/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1326
Epoch 996/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.0622 - mae: 0.1323
Epoch 997/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
Epoch 998/1000
101/101 [==============================] - 0s 1ms/step - loss: 0.0622 - mae: 0.1323
Epoch 999/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
Epoch 1000/1000
101/101 [==============================] - 0s 2ms/step - loss: 0.0622 - mae: 0.1324
```

In [250]:

```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: 1000
Время обучения: 168 секунд
```

```
Функция потерь MSE: 0.062156479805707793
Метрика качества MAE: 0.13214363157749176
```

In [251]:

```python
x = np.linspace(-6, 1, 200)
y = np.linspace(-3, 20, 200)
```

In [252]:

```python
line = np.array(list(itertools.product(x, y)))
xs, ys = np.hsplit(line, 2)
```

In [253]:

```python
predicted = model.predict(line)
```

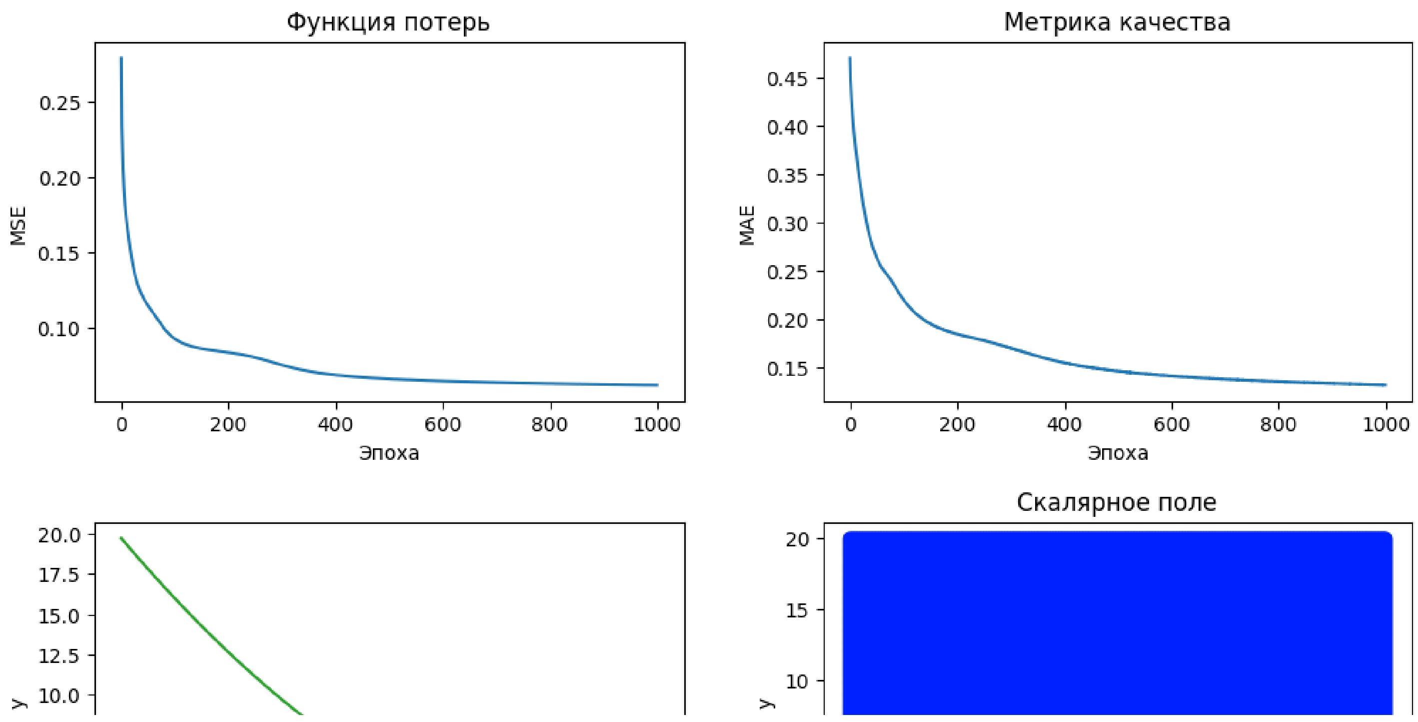
```
1250/1250 [==============================] - 1s 917us/step
```

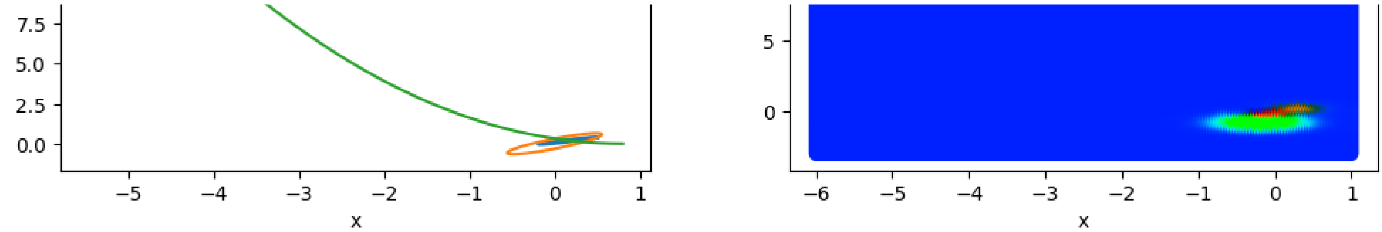
In [254]:

```python
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('x')
axes[1, 0].set_ylabel('y')
axes[1, 0].plot(x1, y1)
axes[1, 0].plot(x2, y2)
axes[1, 0].plot(x3, y3)

axes[1, 1].set_title('Скалярное поле')
axes[1, 1].set_xlabel('x')
axes[1, 1].set_ylabel('y')
axes[1, 1].scatter(xs, ys, c = predicted)
```

Out[254]:

```
<matplotlib.collections.PathCollection at 0x1bbc79ab6d0>
```




7.5
5.0
2.5
0.0
−5
−4
−3
−2
−1
0
1
x
5
0
−6
−5
−4
−3
−2
−1
0
1
x

Лабораторная работа № **4** по Нейроинформатике

Сети с радиальными базисными элементами

Выполнила**:** Тимофеева Наталья М**8**О**-408**Б**-19**

Вариант № **16**

Часть **2**

Аппроксимация

In [81]:

```
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 keras import backend
from timeit import default_timer as timer
```

In [82]:

```
class RBFLayer(keras.layers.Layer):
    def __init__(self, output_dim, **kwargs):
        self.output_dim = output_dim
        super(RBFLayer, self).__init__(**kwargs)

    def build(self, input_shape):
        self.mu = self.add_weight(name = 'mu',
                                  shape = (input_shape[1], self.output_dim),
                                  initializer = tf.keras.initializers.RandomUniform(minv
al = 0, maxval = 3.5),
                                  trainable = True)
        self.sigma = self.add_weight(name = 'sigma',
                                     shape = (self.output_dim,),
                                     initializer = 'random_normal',
                                     trainable = True)
        self.sw = self.add_weight(name = 'sw',
                                  shape = (self.output_dim,),
                                  initializer = 'random_normal',
                                  trainable = True)
        super(RBFLayer, self).build(input_shape)

    def call(self, inputs):
        diff = backend.expand_dims(inputs) - self.mu
        output = backend.exp(backend.sum(diff ** 2, axis = 1) * self.sigma)
        output = output * self.sw
        return output
```

In [83]:

```
def fun(t):
    return np.sin(np.sin(t) * t ** 2 + 3 * t - 10)
```

Создание модели

In [84]:

```
model = keras.models.Sequential()
model.add(RBFLayer(25, input_dim = 1))
model.add(layers.Dense(16,
                       activation = "relu"))
model.add(layers.Dense(15,
                       activation = "tanh"))
```

```
model.add(layers.Dense(1,
                       activation = "tanh"))
model.compile(loss = 'mse', optimizer = 'adam', metrics = ['mae'])
```

In [85]:

```
t1 = np.linspace(0, 3.5, 100)
y1 = fun(t1)
```

In [86]:

```
epochs = 2000
batch_size = 7

time_start = timer()
hist = model.fit(t1, y1, batch_size = batch_size, epochs = epochs)
time_end = timer()
```

```
Epoch 1/2000
15/15 [==============================] - 0s 997us/step - loss: 0.4985 - mae: 0.6302
Epoch 2/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4860 - mae: 0.6236
Epoch 3/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4836 - mae: 0.6244
Epoch 4/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4836 - mae: 0.6240
Epoch 5/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4827 - mae: 0.6235
Epoch 6/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4811 - mae: 0.6226
Epoch 7/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4798 - mae: 0.6205
Epoch 8/2000
15/15 [==============================] - 0s 784us/step - loss: 0.4829 - mae: 0.6240
Epoch 9/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4744 - mae: 0.6181
Epoch 10/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4761 - mae: 0.6185
Epoch 11/2000
15/15 [==============================] - 0s 784us/step - loss: 0.4810 - mae: 0.6206
Epoch 12/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4741 - mae: 0.6170
Epoch 13/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4702 - mae: 0.6151
Epoch 14/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4684 - mae: 0.6127
Epoch 15/2000
15/15 [==============================] - 0s 825us/step - loss: 0.4636 - mae: 0.6099
Epoch 16/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4594 - mae: 0.6074
Epoch 17/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4575 - mae: 0.6055
Epoch 18/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4571 - mae: 0.6034
Epoch 19/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4575 - mae: 0.6066
Epoch 20/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4526 - mae: 0.6019
Epoch 21/2000
15/15 [==============================] - 0s 864us/step - loss: 0.4551 - mae: 0.6028
Epoch 22/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4452 - mae: 0.5957
Epoch 23/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4463 - mae: 0.5963
Epoch 24/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4396 - mae: 0.5907
Epoch 25/2000
15/15 [==============================] - 0s 997us/step - loss: 0.4400 - mae: 0.5914
Epoch 26/2000
15/15 [==============================] - 0s 926us/step - loss: 0.4409 - mae: 0.5861
Epoch 27/2000
15/15 [==============================] - 0s 855us/step - loss: 0.4301 - mae: 0.5823
```

```
Epoch 1972/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0021 - mae: 0.0318
Epoch 1973/2000
15/15 [==============================] - 0s 2ms/step - loss: 0.0023 - mae: 0.0320
Epoch 1974/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0017 - mae: 0.0289
Epoch 1975/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0019 - mae: 0.0315
Epoch 1976/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0020 - mae: 0.0304
Epoch 1977/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0028 - mae: 0.0335
Epoch 1978/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0016 - mae: 0.0287
Epoch 1979/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0014 - mae: 0.0255
Epoch 1980/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0017 - mae: 0.0268
Epoch 1981/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0020 - mae: 0.0304
Epoch 1982/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0014 - mae: 0.0241
Epoch 1983/2000
15/15 [==============================] - 0s 3ms/step - loss: 0.0017 - mae: 0.0274
Epoch 1984/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0016 - mae: 0.0261
Epoch 1985/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0017 - mae: 0.0288
Epoch 1986/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0014 - mae: 0.0259
Epoch 1987/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0016 - mae: 0.0273
Epoch 1988/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0018 - mae: 0.0285
Epoch 1989/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0028 - mae: 0.0362
Epoch 1990/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0023 - mae: 0.0331
Epoch 1991/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0019 - mae: 0.0298
Epoch 1992/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0016 - mae: 0.0282
Epoch 1993/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0018 - mae: 0.0299
Epoch 1994/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0015 - mae: 0.0252
Epoch 1995/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0036 - mae: 0.0404
Epoch 1996/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0039 - mae: 0.0454
Epoch 1997/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0028 - mae: 0.0385
Epoch 1998/2000
15/15 [==============================] - 0s 997us/step - loss: 0.0021 - mae: 0.0312
Epoch 1999/2000
15/15 [==============================] - 0s 2ms/step - loss: 0.0014 - mae: 0.0267
Epoch 2000/2000
15/15 [==============================] - 0s 1ms/step - loss: 0.0017 - mae: 0.0288
```

In [87]:

```
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: 2000
Время обучения: 43 секунд
Функция потерь MSE: 0.0013011953560635448
Метрика качества MAE: 0.02217509038746357
```

In [88]:

```
t2 = np.linspace(0, 3.5, 2000)
y2 = model.predict(t2)
real_y = fun(t2)
```

```
63/63 [==============================] - 0s 611us/step
```

In [89]:

```
mu_x = model.get_layer(index = 0).get_weights()[0][0]
mu_y = model.predict(mu_x)
```

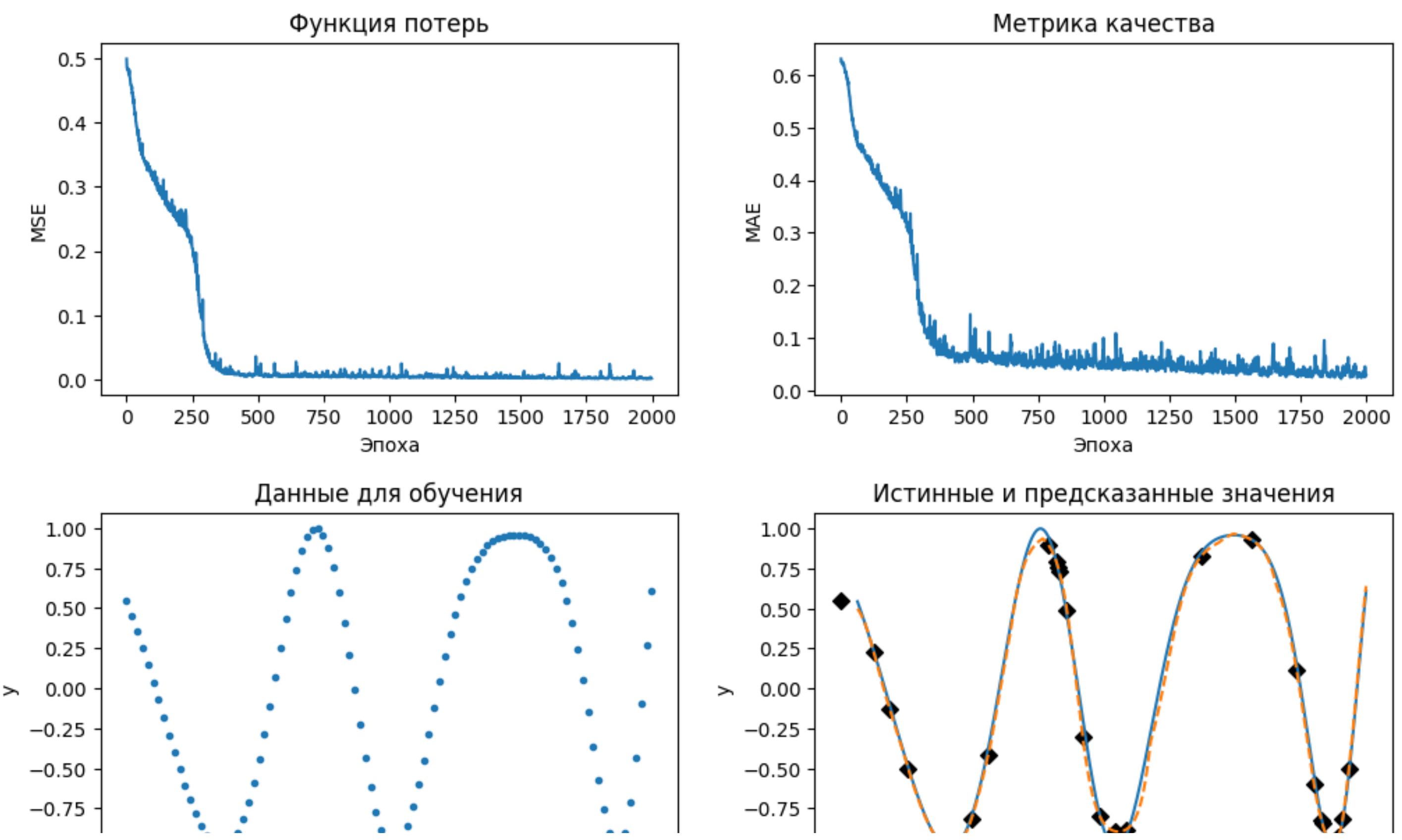
```
1/1 [==============================] - 0s 16ms/step
```

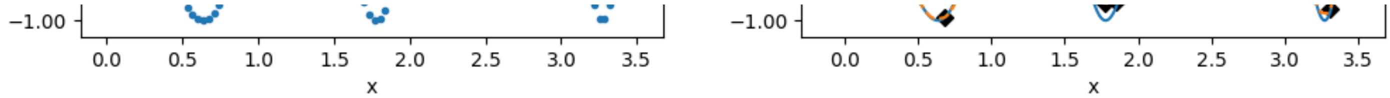
In [90]:

```
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_title('Данные для обучения')
axes[1, 0].set_xlabel('x')
axes[1, 0].set_ylabel('y')
axes[1, 0].plot(t1, y1, '.')

axes[1, 1].set_title('Истинные и предсказанные значения')
axes[1, 1].set_xlabel('x')
axes[1, 1].set_ylabel('y')
axes[1, 1].plot(t2, real_y)
axes[1, 1].plot(t2, y2, '--')
axes[1, 1].scatter(mu_x, mu_y, color = "black", marker = "D")
```

Out[90]:

```
<matplotlib.collections.PathCollection at 0x1c731f857b0>
```




−1.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
x
−1.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
x